

**MULTIMEDIA UNIVERSITY OF KENYA**

**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY**

**PROJECT DOCUMENTATION**

**SIMULATION OF STOCK MARKET PREDICTION SYSTEM USING DEEP LEARNING**

**BY NAME: SOLOMON NGATAARA NDUNG’U**

**REG. NO: CIT-221-011/2015**

**SUPERVISOR: KIIRU**

**8th APRIL 2019**

Submitted in partial fulfilment of the requirements for the award of Bachelor of

Science in Information Technology

# DECLARATION

I hereby declare that this Project [Proposal] is my own work and has, to the best of my knowledge, not been submitted to any other institution of higher learning.

**Student: Registration Number:**

**Signature:……………………………….. Date:…………………………….**

This project documentation has been submitted as a partial fulfilment of requirements for the Bachelor of Science in Information Technology of Multimedia University of Kenya with my approval as the University supervisor.

**Supervisor:**

**Signature: Date:**

# ACKNOWLEDGEMENT

I would like to thank my fellow students for aiding me in completing this project in terms of research and testing of data. Kelvin Akidiva; you are one of the guys who has really helped me in making this project a success.

I would like to also thank my supervisor for guiding me towards making the data output more accurate and more realistic in accordance with the said objectives of the study.

I thank the school for giving me an opportunity to research and implement my own project. The course offered by the school is quite an engaging one and thus doing the project allows a more practical approach to be applied and thus builds skills and aligns one’s mind perfectly in the course.

# ABSTRACT

The project talks of the need for artificial intelligence in companies because AI is slowly taking over the market and improving services.

Chapter two reviews various studies carried out, their strengths and also their weaknesses. The studies talk about various models used in stock market prediction.

Chapter three talks of the methodology of the Recurrent Neural Network used, more specifically, Long Term Short Term Memory model used in studying a company’s data and predicting the future stock position in the market. Whether the price will rise or fall. There is also justification of the study and references.

# TABLE OF CONTENTS

Table of Contents

[DECLARATION I](#_Toc5621051)

[ACKNOWLEDGEMENT II](#_Toc5621052)

[ABSTRACT III](#_Toc5621053)

[TABLE OF CONTENTS IV](#_Toc5621054)

[LIST OF ABBREVIATIONS VII](#_Toc5621055)

[LIST OF TABLES VIII](#_Toc5621056)

[LIST OF FIGURES IX](#_Toc5621057)

[CHAPTER ONE - INTRODUCTION 1](#_Toc5621058)

[1.1 BACKGROUND STUDY 1](#_Toc5621059)

[1.2 PROBLEM STATEMENT 1](#_Toc5621060)

[1.3 AIM OF THE STUDY 2](#_Toc5621061)

[1.4 OBJECTIVES 2](#_Toc5621062)

[1.5 RESEARCH QUESTIONS 2](#_Toc5621063)

[1.6 SIGNIFICANCE OF THE STUDY 2](#_Toc5621064)

[1.7 SCOPE OF THE STUDY 3](#_Toc5621065)

[1.8 ASSUMPTIONS 3](#_Toc5621066)

[1.9 LIMITATIONS 3](#_Toc5621067)

[CHAPTER TWO: LITERATURE REVIEW 4](#_Toc5621068)

[2.1 INTRODUCTION 4](#_Toc5621069)

[2.2 CHALLENGE OF REVIEWED SYSTEM 5](#_Toc5621070)

[2.3 PROPOSED SOLUTION 6](#_Toc5621071)

[CHAPTER THREE: METHODOLOGY 7](#_Toc5621072)

[3.1 INTRODUCTION 7](#_Toc5621073)

[3.2 THE METHODOLOGY 7](#_Toc5621074)

[3.2.1 RAD model 7](#_Toc5621075)

[3.3 JUSTIFICATION OF THE STUDY 9](#_Toc5621076)

[CHAPTER 4: SYSTEM ANALYSIS 11](#_Toc5621077)

[4.1 BUSINESS MODELLING 11](#_Toc5621078)

[4.1.1 Long Short Term Memory (LSTM) models 11](#_Toc5621079)

[4.2 DATA MODELLING 13](#_Toc5621080)

[4.2.1 Data Generator 13](#_Toc5621081)

[4.2.2 Data Augmentation 14](#_Toc5621082)

[4.3 PROCESS MODELLING 15](#_Toc5621083)

[4.3.1 Defining Inputs and Outputs 15](#_Toc5621084)

[4.3.2 Defining Parameters of the LSTM and Regression layer 16](#_Toc5621085)

[4.3.3 Calculating LSTM output and Feeding it to the regression layer to get final prediction 16](#_Toc5621086)

[4.3.4 Loss Calculation and Optimizer 16](#_Toc5621087)

[4.3.5 Prediction Related Calculations 16](#_Toc5621088)

[4.3.6 Running the LSTM 17](#_Toc5621089)

[4.3.7 Visualizing the Predictions 18](#_Toc5621090)

[4.4 APPLICATION GENERATION 19](#_Toc5621091)

[4.4.1 System Requirements 19](#_Toc5621092)

[4.5 TESTING AND TURNOVER 21](#_Toc5621093)

[CHAPTER 5: SYSTEM DESIGN 22](#_Toc5621094)

[5.1 ARCHITECTURAL DESIGN 22](#_Toc5621095)

[5.2 DATABASE DESIGN 22](#_Toc5621096)

[5.3 USER INTERFACE DESIGN 22](#_Toc5621097)

[CHAPTER 6: IMPLEMENTATION AND TESTING 24](#_Toc5621098)

[6.1 Development Environment 24](#_Toc5621099)

[6.2 SYSTEM COMPONENTS 24](#_Toc5621100)

[6.2.3 Acquiring the data 24](#_Toc5621101)

[6.2.4 Splitting data into a set for training and testing 25](#_Toc5621102)

[6.2.6 Long Short Term Memory (LSTM) 26](#_Toc5621103)

[6.2.5 Normalizing the data 26](#_Toc5621104)

[6.2.7 Visualizing the predictions 29](#_Toc5621105)

[CHAPTER 7: CONCLUSION 30](#_Toc5621106)

[7.1 Achievements and lessons learnt 30](#_Toc5621107)

[7.2 Conclusions 30](#_Toc5621108)

[7.3 Recommendations 30](#_Toc5621109)

[7.4.1 REQUIREMENTS 30](#_Toc5621110)

[REFERENCES 32](#_Toc5621111)

[APPENDICES 34](#_Toc5621112)

[APPENIDX 1 34](#_Toc5621113)

[APPENDIX 2 34](#_Toc5621114)

