UNIVERSITY OF HERTFORDSHIRE

DEPARTMENT OF COMPUTER SCIENCE

BSc Honours Computer Science (Online)

6WCM0029 Computer science Project (COM)

Final Project Report

**Title: Algorithmic Trading**

NAME: SHEHRYAR HASSAN KHAN

STUDENT ID: 20033008

Academic Supervisor: Dr Jean E. Martina

YEAR: 2021

# ABSTRACT:

Since the latest boom in the cryptocurrency market, there has been a rush of retail traders who aim to make a fortune despite of the fact that they are new and inexperienced, this was my case. This report is for the purpose of creating an algorithmic trading system. The focus is mainly to deploy an algorithmic trading system solution that includes **price action** and **technical indicators**. The project is not only restricted to the above goals, but an attempt also to create an advance feature which is for the forecasting of a pair. The strategies deployed can be used for any financial trading markets but for this project the specific market that is targeted is the crypto-currency market with using BINANCE Exchange.

The project aims to help a retail crypto-currency **trader to make better decisions in terms of their due diligence & save screen time** and after the completion of this project, the system is to be made available for other algorithmic traders/developers to further improve as per there needs from the given prototype. This prototype is based on the experience of 2 years of constant trading.

Contents

[ABSTRACT: 2](#_Toc80303986)

[INTRODUCTION CHAPTER: 5](#_Toc80303987)

[FUNCTIONAL REQUIREMENTS 5](#_Toc80303988)

[NON-FUNCTIONAL REQUIREMENTS 5](#_Toc80303989)

[METHODOLGY & OBJECTIVES OF THE PROJECT 5](#_Toc80303990)

[CONTEXT OF CHAPTERS 6](#_Toc80303991)

[LITERATURE REVIEW (BACKGROUND CHAPTER): 7](#_Toc80303992)

[CONSIDERATIONS FOR LEGAL, SOCIAL, ETHICAL & PROFESSIONAL ISSUES 9](#_Toc80303993)

[RISK 10](#_Toc80303994)

[ANALYSIS: 10](#_Toc80303995)

[Questions (Litt review) 11](#_Toc80303996)

[DESGIN 11](#_Toc80303997)

[DATABASE: 11](#_Toc80303998)

[BINANCE API: 11](#_Toc80303999)

[TECHNICAL ANAYSIS: 12](#_Toc80304000)

[Whale SEARCH API: 12](#_Toc80304001)

[FORCASTING ALGORITHM: 12](#_Toc80304002)

[NOTIFICATION TO USER 12](#_Toc80304003)

[SOFTWARE DEVELOPMENT & TESTING 12](#_Toc80304004)

[Development Environment: 12](#_Toc80304005)

[Approach to version control: 12](#_Toc80304006)

[Discussion on decisions & analysis of DATABASE, System structure & issues: 12](#_Toc80304007)

[DATABASE: 13](#_Toc80304008)

[STRUCTURE OF SYSTEM: 13](#_Toc80304009)

[BINANCE-API management: 13](#_Toc80304010)

[Technical Analysis: 14](#_Toc80304011)

[Data handling: 14](#_Toc80304012)

[Machine Learning / Deep learning: 14](#_Toc80304013)

[WHALE API: 15](#_Toc80304014)

[Version control: 15](#_Toc80304015)

[Uncertainties’ and resolvents of project: 15](#_Toc80304016)

[Software tools: 15](#_Toc80304017)

[PANDAS: 16](#_Toc80304018)

[PANDAS\_TA: 16](#_Toc80304019)

[SQL PYTHON CONNECTOR: 16](#_Toc80304020)

[YAHOO FINANCE: 16](#_Toc80304021)

[TENSORFLOW: 16](#_Toc80304022)

[MATPLOTLIB: 16](#_Toc80304023)

[BINANCE API: 16](#_Toc80304024)

[LOGGER: 16](#_Toc80304025)

[JSON: 16](#_Toc80304026)

[MySQL workbench: 17](#_Toc80304027)

[WEBSOCKETS: 17](#_Toc80304028)

[PUSHBULLET: 17](#_Toc80304029)

[WHALE ALERT API: 17](#_Toc80304030)

[PYSIMPLEGUI: 17](#_Toc80304031)

[PYINSTALLER: 17](#_Toc80304032)

[Intermediate results: 17](#_Toc80304033)

[Strategy for testing Program: 17](#_Toc80304034)

[EVALUATION FOR RESEARCH QUESTIONS: 18](#_Toc80304035)

[LOSS FUNCTION MEAN SQUARED ERROR: 18](#_Toc80304036)

[OPTIMIZER FUNCTION ADAM: 18](#_Toc80304037)

[NUMBER OF EPOCH: 19](#_Toc80304038)

[ACTIVATION: 19](#_Toc80304039)

[The Bi-LSTM results: 19](#_Toc80304040)

[CONCLUTION 20](#_Toc80304041)

[GLOSSARY 21](#_Toc80304042)

[APPENDICES 1 22](#_Toc80304043)

[APPENDICES 2 26](#_Toc80304044)

[APPENDICES 3 41](#_Toc80304045)

[READERSHIP 51](#_Toc80304046)

[References 51](#_Toc80304047)

[TIME MANAGEMENT 52](#_Toc80304048)

# INTRODUCTION CHAPTER:

**Algorithmic trading** has been round for more than a decade, but such awareness of leveraging technologies only exists in the minds of the commercial sector. The purpose of this project is to make an algorithmic trading system for the retail traders providing them with tools to improve their due diligence. After the completion of this project, it will be available for other algorithmic traders/developers so they can upgrade as per their needs of the system. An important note is that as a retail trader for 2 years, the development of the project is based on the experience of trading that I have learned from and is developed for my tailored needs.

The that the projects prototype addresses: (**EMOTIONAL / IMPLUSIVE TRADING** & SAVES TIME)

If a trader decides to use this prototype, he or she will be able to use the given **strategies** without sitting on the screen for hours and days but simply getting a notification if the trade conditions are met, this way they can check a chart with a fresh mind and decide to enter or exit a trade. This eliminates the emotional trading factor to boost their due diligence and save time by not sitting in front of screens. Another goal is to encourage users to switch to an algorithmic trading approach so they can use this prototype and built on it after learning programming.

