# STOCK INDICATOR SCANNER CUSTOMIZATION TOOL

BY

CHEAH SHING DHEE

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2022

# REPORT STATUS DECLARATION FORM

**Title**: ______Stock Indicator Scanner Customization Tool__________

**Academic Session**: ___JUNE_2022_

I __________CHEAH_SHING_DHEE______________________

**(CAPITAL LETTER)**

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature) (Supervisor's signature)

**Address**:

_No 80, Persiaran Bekor 9,____

_Taman Pertama____________

__30100, Ipoh, Perak________

__ Ts Dr Ku Chin Soon ___

Supervisor's name

**Date**: ___5_September_2022__

**Date**: ___5_September_2022_

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: 5 September 2022

**SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS**

It is hereby certified that ***CHEAH SHING DHEE*** (ID No: __***18ACB05004***__*)* has completed this final year project entitled "Stock Indicator Scanner Customization Tool" under the supervision of Ts Dr Ku Chin Soon (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

________ ________

*CHEAH SHING DHEE*

# DECLARATION OF ORIGINALITY

I declare that this report entitled "**STOCK INDICATOR SCANNER CUSTOMIZATION TOOL**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ____________________________

Name : ___CHEAH SHING DHEE__

Date : ______5_September_2022___________

# ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors Mr Ku Chin Soon who has given me this bright opportunity to engage in a time-series machine learning project. It is my first step to establish a career in data analysis and machine learning field. A million thanks to you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

# ABSTRACT

Nowadays, stock trend prediction has become a famous topic among financial analysts and computer scientists. As developing a model that could accurately predict the directional changes of stock prices will bring huge benefits to investors. The prediction model can act as an assisting tool to helps investors in making trading decisions. Analysing historical prices to make stock prediction is a traditional approach called technical analysis. Technical analysis required investors to identify the trading opportunities of the stock by analysing the historical price manually. To make more accurate prediction in the stock movement, analysis of large amount of historical data is required; however, analysing large amount of historical prices manually by investors is time-consuming and complicated, and this is where machine learning come into handy. Machine learning algorithms able to analyse and extract useful pattern from the huge amounts of data in a short period of time. However, lack of flexibility and dynamicity of the prediction model is the major limit in most previous works. Allowing investors to customize what technical indicators to use in the prediction model is an important factor as experienced investors would know what technical indicators are useful for certain types of stocks.

Therefore, this project is proposed to build a stock analysis website that allow investors to customize the inputs of the stock prediction model. This is mainly to combine the power of the machine learning algorithms with the domain knowledge of investors to make more meaningful prediction rather than supply random technical indicators as input of the model like previous work. Since stock prediction problem is considered a time-series problem, long short-term memory (LSTM) will be used as the algorithm in the proposed model, and LSTM has been proven to be good stock prediction model in previous works. LSTM is the improvised version of the recurrent neural network (RNN). Instead of including all past information in the model like RNN, LSTM has the power to include only the useful information by utilizing the gates such as forget gate. This will not only increase the generalizing capability of LSTM model but also solve the vanishing gradient issue in RNN.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS

| | |
|---|---|
| + | Plus Sign |
| & | And Sign |

# LIST OF ABBREVIATIONS

| | |
|---|---|
| *GA* | Genetic Algorithm |
| *ANN* | Artificial Neural Network |
| *LSTM* | Long Short-Term Memory |
| *SVM* | Support Vector Machine |
| *EMH* | Efficient Market Hypothesis |
| *AMG* | Adaptive Market Hypothesis |
| *SMA* | Simple Moving Average |
| *RSI* | Relative Strength Index |
| *TA* | Technical Analysis |
| *GDP* | Gross Domestic Product |
| *ROI* | Return on Investment |
| *FA* | Fundamental Analysis |
| *EMA* | Exponential Moving Average |
| *MACD* | Moving Average Convergence Divergence |
| *OBV* | On-Balance-Volume |
| *KLSE* | Kuala Lumpur Stock Exchange |
| *MSE* | Mean Square Error |
| *NMSE* | Normalized Mean Square Error |
| *ROE* | Return on Equity |
| *P/S* | Price/Sales Ratio |
| *EPS* | Earnings Per Share |
| *DNN* | Deep Neural Network |
| *ICSPI* | India Cement Stock Price Index |
| *P/E* | Price/Earnings Ratio |
| *PCA* | Principal Component Analysis |
| *FCM* | Fuzzy c means clustering |
| *SMOTE* | Synthetic Minority Over Sampling |
| *RNN* | Recurrent Neural Network |
| *ARIMA* | auto-regression integrated moving average |
| *NKE* | Nike |
| *API* | Application Programming Interface |

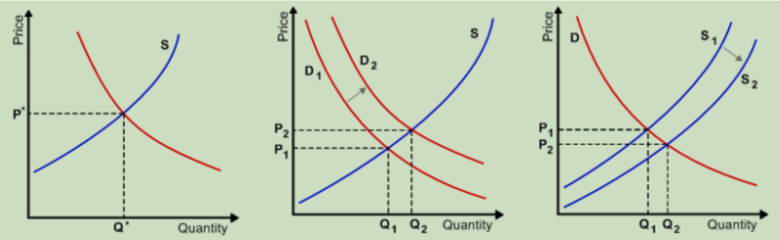
| | |
|---|---|
| *RMSE* | Root Mean Square Error |
| *UA* | Uncertainty-Aware Attention |
| *ATR* | Average True Range |
| *RMSE* | Root Mean Square Error |
| *FC* | Fully Connected |
| *GOOGL* | Alphabet |
| *CAGR%* | Compound Annual Growth Rate |

# Chapter 1 Introduction

## 1.1 Problem Statement and Motivation

Before diving deep to the report, it is important to introduce the basics of stock markets, how stock markets work and how profits can be generated from it. Trading stocks is a process that involved buying and selling company stocks on the stock exchange center with the goal of generating maximum profits. Generally, the trading process in stock exchange is similar as any other economic markets as it will gathers buyers and sellers together to trade stocks; If the buyers want to buy some quantity of a particular stock at a certain price, then there must be sellers who willing to sell the stock at the offered price. In fact, traders often want to buy stocks at relatively low prices and sell them at relatively high prices so that they can generate maximum profits in the stock trading process. This scenario can be expressed by a famous adage among investors: “buy low and sell high “. Because of this, stock trading process is thus governed by the supply and demand principle in economy. Supply and demand principle is the most fundamental economic principle as well as the backbone of economic forecasting. Figure 1.1.1 demonstrates two kinds of curves that represent demand (in red color) and supply (in blue color). Demand is provided by buyers, whereas supply is provided by sellers. The part where the demand curve intersect with supply curve represents the price equilibrium (the transaction price agreed by buyers and sellers). In term of stock trading, the x-axis and y-axis of supply demand graph represent number of shares outstanding and stock price respectively. In the Figure 1.1.1(b), we can see that there is a right shift in demand curve. The right shifting of demand curve implies that the demand in the market has increased, whereas the left shifting of demand curve represent the demand has decreased. When the demand in the market increased, it will result a situation where buyers are more than sellers. When this situation happens, the equilibrium price of the products will be increased (P2). Investors often aim to predict this kind of pattern/shifts so that the stocks can be purchased at lower price (P1) and be sold at (P2), earning positive return of (P2-P1). Similar concept applies when the demand is decreased (sellers are more than buyers).

In this case, investors will aim to minimize the losses, which investors sell the stocks before the price dropped.




(a) Price equilibrium at price $P^*$ and quantity $c$.
(b) An increase in demand causes an increase in price.
(c) An increase in supply causes a reduction in price.

Figure 1.1.1 Supply and Demand Curve

One often question asked by the researchers is whether the stock price or stock trend can be predicted. Market's predictability has been one of the most debated topics in finance world. According to the efficient markets hypothesis (EMH) and Random Walk hypothesis state that the prediction of stock price movements would be an impossible task as the stock prices are always fully reflect all available information in the market, and stock prices will not move in neat patterns which implies that current stock prices are not correlated with future stock prices [1]. However, adaptive market hypothesis (AMH) was then proposed by a researcher to challenge the point of view of EMH and random work theory. AMH says that the conditions of EMH do not match and contradict with the real-world conditions [2]. AMH attempt to associate EMH with behavioral finance. Behavioral finance is the study of finance in psychology and sociology perspective. AMH states that human behavior such as overreaction, overconfidence, information bias, greed, and fear are the factors that contradict with EMH principles. All these irrational human behaviors in the market will therefore lead to profitable conditions in the market.

Due the existence of irrational human behaviors in the stock markets, stock prices forecast through technical analysis are feasible [2]. Technical analysis is a stock forecasting method that studies on the statistical patterns in historical data**.** Technical analysis believes that the behavior of investors in stock trading are often repeated by the latter investors. Thus, by studying the statistical patterns of stock prices and comparing them with the current stock movement in the market, it is feasible to discover the similar profitable positions that had occurred in the past. There are many technical indicators that have been introduced such as Simple Moving Average (SMA), Relative Strength Index (RSI) and so on. With the help of indicators, investors will know whether the current stock movement trend is uptrend or downtrend, whether the observed stock is in overbought condition or oversold condition, and whether the momentum of stock price movement is strong or weak. In most previous works, technical analysis indicators have been extensively used as input features in machine learning techniques to forecast stock price movements in the stock markets. Another popular technique of stock markets forecasting is fundamental analysis, Fundamental analysis study the stocks movement by measuring the intrinsic value of companies. Instead of analysing statistical patterns of stocks prices, it studies from a wide perspective such as the overall conditions of economy, the financial performance of individual companies, the supply chain relationship of companies as well as industries conditions. Earnings, debts, GDP, ROI and so on are reviewed in fundamental analysis to predict and evaluate whether the value of a particular stock will increase in future. The main difference between technical analysis (TA) and fundamental analysis (FA) is that TA is suitable for short-term prediction whereas FA is suitable for long-term prediction. Most recent works have proven technical indicators are useful factors to be considered in short-term stock price forecasting algorithm such as ANN and LSTM.

Due to the high trading volume, noise, volatility and non-stationarity of stocks, stock price prediction is considered as a difficult problem. Moreover, stock market movement can be easily affected by political, macroeconomic environment as well as human emotions. Nonstationary in stocks implies that the distribution of stock prices will change according to the time change. Because of the high volatility of stock price, it is clearly impossible to rely only on the trader’s experiences in analysis and decision making. Investors need some kinds of powerful analysis tools that can detect the nonlinear relationship between the stock prices. This is where machine learning model in stock prediction come into picture. Most recent works have made extensive use of machine learning techniques in stock price prediction. Because advanced machine learning algorithms are capable of handling huge volume and complex nonlinear data as well as disclosing complex relationship and hidden patterns in the complex data. Machine learning techniques in stock price prediction have proven to be more useful and efficient than the traditional stock price forecasting methods (FA, TA). Recent research has shown that stock market prediction using machine leaning algorithm along with technical indicators and fundamental indicators can improve efficiencies by 60% to 80% [3]. There are some popular machine learning algorithms used in stock prediction such as Artificial neural network (ANN), Recurrent neural network (LSTM), and Genetic Algorithm (GA). ANN can tolerate noise and handle incomplete and overlapped data, which has proven to be a suitable option for stock price prediction [3], LSTM can detect complex long-time dependency present in the data [4], and Genetic algorithm also a good approach to determine the best optimal technical indicators during the training process. Therefore, this project is aimed to develop a stock prediction system using machine learning algorithms. However, there were some limitation and problems were observed in most previous works.

**One problem that has been found from most of the previous works is limited flexibility to customize the input parameter(s) in the feature selection process**. During the feature selection process, most previous works tend to apply large number of input parameters such as technical indicators and employed feature selection algorithms to filter out the best-voted features. They did not provide the flexibility to customers (investors) to choose their favourite parameters (technical indicators/fundamental indicators) when building the prediction model. Giving the flexibility of customizing input parameters is important as investors can combine their investing strategies with the power of machine learning to guide them in better decision making during the stock trading process**.** The is particularly useful for those experienced investors. As most of them have formulated their own profitable trading rules based on their experiences on the stock market. Thus, they will know what technical indicators are having a positive relationship with the stock’s prices. With the flexibility of customizing input parameters of the machine learning model, we can combine the domain knowledge of investors with the machine learning algorithms to make more meaningful prediction rather than randomly supplying large number of technical indicators.

**Another limitation that has been found from most previous work is that the stock prediction models is pre-defined model**. Most previous works pretrained their machine learning models and deployed them as static models in the application for stock price forecasting. When the new data is reached, programmers will need to bring the whole application down to update the prediction models in the offline environment and redeploy them again manually. This way of deploying the prediction models to application is good if the data related to the forecasting problem is almost constant regardless of the time such as cats' breed prediction; however, this deployment is not suitable for stock price forecasting problem as the distribution of stocks data will be changed frequently from time to time. Therefore, the deployment of the prediction models for stocks forecasting should be dynamic so that the prediction models can adapt to the frequent changes of the data. The system should build the model automatically upon customer requests in real-time and update the prediction models consistently and automatically without programmer intervention after deployment.

Lastly**, most of the investing platforms do not provide stock indicators recommendation feature to let user knew which model has better accuracy on the stock prediction result.** As mentioned earlier in this paper that most previous works tend to supply large number of technical indicators and make use of feature selection algorithms to reduce the dimension of the stock indicators, and the reduced features is then input to the model. This could be the reason why the system is not able to provide the recommendation as feature reduction algorithms such as PCA will merge all the features together and form the new synthetic features, which all the original features will be gone. Further, stock indicators recommendation feature is important for newcomers who are not familiar with the new stocks. By telling them the best combination of the stock indicators that achieved the highest accuracy in a particular stock, investors will roughly know which indictors have relationships with the historical stock prices and which indicators will have weak relationship with the historical stock prices.

In short, according to the limitations and weaknesses as described above, lack of flexibility in the stock prediction model is the major gap to be filled in this project. In this project, a more flexible and dynamic model will be built upon customer requests in real-time.

## 1.2 Objectives

**To develop a dynamic prediction model that allow users to customize the input and allow users to update the prediction model automatically when new data is reached**. Increasing the flexibility of the stock prediction model is the major goal in this project. The stock prediction model is expected to adapt to investors' favourite indicators during the model training and able to update itself automatically with a single click without having the backend programmers to update it manually in the server side.

**To develop a LSTM stock prediction model design that can achieve accuracy of 50% to 60% as well as guarantee positive return in most stocks in KLSE**. This project would aim to provide dynamic prediction models that will achieve accuracy around 50% to 60% as well as achieve positive return in most of the selected stocks. Of course, it is hard to ensure the model to generate positive return in all stocks especially the stocks that are in downturn state. Therefore, this project will aim to build a model that guarantee positive return in majority of stocks instead of all selected stocks. To design the architecture of model that can achieve the 50% to 60% performance as well as guarantee the positive return in most stocks, comprehensive research process will be carried out. This process mainly includes studying of the best input features, best architecture design, best pre-processing methods as well as the best hyperparameters (layer, units, epochs, batch size and etc) for the prediction models.

**To recommend the best combination of stock indicators to the investors based on the accuracy of the models**. The project would also aim to provide the recommendation of stock indicators combination that generated the highest accuracy during the model training. The system will display the evaluation results for all possible stock indicators combinations and select the best performing combination as the recommendation to the investors. In this way, investors will have a clear view on how each combination of indicators perform in stock market.

## 1.3 Project Scope and Direction

This project is **aimed to develop a stock indicators customization tool using LSTM algorithm on web-based platforms**. This stock indicator customization tool will incorporate with LSTM algorithm to make prediction on the short-term movement of stock prices. This system allows customers to **customize the input parameters of LSTM model** by using drag and drop functionalities so that investors can adapt to the function easily. After customers have chosen the preferred inputs parameters, historical data for the chosen stock will be collected through Yahoo Finance API call, and LSTM will start to learn patterns from the input parameters in real-time and finally make prediction on the movement of the stock price (uptrend/downtrend) for the next timestep. Investors are able to **save the model once the customized model has been trained**. By giving the ability to save the trained model in the database, consumption of time and computational resources can be saved. Users do not have to retrain the model again and again every time the browser refreshed, and user could reuse the previously trained model to make next prediction. **This saved model also can be updated with latest data if the customers click the "update" button**. Offline models will also be provided to satisfy the need of investors who do not want to waste time on training and want an immediate prediction result. Moreover, after the customized model has been trained, investors can choose to **run a trading simulation using test data to evaluate the usability of model prediction in stock trading environment**. During the trading simulation, a buying operation on the current timestep will be performed if the model predicts an uptrend condition in the next timestep. After the trading simulation, a detailed portfolio report that includes some popular performance metrices such as Sortino Ratio, Sharpe ratio and Maximum Drawdown will be provided to the investors

In addition, the system will also **provide recommendations on the selection of the technical indicators used for individual stock prediction to the investors**. After investors have chosen the preferred stock indicators as the input parameters of the prediction model. The investors could choose to run an in-depth analysis on different combinations of the preferred stock indicators. For example, the SMA-30days and EMA-30days were chosen as the preferred indicators, and it will result 3 possible combinations which are SMA-30days, EMA30days and SMA-30days, EMA-30days. These three combinations will be evaluated, and the combination that achieved highest accuracy will be recommended by the system. Finally, when the investors select a particular stock to predict, the system will provide a graphical representation of the historical stock prices (stock charting) with technical analysis capabilities so that investors can perform some basic technical analysis on the historical stock prices.

The main targeted stocks in this system will be the growth stocks and volatile stocks listed in Kuala Lumper stock exchange (KLSE). Growth stocks are referring to the companies that are currently at growing stage. Growing stage is a stage where company will require a lot of capital to develop the company. Because the companies are in growing state, the growth stocks will be more volatile than dividend stocks [5].

Further, volatile stocks are the stock with high percentage of fluctuation such as banking stocks. Due to the volatility and fluctuation present in these stocks, high returns can be easily gained from the changes of prices if the trend of stock prices are correctly predicted. Therefore, we will aim to make prediction on the stocks with volatility. The maximum numbers of stocks provided in the system will be 100 stocks. Furthermore, the indicators provided will only limit to technical indicators and will not consider any fundamental indicators as we are focusing on short-term prediction. Moreover, the entire system will be built using Node JS framework and Python Jupyter notebook. Python jupyter notebook is mainly for the development of the stock prediction model architecture, whereas Node JS is meant for the full-stack (front-end and back-end) website development. After the best model architecture has been designed in the Python Jupyter notebook, the architecture design of the best model will be migrated and implemented in the Node JS so that we can use the exact same architecture of LSTM model to run the real-time training in Node JS platform. To incorporate the deep learning features with JavaScript, TensorFlow JS will be used. In addition, to save and load the trained models in the system, NoSQL database will be used because the format of the LSTM model's weight file would be in binary (.bin). MongoDB is chosen for this project due to its capability to store binary file format

## 1.4 Contributions

This tool has significant implication to institutional investors and individual investors. Institutional investors are the parties who are hired by mutual funds organizations or insurance companies to invest money on behalf of other people [6]. Institutional investors such as fund managers and individual day traders can make this tool as an extra consideration factor to guide them in decision making, instead of just depending on their own investment strategies. By having the flexibility of customizing the investment strategies in LSTM models, Institutional investors can match their own investment strategies with the LSTM models to make prediction so that better decisions can be made. Customer not only have their own investment strategies to guide them by giving buying/selling signals but also have extra analysis information from machine learning model that tells how current data points will perform in next time frame based on the analysis of historical data.

Secondly, by having this system, institutional investors and individual investors will know what combinations of the technical indicators are useful in the stock trend prediction. They do not have to monitor the performance of the technical indicators manually with respect to each of the stocks anymore, which is time consuming. For example, an investor has formulated a trading strategy with 5 technical indicators as the decision-making condition, and he/she have to manually monitor whether these 5 stocks are closely related to the stock price changes. However, by running the in-depth analysis on the system, the system will automatically recommend the best combination of the stock indicators that achieved highest accuracy in prediction which only contain 4 technical indicators. In this way, investors will know that only these 4 technical indicators are possibly associated to the changes of stock prices in the history, and the extra 1 technical indicator might not have strong association to the stock prices. Also, after the in-depth analysis, model accuracy for all possible stock indicators combinations will be presented to the investors. Thus, investors will have a better understanding on how different combinations of stock indicators are performing. This gives investors a clear direction in building their portfolio strategy.

## 1.5 Report Organization

In chapter 2, some important questions and doubts that related to stock prediction will be answered such as the predictability of the stock market, the common input features used in stock prediction as well as the common machine learning algorithms used in stock prediction. Critical review of some popular trading platforms will also be carried in this chapter. Further, in chapter 3, the methodology of the proposed system will be thoroughly discussed. Moreover, the design of the proposed system will be discussed in chapter 4. The system design includes the system architecture design, prediction models architecture design, database design, parallel processing design as well as the system functionalities and interfaces design. Furthermore, chapter 5 will discuss the system testing and evaluation. This chapter will discuss the testing methods to prove the project objectives as well as the outcome of the testing. Chapter 6 includes the project discussion in various perspectives such as the limitations of the project, the future enhancements, project challenges and so on. Lastly, chapter 7 will conclude the whole project by summarizing the project motivation, project objectives and the novelties of the projects.