# LIST OF ABBREVIATIONS

**LSTM :** long-short-term-memory

**NSE:** National Stock Exchange

**ANN:** Artificial Neural Networks

**ARIMA:** Autoregressive Integrated Moving Average Model

**CRISP-DM:** Cross-Industry Standard Process for data mining

**df:** DataFrame

**w.r.t.:** with respect to

**API:** Application Programming Interface

**MSE:** Mean Square Error

**RAD:** Rapid Application Development

**SDLC:** Software Development Life Cycle

# LIST OF TABLES

[Table 1 34](#_Toc5620781)

# LIST OF FIGURES

[Figure 1 8](#_Toc5620761)

[Figure 2 12](#_Toc5620762)

[Figure 3 14](#_Toc5620763)

[Figure 4 18](#_Toc5620764)

[Figure 5 19](#_Toc5620765)

[Figure 6 21](#_Toc5620766)

[Figure 7 22](#_Toc5620767)

[Figure 8 23](#_Toc5620768)

[Figure 9 23](#_Toc5620769)

[Figure 10 25](#_Toc5620770)

[Figure 11 25](#_Toc5620771)

[Figure 12 26](#_Toc5620772)

[Figure 13 26](#_Toc5620773)

[Figure 14 27](#_Toc5620774)

[Figure 15 27](#_Toc5620775)

[Figure 16 28](#_Toc5620776)

[Figure 17 28](#_Toc5620777)

[Figure 18 29](#_Toc5620778)

[Figure 19 29](#_Toc5620779)

[Figure 20 34](#_Toc5620780)

# CHAPTER ONE - INTRODUCTION

## 1.1 BACKGROUND STUDY

Artificial intelligence refers to computer systems that are able to perform tasks that historically required human intelligence, such such as recognizing images, understanding speech, translating languages and making decisions. Some examples of artificial intelligence in financial services are mobile checking deposits that read checks, custom notifications that flag high payments and specific transfer reminders.

Financial Services these days are slowly being taken over by artificial intelligence. Thus, forecasting stock returns is one of the areas being ventured into by several companies.

Forecasting stock return is an important financial subject that has attracted researchers' attention for many years. It involves an assumption that fundamental information publicly available in the past has some predictive relationships to the future stock returns. This study tries to help the investors in the stock market to decide the better timing for buying or selling stocks based on the knowledge extracted from the historical prices of such stocks.

Technical analysis as expounded in: Tsang’s paper of Hong Kong stock price forecasting and Murphy’s paper of Technical Analysis of the Financial Markets; refers to various methods that purpose to predict future price movements using past stock prices and volume information. It is based on the assumption that future market directions can be determined by examining historical price data.

Thus, it is assumed that price trends and patterns which exists can be identified and utilized for profit.

## 1.2 PROBLEM STATEMENT

Investors have been trying to find a way to predict stock market prices and to find the right stocks and right timing to buy or sell. Thus, this study tries to solve the problem of accurately predicting the prices using neural networks in deep learning.

## 1.3 AIM OF THE STUDY

The aim of the study is to analyze historical data available on companies’ stocks using Long-Short-Term-Memory in Recurrent Neural Networks in order to help investors to know when to buy new stocks or to sell their stocks.

## 1.4 OBJECTIVES

1. To develop a framework that will acquire stock data from time series websites like Yahoo finance using API key.
2. To develop architecture that will enable training and testing of the data.
3. To design architecture that will perform data normalization.
4. To develop a simulation that will predict and visualize future stock market with acquired data.

## 1.5 RESEARCH QUESTIONS

1. How will the system acquire company data that will enable prediction of stocks?
2. How will the data be split into training and testing data?
3. How will data normalization be performed? Is it significant for the study?
4. How will the prediction be visualized?

## 1.6 SIGNIFICANCE OF THE STUDY

1. This study will aid the Kenyan youths towards accurate investment of their money thus maximize on gaining dividends in their investments.
2. Attract various people in Kenya towards stock market investment and thus contribute towards gaining of capital in companies thus allow for economic growth of country.
3. Look at the accuracy of the model and try and compare it with the real-life data or change in the stocks.
4. Help the presenter of the project gain skills in recurrent neural networks and learn how to code in python and model statistical data.

## 1.7 SCOPE OF THE STUDY

The study shows previous studies undertaken in predicting stock prices and their accuracy. It also explains briefly what long-short-term-memory **(LSTM)** is and the various stages undertaken so as to apply the methodology and get results.

## 1.8 ASSUMPTIONS

1. The project was done within two-three months after finishing the project proposal.
2. The required hardware and software resources are available and responds to researcher’s needs.
3. The data collected will be trained and produce accurate results which can be used to forecast the stock prices.
4. The researcher has clear knowledge of the field and is competent enough with Neural networks and python programming.
5. The stocks used are from a United States of America company; but it would act as a representation of the Kenyan market.

## 1.9 LIMITATIONS

1. The hardware needed for the study is not sufficient enough to make the study effective.
2. Neural networks is a bit technical thus learning and using it would require patience.
3. Internet is costly and slow.
4. The student has to balance between project and other seven units that he is engaged in.

# CHAPTER TWO: LITERATURE REVIEW

## 2.1 INTRODUCTION

This paper by Wu:”An effective application of decision tree to stocking trading”, presents the design and implementation of low cost, simple, user-friendly and effective National Stock Exchange (NSE) stock prediction and analysis for short term quick gains. Their system collects raw history data from various locations; mainly from National Stock Exchange website and store it in their database. Various data mining methodologies (they did not mention the methodologies) are used on filter, classify and categorize data by extracting useful patterns or rules to represent it in simple form for investor. Investors, as per his investment goals, can choose stocks analysed by this system.

This other paper by Kihoro: “Stock market prediction using artificial neural network”; looks at the application of Artificial Neural Networks (ANN) in predicting future Equity Bank share prices using historical data. It assumed that only previous prices affect future prices, then fitted Autoregressive Integrated Moving Average Model (ARIMA - used for prediction of non-stationary time series when linearity between variables is supposed) models to the stock prices data in order to identify the best input lags into the ANN model. The best combination of lags was taken for input lags and led to optimal results in terms of the least mean squared error between the predicted values and the test data. The 3-3-1 network architecture gave the best results in terms of the Mean Squared error.

In this other paper by Al-Radaideh: “Predicting stock prices using data mining techniques”;, the main objective is to analyse the historical data available on stocks using decision tree technique as one of the classification methods of data mining in order to help investors to know when to buy new stocks or sell their stocks. To build the model that analyses the stock trends using the decision tree technique, the CRISP-DM (Cross-Industry Standard Process for data mining) is used.

This model consists of the following six steps:

* Understanding the reason and objective of mining the stock prices.
* Understanding the collected data and how it is structured.
* Preparing the data that is used I the classification model.
* Selecting the technique to build the model.
* Evaluating the model by using one of the well known evaluation methods (the **K-Fold Cross Validation** where K = 10 folds and the **percentage split method** where 66% of the data was used for training and the remainder for testing).
* Deploying the model in the stock market to predict the best action to be taken, either selling or buying stocks.
* Understanding the reason and objective of building the model.