## FUNCTIONAL REQUIREMENTS

1. Connect to account (BINANCE SPOT ACCOUNT)
2. Technical analysis strategies on streaming data
3. Searching Big cryptocurrency transaction on blockchain
4. Sending notifications when trigger trade conditions are met
5. Predict close price of pair (forecasting algorithm)

## NON-FUNCTIONAL REQUIREMENTS

1. Speed of the internet connection
2. Windows version
3. Amount of capital in BINANCE account
4. Operating System of mobile phone

## METHODOLGY & OBJECTIVES OF THE PROJECT

The project methodology chosen is the hybrid approach due to the given nature of the project. It’s better to perfect the stages(objectives) in the respective time range and tailoring of each stage will be required. With that being said:

The objectives for this project to provide algorithmic tools so the application can aid traders in their due diligence with the help of including several **APIs** to integrate them.

First and foremost, to start this project is the design of a basic structure by searching APIs, to generally see how the trading system would work and since the project is on hybrid approach, it will help a lot as this type of project needs a lot of tailoring before in the design of the system as it is built/re-built. This is expected to be completed by 8/06/2021.

One of the most critical part of this project is that the trading system should have a database so operations can be handled with efficiency whenever needed. The intended data to be stored is the prices of symbols/pairs with the help of a class development. This phase will also construct the BINANCE CLASS. Expected to be completed by 3/07/2021.

For the actual signals, there will be two core parts that need to be addressed. First is price action pattern and technical indicator. Each of the concepts will contain specific trigger condition to indicate or notify the user. After that, the above two concepts will be used to generate a trading signal when required, but for this project it is important to give clarity to generate signals so user can be notified. The time allotted is 24/07/2021.

Another functionality to be provided is to check big transactions which could move the markets. This will be done by the user choosing the **blockchain** name and see several transactions. This can help when a market is in a curial point, a user enters the name of blockchain and see the transactions (target BTC & ETH). This will be developed by 2/08/2021

Next up is the advanced functionality, in which it will be referred from the literature review to make a time-series forecasting of the pair. This is important as it will give a user prediction tool of where the trend is going. The literature reviews will also be used to derive research questions and use information that are already defined & tested for building an algorithm. The time allotted till 14/08/2021.

Next it is important again, to design the whole trading system after the above crucial components being developed. This in objective, the focus will be on the flow of operations set by the user and interface is not the top priority. This objective is to be decide whether what shape or form should the prototype finish in. Each major component is to be tested and then the project will be considered completed. After the completion of the project the results are to be documented. This will be done by 19/08/2021.

## CONTEXT OF CHAPTERS

To give a brief context of chapters, the next chapter is solely based on research of machine learning algorithms for the purpose of using that research to aid in the development of the advance functionality. Few methods will be studied through different literature reviews to decide which algorithm is to be used for the development.

The next chapter will take a brief look at Social, ethical, legal, and professional issues regarding this project.

Later in the next chapter, a brief look at the associated risks that this project brings to the table from the start, the development phase and then to the end. This is an important chapter as the risks mentioned also include mitigation plans to make this project finish.

The next 2 chapters which are the analysis, and the design chapter will are dependent on each other. In the Analysis chapter, a lot of factors are covered to persuade this project ranging from the where to get the data to how to use the data. The design chapter will show the flow of data, decisions and more. The design chapter includes the Data flow diagram, and the UML class diagrams.

One of the last and important from all the chapters is then present which is the software development and testing chapter. It includes sections of information which include the development environment, tools of software used, results (intermediate), refinements and more. This chapter is where all the development decisions are proven why there were chosen in the first place.

Then come the ending chapters which are conclusion chapter, which is the reflection of the project being critically evaluated. The Glossary chapter where key terms are mentioned. References/bibliography section, appendices chapter and lastly the time management chapter where the flow of work is broken down in the Gannt chart.

# LITERATURE REVIEW (BACKGROUND CHAPTER):

The reason to research on the Machine learning algorithms is to find the appropriate model to apply in the projects advance functionality. The aim is to leverage such algorithms to give somewhat of an edge of where the trend of a pair might go. The main topic evolves around time-series forecasting and the purpose is to explore with what architecture to achieve the best forecasting.

**2.0 - Combinations of Machine Learning algorithms for Prediction of Trends**

In this section, several methods of machine learning are looked at to determine how to get close to the decent forecast accuracy. Long Short-Term Memory is chosen as a main method as it has the capability to remember inputs which are previously entered and not letting that information to be lost (Ling Qi, Matloob Khushi, Josiah Poon, 2021).

Using Long short-term memory will help in capturing long term dependences. Looking at the data to be forecasted, it comes in a range of dates, ranging from the start price when the pair was being introduced to the current time.

**2.1 – LONG SHORT-TERM MEMORY WITH REINFORCEMENT LEARNING LAYERS**

Following are some key points from a source, LSTM with reinforcement layers for financial trend prediction. Point over here is to see what the authors used as methods of Machine Learning to get an accuracy of a trend on different pairs.

The main way to identify the trend is to get a **correct estimate of medium- and long-term trends**. (Rundo, 2021).

Using LSTM was proven to be the outstanding one compared to the other approaches (decision tress, random forest, SVM, etc.) giving 80 percent plus accuracy (Rundo, 2021).

**2.2 – PREDICTION OF PRICE ACTION THROUGH EVENT-DRIVEN BASED LSTM**

Price action is a term used when technical analysis is done on pairs to decide on whether to place a buy or sell trade. Since patterns of historical data do repeat itself in one way or another, history of these movements of price is key for trend prediction of price in the future. This can be done by technical indicators and historical data.

Technical indicators are vast at number, each one is made with a purpose. Some are made to measure the volume of trades; how volatile the current price is and so on. Since these indicators can somewhat give an accurate prediction of what is going to happen, this can be great to include with LSTM to increase the accuracy and pinpoint events that occur in the market. Events can be classified through technical indicators (Ling Qi, Matloob Khushi, Josiah Poon, 2021).

A handful set of indicators are needed to get excess returns. This is proven in a research where the main objective was to reject Efficient market hypothesis where one-part states excess returns are not possible in the stock market. Technical and fundamental indicators used as a set to return excess returns(Jaideep Singh and Matloob Khushi, 2021)(Sidra Mehtab, Jaydip Sen, Subhasis Dasgupta, 2021)(Wang, 2021)**.**

This approach can be complex and may not return desired forecasting results as there are a lot of conditions with different sets of data.

**2.3 - Recurrent Neural network with LSTM for time series forecasting**

Let’s start off with the definition of recurrent neural networks, which basically are an artificial neural network type, and which are the best fit for time series data as there are many examples found on the internet for its use with correlation time-series or sequential data. This type of Neural network requires to learn with the utilization of data for intended purpose of training. (IBM Cloud Education, 2021)

Any length of inputs can be processed with the recurrent neural network, but the computation is relatively slow, and models of this type can tend to be difficult to train. One of the major drawbacks of using this type of network is that it has trouble in dependencies capturing of long term with respect to a sequence. This is where Long-short term memory comes in hand. As mentioned before, in section 2.0 of this chapter, long short-term memory has the ability of remembering a lot of data from any long range, hence it is a better option to include it with the recurrent neural network.