# Chapter 2 Literature Reviews

## 2.1 Market Predictability

To determine whether stock markets can be predicted, few papers have been reviewed. Market predictability is one of most debated topics in the financial field. Efficient market hypothesis (EMH) suggested that the stock prices always fully reflect all available information in the market, and thus no one could consistently perform well in stock markets and the prediction of stock prices is not feasible [1]. There are few assumptions made under EMH, which are large number of rational and profit-maximizing investors who are always actively participate in the market, information is freely available, and stock price will quickly and fully respond to the arrival of random information. With these assumptions made in EMH, technical analysis and fundamental analysis could not constantly make abnormal profits in the stock markets. The market is considered as efficient market if the stock prices reflect all information quickly when new information arrives in the stock market [7]. On the contrary, adaptive market hypothesis (AMH) was then proposed to against the assumptions made in EMH from human behavioral perspective. Adaptive market hypothesis proposed that stock markets are not always efficient, and degree of market efficiency is depending on the environmental conditions in stock markets such as the adaptability of stock traders and the number of competitors. In simple words, stock markets are adaptable, and the markets will move between inefficiency and efficiency in different time variation and different market conditions, and thus positive return can be made through analysis. [2] suggests that there are no such thing as rational investors and human do make mistakes during trading process, but investors will learn through the mistakes and adapt their behavior accordingly. Because of the existence of human irrational behavior in the stock market, prediction of stock market movement is feasible.

Therefore, 2 papers that investigated the stock markets' predictability based on the theory of adaptive market hypothesis have been reviewed. Based on the evidence given by both papers, prediction of stock prices is possible, and the predictability of stock prices is high when the degree of efficiency of market is low. The stock markets involved are US stocks market (Dow jones), India stock markets (Sensex) and Nifty index in National Stock Exchange. Firstly, [8] conducted a study to find out whether the emerging markets such as Sensex and Nifty follow the principle proposed by AMH and study the predictability of these stock markets. This study used linear and non-linear tests such as autocorrelation test, variance ratio test, runs test and so on to measure the linear and nonlinear relationship in time series stock prices between the year of 1991 and 2013. High linear dependency between the time series stock prices was found in Nifty index during the time from 1994 to 1996 and 2003 to 2006, whereas other time periods showing less linear dependency, and the similar patterns also have been found in Sensex index. However, in term of nonlinear relationship, both Nifty and Sensex showed strong nonlinear relationship during the year between 2003 to 2008. Existence of strong linear and non-linear relationship in historical prices of both stock markets are the strong evidence that prove the markets are predictable. During markets crashes no predictability was observed in both Nifty and Sensex indexes. The authors concluded that both stock market indexes follow the principle of AMH which the degree of efficiency market switched between inefficiency and efficiency over different period of time and change in different market conditions.

Secondly, [9] also make use of similar tests such as autocorrelation test and variance ration tests in order to test whether Dow jones index in US is predictable based on the principle of AMH. In this study, the authors used Dow jones index from 1900 to 2009. The results have proven that the Dow jones index is highly predictable during political crisis, whereas during market crashes the predictability of Dow joins is unfeasible, which showing consistent result as [8]. Based on the results, the authors have concluded that the degree of market efficiency in Dow jones market is switched between efficiency and inefficiency over time. In short, both research have proven that the degree of efficiency of stock markets in India and USA switched between inefficiency and efficiency over different period of time, and thus leading a conclusion that stock market prediction is possible during the inefficiency of stock market.

## 2.2 Predictive Inputs

Based on the recent works, predictive inputs for machine learning models in stock price prediction can be categorized into 3 groups: Technical analysis, Fundamental analysis and combined analyses. A systematic review of total one hundred and twenty-two research papers in the area of stock price prediction have been carried out by researchers and revealed that 66% of the research have made extensive use of technical analysis as the input parameters, while 23 % of works were based on fundamental analysis and the remaining 11% were based on combine analyses [10]. This systematic review also found out that neural networks are the popular techniques used in the field of stock price prediction.

Technical analysis is aimed at studying the statistical behavior of historical data and predicting the shifts of prices. Technical analysis is focus solely on the price patterns as well as the volume of data [11]. Because the price data and volume data are available daily in stock markets, technical analysis is suitable for short-term predictions. In technical analysis, technical indicators are the main statistical functions that applied in the analysis of stock prices. These technical indicators can be divided into 4 groups which are momentum indicators, volatility indicators, trend indictors and volume indicators. Each category gives different information about the stock prices, and all these indicators are computed from the historical prices of stocks. Momentum indicators talk about the strength of stock price’s movement. Trend indicators represent whether a stock price is going uptrend or downtrend. Volatility indicators measure the amount of trading activity. If the volatility of stock prices is strong, then it implies that there is a drastic change between buying and selling force. Volume indicators represent the amount of stock that have been traded among investors and sellers. If the volume is high, then it means the trading process have happen many times for a given time period [12].

Some famous technical indicators that have been used as input parameters in most previous works are Simple moving average (SMA), exponential moving average (EMA), moving average convergence divergence (MACD), relative strength index (RSI) and on-balance-volume (OBV).

**SMA** is an indicator that represent whether the movement of a stock prices is in uptrend or downtrend. SMA is just a simple statistical model that calculate the average of the prices on a given period [13]. Figure 2.2.1 shows the mathematical formula of SMA.

$$\text{SMA} = \frac{A_1 + A_2 + \ldots + A_n}{n}$$

**where:**

$A_n$ = the price of an asset at period $n$

$n$ = the number of total periods

Figure 2.2.1 Mathematical Formula of SMA

**EMA** is almost like SMA, but the difference is that it places greater wight on the most recent stock prices. Figure 2.2.2 shows the mathematical formula of EMA.

$$EMA = Price(t) \times k + EMA(y) \times (1 - k) \tag{1}$$

where t and y represents today and yesterday respectively, N is the number of days in EMA and k (smoothing) = 2/(N + 1).

Figure 2.2.2 Mathematical Formula of EMA

**MACD** is a momentum indicator that give investors signal about the movement's strength of the stock price. It is calculated by subtracting EMA-26 days with EMA-12 days. After calculating the MACD, it is then compared with EMA of 9 days period. If the MACD is greater than EMA-9 then it will give a buy signal. If the MACD is lower than EMA-9, then it will give a selling signal [14]. Figure 2.2.1 shows the mathematical formula of MACD.

$$MACD = \sum_{i=1}^{n} EMA_k - \sum_{i=1}^{n} EMA_d$$

where k = 12 (number of days) and d = 26 reflect the number of days in EMA.

Figure 2.2.3 Mathematical Formula of MACD

**RSI** is a momentum indicator that talks about the strength of trend of the stock prices. By using this indicator, investors will know whether the stock is overvalued or undervalued. RSI is calculated by determining the average gain and average in past n days. The trend is strong (momentum) when the value of RSI is falling between 30% to 70%. The RSI value below 30% implies that the price might be undervalued which indicating a buying signal, whereas if the RSI rise above 70% means the price is overvalued which indicating a selling signal [14]. Figure 2.2.4 shows the mathematical formula of RSI.

$$RSI = 100 - \left(\frac{100}{1} + RS\right)$$

where RS = average gain/average loss

Figure 2.2.4 Mathematical Formula of RSI

**OBV** is also a momentum indicator that make use of volume flow as the signal to detect the changes of stock prices. If the OBV line is falling, it means that the stock price is expected to be decreased in future. If the OBV line is rising, then the stock price will increase in future. Figure 2.2.5 shows the mathematical formula of OBV.

$$\text{OBV} = \text{OBV}_{prev} + \begin{cases} \text{volume}, & \text{if close} > \text{close}_{prev} \\ 0, & \text{if close} = \text{close}_{prev} \\ -\text{volume}, & \text{if close} < \text{close}_{prev} \end{cases}$$

**where:**

OBV = Current on-balance volume level

$\text{OBV}_{prev}$ = Previous on-balance volume level

volume = Latest trading volume amount

Figure 2.2.5 Mathematical Formula of OBV

Because there are many technical indicators available in the technical analysis, it is important to determine which indicators are relevant to the stock prediction models and which hyperparameters (days) of technical indicators are relevant. Most previous works tend to apply large number of technical indicators as input parameters into machine learning algorithms such as artificial neural network (ANN). However, supplying irrelevant input parameters into ANN might cause difficulty in generalizing the data [15]. Thus, it is better to determine the relationship between the technical indicators and stock prices before training the model. According to [15] from Ohio University, he has conducted a study on investigating the sensitivity of common technical indicators to the changes of stock prices. He has shortlisted technical indicators that are sensitive to the changes of Ford company stock prices by using self-organizing map neural network. Self-organizing map neural network is a feature extraction technique, and it will discretize the input space of training samples. After the training process, the indicators that are sensitive to the stock prices were then fed into artificial neural network to predict the movement of Ford company's stock in future. The validation result shows that, the prediction errors (MSE, NMSR etc) were relatively low, which was about 0.1 in average. Thus, the author has concluded that supplying machine learning model with relevant technical indicators is important to increase the performance of model and reduce the complexity of model at the same time. One important point that need to take note is that different stocks will behave differently, and thus different technical indicators might result after applying the feature extraction techniques. Thus, similar analysis will be done in this project so that different sets of technical indicators that are correlated to stocks listed in Kuala Lumpur stock exchange (KLSE) can be determined.

Another type of input parameter is based on fundamental analysis. Fundamental analysis is a study of companies in economy, industry, and firm perspective. Companies' fundamental such as company supply chain relationship, current market industrial positions, annual financial performance, future performance, economy conditions, company management [16]. Fundamental analysis is mainly used to compute the intrinsic value of companies based on various formulas and then, use the computed intrinsic value to determine whether the company stock is undervalued or overvalued. If a company is overvalued, the stock price might be decreased in the long-term future, and company stock prices is expected to be increased in long-term future if the company is undervalued. In most recent works, fundamental indicators that are being used widely in stock prediction models are return on equity (ROE), Price/sales ratio, price/earnings ratio (P/E) and earning per share (EPS).

**ROE** represent how well the company resources were used to generate profits for a given year. Figure 2.2.6 shows the mathematical formula of ROE.

$$ROE = \frac{PTP}{SE}$$

where PTP = post-tax profit and SE = shareholder equity

Figure 2.2.6 Mathematical Formula of ROE

**P/S** represent the stock value of a company divided by the company market value. If the P/S is lower, then it means company is undervalued; while company is overvalued if P/S is high. Figure 2.2.7 shows the mathematical formula of P/S.

$$\frac{P}{S} = \frac{Share\,Price}{(\text{Returns over a 12 month time frame})}$$

Figure 2.2.7 Mathematical Formula of P/S

**P/E** represent the attractiveness of company current stock price compared to the firm per share earnings. Low P/E is better as investors can buy the stock with relative low price but with high earning. Figure 2.2.8 shows the mathematical formula of P/E.

$$P/E = \frac{\text{Market value per share}}{EPS}$$

Figure 2.2.8 mathematical Formula of P/E.

**EPS** represent the profitability of a company by dividing the total earning with the total shares of stocks. The higher the EPS, the better the company's financial performance. Figure 2.2.5 shows the mathematical formula of OBV. Figure 2.2.9 shows the mathematical formula of EPS.

$$EPS = \frac{\text{Net Income after Tax}}{\text{Total Number of Outstanding Shares}}$$

Figure 2.2.9 Mathematical Formula of EPS

Because fundamental indicators only available annually, it is suitable for long-term prediction instead of short-term predictions [17]. The main reason why automation of fundamental analysis is challenging is because of the structural instability [10]. Due to the introduction of text mining techniques in deep learning, recent works have started to consider fundamental analysis as a part of predictive input. However, there are some problems associated with text mining techniques been introduced such as multi-lingual dependency [18]. Since fundamental indicators is not suitable from short-term prediction, fundamental indicators will not be considered in this project.

## 2.3 Current Approaches on Stock Prediction

In this new media age, the movement of stock prices are becoming more dynamic and volatile than ever these day as investors can easily obtain available information through information technology such as social media and online news. Due to easy access of information, there will be more reactions to the stocks, leading high volatility of stocks price in the markets. Therefore, prediction of stock prices heavily depends on the analysis of vast volume of data, and this is where machine learning is needed. After having reviewed the literatures, the machine learning algorithms that are widely used as stock price prediction model are support vector machine (SVM), evolutionary algorithm (Genetic algorithm), and neural networks (ANN, LSTM).

### 2.3.1 Genetic Algorithm

Genetic algorithm is an optimization approach that aim to find the perfect functional and effective solution with respect to the problem, and it is a form of evolutionary algorithm. In recent works, Genetic algorithm is often used by the researchers to search for the optimized combinations of technical indicators that yields the maximum trading profits during the training process. GA is often incorporated with trading simulation in order to measure the trading profits of the parameters. Two papers that used GA algorithm in stock prediction have been reviewed.

Figure 2.3.1.1 shows the complete process of genetic algorithms. The main mechanisms in GA are chromosome encoding, fitness evaluation. reproduction and mutation. During the initialization phase, there will be a population that is randomly distributed, and then GA will start with the random selection of “parents” from the population. These “parents” are called as chromosomes, and they can be encoded with characters, real number or predefined rules (parameters). After the initialization process, GA will evaluate the performance of each selected chromosome with user predefined fitness function. After the evaluation process, chromosomes with higher fitness value or larger than the predefined threshold will be selected for mating process. The mating process is mainly to produce new chromosomes that will preserve the worthy characteristics from the upper generation (previous selected chromosomes with high fitness value). The mating process is often called as “Crossover’’. The mating process will operate by exchanging the parameters that encoded in the chromosomes with other chromosomes’ parameters so that the new chromosomes that contains the best-performing characteristics from “good parents” can be produced. This exchange rate will be varied depending on the programmers. Although the new chromosomes inherit the best-performing parameters from their parents, they do not introduce any new information that might perform good in the fitness function. This is where mutation process comes in, it will change one of the parameters of chromosomes to random value in the range defined, and thus increase the probability of reaching the best solution. After the mutation process, evaluation of new chromosomes will take place again, and then mating of new chromosomes to produce next generation, and the entire process will be repeated until maximum interactions have reached or reached a satisfactory fitness level.

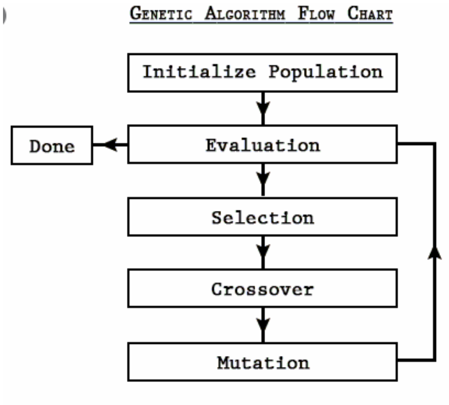



Figure 2.3.1.1 Genetic Algorithm Mechanism

Firstly, [19] has proposed a stock trading model based on the optimized technical indicators. Since the study of the optimization of technical indicators using GA has not been extensively studied in the past, and they aim to fill up the gap in this paper. They applied GA technique to find out the best combinations of hyperparameters in technical indicators. They used historical stock prices between 1997 and 2017 of Dow Jones 30 stocks as training data to determine the best parameters for RSI, SMA combinations in the time series of Dow Jones 30 stocks. After they have found out the 30 sets of optimized technical indicators, they use those optimized parameters as training features in deep neural network to predict the buy/sell position. Based on the trading simulation results, the deep learning model that uses the optimized input parameters (GA+DNN) has achieved 11.93% of average annualized return of 30 stocks, which is better than the model that used random technical indicators (average annualized return of 10.3%). The strength of the approach proposed by the authors is that the GA algorithm is able to find out the important technical indicators that yield the highest results in the past and also solved the difficulty in setting the hyperparameters for different stock indicators. Optimizing the hyperparameters of technical indicators before training the models is important to increase the generalization potential of neural network. However, one limitation of this approach is that the selection of hyperparameters of GA algorithm such as cross over rate and mutation rate should be determined carefully. Random selection of these hyperparameter might results meaningless combinations of technical indicators. Another limitation is that when GA is cosidered in the feature optimization process, more hyperparameters (GA's hyperparemeters and neural network's hyperparameters) will need to be fine-tuned during the training process, and thus more time is needed to compelete the training process. Therefore, some feature extraction techniques which requried less hyperparameters can be used such as PCA and correlation analysis to achieve similar results. PCA has less hyperparameters as compared to GA, and PCA also has been proven to be a good technique in extracting the important technical indicators in stock price prediction. In fact, because of the robutness and complexity of GA technique, GA might able find out the best optimal indicators as compared to PCA technique, but the time taken will be much more longer.

Secondly, [20] also conducted research to adopt GA technique in optimizing the hyperparameters of different technical indicators. One of the purposes of this research is to tackle the ambiguity in setting up the hyperparameters of technical indicators as different combinations of hyperparameters in technical indicators might give different profit. They adopted GA to find out a trading rule (combinations of different technical indicators) that generated the highest profit in the past. They used India cement stock price index (ICSPI) from 2011 to 2012 as the training data and testing data. They used genetic algorithm software package (evolver) to simulate the trading process and calculate the fitness function (profits) over the history. During the validation process, the authors used the financial evaluation methods (calculate the trading profit) to measure the performance of the optimized trading rule generated by the GA algorithm. The results shows that the derived strategies generated about average profit of 21% to 54% even though the index of ICSPI was decreasing during validation period. The strength of this paper is that they considered a total of 6 technical indicators that different meaning such as the trend, momentum, volatility and the volume of stock prices. This will help the model in generalizing the historical data and leading to better results as different kinds of factors were considered. This research focused more on parameter optimization. However, this approach does not account the movement behavior of stock in history, it only tells investors about the best entry points and exit points that generated maximum profits in the past, which is not practical among investors. Telling the best exit and entry timing might not ensure average positive returns in long future as it doesn't study the moving pattern of the stock prices. Forecasting stock directional changes is an important factor to be considered in financial decision making, and algorithm trading [21]. Handling stock movement prediction without considering the movement behavior of the stock price might not help investors in decision making as information about trend behavior of stock is very crucial to assists investors in decision making. Therefore, it is better to include the optimized trading rules as input parameters for machine learning models to learn the movement behavior of stocks.

In short, both papers have shown that GA algorithm is able to find out the best hyperparameters of technical indicators that generate positive profit, which showing consistent results. However, some limitations have been found in both papers such as the difficulty in choosing the hyperparameters and the lack of stock movement information.

### 2.3.2 Artificial Neural Networks

Artificial neural network was designed to simulate how human brain cells process information. Artificial neural network is made up of large number of highly interconnected neurons that are arranged in the form of network layers. ANN contains multiple hidden layers, one input layer and one output layer, and each neuron contains weighted connection which are similar to human’s synapses [15]. Each neuron will be inputted with multiple independent variables along with their associated weight, and then the neuron will sum up all the outputs given by the multiplication of each independent variable and its weight. The value computed from the summation is input to the nonlinear activation function such as Sigmoid function, and the final output will be computed. The backpropagation method will be used to update all individual connection weights of each neuron to achieve lower prediction errors. Because of the nonlinear activation that applied in all neurons, ANN has the ability to learn complex nonlinear relationship between data. Figure 2.3.1.2 illustrates the structure of ANN model.

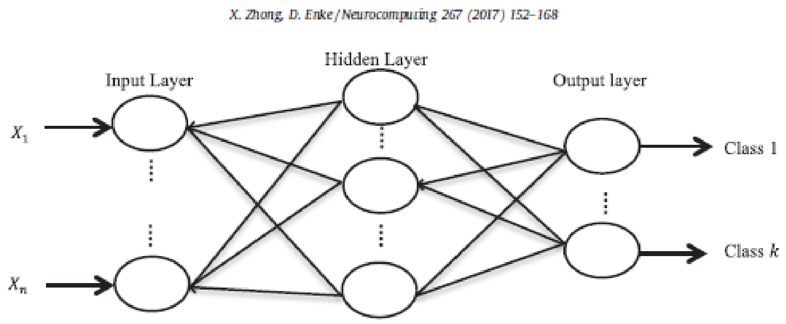



Figure 2.3.2.1 Structure of ANN

According to the systematic review done by [10], ANN is one of the most used machines learning models in stock price prediction among the 122 papers. The study also revealed that the outcome provided by the recent works prove that ANN and SVM have better generalization potential.