## 2.2 CHALLENGE OF REVIEWED SYSTEM

1. In Tsang’s paper, the documentation is shallow and un-comprehensive. The various methodologies are not specified. Plus the system is only able to get short-term quick gains; it does not include a study for long-term quick gains.
2. In Kihoro’s paper, there were some missing values as trading did not occur in some days. The missing values were imputed (taking the average of the previous week’s and following week’s share prices). The overall dwindling of the Nairobi Stock Exchange 20 share index and all share index, caused by tough economic times; did affect the price outcomes thus accuracy of study. The study also dwelt on one company: Equity Bank. It would have been interesting if study could have extended to other companies while incorporating other inputs to the network.
3. In Al-Radaideh’s paper, the reason for such a low accuracy is that the company’s performance in the stock market is affected by internal financial factors such as; news about the company, financial reports, and the overall performance of the market. Also, external factors can affect the performance of the company in the market such as; political events and political decisions. Thus, it can be difficult to have a model that gives a high accuracy classification for all the companies at the same time because the performance of these companies differ.

## 2.3 PROPOSED SOLUTION

1. Evaluation of a larger collection of learning techniques such as neural networks, genetic algorithms, and association rules.
2. LSTM RNN ensures that a constant error is maintained to allow the RNN to learn over long time steps which enables it to associate problems and its effects remotely (as per research by Bedaiko’s paper).

# CHAPTER THREE: METHODOLOGY

## 3.1 INTRODUCTION

Recurrent neural networks are networks with loops in them, allowing information to persist.

**Long Short Term Memory Networks (LSTM)** - are a special kind of RNN, capable of learning long-term dependencies. They are explicitly designed to avoid long-term dependency problem. Remembering information for long periods of time is practically their default behaviour.

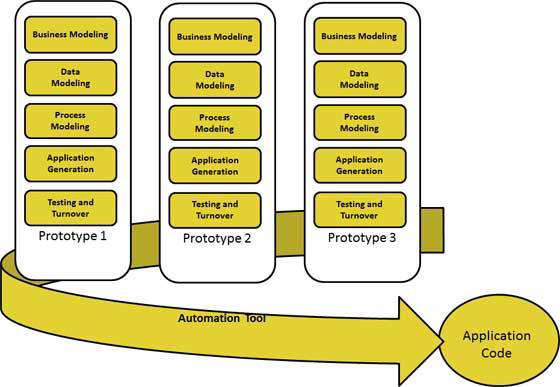
## 3.2 THE METHODOLOGY

### 3.2.1 RAD model

RAD model is **Rapid Application Development model**. It is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

**Diagram of RAD-Model:**

Following image illustrates the RAD Model:



Figure

The phases in the rapid application development (RAD) model are:

**Business modeling:** The information flow is identified between various business functions.

**Data modeling:** Information gathered from business modeling is used to define data objects that are needed for the business.

**Process modeling**: Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects.

**Application generation:** Automated tools are used to convert process models into code and the actual system.

**Testing and turnover:** Test new components and all the interfaces.

**Advantages of the RAD model:**

*  Reduced development time.
*  Increases reusability of components
*  Quick initial reviews occur
*  Encourages customer feedback
*  Integration from very beginning solves a lot of integration issues.

**Disadvantages of RAD model:**

* Depends on strong team and individual performances for identifying business requirements.
* Only system that can be modularized can be built using RAD
* Requires highly skilled developers/designers.
* High dependency on modelling skills.
* Inapplicable to cheaper projects as cost of modelling and automated code generation is very high.

**When to use RAD model:**

* RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time.
* It should be used if there’s high availability of designers for modelling and the budget is high enough to afford their cost along with the cost of automated code generating tools.
* RAD SDLC model should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).

## 3.3 JUSTIFICATION OF THE STUDY

Long-Term-Short-Term are able to predict stock price behaviour correctly most of the time; even though it is not perfect.

We are making predictions roughly in the range of 0 and 1.0 (without using the true stock prices. It is still correct though, since we are predicting stock price movement, not the price themselves.

# CHAPTER 4: SYSTEM ANALYSIS

## 4.1 BUSINESS MODELLING

* Information flow is identified between various different business functions.
* Therefore, business modelling is the graphical representation of a company’s [business processes](https://tallyfy.com/business-process) or [workflows](https://tallyfy.com/what-is-a-workflow/), as a means of identifying potential improvements.
* This is usually done through different graphing methods, such as the flowchart, data-flow diagram.

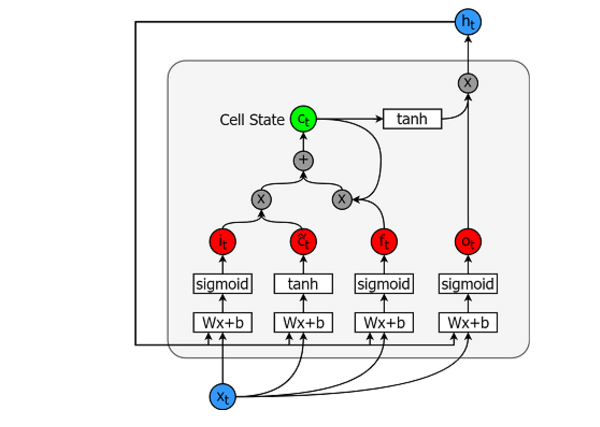
### 4.1.1 Long Short Term Memory (LSTM) models

Are models that can predict an arbitrary number of steps into the future.

An LSTM module or cell has five essential components which allows it to model both long-term and short-term data:

1. **Cell state ($c\_t$)** - This represents the internal memory of the cell which stores both short term memory and long-term memories.
2. **Hidden state ($h\_t$)** - This is output state information calculated w.r.t. (with respect to) current input, previous hidden state and current cell input which you eventually use to predict the future stock market prices. Additionally, the hidden state can decide to only retrieve the short or long-term or both types of memory stored in the cell state to make the next prediction.
3. **Input gate ($i\_t$)** - Decides how much information from current input flows to the cell state.
4. **Forget gate ($f\_t$)** - Decides how much information from the current input and the previous cell state flows into the current cell state.
5. **Output gate ($o\_t$)** - Decides how much information from the current cell state flows into the hidden state, so that if needed LSTM can only pick the long-term memories or short-term memories and long-term memories.

A cell is pictured below:



Figure

And the equations for calculating each of these entities are as follows:

Input Gate:[8]

First Equation

Old Cell State:[8]

Third Equation

New Cell State:[8]

Fourth Equation

Forget Gate:[8]

Sixth Equation

Output Gate:

Seventh Equation

Hidden state:

Eighth equation

TensorFlow provides a nice sub API (Application Programming Interface), called RNN API, for implementing time series models. We will be using that for our implementations.