The machine learning algorithm that will benefit from this section is the supervised learning in which the later stages include Recurrent neural network and long short-term memory.

But why use time-series in the first place to make forecast? The answer is simply that time-series can be used to demonstrate specific patterns, in this case it can help in the identification of the trend. As the literature review mentions in the introduction chapter that time series data, for the past 50 years or more have shown the best possible predictions.

(Adrian Kjaerran, Erling Stray Bugge, Christian Bakke Vennerod , 2021)

The literature review proposed a Long short-term memory architecture framework where the authors constructed functions to reach the desired goal which was to predict the Indian stock market sectors’ companies. The relevant necessary information that might be helpful for the advance functionality have been gone over and extracted below:

The programming language used to construct this architecture was the Python programming language.

The best way to get data is from the YAHOO FINANCE API as it will help extracting historical data like done in the current literature review. The literature review also focuses on scaling columns because it is needed, as mentioned, for the faster convergence of neural networks. This will be done as suggested by the authors, by the SKLEARNS module of the MinMaxScalar function.

A function was created for the purpose of constructing a deep learning model. Where there are 10 arguments needed. Starting off with the length of sequence which implies the data. In context of the literature review, the default length was set to 50 meaning that consecutive 50 time-period selected interval data is selected as input. This value can be changed as per requirement. Then is the number of features which refers to the price features such as the high, open etc. But as noted the target feature is the Close/adjusted close of the symbol/pairs price. Then come the cell type argument which in this case will always be LSTM cells. After that are the number of layers argument implying how many layers to set, then the drop out argument which was for the purpose of training and regularizing it. The higher the value used for this argument, the lower the chance according to the authors of overfitting. After that is the loss argument which refers to the use of function to help evaluate the losses occurred by the deep learning model when the model was going through the training and testing phase (mean absolute error / mean square error / etc). After a few more arguments come the final argument parameter (epochs) which is simply the learning algorithm looping over the set of training data. After the authors used the grid-search method, it was declared that the value that is optimum for epoch is 100.

(Jaydip Sen, Abhishek Dutta, Sidra Mehtab, 2021)

**2.4 – ADVANTAGE OF BIDIRECTIONAL LSTM OVER NORMAL LSTM**

According to another literature review states that the Bidirectional Long short-term memory by almost 38 percent, outperforms the normal long short-term memory in terms error rates reduction by analysis of performance.

Bidirectional Long Short-Term Memory also handles input data differently as mentioned in the literature review: the model will utilize the input data twice for the purpose of training. (Sima Siami-Namini, Neda Tavakoli, Akbar Siami Namin, 2021)

# CONSIDERATIONS FOR LEGAL, SOCIAL, ETHICAL & PROFESSIONAL ISSUES

There are a few concerns regarding ethical issue that is **market manipulation**. In context of this project, if a lot of users start using an automated trading process that would place trades for all users at the same time on one **symbol/pair**. Such scenario can occur, it would be profitable for the users of the project, but other market participants will not benefit from it.

**For example:**

Many user/users with the **volume,** which is significant for a pair, trades automated and synchronized for a condition(s) will defiantly cause market manipulation. That being stated, the final artefact will not be live, it will be a prototype for a single user.

This still doesn’t imply to this project as it is designed for a single user and doesn’t allow to place trades in an automated process.

As per professional duty: A lot of literature reviews will be used to (if required) improve, innovate, and optimize the standards on how they can improve the forecasting algorithm. No Primary research is being taken to involve human participants’ so no ethical issue.

# RISK

The main risk is developing and deploying this trading system in Pakistan. This is a risk as cryptocurrency is not regulated in Pakistan, so there is a chance of the government banning BINANCE as it is used here. To mitigate the risk, if the government does decide to ban BINANCE, there will always be a way to access BINANCE and its API Management. The options include creating a proxy chain in the program or using a VPN. The drawback is the speed of the trading systems operations will be affected.

# ANALYSIS:

As per the project domain “Trading system”, it is important to understand which approach would fit best for the project. Taking an automated approach where everything is automated, the results of the trade may not be up to expectations. The goal is to have an edge. Coding conditions with development being another process can be a complete waste as false signals can destroy trades. Hence the more appropriate approach would be the algorithmic trading approach. The difference can be understood with the following example:

When a condition has met its requirements, the bot/program does not place a trade but instead the bot sends a notification to the user so the user can make the final decision. This way the user has a better chance of winning a trade this way. This approach takes the hybrid method, benefiting from both the algorithms and the users experience.

Putting that into perspective for the project, lets now investigate the strategies’ part. The streaming data will be taken from BINANCE API and for historical data yahoo finance. It is important to make a class with the help of the documentation as it was clear that all the useful data about the account needed to be accessed such as checking account information summary, using the class for streaming of data through BINANCE web sockets and more had to be done too.

As an experienced trader, it is important to mention that the conditions/signals are more accurate on bigger **timeframes** than smaller ones (one day time interval would give better results when compared to seconds or minutes interval). This does not mean that smaller time frames should not be included. The small intervals are essence for testing signals fast.

After the testing, then it is possible to use that method which checks conditions in continuous streaming data. It is expected to make multiple Web socket application methods which would contain separate on message parameter functions as required.

As the above may sound exciting, the above components would be useless without a database. As mentioned, multiple times throughout the report, the project has a range of data complexity. But for the context, the focus will be on the prices storing and retrieving of the streaming symbol/pairs. The database language decided is MySQL, with its workbench program it will be easy to compare results when testing and its connection can be made in python so it can communicate with program.

For the class related to the BINANCE account certain attributes are required to make operations work.

For the forecasting algorithm it has been decided to use the Recurrent neural network with bidirectional LSTMs on time series data as a new beginner in machine learning and the given time allotted.

### Questions (Litt review)

According to the research done on different literature reviews for the purpose of delivering of a forecasting, the most approachable method should be used with respect to the time of the project allotted (**GANT CHART**). The selected approach is using a recurrent neural network with long short-term memory layers on the Historical data.

For simplicity, the data of pairs used in this will be restricted (**ETH/USD | BTC/USD**) for the prototype but will include the process of taking data from yahoo finance.