[22] have conducted a comprehensive study to forecast S&P 500 index based on 60 financial and fundamental features (2003-2013 period) using ANN. The authors used fuzzy c means clustering technique (FCM) and principal component analysis (PCA) in order to reduce the overall complexity of input data and extract the features that have strong relationship with the stock index. FCM is a well-known data clustering technique. Because the input parameters are very large, the authors have divided data into multiple clusters and apply PCA to each cluster for dimension reductions. PCA technique is a powerful unsupervised linear technique for feature extraction and dimension reduction. PCA is aimed to transform high dimensional data into relatively low dimensional data by keeping maximal variance and covariance structure of data. In simple words, PCA will view the inputs data in low dimensional perspective and capture the best angle that retain the maximum information about the input data. The authors applied FCM to divide the inputs into 7 clusters, and each cluster is then supply to PCA to reduce the dimension of data. The pre-processed data is then supply to ANN to make prediction. The results given by ANN model is evaluated based on both computational evaluation method (accuracy) and financial evaluation methods (trading simulation). The authors also compared the result of ANN model that trained with transformed data with the result of ANN model that trained with non-transformed data. Based on the results given, the authors conclude that introduction of PCA technique not only reduce the complexity of data but also increase the prediction accuracy of ANN models. The maximum accuracy achieved was about 59%. The ANN model was further

evaluated in trading simulation to find out whether high predictability of ANN models imply high return in stock trading. The trading simulation results also proved that incorporating PCA with ANN generate higher returns than the benchmark technical analysis buy-and-sell strategy. Incorporating PCA algorithm and FCM in feature extraction process is the main strength presented in this study. Clustering data before applying PCA is a good method as PCA will reduce the dimension based on each cluster rather than the whole set of data, and thus the most influential variables information will be retained as much as possible. Eventually, it transformed 60 types of inputs into 12 types of inputs, which is a gradual decrease in term of dimension. However, one weakness has been found is that information loss will be occurred after applying PCA components on the technical indicators. This is because the output of PCA are the principal components (synthetic outputs), and we cannot tell which technical indicators are correlated to the stock prices. To display the stock indicators that are correlated to stock prices, we can use Pearson correlation matrix to approximate. As PCA extract feature based on the Pearson correlation calculation; thus, we can roughly get the technical indicators that were retained in the principal components by using Pearson correlation matrix. Finding out the relevant technical indicators is important to present the technical indicators recommendations to the investors in this project.

[23] also conducted research on ANN model in stock prediction. This study used NASAQ stock market index during 2017 and 2018 as training data. The authors considered both technical indicator and financial news as extra features in prediction, which results 151 to 611 dimensions of input dataset. The authors have found out that in most previous work, they did not focus on balancing the label in the outputs data and also do not consider the sequential dependency between input data and predicted output data in validation process. Therefore, the authors have considered increasing window cross validation method, and this method will not randomly shuffle the data instead it will validate the prediction error in sliding window form which preserve the time series behavior. Figure 2.3.2.2 shows the increasing window cross validation technique. If normal cross validation method is being used to test the accuracy of the time series data, it might not give the true performance of the models. Moreover, the authors also have made use of Synthetic Minority over sampling technique (SMOTE) to balance the label so that the prediction model will not biased to the majority of the label. This study also focused on introducing dropout layers in the ANN model. Dropout layer will eliminate

part of the neurons while training, and thus avoid overfitting issue. This study also uses two steps evaluation (machine leaning validation technique and trading simulation) to validate the effectiveness of the model. The validation results shows that ANN model achieved 50%-68% accuracy in average and achieved 85.2 % positive annualized return and 4.6% of Sharpe ratio in the trading simulation. The strength of the approach proposed by this study is that the model not only consider technical, fundamental factors and sentiment data from financial news but also focus on retaining the time series behavior of stock prices during the validation process, Because of this, the true performance of model can be measured. Balancing technique that proposed by this study also able prevent model return biased results. One weakness has been found is that the dimension of data used was about 151 to 611 dimensions. They did not consider the approach that exploit the relationship between the input and output data which increase the overall noise in the data. Studying the relationship in input features has been proven to be an important task in stock price prediction. Without studying the input feature relationship not only increase the overall complexity of the models but also might give bad result in term of model’s performance. Feature extraction technique should be applied so that more emphasis can be put on the features that has strong relationship and the overall complexity of data also can be reduced. Another limitation is that time series relationship will not be discovered using ANN model although the authors considered increasing window validation method. Increasing window validation methods only help to maintain the inputs and predicted output dependency during the validation but ANN will not take any sequential information in consideration when making prediction as ANN cannot detect time steps in input data. In order to detect the sequential pattern in the data, LSTM algorithm should be used.

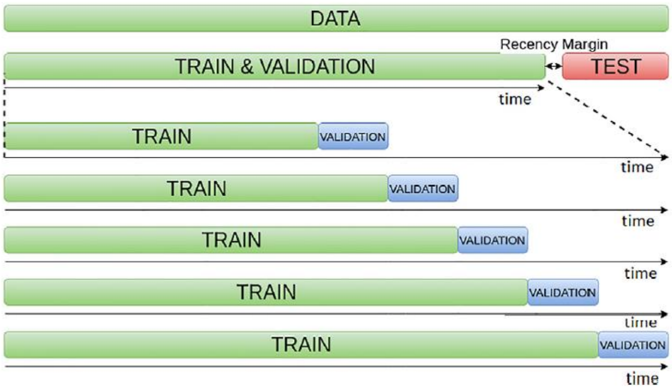



Figure 2.3.2.2 Increasing Window Cross Validation

In short, both papers have proven that ANN model is able to detect nonlinear relationship in complex data and both papers have shown consistent results in both machines leaning validation and trading simulation. To increase the overall performance of the stock prediction model, [22] have applied some feature engineering techniques, such as FCM and PCA. However, ANN is not able to capture the long-time dependency between the historical prices. ANN will not store any sequential information and not consider any time step (time information) in the data instead it treats every input data as independent. Therefore, in this project, we will take LSTM model as our choice as it can capture non-linearity as well as sequential information about the data.

### 2.3.3 Long Short-Term Memory

Long short-term memory (LSTM) is time series model that derived from recurrent neural networks (RNN). LSTM can store temporary information about the past stages and use them as consideration factors in making predictions. LSTM can memorize the sequence of the data which is very suitable for time series problems such as stock prediction [4]. In traditional RNN structure, RNN will concatenate the hidden layer output of the previous time step ($h_{t-1}$) with the input of next time step ($x_t$), and then supply the combined inputs into the neural network and these procedures will be repeated until the latest time frame of input data is reached. Therefore, the neural network can use previous information to predict the next outputs. However, one limitation with this structure is that RNN is not able to store long-time memory, and it will put more emphasis on the recent memory. As RNN will keep multiplying the outputs for each previous time step, and thus only the recent outputs will contain large weightage as compared to earlier outputs. Figure 2.3.3.1 shows the structure of RNN model.

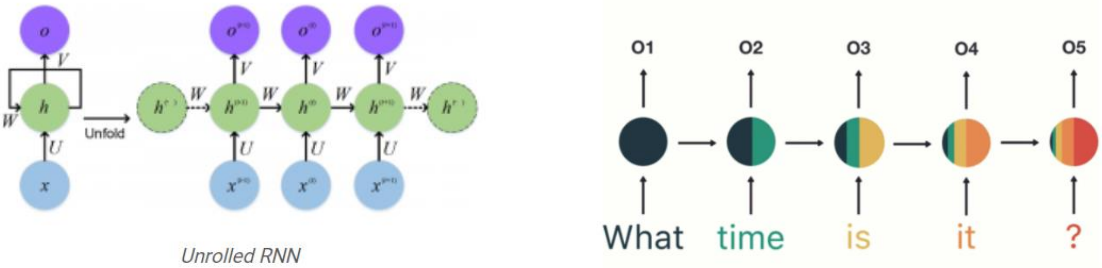



Figure 2.3.3.1 Structure of RNN Model

Because of this limitation, LSTM has been introduced to solve this problem. LSTM can store valuable long-term memory about the data by using extra memory line and control gates, and it has proven to be very useful to deal with long time data. Figure 2.3.3.2 illustrates the internal structure of LSTM model.

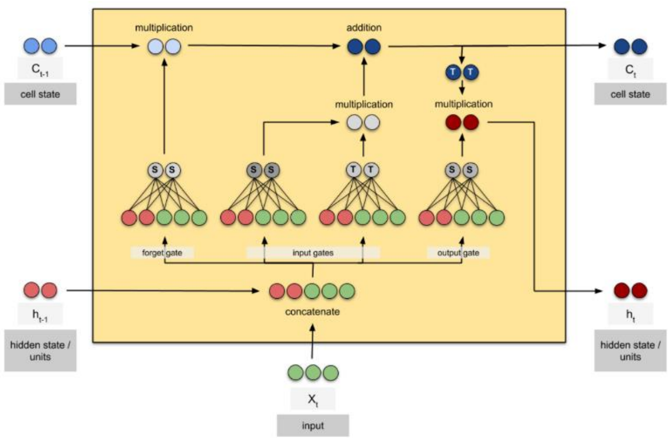



Figure2.3.3.2 Internal Structure of LSTM Model

LSTM contains forget gate, output gate, memory gate (input gate). Forget gate is used to determine whether the information of a particular time frame should be stored in the memory. The forget gate will be employed with sigmoid function, so that value between 1 and 0 can be returned. 1 implies that LSTM should store this information; while 0 implies that LSTM should forget the information in memory. The output gate will decide whether the value stored in memory should concatenate with the next inputs. The input gate is where the current time step data will be computed, and then the input gate will determine whether the computed data need to be stored in the memory cell. By having the extra gates to control the value stored in memory cell, LSTM can decide which past information is valuable in making prediction, instead of multiplying all past information like RNN, and thus LSTM is able to learn long-term dependencies in the data. Further, the figure 2.3.3.3 shows the working procedure of a single unit of LSTM. Each time-step Xt-1, Xt, Xt+1 will go through the LSTM unit that named with "A" (only single unit is present in the left  digram, the right digram is just an unfolded version of LSTM unit present in left diagram. Eventually, all time step will go through the same unit), and each time step will results a hidden state output ht-1, ht, ht+1 repectively. The hidden state ouput poduced in each time-step would also be brought to another time-step, so that the time dependecy can be learned by the model.

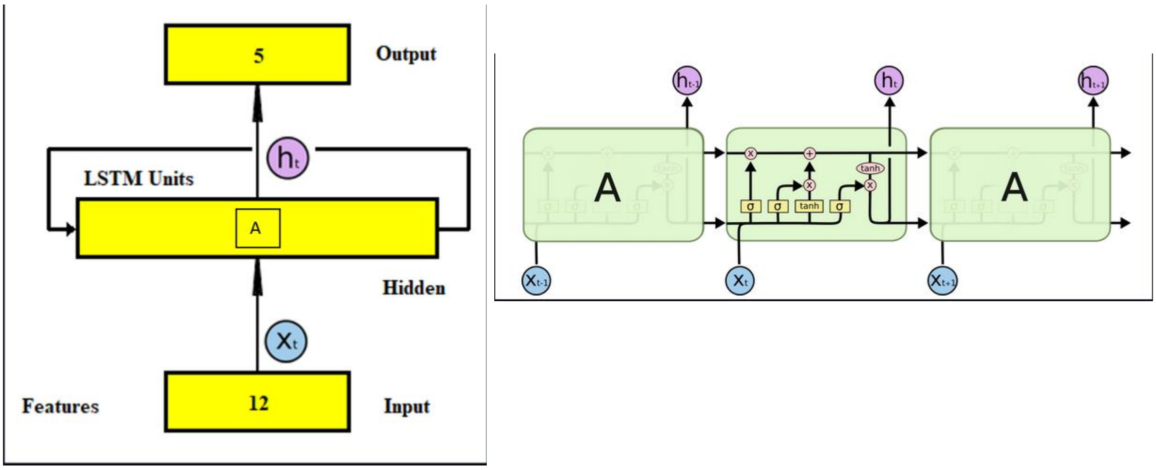



Figure2.3.3.3 Single Unit of LSTM Cell (Folded & Unfolded)

The following studies shows the implication of LSTM in stock price prediction. First, [4] has conducted research to investigate the performance of LSTM in stock price prediction and to determine how much epochs and data time span can achieve better results. Based on the papers that have been reviewed by the authors, they found out that previous research has proven LSTM is far more superior than other traditional time series prediction models in financial field such as auto-regression integrated moving average (ARIMA) when dealing with time series prediction. The input data used in this research are the closed prices of Alphabet (GOOGL) from 2004 to 2009 and closed prices of Nike (NKE) from 2010 to 2019.The authors used of 4 dropout layers and 4

LSTM units in the neural network to avoid overfitting issues. The authors only make use of opening prices of stock for forecasting, they did not consider any technical and fundamental indicators. During the validation process, the performance for different number of epochs and different number of periods used in training data were noted. Based on the results, the authors found out that increasing the data time span for both stocks (1980-2019) will gradually reduce the accuracy for the future prediction and also found that LSTM will perform better and achieve low testing errors if more epochs is set. The average error for both stocks was about 0.01, which is a good result. Thus, the authors conclude that short data time span with higher epochs will improve the performance of LSTM models in short-term prediction. The strength of this is study is that the authors introduce drop out layers after each LSTM layers and the results has proven that this way of structuring the LSTM neural network can reduce prediction error. The study focuses on the way to prevent the overfitting of the data as most previous works on LSTM suffer from overfitting. Figures 2.3.3.4 show the LSTM model structure used in this research. By arranging the neural network in this way (stacked each LSTM unit with a drop out layer), overfitting of data can be avoided. Adding more drop out layers will increase the generalization potential of LSTM model, but the downside is the time taken for such complex structure of LSTM model is too long if more input parameters are added into the model. In this study, the minimum time taken for the models to complete is about 2 minutes for 12 epochs as they only considered one input parameter. However, when there are more input factors to considered, this approach might not work well due to the complex structure of model. This model is good if the model is deployed as offline model, but if the model is use as online model, then it will not be a good choice as the time and computational consumption is too high.

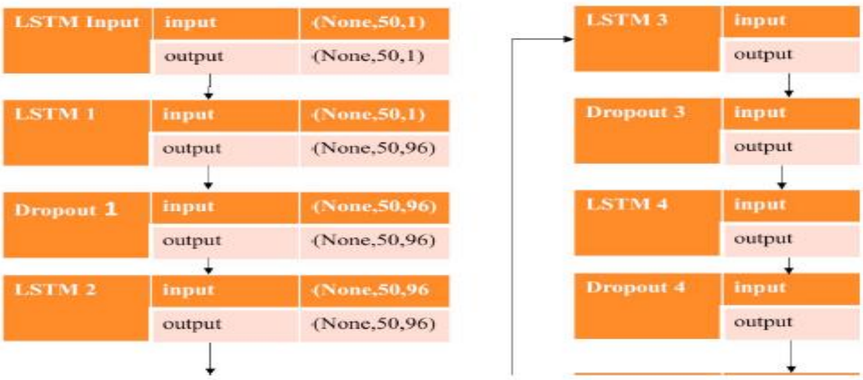



Figure 2.3.3.4 LSTM Model with Drop Out Layers

Secondly, [24] conducted research on combining the optimal technical indicators with LSTM in stock prediction. The dataset used in this research are the closed prices of HDFC bank, SBI bank and Yes bank between 2016 to 2018.This research aim to find out the optimal technical indicators that have strong positive correlation with the stock prices, and then supply the chosen indicators to LSTM neural network. This research also compared the performance of LSTM with ANN model and SVM model. Before the training process begin, the authors studied the correlation relationship between the technical indicators and the historical stock prices. Based on the correlation analysis, they filtered out the best-voted technical indicators as the optimal inputs for the LSTM, ANN and SVM models. The results show that using optimal technical indicators again produce promising result with the highest accuracy of 64%. The result obtained from LSTM is then compared with ANN and SVM models. After comparing with other models, the results show that using optimal technical indicator in LSTM model outperform than other models such as ANN. They also concluded that MA and MACD technical indicators are highly correlated with the stock prices of 3 banks. The strength of this research is that the relationship between technical indicators and stock prices has been thoroughly studied before training the LSTM model. Supplying relevant input parameters to the LSTM model will increase the performance of model as well as reduce the complexity of model. Another advantage of this study is that LSTM model not only able to capture relevant information contains in technical indicators but also able to capture time series pattern in the data.

Thirdly, [25] has conduct comprehensive research that aim to compare the performance of common machine learning models such as LSTM, ANN and uncertainty-aware attention (UA) in stock price prediction. UA is a model that derived from RNN, which is also a time series prediction model. To test the robustness of the machine learning models in stock prediction, different types of market environments were considered. The authors have selected CSI300 index, S&P500 index and Nikkei255 index. The CSI300 index is considered a dataset representing developing market in China. The S&P500 index represents a well-developed market. The Nikkei255 index of the Tokyo Stock Exchange is a market between developed markets and developing markets. The time span of these 3 stock market indexes is from 2008 to 2016. In each market, some common technical indicators (MACD and ATR) and fundamental indicators (exchange rate and interest rate) were also considered as input data. Based on the result given, the authors found out that the overall performance of ANN is the worst as compared to other models such as LSTM and UA. The main reason is that ANN is not capable of learning complex long-time dependency relationship between time series data. However, all machine learning models achieved low prediction error (RMSE of 30% in average) in the developed market; whereas in developing and less developed markets, time series algorithms (LSTM and UA) performed better as compared to ANN, which was about 40% of RMSE in average. The results reflect that RNN-based architecture could handle time series data properly. A limitation that has been addressed by the authors in this paper is that analysis of technical indicators has not been caried out during the training process. Training the model by giving meaningful inputs will generate better results. There might be many technical indicators that are correlated to the changes of stock prices can be considered. In short, this study again prove that LSTM performed better than ANN model in term of time series data. However, the downside about LSTM is more pre-processing technique need to be implemented before training the LSTM network as compared to ANN. LSTM will only accept 3-dimensional dataset that are arranged in time series behavior. Further, there are two extra hyperparameters need to be considered during the model development which are window size and forecast horizon. Window size is the number of previous days' prices to be considered to predict the closing price of forecast horizon, whereas horizon is the forecast period. For example, the window size is 5 and the forecast horizon is 2 days, this mean that the historical prices of 1-st day to 5-th day will use to predict the price of 7-th day. Having two extra hyperparameter is not a big deal, and it can be solved easily buy running a fine-tunning process.

In short, both papers that compare the performance of various models such as ANN and LSTM have shown consistent result. [25] and [24] has proven that LSTM is able to handle time series data well as compared to the ANN models. This is because LSTM not only capture the hidden pattern underlying the inputs parameters but also capture the long-time dependency relationship between data. However, one situation that can be observed from most papers is that the performance for stock trend prediction is around 50-60% accuracy. It is again proved that stock trend prediction is indeed considered as a difficult problem to be solved. Therefore, this project is expected to achieve similar accuracy as the reviewed papers.

## 2.4 Comparison of the Stock Prediction Techniques

Most of the articles and journals that have been reviewed were focusing on the optimization of technical indicators before training the machine learning models. The approaches in optimizing the technical indicators include correlation analysis, GA technique, and PCA algorithm. Table 2.4.1 summarize the stock indicators optimization approach used by the recent works. Based on the results given by the articles, supplying optimal technical indicators is a significant factor to reduce the overall complexity of machine learning models without compromising the accuracy of the models. They also proved that using optimal technical indicators in machine learning models will possibly generate positive return in trading simulation. Analysis of the relationship between inputs and output variable is extremely important as it will help us identify the relevant features and irrelevant features. When the irrelevant features are not included in the prediction models, noise of the input data will be gradually reduced [26]. Some studies also found that most beginners in neural network think that more input variables they provide to the neural networks, more information it will have and leading better performances [15]. However, supplying irrelevant features not only cause long training time but also might cause some confusion to the machine learning model. Table 2.4.2 summarize the performance of prediction models used in the reviewed papers. From the table 2.4.2, we can see that the prediction model with only optimal indicators as input, [24] can achieve similar accuracy as the prediction model with news and random technical indicators as input [23].

Moreover, most of the articles have shown that time series prediction models such as LSTM and UA performed better than non-time series prediction models like ANN in stock trend prediction. This is because time series prediction models can learn long-time dependency as well as nonlinear relationship in stock data. However, non-time series prediction models are not capable of learning long-time dependency in stock data. Ability of learning long-time dependency as well as sequential information in stock prediction model is an important factor to achieve low prediction error in any market conditions such as well-developed market, developing market and less developed market [25]. Previous studies that aim to explore the generalization potential of LSTM model in stock prediction have shown consistent results. Based on the results given by the articles, LSTM has been proven to be better than ANN in time series prediction. As LSTM model can capture information about the inputs parameters as well as the long-time dependency between the time series data.

Further, one weakness that has been found in most previous works is that most papers used normal cross validation test to measure the performance of the models; However, in time series data such as stock data, considering the time series behavior in validation process is important as it will give the true picture about the performance of the stock prediction models [23]. Using normal cross validation method in time series prediction models such as LSTM is not a problem as the data in time series models already been arranged in time series multivariate form, which means the input data are arranged in 3-dimensional form (the first dimension is the number of samples, the second of dimension is window size of the data and third dimension is the variables for each window). However, it would be a problem if the normal cross validation is used on the non-time series models such as ANN method. For example, the $1^{st}$ day closed price to $5^{th}$ day closed price is used to predict the $6^{th}$ closed price, and prediction of the $6^{th}$ day closed price is heavily depending on $1^{st}$ day closed price to $5^{th}$ day closed price. So, if normal cross validation method is being applied, they will be shuffled randomly, and the time dependency will be gone. Therefore, increasing window cross validation has been introduced to tackle this problem in time series data. The increasing window cross validation will retain the time dependency about the training data and output data. Since we are using LSTM model in this project, the problem of losing the time dependency between input and output data will not happen.