## 4.2 DATA MODELLING

* Information gathered from business modelling is used to define data objects that are needed for the business.
* Data modelling is the process of documenting a complex software system design as an easily understood diagram, using text and symbols to represent the way [data](https://searchdatamanagement.techtarget.com/definition/data) needs to flow.
* The diagram can be used to ensure efficient use of data, as a blueprint for the construction of new software or for re-engineering a [legacy application](https://searchitoperations.techtarget.com/definition/legacy-application).

### 4.2.1 Data Generator

We are going to first implement a data Generator to train our model.

This enables generation of data for training.

This generator will have a method called **unroll\_batches(…)** which will output a set of **num\_rollings** batches of input data obtained sequentially, where a batch of data is of size [batch\_size, 1].

Then each batch of input data will have a corresponding output batch of data.

For example if num\_unrollings=3 and batch\_size=4 a set of unrolled batches it might look like,

input data: $[x\_0,x\_10,x\_20,x\_30], [x\_1,x\_11,x\_21,x\_31], [x\_2,x\_12,x\_22,x\_32]$

output data: $[x\_1,x\_11,x\_21,x\_31], [x\_2,x\_12,x\_22,x\_32], [x\_3,x\_13,x\_23,x\_33]$

### 4.2.2 Data Augmentation

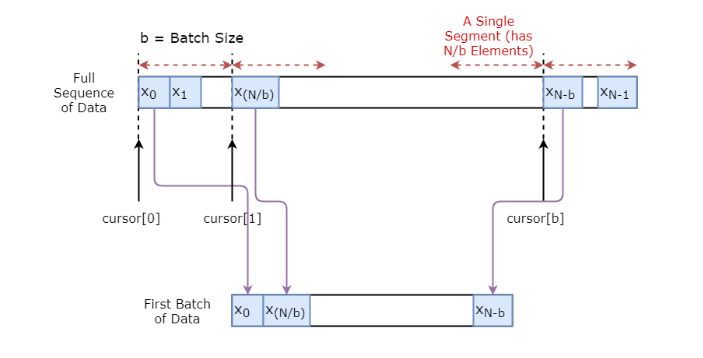
Data augmentation enables data to be random which prevents overfitting or “Cramming”.

Also to make your model robust you will not make the output for $xt$ always $x{t+1}$. Rather you will randomly sample an output from the set $x{t+1},x{t+2},\ldots,x\_{t+N}$ where $N$ is a small window size.

Here you are making the following assumption:

$x{t+1},x{t+2},\ldots,x\_{t+N}$ will not be very far from each other.

Below you illustrate how a batch of data is created visually.



Figure

**Defining Hyperparameters**

In this section, you'll define several hyperparameters.

D is the dimensionality of the input. It's straightforward, as you take the previous stock price as the input and predict the next one, which should be 1.

Then you have num\_unrollings, this is a hyperparameter related to the backpropagation through time (BPTT) that is used to optimize the LSTM model. This denotes how many continuous time steps you consider for a single optimization step.

You can think of this as, instead of optimizing the model by looking at a single time step, you optimize the network by looking at num\_unrollings time steps. The larger the better.

Then you have the batch\_size.

**Batch size** is how many data samples you consider in a single time step.

Next you define **num\_nodes** which represents the number of hidden neurons in each cell.

## 4.3 PROCESS MODELLING

Data objects defined in data modelling are converted to achieve business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects.

### 4.3.1 Defining Inputs and Outputs

Next you define placeholders for training inputs and labels. This is very straightforward as you have a list of input placeholders, where each placeholder contains a single batch of data. And the list has num\_unrollings placeholders that will be used at once for a single optimization step.

### 4.3.2 Defining Parameters of the LSTM and Regression layer

You will have a three layers of LSTMs and a linear regression layer, denoted by w and b, that takes the output of the last Long Short-Term Memory cell and output the prediction for the next time step. You can use the MultiRNNCell in TensorFlow to encapsulate the three LSTMCell objects you created. Additionally, you can have the dropout implemented LSTM cells, as they improve performance and reduce overfitting.

### 4.3.3 Calculating LSTM output and Feeding it to the regression layer to get final prediction

In this section, you first create TensorFlow variables **(c and h)** that will hold the cell state and the hidden state of the Long Short-Term Memory cell.

Then you transform the list of train\_inputs to have a shape of **[num\_unrollings, batch\_size, D],** this is needed for calculating the outputs with the **tf.nn.dynamic\_rnn function**.

You then calculate the LSTM outputs with the tf.nn.dynamic\_rnn function and split the output back to a list of **num\_unrolling** tensors. (the loss between the predictions and true stock prices)

### 4.3.4 Loss Calculation and Optimizer

Now, you'll calculate the loss. However, you should note that there is a unique characteristic when calculating the loss. For each batch of predictions and true outputs, you calculate the Mean Squared Error. And you sum (not average) all these mean squared losses together.

Finally, you define the optimizer you're going to use to optimize the neural network. In this case, you can use Adam, which is a very recent and well-performing optimizer.

### 4.3.5 Prediction Related Calculations

Here you define the prediction related TensorFlow operations.

First, define a placeholder for feeding in the input **(sample\_inputs)**, then similar to the training stage, you define state variables for prediction (sample\_c and sample\_h).

Finally you calculate the prediction with the tf.nn.dynamic\_rnn function and then sending the output through the regression layer (w and b).

You also should define the r**eset\_sample\_state** operation, which resets the cell state and the hidden state. You should execute this operation at the start, every time you make a sequence of predictions.

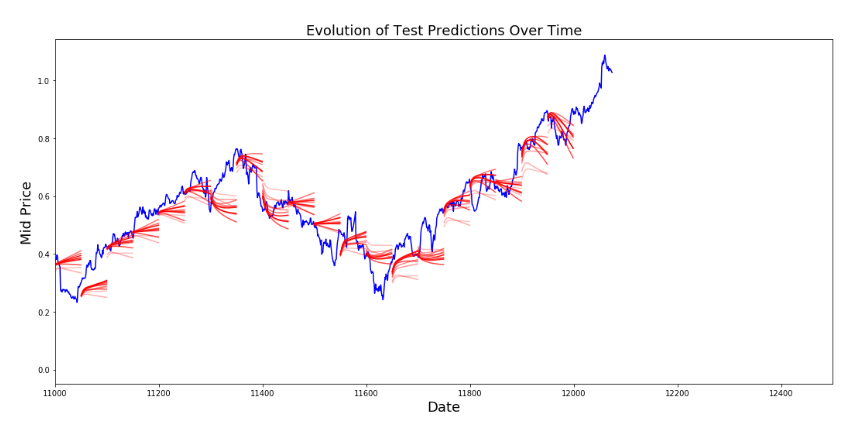
### 4.3.6 Running the LSTM

Here you will train and predict stock price movements for several epochs and see whether the predictions get better or worse over time. You follow the following procedure:

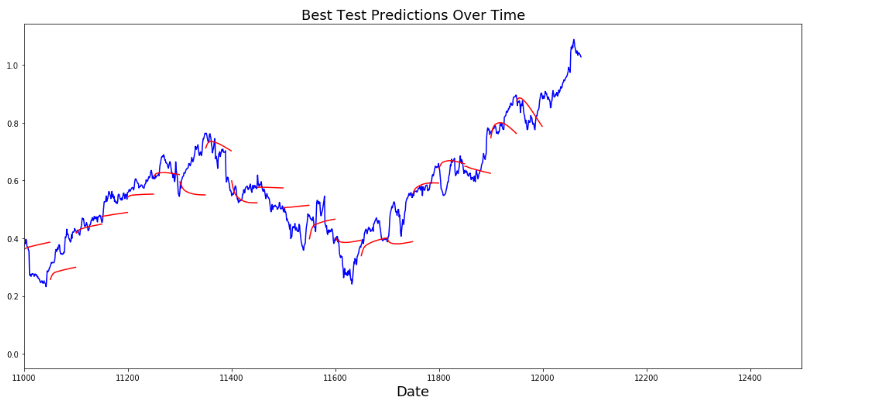
* Define a test set of starting points (test\_points\_seq) on the time series to evaluate the model on.
* For each epoch: For full sequence length of training data Unroll a set of num\_unrollings batches train the neural network with the unrolled batches.
* Calculate the average training loss For each starting point in the test set .
* Update the LSTM state by iterating through the previous num\_unrollings data points found before the test point.
* Make predictions for n\_predict\_once steps continuously, using the previous prediction as the current input.
* Calculate the MSE **(Mean Square Error)** loss between the **n\_predict\_once** points predicted and the true stock prices at those time stamps.

### 4.3.7 Visualizing the Predictions

You can see how the MSE loss is going down with the amount of training. This is good sign that the model is learning something useful. To quantify your findings, you can compare the network's MSE loss to the MSE loss you obtained when doing the standard averaging (0.004). You can see that the LSTM is doing better than the standard averaging. And you know that standard averaging (though not perfect) followed the true stock prices movements reasonably.



Figure



Figure

## 4.4 APPLICATION GENERATION

Automated tools are used to convert process models into code and the actual system.

### 4.4.1 System Requirements

Is a document or set of documentation that describes the features and behaviour of a system or software application.

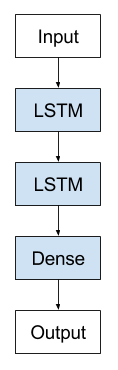
**4.4.1.1 Functional Requirements**

Defines a function of a system or its component, where a function is described as a specification of behaviour between outputs and inputs.

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish.

Arguments required to construct the Recurrent Neural Network model using LSTM cell:

* **sess**: provides access to devices in the local machine, and remote devices using the distributed TensorFlow runtime. Also caches information about the Graph so that you can efficiently run the same computation multiple times.
* **Input**: training data X (or number of stock data to be fed to system)
* **learning\_rate**: the rate at which the network will configure weights so as to be familiar with the trend and know how to predict stock prices.
* **embed\_size (int):** length of embedding vector. Embedding is a mapping from discrete objects, such as words, to vectors of real numbers. It allows for binary processing which enables faster training rather than the computer interacting with the high level programming language which would be heavy for it.
* **keep\_prob(int)** : (1.0 – dropout rate.) for a LSTM cell. This is the maximum gradient to maintain to prevent exploding gradients.
* **Symbols:** a list of stock symbols associated with each sample. A stock symbol is a unique series of letters assigned to a security for trading purposes. e.g, NYSE (New York Stock Exchange)
* **num\_layers**: number of LSTM cell layers. This below is a diagram of LSTM layers.



Figure

**4.4.1.2 Non-functional requirements**

Is a requirement that specifies criteria that can be used to judge the operation of a system.

Non-functional requirements can be divided into two main categories:

* Execution qualities; such as safety, security and usability, which are observable during operation.
* Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the system.

1. It is usable through the FireFox browser.
2. It is testable because it trains the acquired data.
3. It is maintainable.

## 4.5 TESTING AND TURNOVER

Test new components and all the interfaces.

# CHAPTER 5: SYSTEM DESIGN

## 5.1 ARCHITECTURAL DESIGN

Is a process for identifying the sub-systems making up a system and the framework for sub-system control and communication.

Acquiring data

Shaping data and normalizing

Model for training

Training Output

Figure

## 5.2 DATABASE DESIGN

This is a simulation prediction system. Its purpose is to show results of study of data. It only has logs which shows the activities that have taken place and results of the data. It does not have a database design.

## 5.3 USER INTERFACE DESIGN

User interface is the front-end application view to which user interacts in order to use the software. It is: attractive, simple to use, responsive, clear to understand.

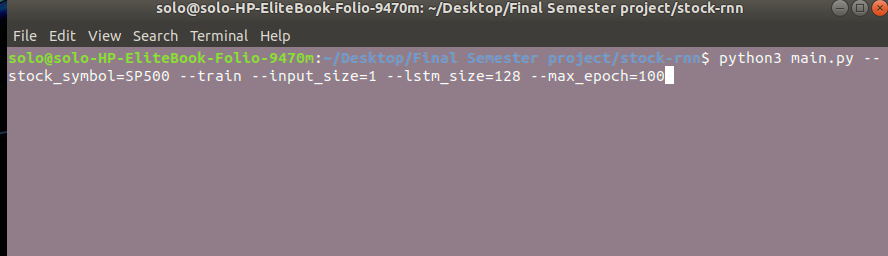
There are two types of User Interface:

1**. Command Line Interface**: provides a command prompt, where the user types the command and feeds to the system.

**2. Graphical User Interface**: provides the simple interactive interface to interact with the system.

This system has both command line interface and graphical user interface.