#### RESEACRH QUESTIONS DERIVED FROM LITERATURE REVIEWS

1. What type of forecasting term on the data with respect to timeframe will yield better results (Short term prediction / Medium term prediction / long term prediction)?

For Reference the terms of predictions’ data to look at:

**Short term prediction:** 5 minutes’ Interval (5m / 5 minutes)

**Medium term prediction:** 1 hours’ Interval (1h / 1 hour)

**Long term prediction:** 1 days’ interval (1d / 1 day)

1. What other factors can affect the forecasting algorithm?

# DESGIN

The trading system is designed as shown in the appendances section image under Design heading. To elaborate further: ­(appendices 1)

### DATABASE:

There is a class for the system database with establishes the connection to MySQL server. This class has methods such as create table in the database, save/read data to the database etc. The database object will be needed in other parts of the system.

### BINANCE API:

Another class is created to connect to the BINANCE account (BINANCE API MANAGEMET). This class has the responsibilities to pull Data requests such as information on withdrawals/deposits history, get open orders, balance etc. The class also has other responsibilities such as functions to trigger trades trade conditions, etc.

The intended process is to save the streaming data from WEBSOCKETS to the database table which can be pulled for technical analysis by converting it to a pandas data frame. Or to use the streaming data for the next title

### TECHNICAL ANAYSIS:

Technical Analysis has two core components to trigger conditions, **price action patterns and technical indicators**. But this depends on the how much time it takes to build one signal. So, for this stage it is decided to find given solutions if available (to get a generic structure) and tailor them to the needs.

The goal with this design is to give a simple but effective algorithmic trading system.

### Whale SEARCH API:

**WHALE transactions** will be developed with the **WHALEALERT API.** This API can help retrieve whale transactions on a specific blockchain. This will be a simple class and may not require a lot of methods.

### FORCASTING ALGORITHM:

Design decided is to take data from Yahoo finance for the 3 terms specified on which the data will be used to test. The model architecture that is going to be developed is the Bidirectional LSTM with 2 pairs’ prices.

### NOTIFICATION TO USER

For this functionality it has been decided to use the **PUSHBULLET** API application which will send a notification to a phone by creating an object and using the API Method.

# SOFTWARE DEVELOPMENT & TESTING

Development screenshots under appendices 2

## Development Environment:

The main development environment is PYCHARM. This IDE is chosen as its useful to have shortcuts and hits while development and can help speed up the process too. Other development environments chosen is the GOOGLE COLLAB which will be used to make the forecasting model.

## Approach to version control:

An attempt with GITHUB with GIT.

## Discussion on decisions & analysis of DATABASE, System structure & issues:

First and foremost, to start this project it is important to find an exchange and register a verified account on which the trading system can connect to, pull necessary information, and send streaming commands. For this the exchange selected is BINANCE exchange as they have a well-maintained Application Programming Interface documentation, and the project can leverage the API to its tailor needs. For example, look-up balance in account, etc.

Then to get other required API keys for PUSHBULLET & WHALE ALERT.

### DATABASE:

As specified in the objectives of this report, one of the curial parts is the DATABASE of the trading system. After a few dos and don’ts, it was understood that there is a big complexity of data that is involved with this project. So, it was important to know what type of data was too be stored. At the starting point it was not sure of what data would be saved. The saved data is the streaming data.

For this purpose, it was decided that there would be a class for interacting with the backend MySQL server & MySQL Workbench and whenever there is a requirement of data to be saved, the database object can be created and called whenever. This meant that the Database class had to be flexible in nature so it could adapt to the requirements, this included firstly saving/retrieving intended data, transforming/converting data (Database type to pandas’ data frame), creating specific query methods, etc.

The main intended data of pair to be used is the **high, open, close & low.**

Due to the short time of the project only a few pairs are chosen to build the prototype upon.

### STRUCTURE OF SYSTEM:

The system structure of this project is heavily dependent on classes as the classes are used one in other to complete functionalities. Such as all data was made linkable so it could be used all-over whenever needed.

In terms of flow of operations of the system structure, that is decided to let the user choose the options provided. This task is let for the end this dependant on the final functions to be built. Some of the intended functions will be used to make a process which are the curial part of the trading system. As far as the flow of operations, that are defined in the main program, will be in the hands of the user.

Separate tasks/processes are required to make a system automated process. For this project it was oriented around gather pieces of information and finally constructing something useful for the system. For this case a few things were noted & addressed:

### BINANCE-API management:

The class was constructed with the purpose to address the requirements needed for the processes of the project. The account type selected was spot as there is negligible **leverage**.

After a few legs pulls with the BINANCE API, the working is understood of how the bot will function in relation with the API, some will require AYSINIO connection and mostly functions will require streaming data through Web sockets. But the testing with bot integration will be done through a separate API, an API which will help us download specific data regarding start/end date & time interval of the pair (YAHOO FINANCE).

In the process of streaming data, a lot of JSON data with the help of JSON library was used for operations like the spot account retrieving status and account information (used with an AYSINIO), streaming of data for technical analysis etc. Json library was key for getting the intended data in the first place.

### Technical Analysis:

For the patterns, conditions need to be hard coded; these solutions are a bit of complex math of understanding the differences of the candlestick price data (High/open/close/low).

As known with a help of patterns and technical indicator it can be used to make trigger trade conditions. The task now was to know how to perform it in code. At first the requirement selected was the price action patterns. After looking and experimenting with the BINANCE class, it was understood that there should be a way to back-test the pattern to see how many times the signals occur and then later include it in the BINANCE class’s function where it is required to be used.

To tackle this task another class was created to identify the **TECHNICAL SIGNALS**.

The class when initialized requires a panda’s data-frame which would be used in the class functions for the desired need.

To test, a few functions were created to check whether they work or not, these functions can be seen as back test.

Most of the work price action work is from the below links (used as a generic structure) which was later integrated as tailored:

Doji: <https://drive.google.com/file/d/1rl3LEAtPUL3FWaGsawzvG_K-blmLfhks/view>

Engulfing: <https://drive.google.com/file/d/112PGLxL4TpoR_DeyCzMi0SgegipSyHWq/view>

### Data handling:

As mentioned before, there is a lot of complexity in the handling of data for this project. It had to be addressed. The following are some of the major needs that were addressed.

Data from MySQL needed to be addressed as the data needed to be saved, retrieved and all data from tables and returning data in a format of a Data-frame.

Data from BINANCE account needed to be chosen to select the correct relevant information, this was extracted from the JSON messages.