Table 2.4.1 Recent works on Stock Indicators Optimization

| Papers | Machine learning model | Feature extraction technique |
|---|---|---|
| [19] | ANN | GA |
| [20] | - | GA |
| [22] | ANN | PCA |
| [24] | LSTM | Correlation analysis |
| [15] | ANN | Self-organizing map |

Table 2.4.2 Performance of Recent Works on Stock Indicators Optimization

| Papers | Technique | Features | Performance |
|---|---|---|---|
| [19] | GA | Optimal technical indicators | 10-11% annualized profit |
| [20] | GA | Technical indicators | 21-54% average profit |
| [22] | ANN | Technical indicators & fundamental indicators | 59% (Accuracy) |
| [23] | ANN | Technical indicators & Financial news | 50-68% (Accuracy) |
| [4] | LSTM | Open price | 0.01 (error) |
| [24] | LSTM | Optimal technical indicators | 53%-64% (Accuracy) |
| [25] | LSTM | Technical indicators & fundamental indicators | 30-40% (RMSE) |

## 2.5 Existing Software

In this section, 4 applications were chosen for review which are JStock, Trading View, Investing.com and KSLE screener. The review will only focus on the technical stock charting features, watchlists, stock scanning features as well as stock recommendation and stock prediction features.

### **2.5.1 Jstock** [27]

Jstock is a stock customization scanner tool that can help investors to track their stock investment. One unique function that provided by Jstock is that customer is able to customize the stock filtering rules based on the technical indicators given by using drag and drop graphics block. Once the investors have formulized the filtering rules, the system will automatically scan through the stocks that stored in the investors' watchlist in real-time periodically. If there is stock that meets the customized filtering rules, the system will push a notification to investors. By having these scanning and notification features, investors can manage multiple stocks at once and do not need to stay closely with the changes of stock prices. Figure 2.5.1.1 shows that the filtering rules that has been customized and named it with "dhee", which locating at the top right corner.

The strength of this system is that investors can customize their favourite technical indicators and formulate them as the filtering rules. The system also contains their own cloud database which make the stock data retrieval much faster instead of calling from API. Moreover, the system would also automatically perform periodical scanning on the stocks added in watchlist so that the system can closely monitor the changes of the stock price and alert users if the status of stock meets the customized filtering rules.

However, the Jstock system is just a filtering tool, it will only compare the customized stock filtering rules with the real-time stock data and detect which stocks are matching. It does not have the capability of making predictions. Further, there are only 5 technical indicators provided in the system which is not enough for investors to perform technical analysis. Technical analysis trading strategy usually required many kinds of indicators to represent different information such as the trend, momentum, and volatility in order to formulate better trading decisions. Some popular indicators such as MACD and Bollinger bands also not provided in the system. Further, the stock graph presented does not have any responsive feature such as zooming in and out. Figure 2.5.1.3 shows the stock charting provided by Jstock.

Therefore, customization capability and machine learning prediction technique will be added in this project, and some useful features such as push notification will be considered as well. Figure 2.5.1.2 illustrate the notification interface of Jstock.

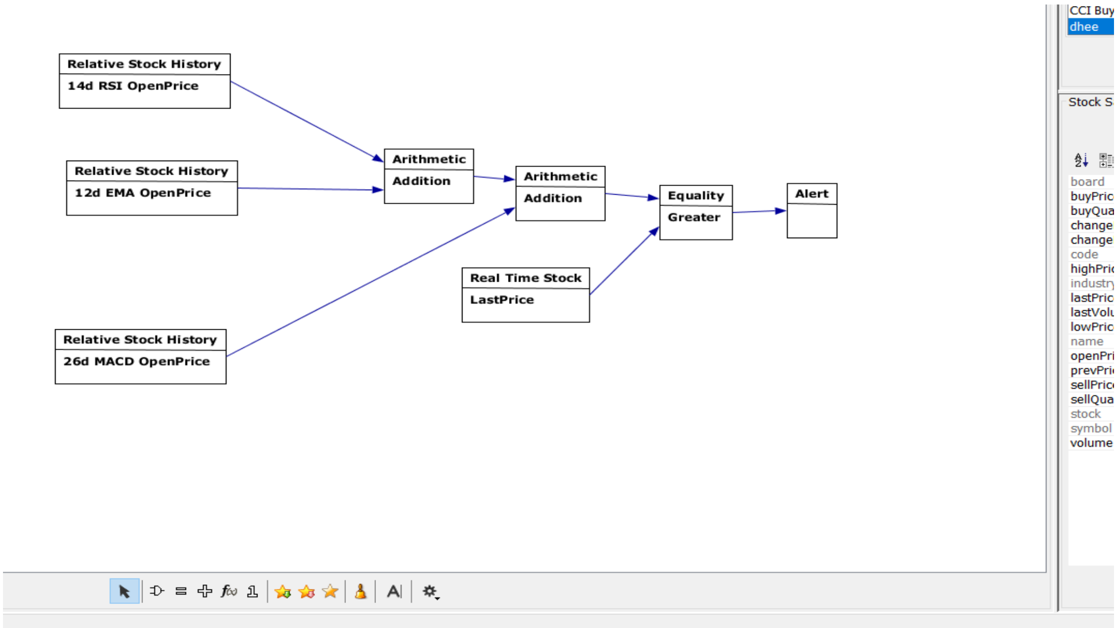



Figure 2.5.1.1 Customization Function of Jstock

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| CCI Sell Signal | 5131.KL | Zhulian Corporation Berhad | 1.92 | 1.95 | 1.95 | 1.95 | 1.92 | 46000 |
| dhee | 5131.KL | Zhulian Corporation Berhad | 1.92 | 1.95 | 1.95 | 1.95 | 1.92 | 46000 |
| MACD Down Trend Signal | 5250.KL | 7-Eleven Malaysia Holdings Berhad | 1.43 | 1.44 | 1.43 | 1.44 | 1.42 | 26300 |
| MFI Up Trend Signal | 5250.KL | 7-Eleven Malaysia Holdings Berhad | 1.43 | 1.44 | 1.43 | 1.44 | 1.42 | 26300 |
| CCI Sell Signal | 5250.KL | 7-Eleven Malaysia Holdings Berhad | 1.43 | 1.44 | 1.43 | 1.44 | 1.42 | 26300 |
| dhee | 5250.KL | 7-Eleven Malaysia Holdings Berhad | 1.43 | 1.44 | 1.43 | 1.44 | 1.42 | 26300 |

Figure 2.5.1.2 Notification Interface of Jstock


Figure 2.5.1.3 Stock Charting with Technical Indicators of Jstock

### 2.5.2 Trading View [28]

Firstly, the technical stock charting features provided in Trading View is extremely responsive and advanced. Figure 2.5.2.1 show the graphical interface of the Trading View platform. Trading view also provides different kinds of drawing tools such as drawing lines and price range boxes to ease the investors in chart analysis. This feature is considered user-friendly as it allows users to perform many flexible functions on the graph, and users can make any notes on the graph based on his/her needs. Figure 2.5.2.2 illustrate the graph with some drawing tools applied on it. Moreover, the system also provides stock prediction and decision recommendation feature. This feature will predict whether to buy or to sell the stock in the next time frame. The recommendation decision is formulated by summarizing different signals generated from multiple technical indicators. Figure 2.5.2.3 show the buying and selling decisions generated from multiple technical indicators. Different kinds of technical indicators will generate different signals; If more buying decisions are generated, then the system will suggest users to buy or sell otherwise. For example, according to the figure 2.5.2.3, an overbought situation will result when RSI value is larger than 70, which indicating the stock price is likely to drop in next few timesteps and thus giving a sell signal. Further, instead of performing technical analysis prediction on a single stock one by one, Trading View also integrate filtering and scanning features with the stock prediction and decision recommendation feature. By applying customized rules on the filtering and scanning feature, users can get the predicted decision for all the matching stocks at once. Figure 2.5.2.4 shows the scanning plus technical analysis prediction feature. Further, the system also provides more than hundreds of technical indicators for investors to choose from. Moreover, Trading View also allows users to upload their own machine learning scripts to make stock prediction. Figure 2.5.2.5 shows the stock prediction based on machine learning algorithms uploaded by a user named "capissimo".

In contrast, the only weakness observed in this system is that the stock prediction feature provided by the platform is only based on the formula of the technical indicators instead of machine learning algorithms. The prediction of stock using machine learning as shown in figure 2.5.2.5 were manually developed by the platform user named "capissimo", and it is not created by the system provider by default. Thus, this feature only limited to the users who subscribed "capissimo". This is allowed as Trading View provide a feature that allow users to upload any kind of scripts (including AI script) to the platform to make stock prediction. Figure 2.5.2.6 shows the section to add the coding script in the platform. Therefore, to improve the weakness, this project will develop own AI algorithm to make stock prediction, which this AI algorithm will be available to all users especially non-technical users; thus, users do not have to create own AI scripts in the platform.


Figure 2.5.2.1 GUI of Trading View Investing Website


Figure2.5.2.2 Stock Charting with Users Customized Notes


Figure 2.5.2.3 Stock Prediction and Decision Recommendation on Technical Analysis


| TICKER 8274 MATCHES 3 ITEMS | MOVING AVERAGES RATING | LAST | SMA20 | SMA50 | SMA200 | BB UP | BB |
|---|---|---|---|---|---|---|---|
| LSPR D LARKSPUR HEALTH ACQUISITION CORP. | Strong Buy | 9.95 USD | — | — | — | — | |
| DUET D DUET ACQUISITION CORP. | Strong Buy | 9.89 USD | — | — | — | — | |
| KNSW D KNIGHTSWAN ACQUISITION CORPORATION | Strong Buy | 9.99 USD | — | — | — | — | |
| AKAN D AKANDA CORP. | Strong Buy | 8.64 USD | — | — | — | — | |
| IVCB D INVESTCORP EUROPE ACQUISITION CORP I | Strong Buy | 9.98 USD | — | — | — | — | |
| PPGH D POEMA GLOBAL HOLDINGS CORP. | Strong Buy | 10.30 USD | 9.961 B | 9.9312 B | 9.8506 B | 10.13419353 S | 9.7878 |
| TETE D TECHNOLOGY & TELECOMMUNICATION ACQUISITION COR... | Strong Buy | 9.98 USD | — | — | — | — | |
| XFIN D EXCELFIN ACQUISITION CORP | Strong Buy | 9.97 USD | 9.89912 B | — | — | 9.9720203 — N | 9.826 |
| SCMA D SEAPORT CALIBRE MATERIALS ACQUISITION CORP. | Strong Buy | 9.96 USD | 9.9105 B | — | — | 9.96965235 — N | 9.8513 |

Figure 2.5.2.4 Stock Prediction and Decision Recommendation on Technical Analysis with Filtering Feature

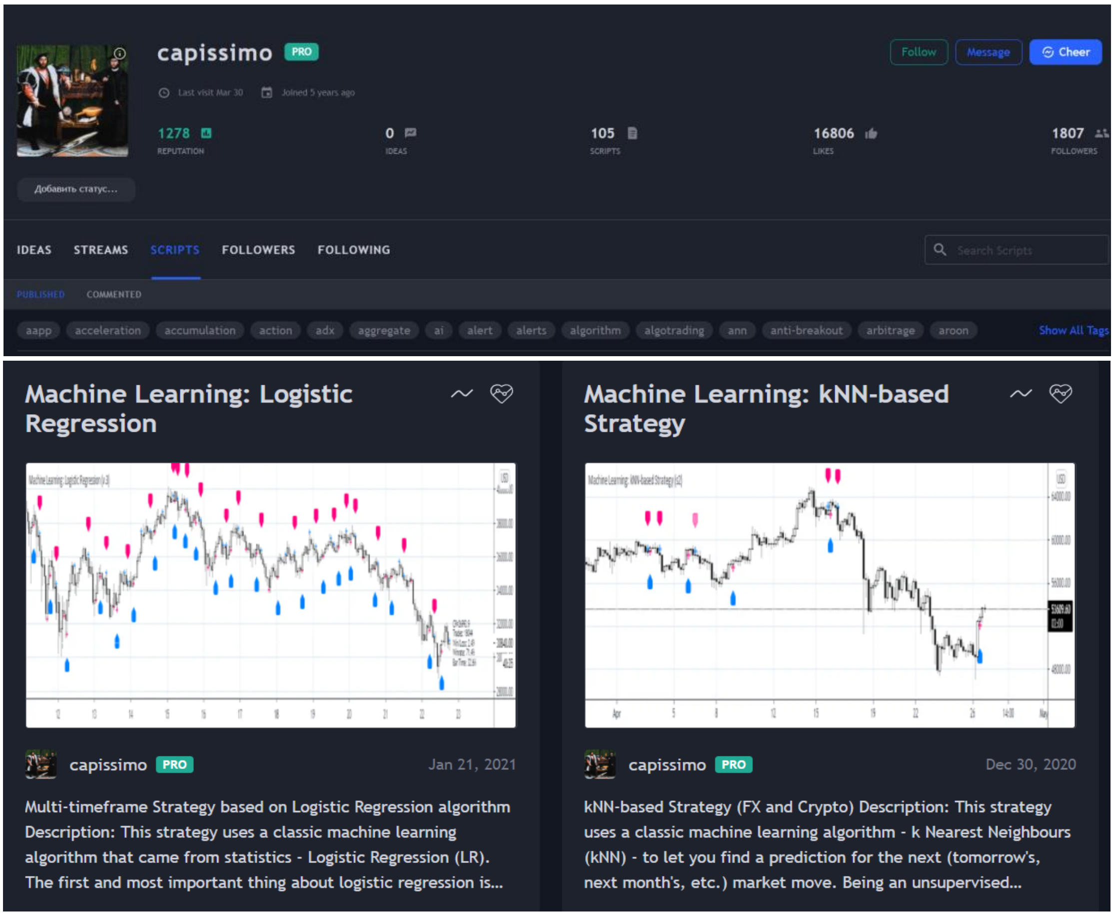



Figure 2.5.2.5 Stock Prediction and Decision Recommendation Using Machine Learning Algorithms

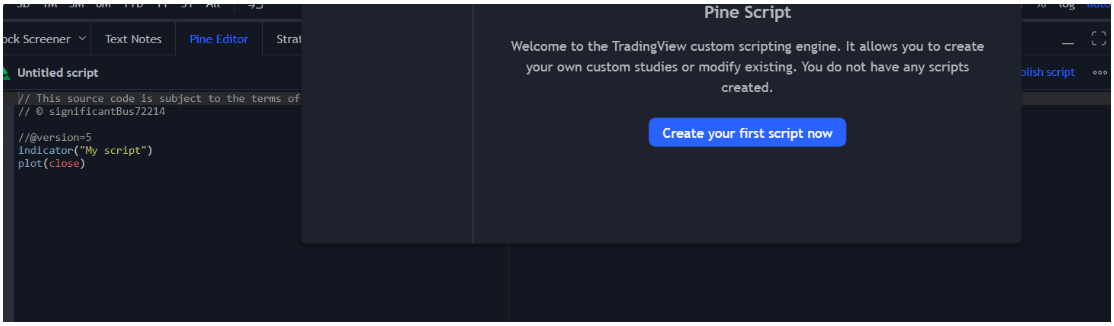



Figure 2.5.2.6 Place to Add Any Customized Scripts for Making Prediction

### **2.5.3 KLSE Screener** [29]

KLSE screener provide a similar feature as the JStock and Trading View which is the filtering and scanning feature. Investors can scan and filter the stocks based on the customized rules in fundamental analysis and technical analysis. Figure 2.5.3.1 show the scanning feature provided in the KSLE. After investors have defined a customized rules in fundamental and technical analysis, then the system will display the matching stocks. Figure 2.5.3.2 shows the filtered stocks after applying the filtering and scanning function. This feature is very convenient and user-friendly as users do not have to search the preferred stocks one-by-one, instead it can be done with a single click. Further, the stock charting provided by the system is powered by the Trading View stock charting function. Figure 2.5.3.3 shows the charting function in KSLE screener. Therefore, KSLE stock charting features will retain all the benefits that the Trading View stock charting function have.

However, one weakness has been found is that the system does not provide any prediction features not even the simple prediction based on the traditional technical analysis like [28]. Further, the scanning feature provided by the system might be confusing for non-technical users as the filtering rules are based on some advanced fundamental analysis knowledge. In order to make full use of the scanning feature, users must have some understanding on the fundamental of financial and accounting theories.

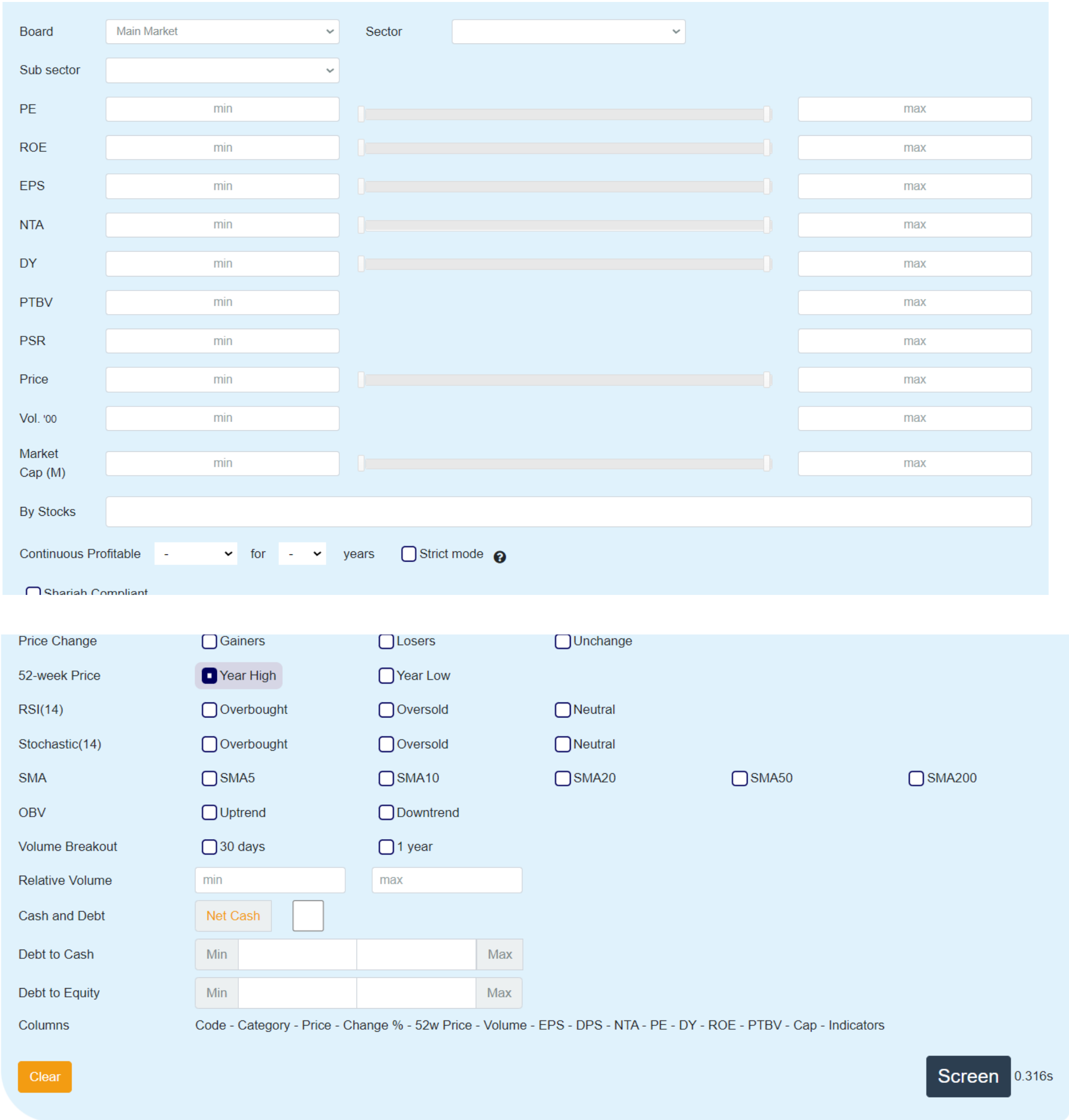
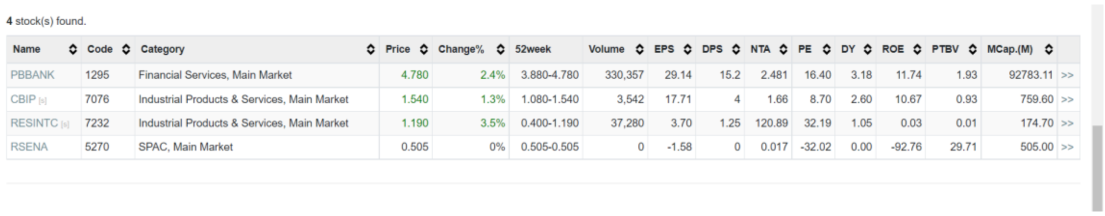