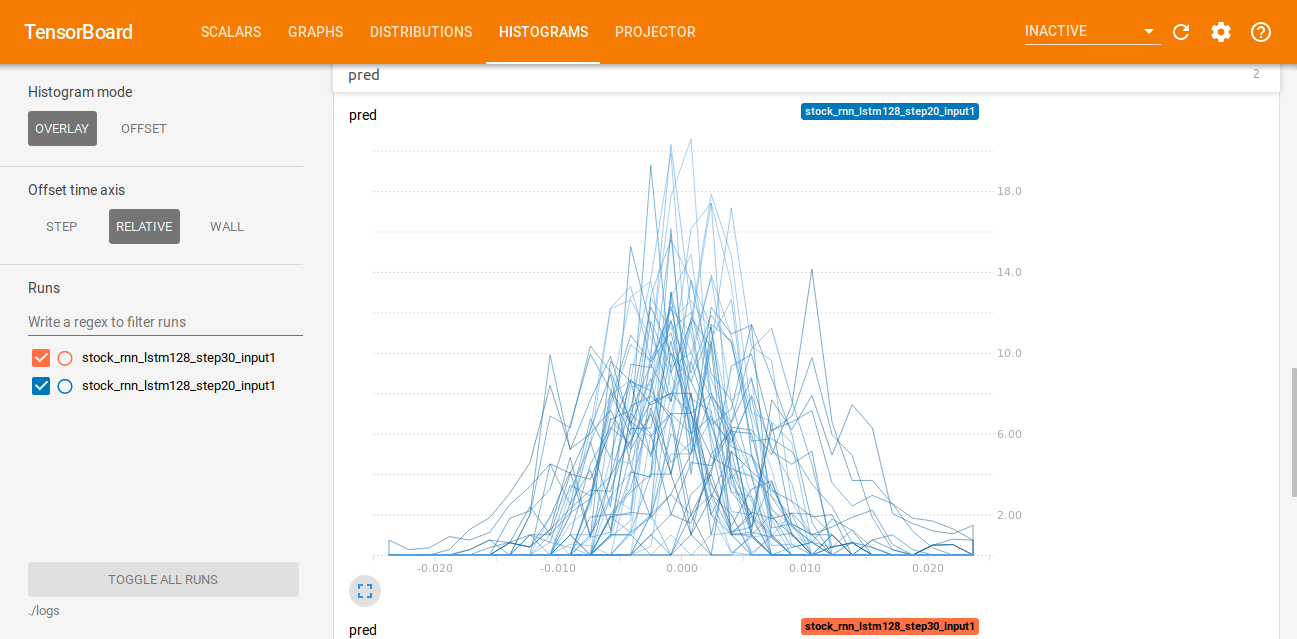
For command line interface:



Figure

This is where some commands are entered and it activates the running of the python codes. It also activates training of the model.

For Graphical User Interface:

Figure

This displays the activities of the training model graphically.

# CHAPTER 6: IMPLEMENTATION AND TESTING

## 6.1 Development Environment

* **Python 3.6.7 :** python is an interpreted, high-level, general-purpose programming language. It provides constructs that enable clear programming on both small and large scales.
* **Numpy 1.16.2** : is the fundamental package for scientific computing with Python.
* **Pandas 0.24.2 :** library providing data structures and analysis tools for python.
* **Tensorflow 1.13.1** : library helping in the development of neural networks architecture.
* **Matplotlib 3.0.3** : is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

## 6.2 SYSTEM COMPONENTS

A model system concept is defined as a collection of interconnected and consistent components that work together for defining, developing, and delivering a software system.

### 6.2.3 Acquiring the data

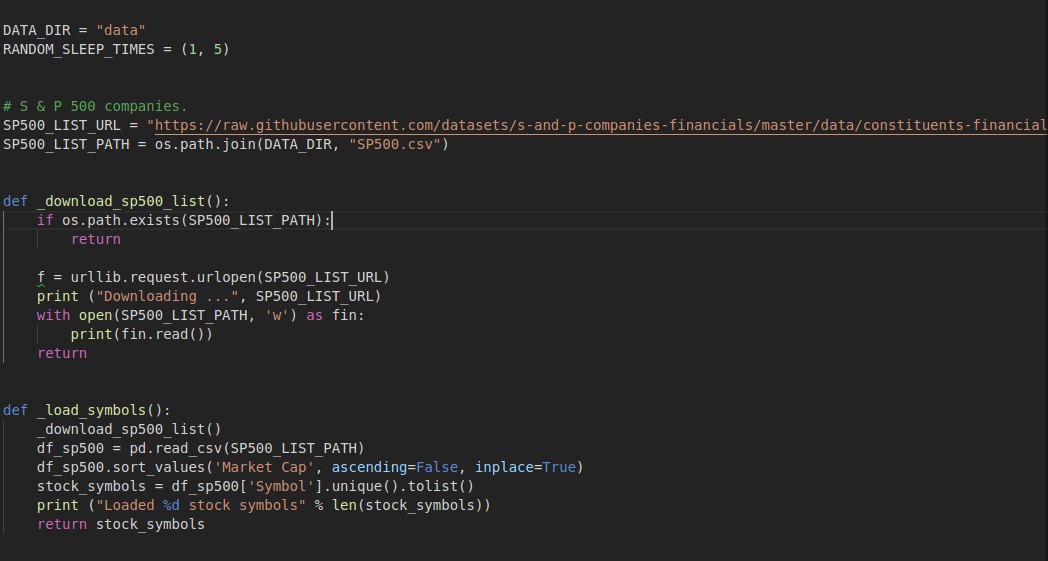
We will be using stock market data acquired from Yahoo Finance but this data has been obtained in GitHub and downloaded so that it may be used in the below project. This is the link to the data: <https://www.sciencedirect.com/science/article/pii/S0167642314005401>

We will save the data as **SP500.csv** and access it on the hard disk via a specified path.

**stock\_symbol** will return a JSON file containing the data for training.

Stock prices come in different categories. They are:

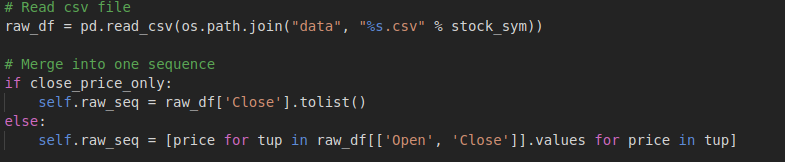
* **Open:** opening stock price of that day.
* **Close:** closing stock price of that day.
* **High:** Highest stock price of acquired data.
* **Low:** Lowest stock price of that day.

Figure 

### 6.2.4 Splitting data into a set for training and testing

One first reads the data, puts it into a **dataframe**. A dataframe is a two-dimensional data structure which is aligned in a tabular fashion in rows and columns.

We then take the opening price and closing price.