JSON Library was curial as most of the APIs sent data in JSON datatype. For specific functions the datatype was used to retrieve specific data. For BINANCE API & YAHOO finance, it got the pair price which was later either converted into a PANDAS DATAFRAME or saved to a database. For WHALE ALERT API it was used to extract transactions.

For the database object, its responsibilities were to either create a table if not existed. Saving data. Converting data types to required types (EXAMPLE: DATAFRAME for Technical analysis and csv for machine learning data)

### Machine Learning / Deep learning:

For this part of the project, it was decided to persuade this on GOOGLE COLLABORATION as at first the library that is intended to use (TENSORFLOW) has separate dependencies.

For simplicity on pair was chosen to build the forecasting model (ETHUSD & BTCUSD). The data selected was from the functions created earlier where the YAHOO FINANCE API was used to download the intended pair price history and later exported into a CSV format which was uploaded on GOOGLE COLLABARATION through google drive.

### WHALE API:

Another class is built which is a simple class with a powerful response. Which has one symbol as an argument and one main function which will search for a transaction on a blockchain.

### Version control:

This task attempt was delegated to git application with the GITHUB website.

## Uncertainties’ and resolvents of project:

The major uncertainty was regarding the advance functionality with the Library of TENSORFLOW. The problem came when installing the library/tool in the projects IDE where this tool uses lower versions of NUMPY (common tool to help in faster and efficient workings). The problem arises when it was installed, directly effecting other libraries (PANDAS, PANDAS\_TA) which most options were dependant on. To resolve this, the TENSORFLOW library was uninstalled and the for the purpose of not leaving the advance functionality, the objective was moved to a different IDE (google collaboration). The only problem now was the integration of the model with the program, but the major issue was resolved.

One major uncertainty was discovered when rechecking the forecasting algorithm after reconnecting to the runtime. The model that was trained had lost the training weights and wouldn’t give a decent prediction when compared to the correct one. To resolve this, a simple method was used which saved the models’ weights so the model wouldn’t lose the weights if the runtime got disconnected.

Another problem faced was with the retrieving of information from the BINANCE servers sometimes as it would time out. After a few days of debugging a simple solution was found that to make the program work with the servers, a simple resynchronizing of the laptop was needed from the settings and the program component was connected.

The streaming method pose the lack of inefficiently and can be a problem as the WEBSOCKET uses the ‘Run forever’ method making it an infinite loop. These methods are the methods for technical analysis so the final prototype will have this bug. Only one strategy can be running without interruption. This opens a way to develop two prototype solutions where one is with the interface which will be inefficient and the other is that the future user must download dependencies (if required) and run from command line or from IDE itself.

## Software tools:

The software tools used where the soul reason why python was chosen in the first place to develop the project. Python has several libraries/tools which aid to any development. Majority of the developers have developed Application programming interfaces (APIs).

### PANDAS:

Mainly used to form a data structure for handling data. This can be seen with in several examples, such as taking data and forming it into a data frame. The data frames were later used for other options (analysis to find signals).

Another use was for the data needed to train the machine learning algorithm. This was done by converting the data frame into a csv format.

### PANDAS\_TA:

Although there were many technical indicator libraries. It was decided to take this tool as the project was already using the PANDAS library.

### SQL PYTHON CONNECTOR:

This was installed for the purpose of creating the backend of data handling. The approach was to create a class whose object can be called whenever needed to save the data and retrieve the data. To confirm operations, MySQL workbench was used.

### YAHOO FINANCE:

This was a critical as data was needed was historical time series data for the intended purpose of using this data for the machine learning algorithm. But most importantly for the historical data.

### TENSORFLOW:

This was the library used for the purpose of building the advance functionality. TENSORFLOW is widely known to help developers make and train models of machine learning. A list comprehensive tools in this library help to deploy machine learning algorithms with robustness which enables research deployment all with building them with ease.

### MATPLOTLIB:

This library is useful for the visualization when plotting data to better understand the relation between data points. This was mainly used to plot the data for the forecasting of the pair which was used in the advanced functionality.

### BINANCE API:

With the help of a well-documented API, a lot of work custom work was done. This was mainly used with the generated API keys from BINANCE API management to send and retrieve intended data to make the whole project work.

### LOGGER:

Not as important as other tools used in this project. But it was used in some parts to help the process of debugging.

### JSON:

One of the important libraries used in terms of returning data from APIs. This was used to understand and extract important information from APIs of yahoo finance and BINANCE API. The main use of this was with WEBSOCKETS in returning intended data from the API which then later was used to scrap the intended data for respective purposes. The uses of this can be seen in BINANCE class for retrieving account information, streaming data of pairs, from the whale search class to get the information required to present to the user.

### MySQL workbench:

This software tool was intended to be used for two purposes which link to the objective of the database. Firstly, the main reason was to connect the backend operations of the program (database object). Second, was to test if the operations were being performed correctly of the database class.

### WEBSOCKETS:

WEBSOCKETS is another important library. The intended purpose was for the streaming of data from the BINANCE API. This library helped to complete the tasks of saving data to the database, using data for the technical analysis (price action and technical indicators) just by the help of using the options of built-in operation methods (User web sockets, web socket application etc.).

### PUSHBULLET:

Push-bullet application is used for the purpose of sending notifications to the user. This was done by installing Push-Bullet mobile application (as the notification is appear there) and by the help of installing the push-bullet library (PUSHBULLET API).

As soon as the condition/conditions were met, with the help of Push-bullet objects methods, a notification was sent

### WHALE ALERT API:

This was used for creating a class that would retrieve transactions of a whales on the given blockchain.

### PYSIMPLEGUI:

This library was used to build a simple interface as it doesn’t require complex work like other python front-end libraries.

### PYINSTALLER:

This library is the most important one as this will help in the conversion of the python program/script into a stand-alone application / provide an executable file.

## Intermediate results:

For the BINANCE class, each functionality worked correctly, the only problem faced was in the streaming data methods where Web socket thread would eventually create a deadlock hence it had to be commented out.

The system database class was one of the most complex hence for the prototype only two symbol/pairs were built in their context. All operations that were intended were working as required.

After creating an executable file, there was an issue where the application won’t run.

## Strategy for testing Program:

Appendices 2

There are a few components that are required to be tested:

It is not a surprise that one of the major testings of this algorithmic trading system is back-testing. This allows the better interpretation on understanding how well the constructed trading system responds. This was done with the PANDAS library to set conditions.