Figure 2.5.3.1 Stock Filtering and Scanning Feature



4 stock(s) found.

| Name | Code | Category | Price | Change% | 52week | Volume | EPS | DPS | NTA | PE | DY | ROE | PTBV | MCap.(M) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PBBANK | 1295 | Financial Services, Main Market | 4.780 | 2.4% | 3.880-4.780 | 330,357 | 29.14 | 15.2 | 2.481 | 16.40 | 3.18 | 11.74 | 1.93 | 92783.11 | >> |
| CBIP [s] | 7076 | Industrial Products & Services, Main Market | 1.540 | 1.3% | 1.080-1.540 | 3,542 | 17.71 | 4 | 1.66 | 8.70 | 2.60 | 10.67 | 0.93 | 759.60 | >> |
| RESINTC [s] | 7232 | Industrial Products & Services, Main Market | 1.190 | 3.5% | 0.400-1.190 | 37,280 | 3.70 | 1.25 | 120.89 | 32.19 | 1.05 | 0.03 | 0.01 | 174.70 | >> |
| RSENA | 5270 | SPAC, Main Market | 0.505 | 0% | 0.505-0.505 | 0 | -1.58 | 0 | 0.017 | -32.02 | 0.00 | -92.76 | 29.71 | 505.00 | >> |

Figure 2.5.3.2 Stock Filtering and Scanning Results


Figure 2.5.3.3 Stock Charting Function Powered by Trading View

### **2.5.4 Investing.com** [30]

Lastly, stock charting provided by the Investing.com also powered by Trading View platform, thus similar advantages from Trading View would also be preserved in the Investing.com. The figure 2.5.4.1 shows the stock charting function provided by Investing.com. Further, Investing.com also provide stock prediction and decision recommendation features based on technical analysis like Trading View. The figure 2.5.4.2 shows the Stock prediction and decision recommendation based on technical analysis. Investing.com also provide stock filtering and scanning feature. The figure 2.5.4.3 shows the filtering and scanning function. After the users have applied some customized rules, the technical analysis prediction will then be applied on the matching stocks. In short, most functions provided in Investing.com are almost similar as Trading View, thus same strengths and weakness that has been mentioned in the Trading View are also observed in the Investing.com


Figure 2.5.4.1 Stock Charting Function Powered by Trading View

**Technical Indicators »** Apr 01, 2022 08:05PM GMT

| Name | Value | Action |
|---|---|---|
| RSI(14) | 45.827 | Neutral |
| STOCH(9,6) | 19.713 | Oversold |
| STOCHRSI(14) | 54.786 | Neutral |
| MACD(12,26) | -0.620 | Sell |
| ADX(14) | 42.657 | Sell |
| Williams %R | -64.474 | Sell |
| CCI(14) | -47.1567 | Neutral |
| ATR(14) | 1.4450 | High Volatility |
| Highs/Lows(14) | 0.0000 | Neutral |
| Ultimate Oscillator | 43.952 | Sell |
| ROC | -2.059 | Sell |
| Bull/Bear Power(13) | -1.4480 | Sell |

Buy: **0** Sell: **6** Neutral: **4**
**Summary:STRONG SELL**

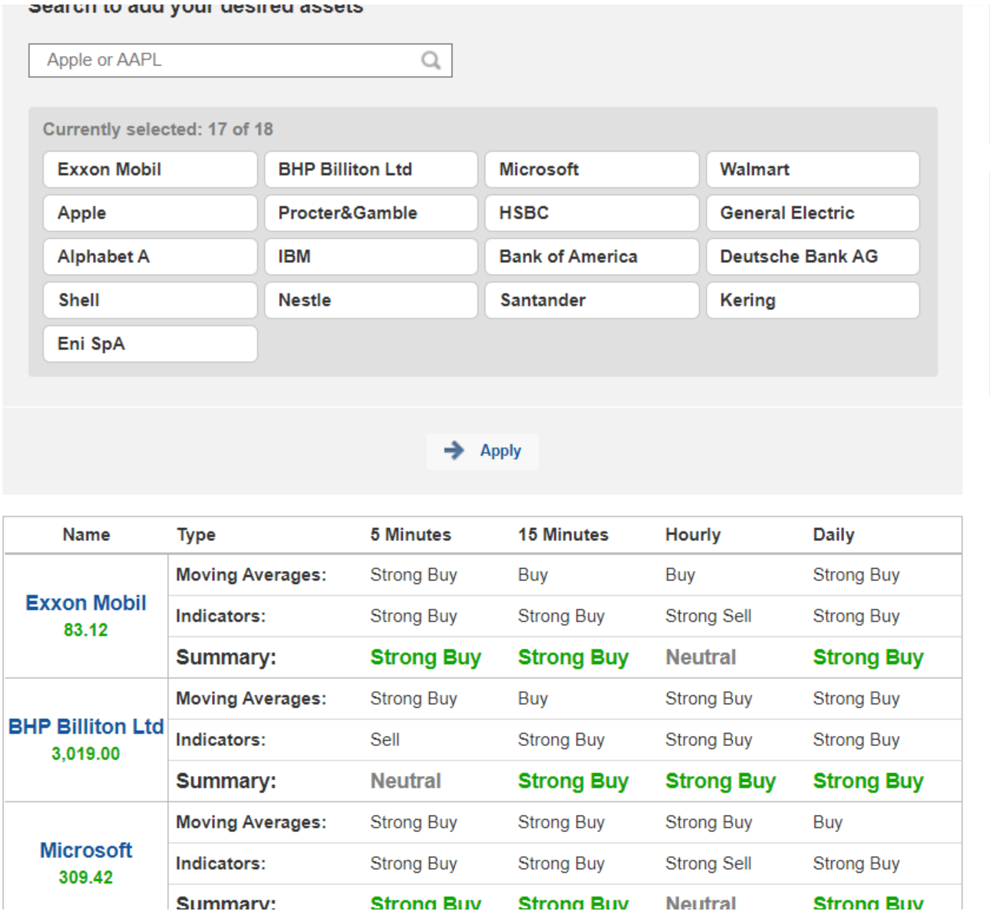
**Moving Averages »** Apr 01, 2022 08:05PM GMT

| Period | Simple | Exponential |
|---|---|---|
| MA5 | 172.87 Buy | 173.50 Buy |
| MA10 | 173.94 Buy | 174.03 Buy |
| MA20 | 175.70 Sell | 174.80 Sell |
| MA50 | 175.03 Sell | 173.49 Buy |
| MA100 | 168.44 Buy | 170.68 Buy |
| MA200 | 165.01 Buy | 168.16 Buy |

Buy: **9** Sell: **3**
**Summary:BUY**

**Disclaimer:** Fusion Media would like to remind you that the data contained in this website is not necessarily real time nor

Figure 2.5.4.2 Stock Prediction and Decision Recommendation on Technical Analysis




| Name | Type | 5 Minutes | 15 Minutes | Hourly | Daily |
|---|---|---|---|---|---|
| Exxon Mobil 83.12 | Moving Averages: | Strong Buy | Buy | Buy | Strong Buy |
| | Indicators: | Strong Buy | Strong Buy | Strong Sell | Strong Buy |
| | Summary: | Strong Buy | Strong Buy | Neutral | Strong Buy |
| BHP Billiton Ltd 3,019.00 | Moving Averages: | Strong Buy | Buy | Strong Buy | Strong Buy |
| | Indicators: | Sell | Strong Buy | Strong Buy | Strong Buy |
| | Summary: | Neutral | Strong Buy | Strong Buy | Strong Buy |
| Microsoft 309.42 | Moving Averages: | Strong Buy | Strong Buy | Strong Buy | Buy |
| | Indicators: | Strong Buy | Strong Buy | Strong Sell | Strong Buy |
| | Summary: | Strong Buy | Strong Buy | Neutral | Strong Buy |

Figure 2.5.4.3 Stock Prediction and Decision Recommendation on Technical Analysis with Filtering Feature

## 2.6 Comparisons of the Existing Systems

Table 2.6.1 show the feature comparisons among the reviewed existing systems. Based on the table 2.6.1, watchlist, stock scanning and stock charting feature seems to be the essential features that must be provided in the trading platform. Therefore, this project would also implement the stock scanning, stock charting with technical analysis capabilities and watchlists features. Next, the 4 trading platforms that have been reviewed do not provide the prediction of stock trend using machine learning algorithms by default, instead they utilize the basic technical analysis calculation to formulate the prediction decisions. Thus, this project will include the stock prediction using machine learning algorithm as the core and default feature for all users.

Table 2.6.1 Feature Comparison among Existing Systems

| **Features** | **JStock[27]** | **Trading View[28]** | **KLSE screener[29]** | **Investing.com [30]** |
|---|---|---|---|---|
| Stock Charing with technical analysis capabilities | ✓ | ✓ | ✓ | ✓ |
| Stock recommendation and prediction based on technical analysis indicators | ✗ | ✓ | ✗ | ✓ |
| Stock scanning and filtering feature | ✓ | ✓ | ✓ | ✓ |
| Watchlist feature | ✓ | ✓ | ✓ | ✓ |
| Number of technical indicators provided | **5** | > **hundreds** | > **hundreds** | > **hundreds** |
| Stock prediction using machine learning algorithms | ✗ | ✓ *Users are needed to develop their own machine learning algorithm scripts and upload to the platform | ✗ | ✗ |

# Chapter 3 System Methodology/Approach

## 3.1 Project Development

The figure 3.1.1 below illustrates the workflow of the stock indicator scanner customization tool development.

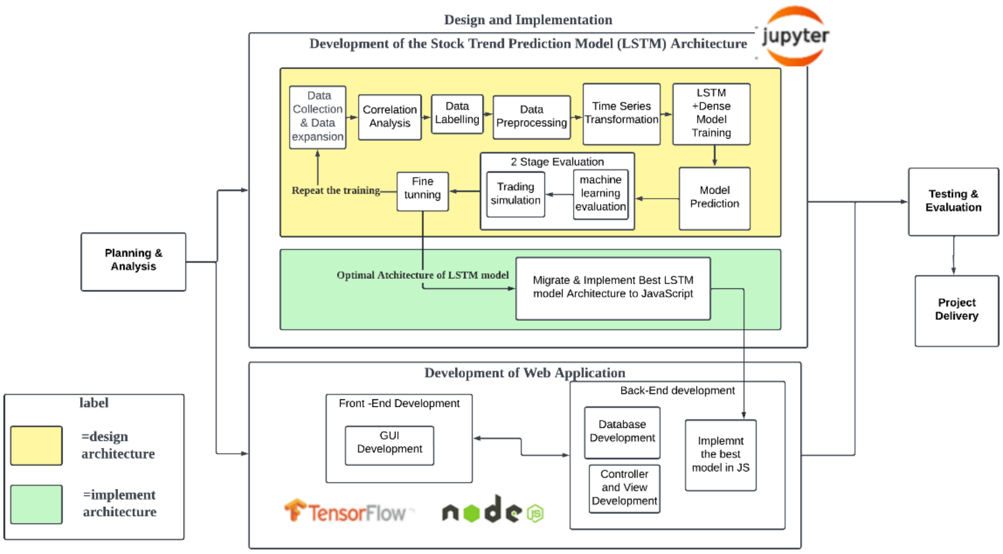



Figure 3.1.1 Workflow of the Stock Indicator Scanner Customization Tool Development

First, to understand the problem domain, a comprehensive critical review of the literatures will be carried out. In the literatures review, some important doubts can be answered such as market predictability, the predictive inputs used in most works as well as the machine learning techniques in stock price prediction. After having reviewed the literature papers, some weaknesses have been found such as most stock prediction models are static models and so on. Other than weakness, some techniques that lead to low prediction errors in stock price prediction models also have been noted.

After reviewing the existing approaches, problem statements were formulated. Project scopes and project objectives were then created based on the problem statements that had defined. To ensure the project can be realized, exploration of technologies and methods was done. TensorFlow JS framework was found, and it allows programmers to run and build machine learning model in the JavaScript platform in real-time. By using TensorFlow JS framework, investors are allowed to customize their favourite trading rules as the input parameters of machine learning model and train the model on the fly.

A review on common technical indicators used in most Malaysia stock trading systems such as Trading View, KLSE Screener and Investing.com was done. These stock trading systems normally will offer a list of common technical indicators to the traders. Therefore, these common technical indicators will be included in this system. This is to ensure the proposed system will provide the technical indicators that match users' need.

The next step will be the analysis of the available dimension reduction and feature extraction algorithms that are suitable for stocks prediction. Feature engineering process is a significant step that must be taken care in this project. This is because the main goal of project is to develop a dynamic model that can be trained in real-time, and thus complexity of dataset must be low.

Next, the design and implementation for this project will be divided into 2 main phases which are the development of LSTM model architecture and the development of web application which they will be carried out parallelly in Python and NodeJS environment respectively. The optimal stock prediction model architecture that is well-suited for 100 stocks will be designed in Python Platform (yellow box). After the optimal architecture has been designed, it will be migrated and implemented to the Node JS backend of the application (green box). The reason why the architecture of the stock prediction model is developed and designed using Python instead of TensorFlow JS machine learning framework in Node JS is that Python is a well-developed and advanced machine learning platform, and it provides many convenient libraries that can ease programmers in pre-processing/post-processing the data. However, there are no such libraries in TensorFlow JS and thus all operations have to be achieved manually using JavaScript arrays and loops. Furthermore, the process of designing the optimal LSTM model architecture that is well-suited for all 100 stocks is considered a work-intensive process as many experiments are needed to be run before the best architecture can be found. Therefore, the architecture design process is better to be carried out in a platform that provide many convenient libraries that can ease the programmers' workload. But, after the optimal architecture of the LSTM model has been designed in python, we will still need to migrate and translate the machine learning codes from

Python to JavaScript using TensorFlow JS as the whole web application will be running in JavaScript.

After the complete project has been developed and designed, testing will be carried out to ensure all functions is working well, ensure no bugs or errors present in the system, and ensure the application meets the objectives. Finally, the project is ready for delivery. Figure 3.1.2 show the timeline of the project development.

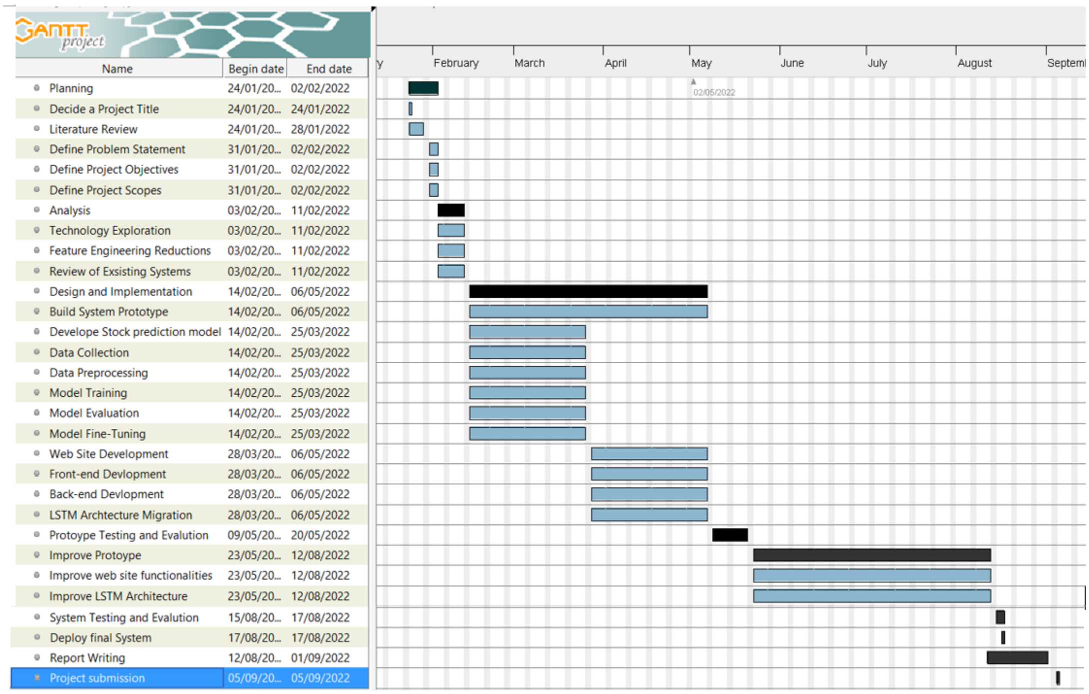



Figure 3.1.2 Timeline of the Proposed System Development

## 3.2 Workflow of Stock Prediction Model Architecture Development

In the development of the stock trend prediction model architecture, there are mainly 6 phases involved. In this process, many possible of technical indicators combinations will be used in studying and developing the optimal LSTM model architecture so that we can closely simulate the real-world situation where many investors will use different indictors combinations as the input to the real-time model. Thus, the goal of this process is to figure out the optimal LSTM architecture design that can accommodate and achieve average good results across all 100 stocks with different kinds of input indicators combinations.

### 3.2.1 Data Collection and Data Expansion

The first step of developing a prediction model is to collect stock data. Historical prices of stocks listed in the Kuala Lumpur Stock Exchange (KSLE) will be collected using Yahoo finance API provided by python platform. The data provided by Yahoo finance API includes open price, close price, high price, low price, volume and adjusted close price. After that, feature expansion will be carried out from these data. Some famous technical indicators such as moving average (SMA) and reflective strength index (RSI) will be calculated based on the historical stock prices. Different combinations of technical indicators will be used in each run of experiments. Calculating the technical indicators manually by code is time-consuming and not effective. Luckily, python provide a useful library called Ta-lib which can automatically generate the required technical indicator values by just calling a function. By using the Ta-lib library, human mistake in calculation can be avoided and much time can be saved.

### 3.2.2 Correlation Analysis and Data Labelling

After the inputs (historical stock prices + some technical indicators) have been formulated, correlation analysis will be carried out to study the correlation between stock indicators and stock closing prices. Based on the reviewed papers, studying the features is an important step to reduce the noise present in the data as well as reduce the dimension of the data [15] [19] [20] [22] [24]. Therefore, correlation analysis will be carried to ensure only the correlated features are included in the input of the model. After the correlation analysis, data labelling process will be carried out. The stock trend prediction problem will be formulated as the binary classification problem which the data will be labelled either with "Up" or "Down". The labelling condition is provided in the figure 3.2.2.1. According to the formula, the data will be labelled with "Up" if the price of the forecast horizon is greater than the close price of its previous time step. For example, the 1$^{st}$ day to 5$^{th}$ days' stock prices will be used to predict the close price of 6$^{th}$ day (window = 5, forecast horizon =1). If the 6$^{th}$ day close price is larger than the 5$^{th}$ day close price, then the data will be labelled with "Up" or "Down" otherwise. Since text cannot be included in the calculation during the model training, “1” will be used to represent “Up” and “0” will be used to represent “Down”.

$$Label_{2CL}(p_i) = \begin{cases} 'Up' \quad , \ if \ (C_{t+s} - C_t)/C_t > 0; \\ 'Down', \ if \ (C_{t+s} - C_t)/C_t \leq 0, \end{cases}$$

Figure 3.2.2.1 Condition of Labelling [21]

### 3.2.3 Data Pre-Processing and Time-Series Transformation

After the data labelling, Min Max scaling will be used to scale the data into the range of 0 to 1. Scaling the data is important as neural network tend to work better with the data that have the range of 0 to 1 [15]. One mistake that was commonly made by the beginners is that they tend to compute the scaling factors on the whole dataset instead of computing in sperate manner. In sperate manner means the whole dataset will first be divided into 2 sets namely train set and test set, and min-max scaling factors is calculated based on the train set only, then the transformation is applied in both train and test set [15]. It is important to transform the dataset in the sperate manner as test set is representing the real-world unseen data. If min-max scaling factor is computed from the whole dataset, then the model will have look-ahead bias on the test data as model will have some knowledge about the scale of the test set which the performance of the model is no longer reliable. Therefore, during the pre-processing stage, dataset will be split into training set and test set with the splitting ratio of 70%/30%. After that, min-max scaling factors will be calculated based on the train set, and then the computed scaling factors will be used to transform both training and testing set. The figure 3.2.3.1 illustrates the method of min-max scaling carried out in this project.

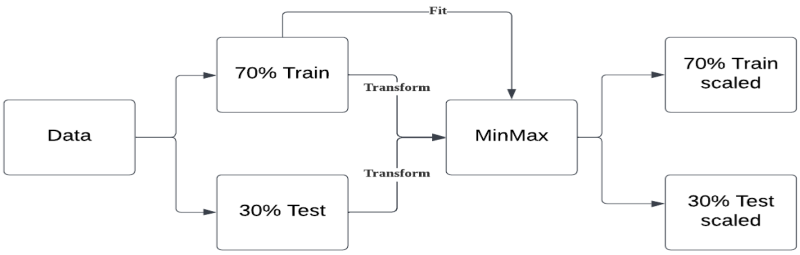



Figure 3.2.3.1 the Min-Max Scaling Process

After that, time-series transformation will be applied. Time-series transformation is needed as LSTM network only accept the input in 3D format which are in (samples, time-step, variable) form. The format of time-series is illustrated in figure 3.2.3.2.