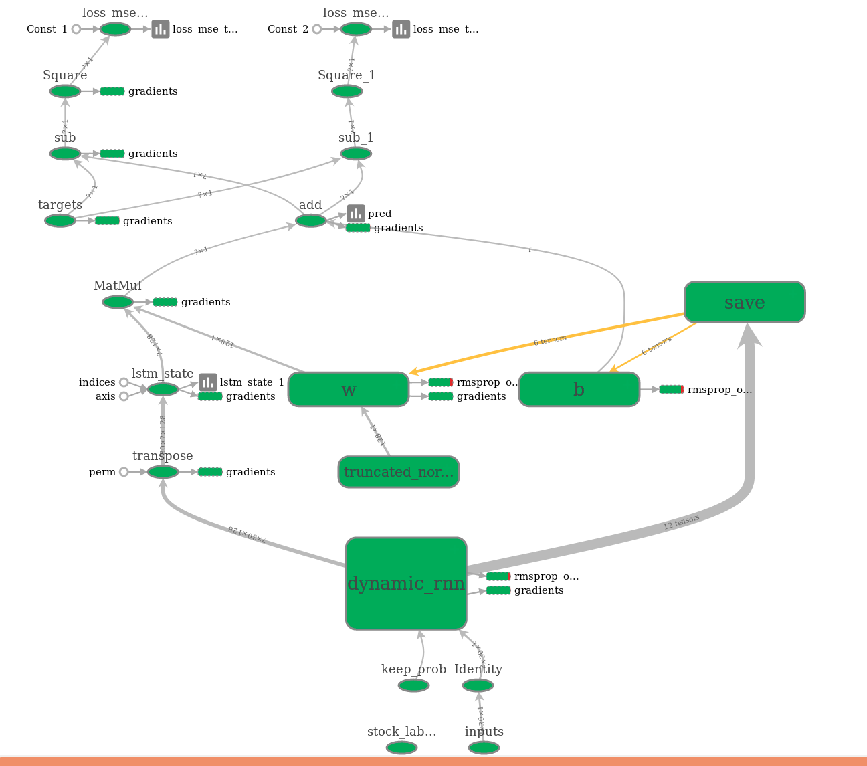
****Figure

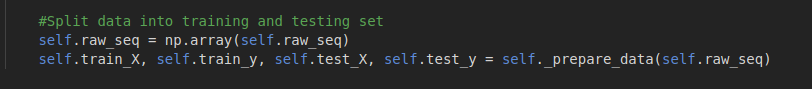
This code below converts the data to array, then splits it into training set and testing set.

### 6.2.6 Long Short Term Memory (LSTM)

Are models that can predict an arbitrary number of steps into the future.

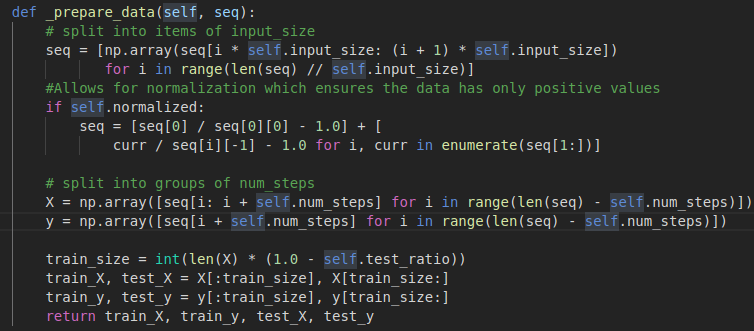
Figure

**train\_X** is for input or feeding into network for training. **train\_y** is expected number of items to be output after training.

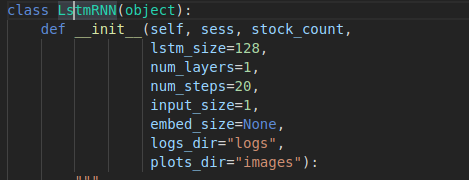
**** Figure

### 6.2.5 Normalizing the data

**Data Normalization** ensures that data has only positive values. It also **reshapes** the training and test data to be in the shape **[input\_size, num\_steps]**. Normalization is a good technique to use when we do not know the distribution of our data.

****Figure

First, we define the Recurrent Neural Network:



Figure

Then, we build a graph that will enable display of graphs and diagrams called histograms which will show the training process and how the biases and weights are changed or lowered to enable accuracy of predictions.

We then train the model that reads stock symbols or data and takes a while to finish.

We type the following commands in the Command Line Interface:

python main.py --stock\_symbol=SP500 --train --input\_size=1 --lstm\_size=128 –max\_epoch=100

**This will activate training of the model.**

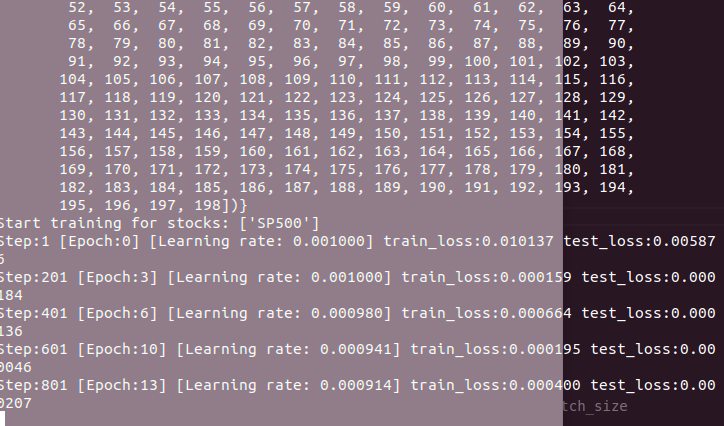
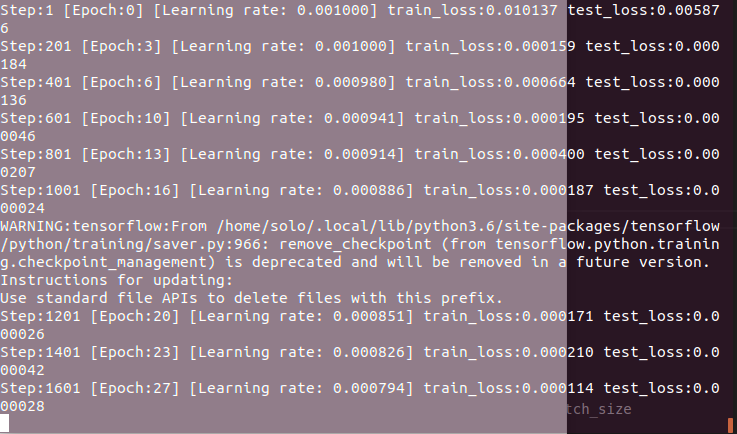
Figure

Figure ****

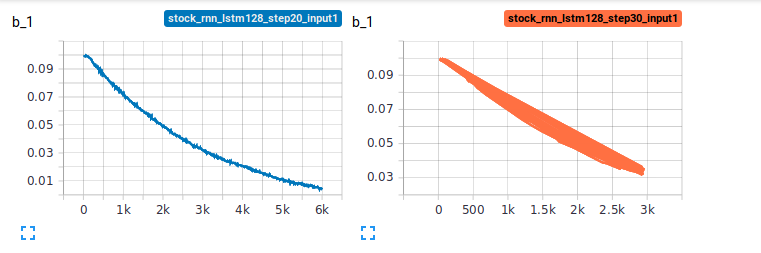
The above diagrams show the training process via the Command Line Interface.

The learning rate reduces because the training loss also reduces.