The main approach was to check the response of the price action patterns on smaller timeframes as the testing could be done in seconds which has saved a lot of time. The image in the Appendices show how the data was returned for testing (under the false positives and back testing given in the appendices). The technical price action patterns returned a list which could later be used to get the last index. So, the image where there is a list of 0s, 1s and 2s, they represent trends of the pattern. 1 is for bearish (downward signal), 2 is for bullish (upwards) & 0 indicates no trend. If the signals functions were working on the smaller timeframe that meant they would be working for larger timeframes too.

Next was the testing the Whale alert API by calling its methods to see transactions. A detailed look is in the appendices under **WHALE ALERT FOR TRANSACTIONS CALLS.**  For simplicity for usage, the whale class took only one argument to be initialized and would return 50 transactions.

# EVALUATION FOR RESEARCH QUESTIONS:

Appendices 3

The main goal was to build a suitable deep learning model for the forecasting algorithm:

The prediction results were better for the Bi-LSTM in the medium-term trend using mean squared error (loss function).

Standards were fixed in terms of the low-level decisions after looking and extracting information from the literature review. These are as follows:

|  |  |
| --- | --- |
| parameters | standard for comparing |
| LOSS FUNCTION | MEAN SQUARED ERROR |
| optimization | ADAM |
| number of epoch | 100 |
| activation | LINEAR |

## LOSS FUNCTION MEAN SQUARED ERROR:

As the algorithm was targeting a problem of regression, this is known to be the standard to be used for such problems. Mean squared Logarithmic Error function can also be considered only if the data is too much for the regression algorithm to handle. After experimenting, the loss was less for the chosen loss function and the results were far better with the MEAN squared error (comparing to mean absolute error [regression problems]).

## OPTIMIZER FUNCTION ADAM:

Taken from the literature review based on prediction of prices based on the Indian stock market.

## NUMBER OF EPOCH:

The number of epoch was a fixed standard taken from the literature review based on prediction of prices based on the Indian stock market.

## ACTIVATION:

As a timeseries problem, it was best to solve it with a linear regressor.

The main difference was dependent upon the factor that the validation and training split of data. The screenshots in the appendices are of the results of 50-50 split as they yielded better results.

### The Bi-LSTM results:

AVERAGE TIME TO TRAIN: 12 minutes

It was clear that the medium-term prediction was the best term prediction with the pair of ETHUSD or BTCUSD as seen in the snippets in the Appendices 3.

According to the answers from the derived research questions:

1. What type of forecasting on the data with respect to timeframe will yield better results (Short term prediction / Medium term prediction / long term prediction)?

The best result came from the medium-term prediction where data that was selected was of the one-hour time frame on the ETH/USD & BTC/USD pair. The rest of the two terms didn’t return better results, the worse out of the three was the long term of the ETH/USD while BTC/USD still gave a better result. For short-term trend results were not that bad for ETH/USD but worse for BTC/USD when compared.

The plotted graphs show the predictions in the appendices section under **MACHINE LEARNING FORECASTING RESULTS. (results of mean squared error)**

1. What other factors can affect the forecasting algorithm?

The noted factors that can affect the prediction are firstly the data in terms of which term to predict.

Secondly changing of hyper parameters specially between the validation and testing spilt of data, in the initial stages the split was 20 to 80 but as a standard it was decided for a fair play that 50 to 50 splits would be set a standard. Better results were produced on a 50-50 split between validation and training.

Epoch was also a factor, as a standard, 100 was the default limit which was taken from a literature review.

One main factor which was noticed from time to time was rerunning the model after reconnecting. All the models training was lost and had to be trained again (if weight/model not saved).

One factor later identified is that in times of uncertainty where the market is volatile the algorithm doesn’t show a good prediction.

Lastly the loss methods have a major affect the prediction of the pair. It was seen that the mean squared error had better predictions when compared to mean absolute error.

# CONCLUTION

It is important to keep in mind that the prototype was built with the existing trading knowledge and so the project could be carried on by other algorithmic developers for the retail level.

All the objectives were met in the given timeline. The forecasting has been completed but the integration couldn’t happen as the integration isn’t possible due to the use of the algorithm’s library and different IDEs.

All the research questions were answered with the best of ability with respect to time. For screenshots of this section: Appendices 3.

Apart from that, the prototype gives users an opportunity to use tools as mentioned in the objectives. (Deploying strategies given, retrieving account information, searching whale transactions, notifying trader). The only inconvenience is when user uses the forecasting algorithm, they have use GOOGLE COLLAB to access the algorithm, provide a dataset and run the code lines themselves.

There are 2 final prototypes, one is an interface and the second is script. Unfortunately, the executable files couldn’t be built by pyinstaller so they are not included in the final submission.

Furthermore, to enhance this project prototype it is highly recommended to use bidirectional LSTMs for other forecasting algorithms with mean squared error, include more recognized price action patterns or include other key components such as open of data pair if further developers who wish to customise as per needs.

Another further work needed to save laptop usage is to run this program with the help of hosting services such as Amazon web services.

Lastly as there was less time, a near to perfect interface couldn’t be built as the focus was around the operations for algorithmic trading. Hence in the further development, an interface with meaningful visualization and which doesn’t end after pressing a button is recommended.

# GLOSSARY

**TECHNICAL ANALYSIS:** term used to make an analysis by looking at price action patterns and technical indicators.

**Price action patterns:** Patterns of the candlestick data that can indicate market directions.

**Candlestick:** description of price at time frame length. Includes High, open, close, low.

**Technical indicators:** based on patterns they produce signals.

**Technical signals / Strategies:** conditions that are derived from price action patterns and technical indicators to make trade decisions.

**Algorithmic trading:** pre-programmed trading conditions.

**Emotional / impulsive trading:** When a trader allows feelings take control of their trading decisions.

**API:** Application Programming Interface.

**BINANCE:** World’s largest & most developed crypto-currency exchange.

**Blockchain:** Next generation technology where the information can’t be changed or hacked due to its complexity, considered ‘bigger than the internet technology’.

**Pair / Symbol:** trading instruments which include 2 trading securities (ETHUSD / BTCUSD)

**BTCUSD & ETHUSD:** Bitcoin against the U.S dollar, Ethereum against the U.S dollar.

**Market manipulation:** Certain moves in a market that are caused by volume which is profitable for the planners and unethical.

**Volume:** period in timemeasurement of trades.

**Time frame:** Simply an area of time on chart options in context to candlesticks. Examples are 1 day, 1 week, 1 month etc.