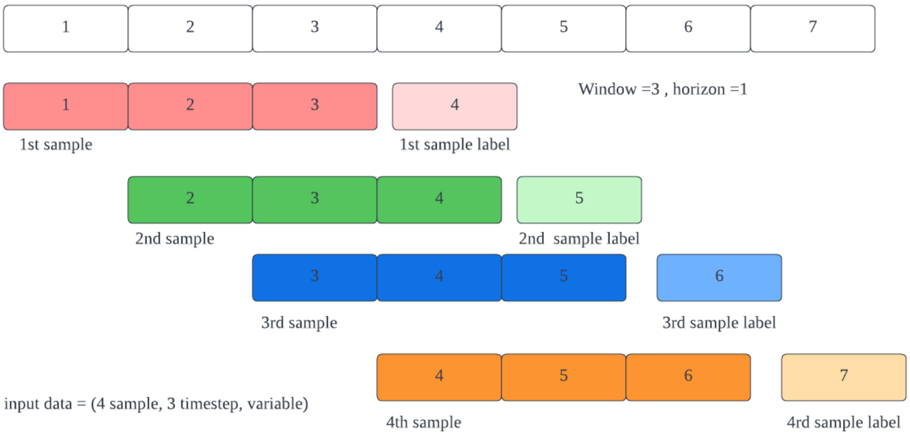



Figure 3.2.3.2 Time Series Input Format

The first dimension of the input will be the number of samples or batches, the second dimension will be the number of time steps in each sample, and the last dimension will be the number variables in each of the time step. Based on the figure, it is showing dataset with 4 samples, 3 timestep, and 1 variable.

### 3.2.4 LSTM Model Training and 2 Stage Evaluation

After the input data have been pre-processed, LSTM model training will be started. After the training, the LSTM model will be evaluated using testing data. Using testing data to evaluate the model will indeed give a reliability measure of the model performance as the model does not have any knowledge about the testing data during the training process. There are 2 stages of evaluation will be carried out namely machine learning evaluation and trading evaluation. In machine learning evaluation, the prediction accuracy of the LSTM model will be computed. Prediction accuracy represents the correctness of the prediction made by the LSTM model on the testing data. In trading evaluation, the LSTM model will be evaluated as a trading system. If the LSTM model made a “Up” prediction in any moment in the testing data, a “buy” operation will be made until the next moment [31]. For example, the LSTM model made a “Up” prediction in moment X, and then the LSTM model will make a “buy” operation on the moment X and sell it in moment X+1, earning price difference of (current price of X+1 – current price X). If it is a correct prediction, then a positive return will be gained through this particular action. Some popular portfolio evaluation metrices such as cumulative return, Sharpe ratio and Maximum drawdown will be calculated to measure the performance of the proposed model in stock trading simulation. Besides that, the trading performance of the LSTM model will also be compared with Buy-and-Hold and Optimistic Strategy. These 2 strategies are the most fundamental strategies used in stock market; thus, the proposed model is expected to perform better than these 2 strategies. Buy-and-Hold strategy will buy the stock at the beginning of the testing data period and sell it at the end of the testing data period [31]. The figure 3.2.4.1 shows the illustration of Buy-and-Hold strategy. Optimistic strategy will buy the stock at the current time step (n) and sell it at the next time step (n+1) if the current time step (n) stock price is higher than previous time step (n-1) stock price (today’s price has increased as compared to yesterday’s price). The Optimistic strategy expects the stock price will continue to increase tomorrow [31]. The figure 3.2.4.2 shows the illustration of Optimistic strategy.

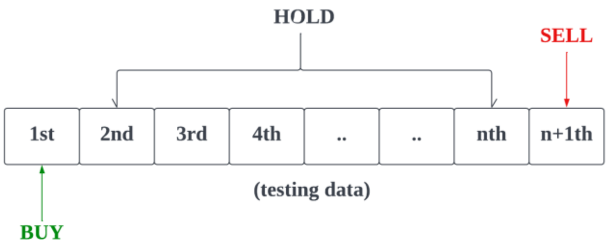



Figure 3.2.4.1 the Illustration of Buy-And-Hold Strategy


Figure 3.2.4.2 the Illustration of Optimistic Strategy

### 3.2.5 Fine-Tuning

In this phase, model training process and model evaluation process will be repeated multiples times with different combination of indicators as model input so that an architecture design of LSTM model that work well in general can be determined. The model will be further fine-tuned to find out the best hyperparameters such as the number of time step, forecast horizon, and decision threshold. After multiple runs of fine-tuning process, the optimal model architecture will be determined.

During the fine-tuning process, the best combination of technical indicators that work well in most of the stocks will be noted so that the system can use this best combination of technical indicators to prepare an offline model that act as the reference models to all users. Offline model is designed for new user who has no idea to customize what indicators to use in the dynamic model and insist to get an immediate prediction for a particular stock. The offline model will be trained in advance and then be stored in the database so that users can get the immediate prediction directly from the offline model stored in database, but the downsides are that the offline model must be updated manually by programmers when new data is reached, and the users cannot customize the input of the offline model. Offline models are just reference models for newcomers who have no idea to customize what indicators as input of the models. For those experienced investors who know what indicators to customize as input of the models, they can ignore the offline models and go ahead to build his/her own customized models.

### 3.2.6 Migrate and Implement Optimal Architecture Design into Node Js Backend

After the fine-tuning process, the optimal LSTM model architecture that is well-suited for 100 stocks will be produced, and it will then be migrated to Node JS backend by using TensorFlow JS framework. After the migration process, the web application is capable of training stock prediction models with optimal architecture design in real-time.

## 3.3 Workflow of Website Development

In this stage, there are two main steps which are front-end and back-end development. In front-end development. The visual looks of the website will be designed which will also include the design of the candlestick stock charting. The stock charting will be developed using Lightweight Chart JS provided by Trading View investment platform [28]. Further, in back-end development, there will be 3 main processes. The first process is to design the database system used on the website. The second process is to develop the controller classes to handle the incoming http request from front-end, and the view classes to formulate the dynamic contents to be displayed in the client-side. Finally, is to migrate the best architecture design of LSTM model from python to the JavaScript backend. After the optimal architecture design of LSTM model has been produced in python environment, the same architecture design of LSTM will be manually implemented in the JavaScript backend using TensorFlow JS, array, and loops. After translating the LSTM model code from python to JavaScript platform, the web application will be equipped with machine learning capabilities

## 3.4 System Functionalities of Stock Indicator Customization Tool

The figure 3.4.1 illustrate the system functionalities of the proposed system. The are 3 major functions in this system namely stock prediction and training function, in-depth analysis indicator recommendation function and model template customization function. One of the project objectives is to allow investors to customize the inputs of the prediction model and it is achieved through the model template customization function. In this function, investors can customize their favourite indicators and save it as a template. Template is mainly to define what technical indicators will be used by the prediction model in the training process. Different templates will contain different combination of technical indicators. After users have created templates, users can use these templates to start a stock model training process. Stock model training and prediction is the second major function provided by the system. To train a prediction model for a specific stock, users must first choose a template for the model to train on, or else the model will not know what input indicators to use in training. After a template has been chosen, the training process for a stock will begin. After model training done, the model will automatically be saved in the database so that users can reconstruct this trained model whenever users want to make a prediction in the future. Of course, if users do not satisfy with the model performance, users can choose to retrain the model. After the customized model has been trained, the model will be evaluated using prediction accuracy rate as well as be evaluated from the trading point of view. This is because users are hard to judge the usability of the prediction model in actual trading environment based on the accuracy alone, and therefore trading simulation feature is provided to users. Trading simulation is mainly to measure the trading performance of the model by deploying it in the past three years data (testing data). This can effectively measure how much profit will be earned if the model is being used as the only tool for decision making in the past three years. To ensure the evaluation in trading simulation is indeed reliable and not biased, the prediction model will not be trained with the past three years data. By doing so, the model will be predicting and performing trading operations on the unseen data which will make the evaluation of the model more convincible. In best case scenario, the model will outperform and generate higher positive return over the past three years. Further, a basic portfolio report that includes some essential performance metrics such as cumulative return, Sharpe ratio will be provided after the trading simulation. Other than that, a more detail, advanced and professional portfolio report that is mainly designed for users who are portfolio managers will also be provided in the system. In-depth analysis is the third major function provided to users. As mentioned, templates defined the types of indicators that will be used in the model training. After users have selected a template for training, users might be wondered if the current selection of the indicators is indeed helpful to the model generalization, will users choose the wrong indicator that will affect the performance of the model, can system suggest a better combination out of the indicators that defined in the selected template. This is where in-depth analysis feature come into handy. Instead of directly using all the technical indicators defined in the template in the model training, the system will perform in-depth analysis on all possible indicators combinations and utilize the best combination to perform stock prediction. In-depth analysis will compute all possible combinations of technical indicators based the original indicators that defined in a specific template, and each combination will run for a training, and the model performance for each combination is noted. Based on the accuracy obtained by each combination, the system will recommend users the best combination of technical indicators that yields the highest accuracy. Last but not least, instead of training or predicting the stocks that stored in watchlist one by one separately, the system will also provide a convenient feature to train and predict multiple stocks together at once with a single click. After the training of multiple stocks completed, the prediction result and the evaluation results for all watchlist stocks will be displayed at a centralized place. By doing so, users are able to monitor their preferred stocks in a single place rather than monitoring them separately. The multiple stocks training process will be carried out parallelly and thus the time taken to train multiple stocks will be slightly longer than single stock training.


Figure 3.4.1 System Functionalities of the Stock Indicator Scanner Customization Tool

### 3.4.1 Design of Stock Trend Prediction for A Single Stock Procedure

In this project, the prediction model for single stock will be categorised into 2 main groups namely offline static model and online dynamic model. Offline static model is the model that have already been trained in the offline environment in advance with the fixed types of stock indicators that are predetermined by programmers, whereas online dynamic model is the model that has the same architecture as the offline static model, but the technical indicators used in this model would be dynamic and depends on the users' choices. Online model will be built upon users' requests on the website in real-time. Further, Offline models are shared by everyone including the non-registered users, whereas online dynamic models are not shared among registered users as different users will customize different type of dynamic models (different indicators used as input parameters). Offline model can be imagined as a global stand-alone API service that is freely available to everyone which everyone can make a call to it to get the today prediction, but no one can make any changes on the API. However, online dynamic models are models that were trained in real-time by specific users which the trained model will be stored in the database under the specific users' account and thus the trained models only belong to that specific user and that specific user can make further changes on the model such as updating the model. The reason of providing the offline static model to the users is that it will act as a reference model for users, and users do not have to go through any model training process and can get an immediate prediction from the system, but the downside is users can't change the indicators input of the offline model as the indicators input used by offline model is predetermined by programmers. The reasons of creating offline model will be thoroughly discussed in chapter 4.5.

In this procedure, the first step is to select a specific stock to analyse. After users selected a stock for prediction, a stock charting (candlesticks) with technical analysis capabilities will be provided to the users so that users can perform some basic technical analysis on the stock chart given. Stock charting will give users a better idea on the recent movement of the stock prices. Further, there will be 3 other options regarding the stock prediction for users to choose from. The options are predicting the stock trend using offline static model, predicting the stock trend using the previous trained online customized model, or training a new customized model in real-time. If the customers select the first option which is predicting the stock trend using offline static model, then the system will make a request to the server-side to get a prediction from reference model. Moreover, if the second option is chosen, then the system will also make a request to server-side to get a prediction, but the only difference is that the prediction will be made by the users' specific customized model that has been trained by the users earlier. If customers choose to train a new customized model, then customers must choose which kind of template that the model will used. The templates can be added by users in other page named "modelling". By having a template chosen, the system will start running the training process which will take roughly 3 to 5 minutes. Once the training is done, the system will display the prediction accuracy, predicted decision, probabilities of stock growing in next time step, and profit gained by the model prediction. After the customized model has been trained, customers can choose to run the trading simulation function to evaluate the model as an actual trading system. A simple interactive trading simulation report will then be generated. In this report, all buying operations made by the model will be displayed in graph and table form. If some users especially the users who are portfolio managers think that the given report is too simple or the performance metrices provided is not enough, users can choose to generate an even more detail and in-depth analytic portfolio report. Further, customers can choose to run the indicators recommendation feature in the in-depth analysis section. Finally, the trained model will be saved automatically in the database so that customers do not need to train the model again for future prediction, instead the trained model can give an immediate prediction based on the future prices. Of course, users can choose to retrain the model if the users are not satisfied with the performance of the model or update the model with the new reached data. Figure 3.4.1.1 illustrate the workflow of the single stock prediction and training feature.

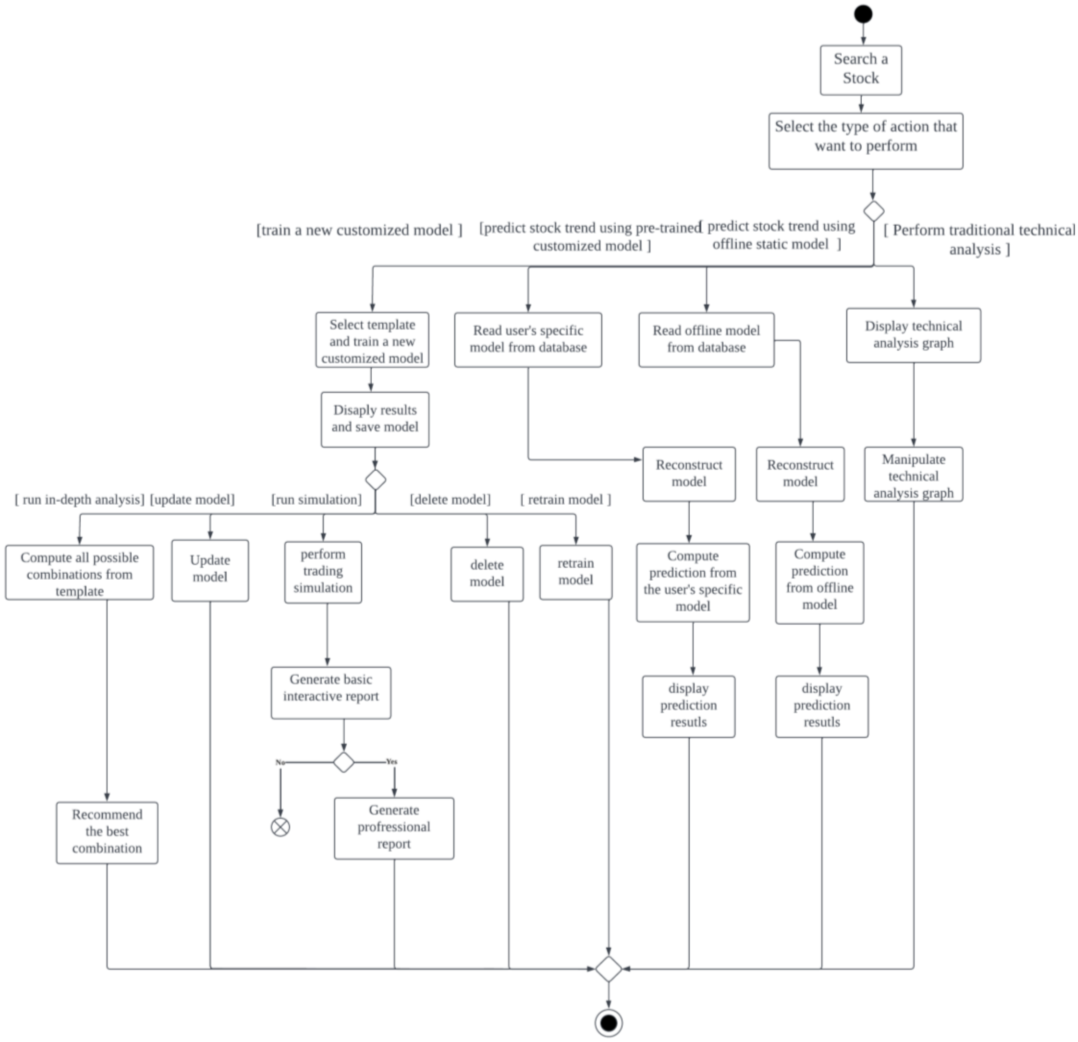



Figure 3.4.1.1 Process Workflow of the Single Stock Prediction and Training Feature

### 3.4.2 Design of The Indicator Recommendation Procedure

As mentioned in the above section, the customers can choose to run the indicator recommendation feature in the in-depth analysis section after the model training. Firstly, the indicator recommendation feature will compute all possible combinations of the stock indicators based on the original indicators that defined in the chosen template and run the model training for each combination. The time taken for in-depth analysis process will be roughly 3 to 5 minutes regardless of the number of combinations. This is because the system will apply parallel processing for all possible combination, thus the training process for all combinations of indicators will be carried out simultaneously. After the training of all combination has been completed, the system will recommend the combination of indicators that achieved highest accuracy in the machine learning evaluation and display the accuracy for all combinations so that investors will have better idea how each combination perform. The best performing model will be automatically saved to database. Figure 3.4.2.1 shows the workflow of the stock indicator recommendation feature.

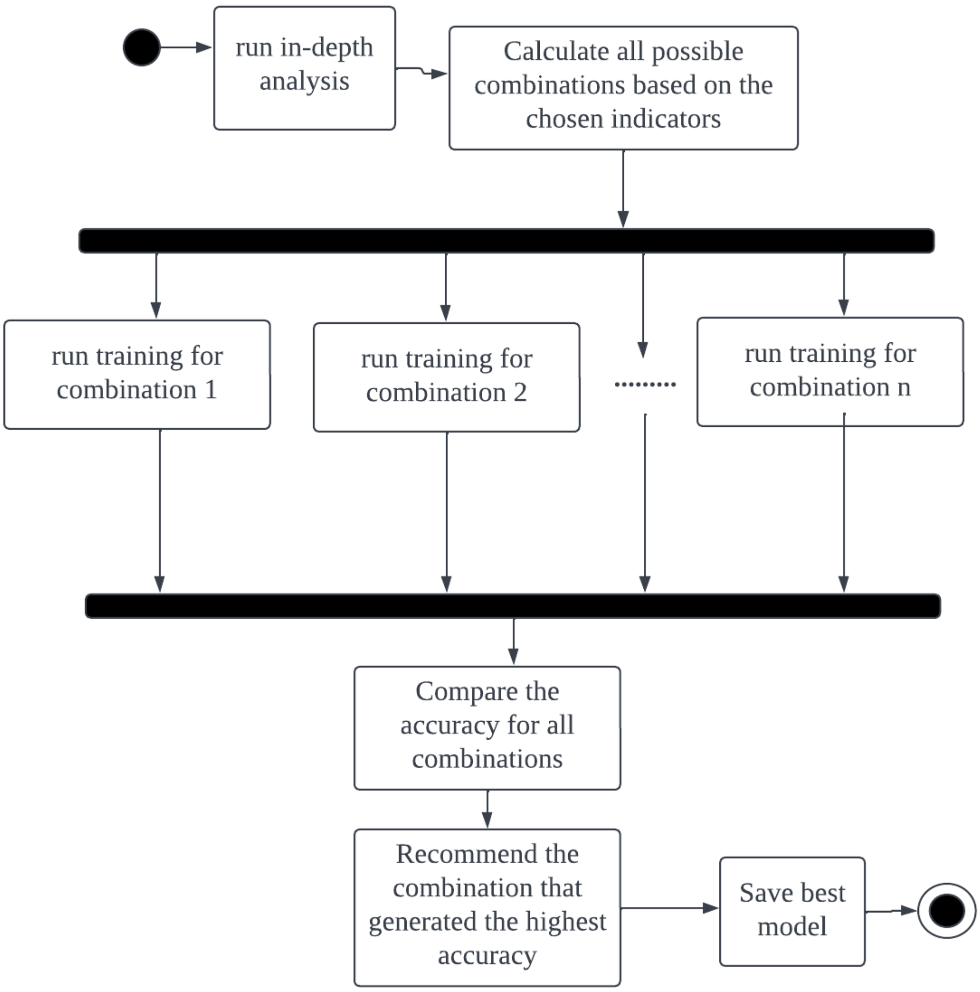



Figure 3.4.2.1 Process Workflow of the Stock Indicator Recommendation Feature

### 3.4.3 Design of Stock Trend Prediction for All Stocks in Watchlist Procedure

This feature is mainly to ease the customers in prediction. Instead of performing the prediction and training on customers' favourites stocks one-by-one separately, this feature allows customers to run the prediction and training on all customers' favourites stocks together at once. As compared to the training and prediction process for single stock, there will be 3 similar options provided for the stock prediction (predicting the trend using offline static model, predicting the trend using the previous trained online customized model, or training a new customized model in real-time), but the difference is that instead of performing the functions on single stock, the system will perform the functions on multiple stocks together at once. If users choose the first option which is to predict the trend of favourites stocks using offline model, the similar concept applies where the system will make request to server-side to obtain the predictions for all favourites stocks from the offline models. If users choose to train new customized models for all favourites stocks in watchlist, users must choose which watchlist the users want to train/predict and what template to be used in model training. After choosing the desired watchlist and template, model training process for all favourites stocks in the chosen watchlist will be carried out parallelly. When the training process is done, system will display the prediction result for all favourites stocks together with the average accuracy of all models. For example, if customers included 5 stocks in the selected watchlist then there will be 5 different models resulted after the training where each model belong to each stock. The last option will be predicting the watchlist stocks using the pre-trained customized models. Predicting all stocks in watchlist using pre-trained customized models will generate a new issue. For example, the customer has 5 stocks in the watchlist, and customer had also trained the prediction model for all 5 stocks, when customers want to run the prediction for the 5 stocks using the pretrain customized models, then 5 prediction results will be displayed. But here is problem come, what if the customer added a new stock in the watchlist which results a total of 6 stocks in watchlist. When customer directly run the prediction for watchlist stocks using the pretrain customized models, there are only 5 stocks that have a trained model associated to each of them, and the new stock added does not have any trained model associated to it. The solution is to train a new model for the new added stock automatically when the pertained model prediction is run, this will eliminate the need of retraining all the stocks again which much time can be saved. Therefore, before running the prediction using pre-trained customized model, the system will always check if the current stock in watchlist has a pre-trained model associated to it. Figure 3.4.3.1 shows the workflow of the watchlist stocks prediction and training feature.