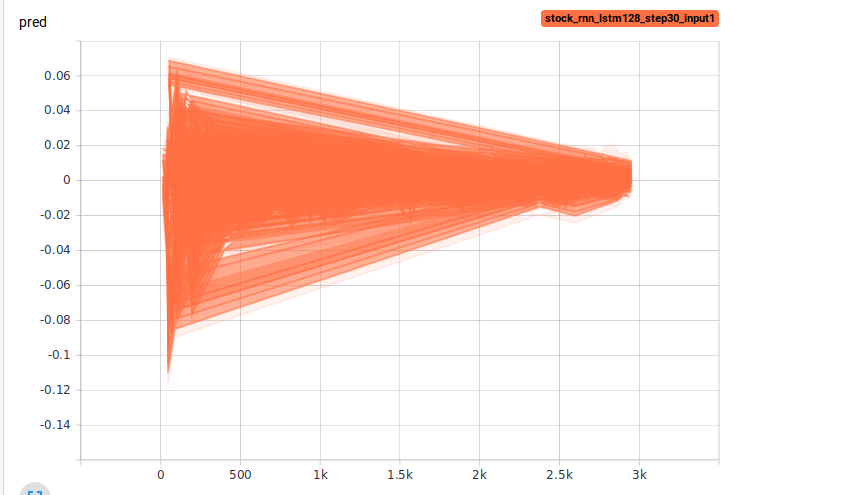
Training loss can be defined as the deviation from the accurate results expected. It reduces as training continues because it nears the expected accuracy of the model.

Learning rate reduces with time to enable fitting of model.

### 6.2.7 Visualizing the predictions

Figure

The bias is changed with time to allow results to be more accurate.

Figure ****

The above is the results prediction. The stock prices are on average around 500 to 1000. It shows how it will project into the future.

# CHAPTER 7: CONCLUSION

## 7.1 Achievements and lessons learnt

## 

Achievements include:

The system was able to train the data.

Python proved its efficiency in terms of being precise and enabling training.

The user interface was responsive and showed various graph outputs.

Lessons learnt:

Coding a recurrent neural network architecture is hard and needs a lot more consultation from people so as to enable progress in the project.

The output is still a challenge thus I should read it or research further on it.

Deep learning is complex and needs more time invested into it.

## 7.2 Conclusions

The project was a success and it taught me how to code artificial intelligence architectures.

## 7.3 Recommendations

The project needs to be assessed in terms of the testing data, because the testing data has not been used in validating the results of the project.

### 7.4.1 REQUIREMENTS

**Software Requirements**

* Python 3.5.x
* TensorFlow 1.16.0
* Numpy 1.15.0
* Matplotlib 2.2.2

**Hardware Requirements**

* 4GB (GigaByte) RAM (Random Access Memory)
* 500 GB Hard Disk.
* 30 GB SSD (Solid State Disk).

# REFERENCES

**[1]**Tsang, P.M., Kwok, P., Choy, S.O., Kwan, R., Ng, S.C., Mak, J., Tsang, J., Koong, K., and Wong, T.L. (2007) “Design and implementation of NN5 for Hong Kong stock price forecasting”, Engineering Applications of Artificial Intelligence, 20, pp. 453-461.

**[2]**Murphy, J.J., (1999) Technical Analysis of the Financial Markets: a Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance.

**[3]** Wu, M.C., Lin, S.Y., and Lin, C.H., (2006) “An effective application of decision tree to stock trading”, Expert Systems with Applications, 31, pp. 270-274.

<https://github.com/jaungiers/LSTM-Neural-Network-for-Time-Series-Prediction>

Reinicke, C. (2018). *Don’t be afraid of artificial intelligence managing your money. Article of Personal Finance.*

**[4]**Kihoro, J., Okango, E. (2014). *Stock market price prediction using artificial neural network: An application to the Kenyan Equity Bank share prices. Journal of Jomo Kenyatta University of Agriculture and Technology.*

**[5]**Al-Radaideh, Q., Alnagi, E. (2013). *Predicting stock prices using data mining techniques. Journal of ResearchGate.'*

**[6]**Bediako, P. (2017). *Long short-term memory recurrent neural network for detecting DDoS flooding attacks within tensorFlow implementation framework. Thesis of Lulea University of Technology.*

**[7]** https://www.tissuegroup.chem.vt.edu/chem-ed/data/gaussian.html

Ganegedara, T. (2018). *Stock market predictions with LSTM in python. Journal of DaaCamp.*

**[8]**http://colah.github.io/posts/2015-08-Understanding-LSTMs/

**[9]** <https://tallyfy.com/business-process-modeling/>

**[10]**<https://searchdatamanagement.techtarget.com/definition/data-modeling>

**[11]**https://www.inflectra.com/ideas/topic/requirements-definition.aspx

**[12]**<https://machinelearningmastery.com/stacked-long-short-term-memory-networks/>

**[13]**<https://en.wikipedia.org/wiki/Functional_requirement>

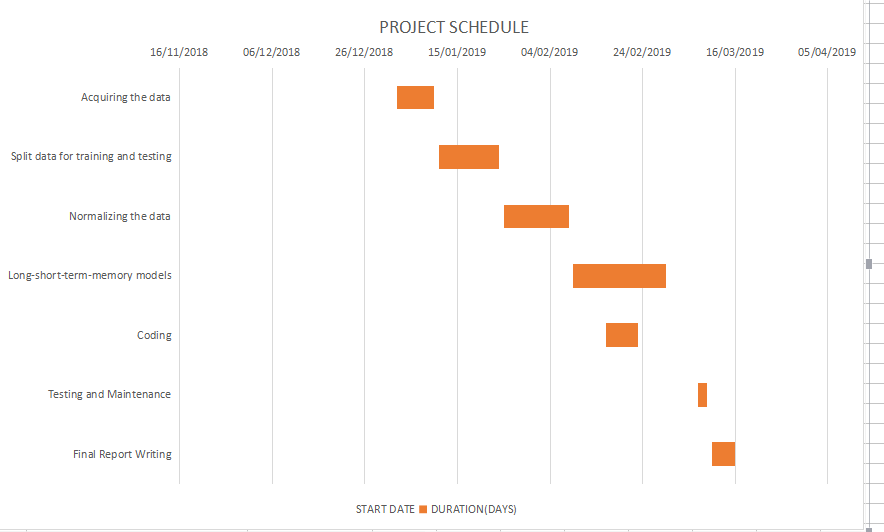
**[14]**<http://www.cs.ccsu.edu/~stan/classes/CS410/Notes16/06-ArchitecturalDesign.html>

**[15]**<https://www.geeksforgeeks.org/software-engineering-user-interface-design/>

**[16]**https://www.sciencedirect.com/science/article/pii/S0167642314005401

# APPENDICES

## APPENIDX 1



Figure

## APPENDIX 2

BUDGET

|  |  |
| --- | --- |
| Internet Cost | 2000 |
| Printing of documentation | 700 |
| Transport for consultation | 1300 |
| Food for the work | 2000 |
| **Total** | **6000** |

Table