**Whale:** A person or an entity that has a large amount of a crypto-currency pair.

**High / open / close / Low:** The body shape of a candlestick.

**Leverage:** Allows traders to use big trading positions on accounts.

**RNN:** Recurrent Neural Network.

**LSTM:** Long short-term memory.

**BI-LSTM:** Bidirectional Long Short-term memory.

# APPENDICES 1

**INITIAL DESGIN OF THE SYSTEM:**

**Diagram

Description automatically generated**

**UPGRADED DESGIN OF THE SYSTEM:**

Diagram

Description automatically generated

**FINAL DESGIN OF SYSTEM:**

Diagram

Description automatically generated

**CLASS DIAGRAMS:**

**Binance:**

Text

Description automatically generated

**System database:**

Text, table

Description automatically generated

**Signal (technical analysis class):**

Text

Description automatically generated

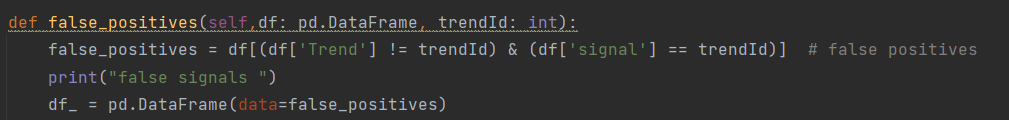
**WHALE ALERT (WHALESEARCH CLASS):**

Graphical user interface, text, application, table

Description automatically generated

# APPENDICES 2

**FASLE POSITIVES & BACKTESTING:**



Graphical user interface, text

Description automatically generated

A picture containing text, black

Description automatically generated

**TESTING SIGNALS ON SMALLER TIMEFRAMES:**



**ENGULFING CANDLES PRICE ACTION BACKTEST (TIMEFRAME = 1 DAY):**

Text

Description automatically generated



A picture containing table

Description automatically generated

Text

Description automatically generated

**DOJI CANDLES PRICE ACTION BACKTEST (TIMEFRAME = 1 DAY):**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**WEBSOCKETS USEAGE:**

Text

Description automatically generated

Text

Description automatically generated

**Continuation of the above screenshot method:**

Text

Description automatically generated

**DEADLOCK BT WEBSOCKECT THREADS SOLUTION:**

Text

Description automatically generated

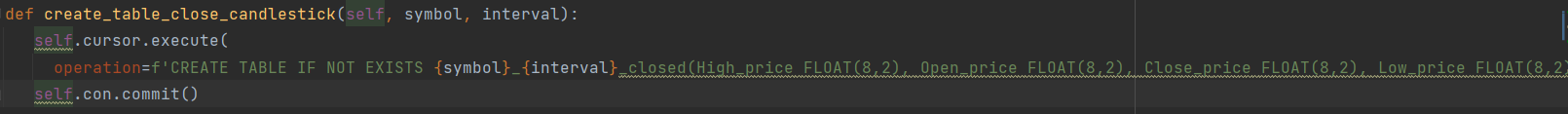
**DATABASE COMPONENT:**

Text

Description automatically generated

Text

Description automatically generated



Text

Description automatically generated

**BACKEND MYSQL CHECKING:**

Table

Description automatically generated with medium confidence

Graphical user interface, table

Description automatically generated

**WHALE SEARCH CLASS:**

Text

Description automatically generated

**TECHNICAL ANALYSIS CLASS:**

Text

Description automatically generated

**DEVELOPING STRATEGIES**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

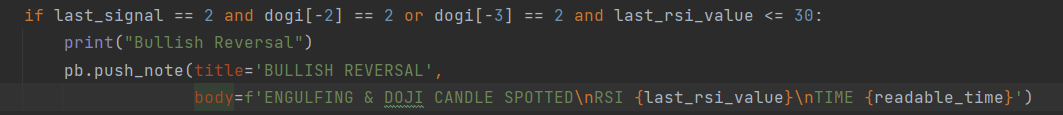
Text

Description automatically generated

**PUSHBULLET NOTIFICATION ON SMARTPHONE:**

Text

Description automatically generated



Graphical user interface, text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