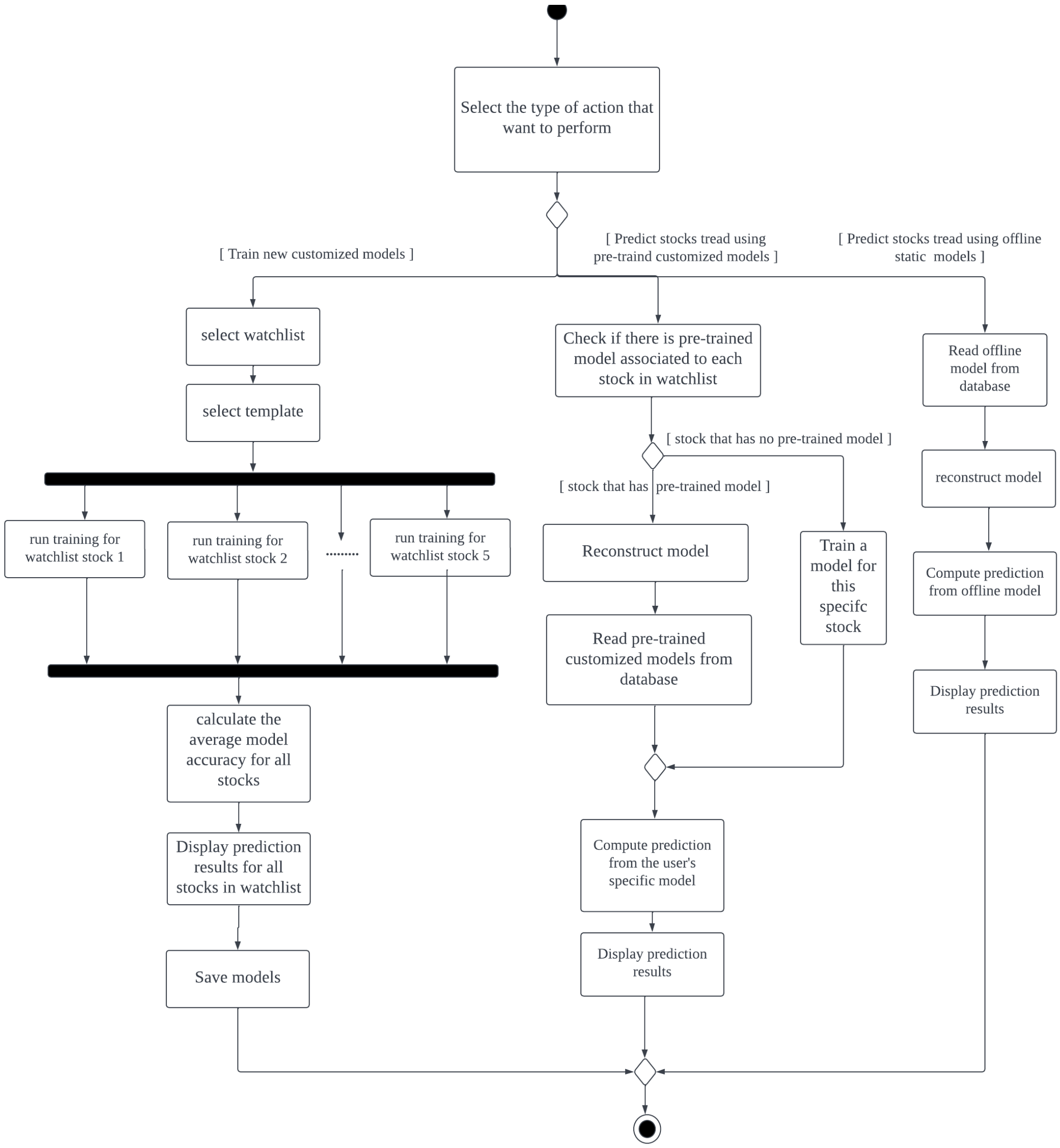



Figure 3.4.3.1 Process Workflow of the Watchlist Stocks Prediction and Training Feature

## 3.5 Hardware and Software Requirements

Computer or Laptop is the only hardware involved in the project development. The table 3.5.1 shows the specification of the hardware used.

Table 3.5.1 Specification of the Computer

| Description | Specifications |
|---|---|
| Model | Asus TUF Gamming FX505DT |
| Processor | AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx 2.30 GHz |
| Operating System | Windows 10 |
| Graphic | NVIDIA GeForce GT 1650 |
| Memory | 8GB DDR4 RAM |
| Storage | 500 GB SSD |

As mentioned in chapter 3.1, the development of project will be divided into 2 phases which are stock prediction model architecture development and web application development.

Anaconda Python Jupyter notebook will be used as the development platform for the stock prediction model architecture. There are some useful libraries provided by the Jupyter notebook such as Numpy, Pandas and Talib. Numpy and Pandas libraries will be used for the manipulation of the stock data such as splitting the data, scaling the data, and transforming the data into time-series, whereas Ta-lib library will be used to compute the technical indicators value such as MA, SMA and RSI.

Further, Visual studio code will be used as the development platform for the web application. In front-end web development, HTML, CSS and JavaScript will be used to design the graphical interface of the application. To achieve stock charting capabilities in the front-end, Lightweight Chart JS will be used. Moreover, in back-end web development, Node JS framework will be used to develop the backend code to handle the requests coming from front-end. To incorporate deep learning feature in Node JS backend, TensorFlow JS will be used. Using TensorFlow JS, the architecture of stock prediction model developed in Jupyter notebook can be migrated to the Node JS platform.

Lastly, to use the system, there are no specific hardware or software restrictions in the user environment as the system is in the form of web application where all the functionalities will be hosted in a public server. Thus, users can access the system using any computers with web browsers and internet connectivity.

# Chapter 4 System Design

## 4.1 System Architecture Design

### 4.1.1 System Architecture Design of Web Application

The architecture design of the web application in the project is Client-Server architecture. The design pattern used for the poject is Model-View-Controller pattern. The Controller module will mainly handle the Http reqeust generated from the client-side. The view module will formulate the personalized dynamic content based on the data that passed by the controller, and model module will be responsible for the database query operations. Figure 4.1.1.1 shows the system Architecture design of the web application

When a request is coming from the client-side, controller will first identify the user's identity so that only the information that related to the particular user will be retrieved from the model module. The model module will then query the specific user's data from the MongoDB database. After controller obtained the user's data from the model, the controller will pass the data to the view module so that view module could formulate the personlized dynamic content and then return the personlized page to the client-side. Since different users will contain different data in watchlist, lstm model template as well as the pretrain models, different dynamic views will be created according to the user's identity.

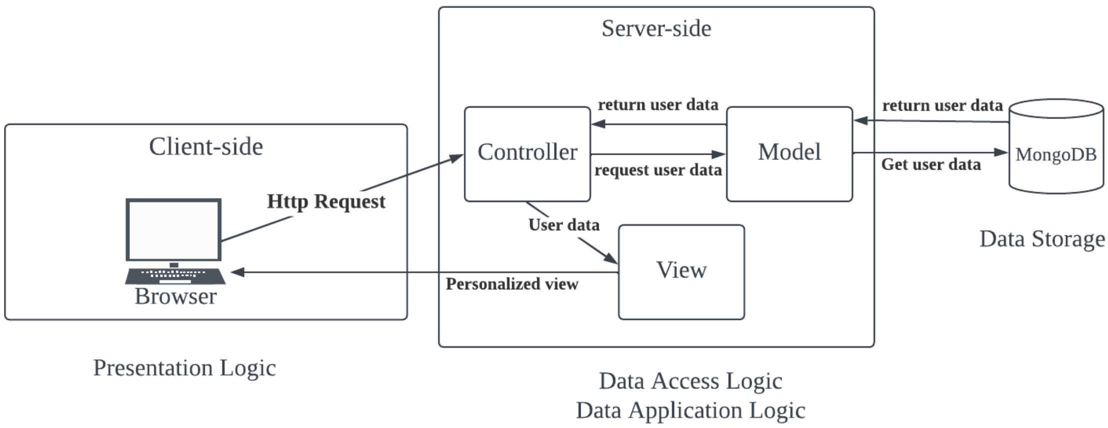



Figure 4.1.1.1 System Architecture Design of the Web Application

### 4.1.2 System Architecture Design of Proposed Stock Prediction Model

Figure 4.1.2.1 shows the system architecture design of the proposed Stock prediction model, and the gray box represent the output size of each phase. The model will accept input in the form of time series as metioned in figure 3.2.3.2. The proposed model consist of different types of layers. The first hidden layer is the LSTM layer with 64 units. The LSTM layer will learn the time-series depedency present in the data. Each time step in each input data will go through all 64 LSTM units and results ( number samples= m, 64) as output size, and 64 represent the number of the 64 hidden states outputs accumulated from the first time step to the last time step in each sample. Next, the data will be passed to dropout layer. In dropout layer, 5 % of the LSTM hidden nodes will be turned off randomly during each training operation to increase the generalization capability of the model and reduce overfitting issues. Finally, the data will be passed to the dense layer with 32 neurons. The purpose of adding fully-connected layer (FC) is to perform classification learning based on the LSTM hidden states data. According to [4], the author only made use of LSTM layer inside the neural network to learn the temporal features/time dependency of the data but did not include any layers for the classification learning. Therefore, a LSTM + FC layer model is proposed in this project. The proposed model not only contain the LSTM layer to learn and extract the temporal feature but also contain the FC layer to learn the classification of data. The final output of the model will be in the form of probabilities. The model will predict the data as “1” if the probabilities is greater than the threshold and “0” otherwise.

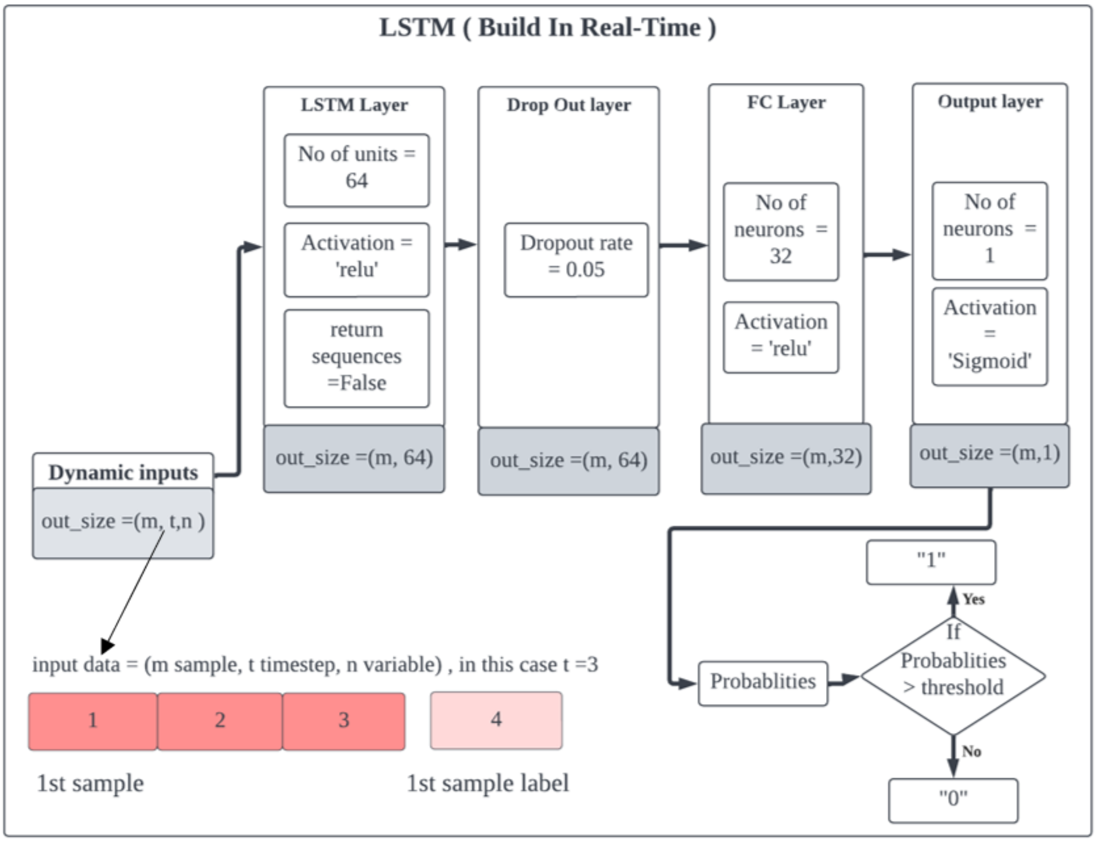



Figure 4.1.2.1 System Architecture Design of the Stock Prediction Model

## 4.2 Data Storage Design

First of all, the database used in this system is the NoSQL MongoDB. The main reason to use MongoDB is that LSTM models built in python platform and javascript paltform can only be saved in Binary and JSON file format. MongoDB is a document database that allowed users to store JSON and Binary file. In the database design, the database will be divided into 2 schemas. Schema 1 will be used to store the information that is for public which mean the information does not belong to any specific users accounts (non user-sepcific information), wherease schema 2 will be used to store user-specific information such as users' credential, model templates and watchlist.

Schema 1 will store two kinds of information which are offline models and latest stock prices. All offline models that have been trained in advance in the offline environment will be saved in the schema 1. In order to store the entire offline model into the database, 3 entities are needed which are offline_model, offline_scalling and offline_model_structure. Offline_model entity is used to store the basic information about offline models such as the name of the offline model, the primary key, and the trained weights of the model. Further, offline_model_structure entity is used to store the topology (number of layers, neurons contain in each layer, and etc) of the model. By using topolgy data and weight data, system can reconstruct back the model easily in the application and utilize it in the prediction. Besides that, scalling factors for the input of the model also required to be stored in the database. The scalling factors is used to scale the input of the model to the range that determined in the training process. Since the input to the model will contain multiple attributes, scalling factor for each attribute will be stored in a sperate child table named “offline_scalling” and thus one model will contain one or many scalling factors. In addition, in schema 1, latest stocks’ price information will also be stored in the entity that named “allstocks”. The stocks’ price information includes open price, close price, high price, low price, volume and the market time. This application will provide users with the latest stocks’ price information that is almost in sync with the currect stock market conditions. To achieve this, the system will call the latest data from the Yahoo finance API everytime users refreshed the page and save the returned data into the “allstock” entity. Because this system will provide 100 stocks to users, fetching real-time data for all 100 stocks might take longer time to complete. Whenever users are waiting for the newest data from the API, the old data that were stored in the “allstock” entity will be used as the content in the page temporarily, and when the newest data is reached, the sysetm will then update the “allstock” entity with the newest data. Since the data stored in the “allstock” entity are updated, users are able to switch the page content to the latest data by refreshing the page again. Because of this repeated mechanism, users will always obtain the latest data by refreshing the page.

The schema 2 will be used to store the user-specfic information. There are total of 8 entities in schema 2. Firstly, the users entity is used to store the users’ credential information such as email and password for authentication purposes. Further, since the system allowed users to create main watchlist and alternative watchlists, main_watchlist and extra_watchlist entities will be created. Main_watchlist entity is used to store the main watchlist information such as owner’s id. Users can store one or many stocks into the main watchlist, thus a child table called main_watchlist_stock is created to store the information of all stocks in the main watchlist. Users are limited to create 1 main watchlist only. Further, extra_watchlist and extra_watchlist_stock store the similar things as the main watchlist, but the only difference is that each user can create multiple extra watchlists. Futher, the lstm_model_template entity is used to store the information about model templates created by different users. Each user can create multiple templates in the lstm_model_template entity.The lstm_model_template entity will contain some basic information that related to a template such as the name of the template, the owner of the template, the input technical indicators, and  the time created. Lastly, after users have trained an online model on the website in real-time, to save the whole online model in the database, 3 entities are needed. The structure of these 3 entities are almost similar as the 3 entities mentioned in the offline model. The online model's weight, the name of the online model, the owner of the online model will be stored in Online_model entity, attributes'scalling factors of the online model will be stored in the Online_model_scalling, and topolgy of the online model will be stored in Online_model_strcutre. The Figure 4.2.1 illustrate the data storage design of the system.

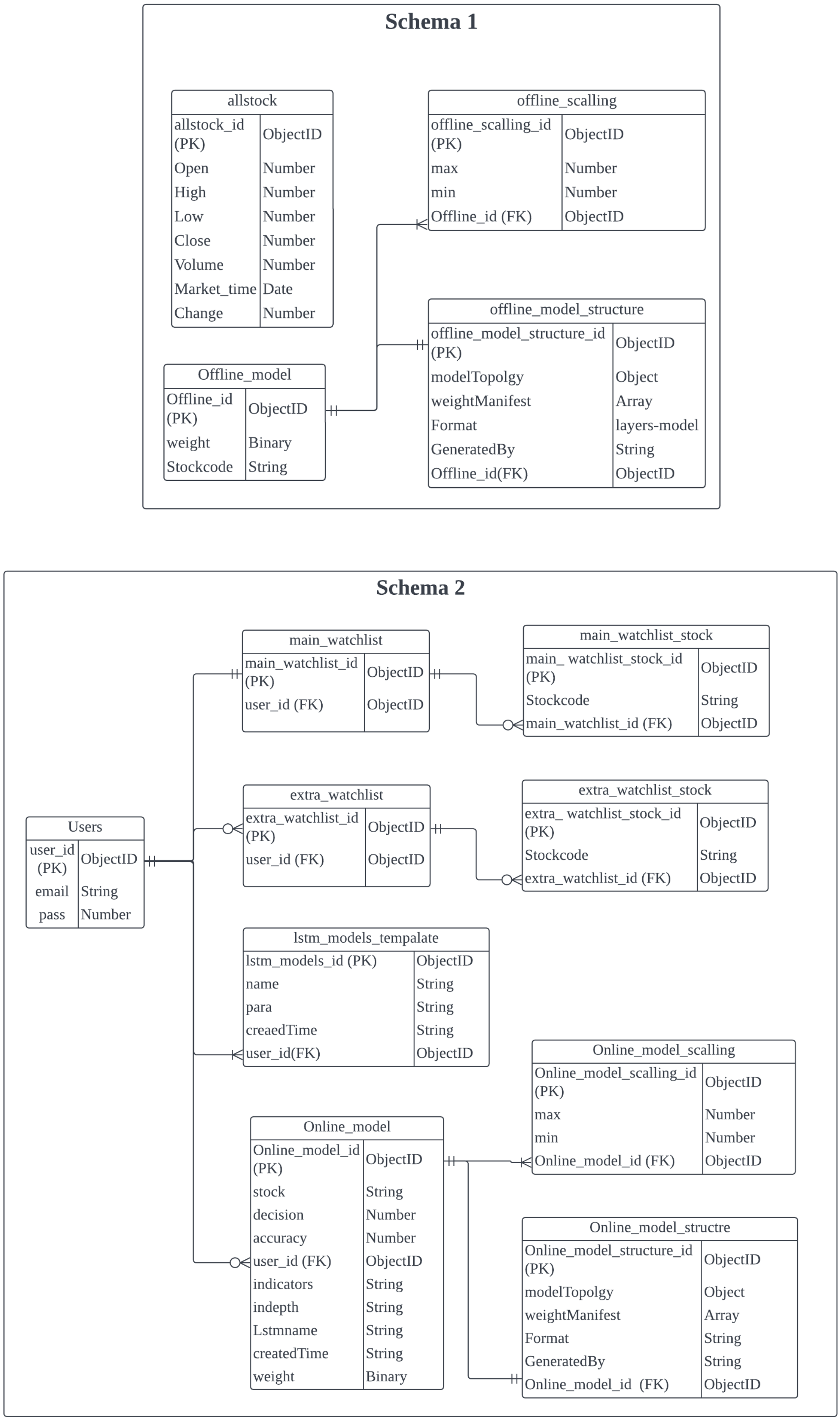



Figure 4.2.1 Data Storage Design of the System

## 4.3 Graphical User Interface Design

### 4.3.1 Chart Page

The figure 4.3.1.1 show the top section of the “chart” page. When a user clicks into the chart page, a candlestick chart with technical analysis capabilities will be appeard in the upper-right section of the page together with the historical stock prices attached at the bottom of the candlestick chart. Users are allowed to hover over to the graph at any points, and the repective prices and the technical indicators value for the selected day will be displayed. Further, Users can also perform ascending and descending sorting on the historical price table by clicking the header of each column, and users are also allowed to perform searching on the table. The figure 4.3.1.2 shows the detailed illustration of the functions available on the candlestick chart and historical table. In the upper-left section, there will be 4 divisions which are Top gainers, Stock Info, Stock indicator manipulation section and Watchlist. The figure 4.3.1.3 show the detail view of the 4 divisions. Top gainers section will display the stocks that have increased the most percentage at the current timestep. By having Top gainers section displayed in the page, investors will have a clear view on the stocks that are currently in the bull trend. This section is coded in an animated vertical tickers way where the prices will be autmotically moved downward and upward within certain time interval so that the prices of all Top gainer stocks can be displayed and fitted in this small section. Further, the stock info section will display some basic information about the current selected stock such as the stock code. Stock indicator manipulation section is the section to add, delete and modify the stock indicators on the candlestick chart. Lastly, the watchlist section will display the current stocks stored in the main watchlist together with offline model prediction and delete function.