**EXTRACTING DATA WITH JSON:**

Text

Description automatically generated

Text

Description automatically generated

**BINANCE DOUCUMENATION:**

Graphical user interface, text, application

Description automatically generated

**GIT:**

Text

Description automatically generated

**WHALE ALERT FOR TRANSACTIONS CALLS**

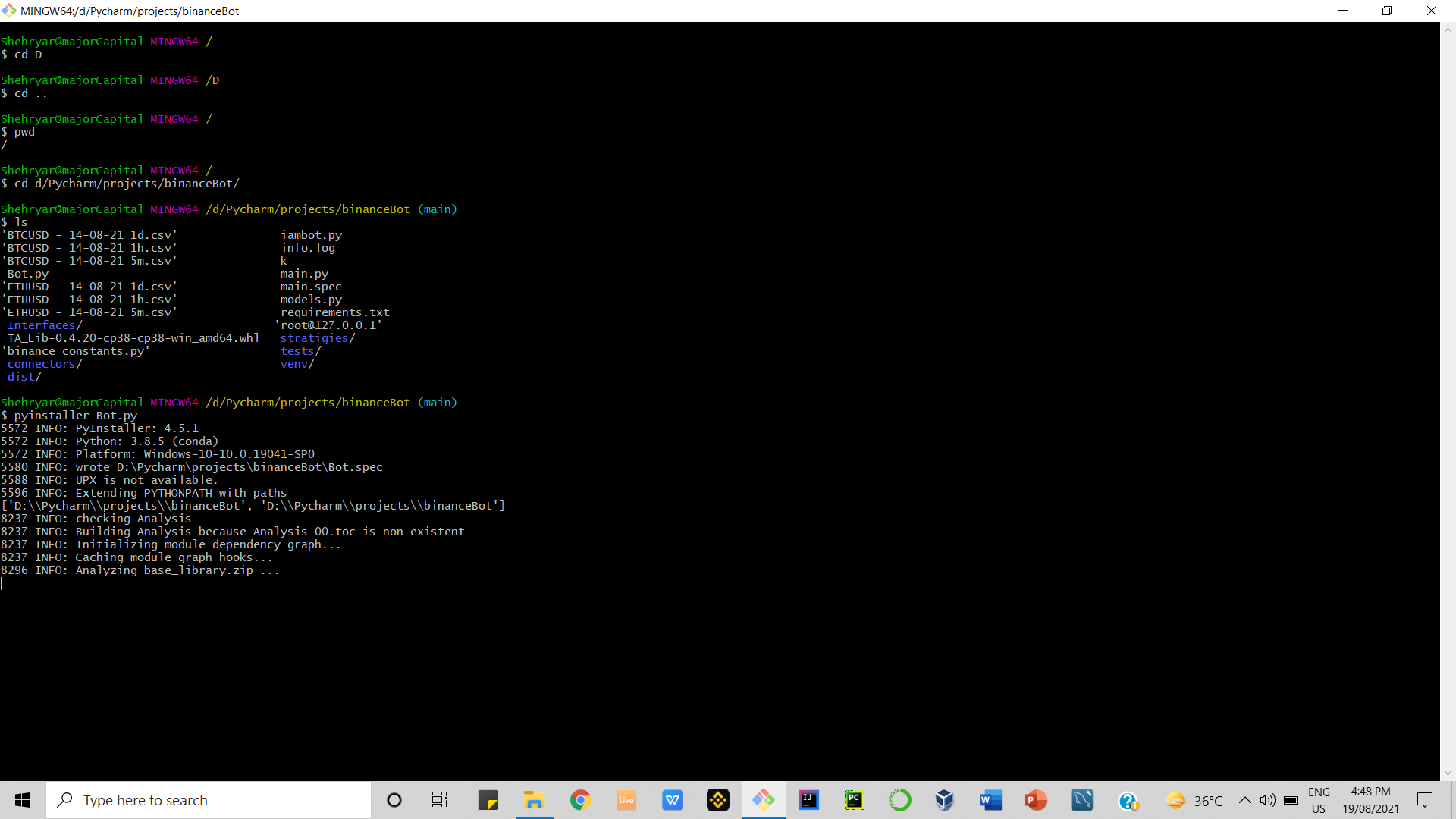
Text

Description automatically generated

**CONVERTING PYTHON PROGRAM INTO EXECUTABLE STAND-ALONE APPLICATION:**

Text

Description automatically generated



# APPENDICES 3

**IMAGE FROM CHOOSEN LITERATURE REVIEW:**

Diagram

Description automatically generated

**MODEL DEVELOPMENT:**

Graphical user interface, text, application, email

Description automatically generated

**LOW LEVEL FUNCTIONS & TRAINING:**

Text

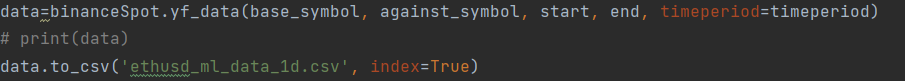
Description automatically generated

Text

Description automatically generated

**MACHINE LEARNING GETTING & LOADING DATA:**

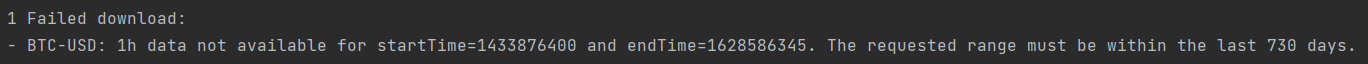
**For Long term trend:**



Graphical user interface, text, application

Description automatically generated

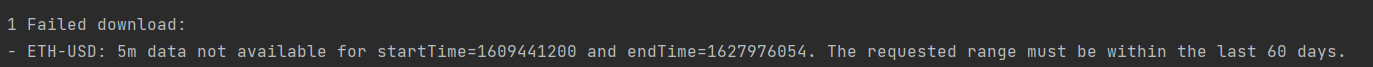
For medium term trend:



A screenshot of a computer

Description automatically generated with medium confidence

For short term trend:



Text

Description automatically generated

CSV type data successfully stored:

Text

Description automatically generated

Text

Description automatically generated

**MACHINE LEARNING FORECASTING RESULTS:**

**MACHINE LEARNING (BiLSTM) FORECASTING RESULTS (LONG TERM-TREND):**

Chart, line chart

Description automatically generated

Chart, line chart, histogram

Description automatically generated

**MACHINE LEARNING (Bi-LSTM) FORECASTING CHANGING DATA RESULTS (MEDIUM TERM-TREND):**

Chart, line chart, histogram

Description automatically generated

Chart, line chart

Description automatically generated

**MACHINE LEARNING (BiLSTM) FORECASTING CHANGING DATA RESULTS (SHORT TERM-TREND):**

Chart, histogram

Description automatically generated

Chart, scatter chart

Description automatically generated

**MEAN ABSOLUTE ERROR PERDICTIONS**

Chart, line chart, scatter chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart

Description automatically generated

Chart, histogram

Description automatically generated

# READERSHIP

It is expected that a person should at least have some programming knowledge and experience or a computer science degree in process.

# References

Adrian Kjaerran, Erling Stray Bugge, Christian Bakke Vennerod . (2021, 07 25). *Long Short-term Memory RNN.* Retrieved from https://arxiv.org/pdf/2105.06756.pdf

Desjardins, J. (2021, 03 24). *12 Types of Technical Indicators Used by Stock Traders.* Retrieved from https://www.visualcapitalist.com/12-types-technical-indicators-stocks/

IBM Cloud Education. (2021, 07 25). *Recurrent Neural Networks.* Retrieved from https://www.ibm.com/cloud/learn/recurrent-neural-networks

Jaideep Singh and Matloob Khushi. (2021, 03 29). *Feature learning for stock price prediction shows a significant role of analyst rating.* Retrieved from https://arxiv.org/ftp/arxiv/papers/2103/2103.09106.pdf

Jaydip Sen, Abhishek Dutta, Sidra Mehtab. (2021, 07 25). *Profitability Analysis in Stock Investment Using an.* Retrieved from https://arxiv.org/ftp/arxiv/papers/2104/2104.06259.pdf

Ling Qi, Matloob Khushi, Josiah Poon. (2021, 03 23). *Event-Driven LSTM For Forex Price Prediction.* Retrieved from https://arxiv.org/ftp/arxiv/papers/2102/2102.01499.pdf

Rundo, F. (2021, 03 21). *Deep LSTM with Reinforcement Learning Layer for Financial Trend Prediction in FX High Frequency Trading Systems.* Retrieved from https://www.mdpi.com/2076-3417/9/20/4460

Sidra Mehtab, Jaydip Sen, Subhasis Dasgupta. (2021, 03 29). *Robust Analysis of Stock Price Time Series Using.* Retrieved from https://arxiv.org/ftp/arxiv/papers/2011/2011.08011.pdf

Sima Siami-Namini, Neda Tavakoli, Akbar Siami Namin. (2021, 08 03). *A Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM, and BiLSTM.* Retrieved from https://arxiv.org/pdf/1911.09512.pdf

Wang, L.-X. (2021, 03 29). *Dynamical Models of Stock Prices Based on.* Retrieved from https://arxiv.org/ftp/arxiv/papers/1401/1401.1888.pdf

# TIME MANAGEMENT

BREAK DOWN:

Table

Description automatically generated