Figure 4.3.1.1 Top Section of the “chart” Page

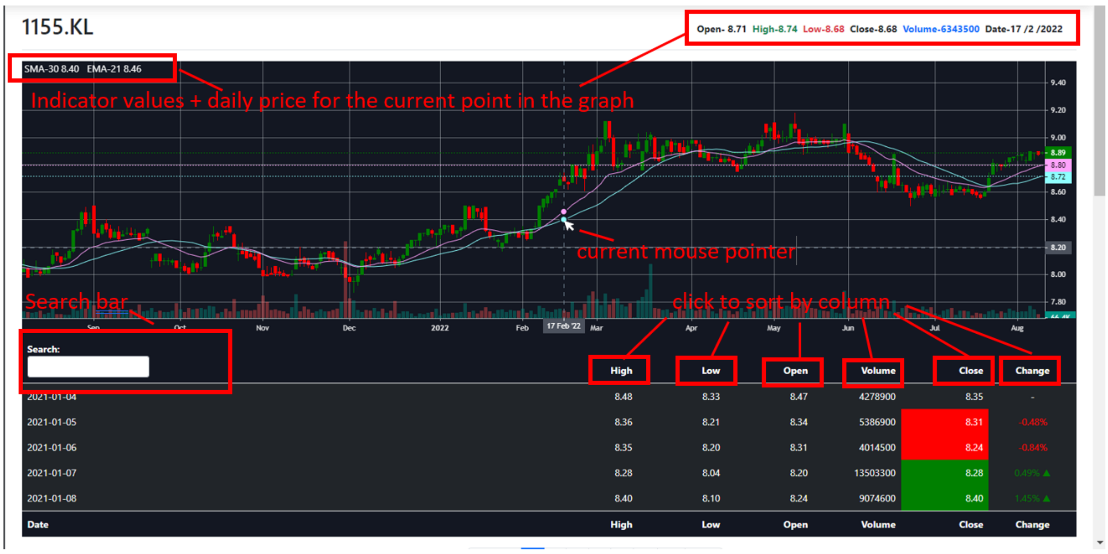



Figure 4.3.1.2 Detailed Illustration of the Functions Available on the Candlestick Chart and Historical Table

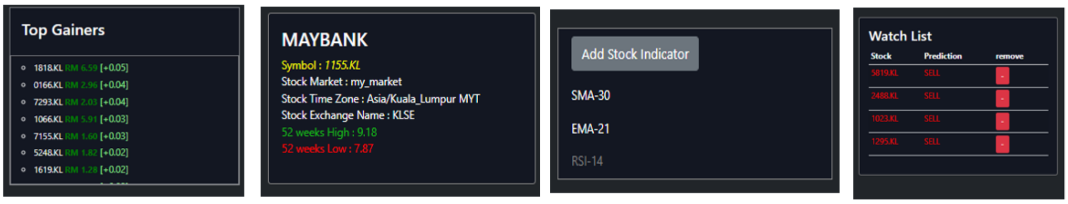



Figure 4.3.1.3 Detail View of the 4 Divisions in the Top-Left Section

The figure 4.3.1.4 shows the indicator manipulation function on the chart. On the candlestick chart, there will be 3 default indicators provided which are SMA-30days, EMA-21days and RSI-14days. The default indicators can be turned on or off in the graph. From the figure 4.3.1.4, the SMA-30day and RSI-14days are turned on, wherease the EMA-21days is turned off. Of course, other than the 3 default indicators, users also allowed to add other indicators.


figure 4.3.1.4 the Indicator Manipulation Function on the Chart

The figure 4.3.1.5 show the pop-up window when users add a new stock indicator. There will be total of 22 stock indicators provided to users, and the indicators will be categorized into four categories namely trend indicator, momentum oscillator, volume & volatility, and statistic indicator. After users added an indicator in the graph, they are allowed to turn on/off, delete, and change the parameters of the indicators. Figure 4.3.1.6 show the process of changing the indicator’s parameters. Based on the figure 4.3.1.6, initially, Bollinger bands (BBands) with 10 days period was added. When users change the parameter of the BBands-10days to BBands-30days-1std, the graph showing will also be updated automatically.

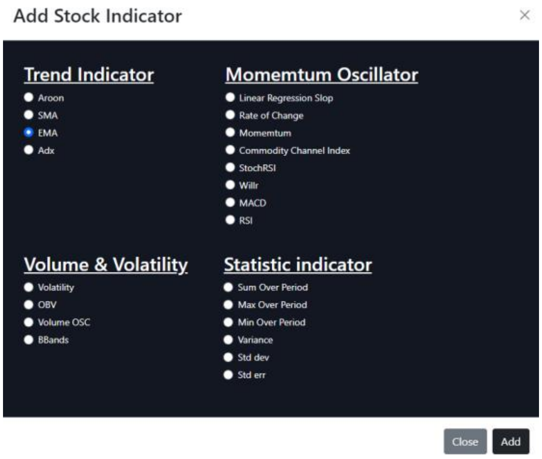



Figure 4.3.1.5 Pop-Up Window When Users Add a New Stock Indicator

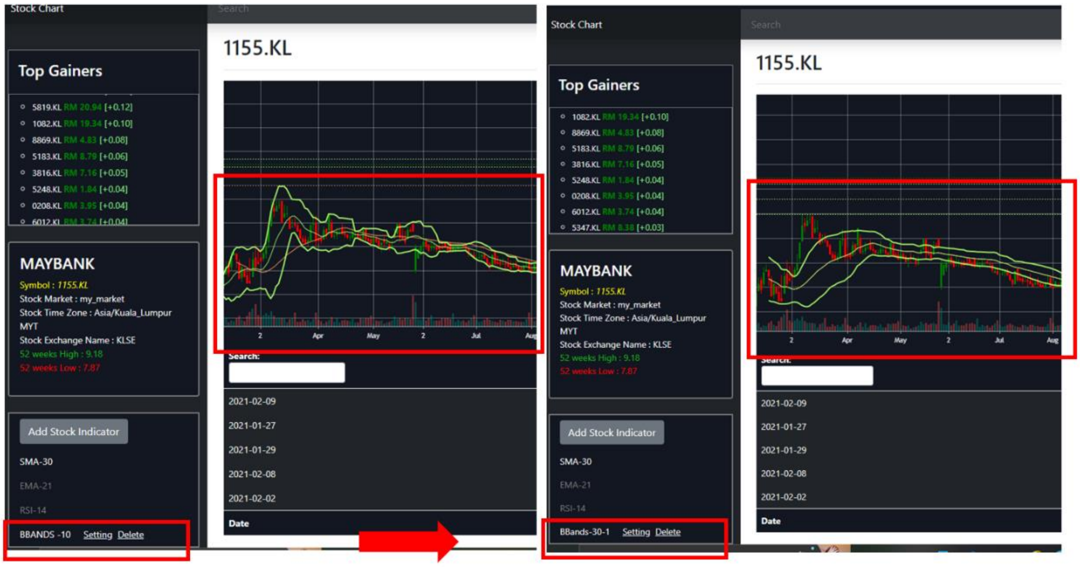



Figure 4.3.1.6 the Process of Changing the Indicator's Parameters.

Moreover, the bottom part of the "Chart" page will contain the prediction section and model training section. In prediction section, users are provided with 2 types of prediction model to predict the trend of the stock namely offline static model and online pre-trained customized model. For all registered users and non-registered users, there will be 1 fixed offline static model provided for prediction by default, it is a reference model provided by the system. This offline static model will be trained in advance and stored in the database. In this way, users can get a stock prediction immediately from the system without going any model customization and model training process in the system. This offline static model allows newcomers to have an experience on the prediction function provided by the system even they are not registered. After having tried on the feature, it might possibly trigger the interest of the newcomers to join as the registered users. Further, the offline static model will also be updated periodically by the programmers with the new data arrived so that the performance of the model can be maintained. The figure 4.3.1.7 shows the prediction section of new users who have not registered in the system. As compared to non-registered user, registered users are given with an extra privilege to build customized models in real-time. For the existing users who have trained few customized models, the prediction section will contain multiple models. The table in prediction section will be sorted in descending order by accuracy so that the customized model with higher accuracy will be first appeared to customers. The figure 4.3.1.8 shown the prediction section of existing users.

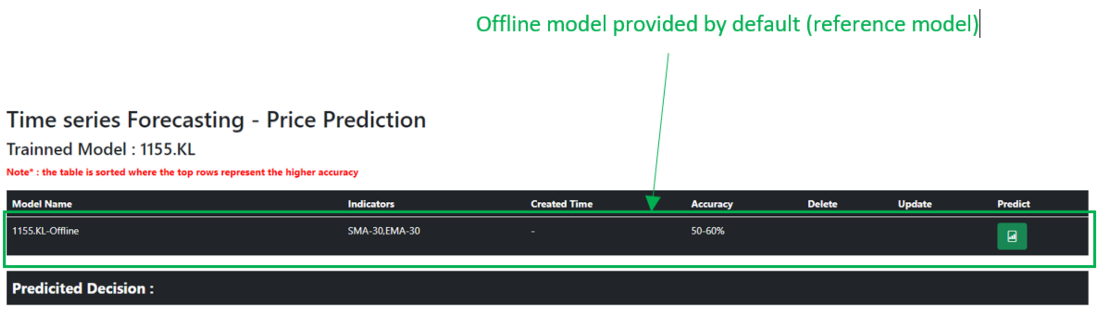
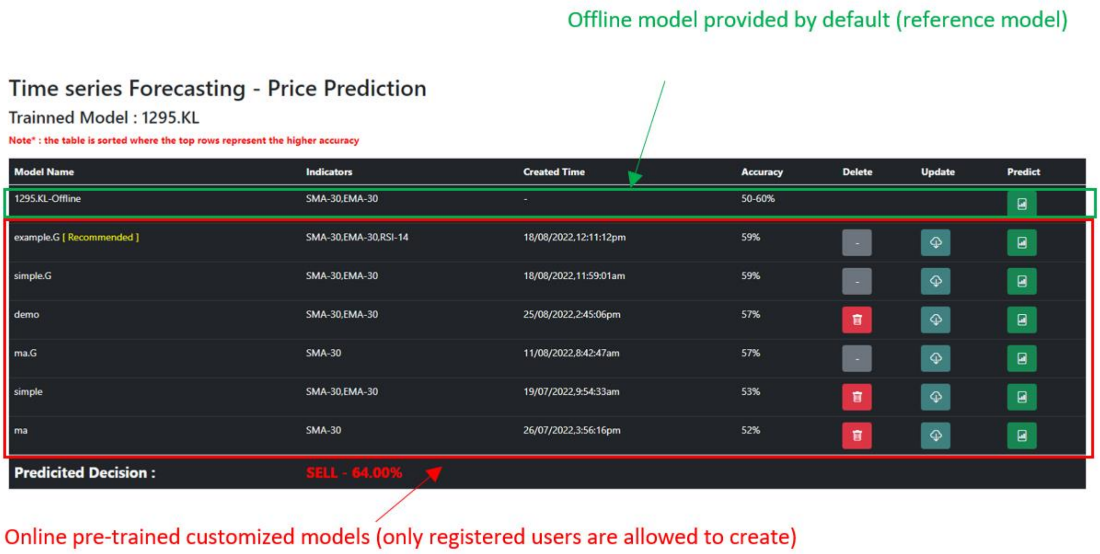



figure 4.3.1.7 Prediction Section of New Users Who Have Not Registered in the System.




The figure 4.3.1.8 the Prediction Section of Existing Users.

Further, in training section, there will be 2 different parts which are model building section and indicator suggestion section. The figure 4.3.1.9 shows the model training section. The first part, model building section, is the section to build and train a customized model in real-time. In order to train a customized model, users must first choose a template for the customized model they want to train on. Templates represent different combinations of technical indicators that the users want to input to the model. Users can add or delete templates in the “Modelling” page. After the training is done, a model performance report will be displayed to the users. The figure 4.3.1.10 shows performance report of “demo” customized model. This report contains 3 kinds of evaluation metrics which are model accuracy, cumulative return in percentage and earnings from price change (1 unit). Model accuracy represents the correctness of the model prediction, cumulative return represents the total percentage change of investment with respect to the initial investment amount, and earnings from price change (1 unit) represents the cumulative earnings gained from each buying operation. A well-performing model is expected to achieve high positive cumulative return and high positive earnings from price change (1 unit). In addition, user can also choose to train other models by selecting another model template in the drop box and press “Start Training”, and users also allowed to retrain the model if the performance of model is not satisfied.

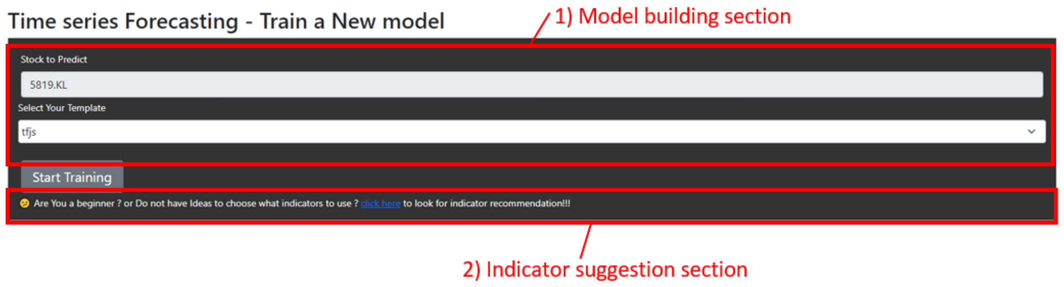



Figure 4.3.1.9 the Model Training Section

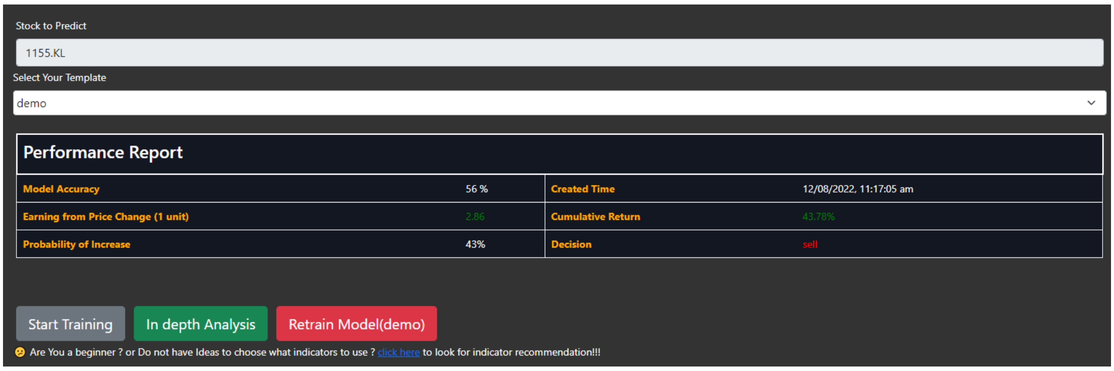



Figure 4.3.1.10 Performance Report for the “demo” Model

After the customized model has been trained, the users can also choose to evaluate the prediction model as a trading system, 3 years of trading simulation with initial amount of RM1,000 will be provided to the users. The figure 4.3.1.11 show the trading simulation report. All buying operations performed by the model will be tabulated in the top-left section of the report and graphed in the top-right section of the report. By showing each buying operation in the table and graph format, users can better visualize how the model is performing overall. Each trading operation will be highlighted with red or green colour where red represent the negative return (predicted buy but the price dropped), and green represent the positive return (predicted buy and the price is indeed increased). Further, some popular evaluation metrices will also be presented in the bottom-left section of the report. Cumulative return (amount) represents the total profit gained over the trading period, and average return represent the average percentage gained per trading operation. Standard deviation of return represents the volatility of the trading strategy or in other words the overall risk of the trading strategy. The lower the standard deviation of return, the lower the risk of the trading strategy. Further, Sharpe ratio is the measurement of the risk-adjusted trading performance or in other words, it tells how much reward associated to a single risk during the investment. High Sharpe ratio implies that high reward per risk [21]. The model performance will also be compared with the Buy-and-hold and Optimistic trading strategy which these 2 trading strategies are the most fundamental strategies that used for comparison [31]. In the bottom-right section, a daily return and cumulative return graph will be displayed. Further, a button to generate even more in-depth portfolio performance report will also be provided to users. This report will be generated by an open-source Python library called Quantstats, and it is mainly designed for portfolio managers and fund managers to compare the performance of the trading strategy. Thus, the report will include a lot of advanced risk metrics and in-depth analytics. The figure 4.3.1.12 shows the in-depth performance report provided by the Quantstats. In the report, the information will be divided into 2 sections which are graph section and table section. In graph section, there will be total of 7 graphs displayed, and these 7 graphs can be further categorised into 3 different groups namely cumulative return-related graphs (highlighted in green), return-related graphs (highlighted in blue), and drawdown graphs (highlighted in orange). Drawdown is the measure of how many percent of stock price has dropped before it recovers to its new peak value, or in other words, how much losses the trading strategy experienced before the price increased again. For example, the latest peak of stock price is RM1.00, and the price has increased to a new peak of RM2.00 in the end of the month; however, before the stock price achieved RM2.00, it experienced a drop to RM0.50 during the middle of the month, thus the Drawdown will be -50%. Drawdown of -50% indicate the trading strategy experienced a 50% of decrease in the history. Although the strategy achieved 100% return as the final return, it will still be considered as a risky strategy. The higher the drawdown value, the greater the efforts required to recover the losses. Further, in the table section, information can be categorized into 4 different categories namely basic info, overall return info, risk metrics as well as the advanced performance metrics. Firstly, the basic info section will include the risk-free rate and the time in market. Risk-free rate is defined as the minimum percentage of return that will be guaranteed by the market, and it is usually referred to the overnight policy rate set by the central bank of Malaysia or short-term government bond rate [32]. Since the risk-free rate will be changing from time to time, 0% of risk-free rate will be considered to make the calculation simpler. Secondly, the overall return section will include the cumulative return percentage gained by the portfolio and the compound annual growth rate (CAGR%). CAGR% is mainly to smooth the cumulative return by eliminating the compounding information. The smoothing of the cumulative return is particularly important when users want to compare the performance of a trading strategy on 2 uncorrelated stocks such as comparing between growing stock and falling stock. Because of the rate of return is smoothed, the CAGR% value generated for both growing stock and falling stock will be comparable and will not be biased [33].

Thirdly, the risk metrics section indicates the possible risks present in the portfolio. Because there are too many risk metrics in the report, only the most common and significant metrices will be discussed. As mentioned in above chapters that Sharpe ratio is the measure of reward per risk, the higher the Sharpe ratio the better the trading strategy as more reward is associated to a single risk. Prob. Sharpe and Smart Sharpe are the variation of the Sharpe ratio that include even more information such as time-series autocorrelation relationship between the prices. Further, Sortino ratio is also considered as one of the popular risk metrics being used in the market. Sortino ratio is an improved version of Sharpe ratio. Instead of calculating the reward per risk like Sharpe ratio, it calculates the reward per downside risk which implies how much reward per bad risk. Sortino ratio put more focus on the bad risk instead of the overall risk of the portfolio. Bad risk is defined as the negative return that caused by the large fluctuation of stock price. A portfolio with high Sortino ratio is better than a portfolio with high Sharpe ratio as it indicates more reward are earned per single bad risk. Further, as mentioned above, drawdown indicates the risk that experienced by the portfolio before it recovers back to the next new peak in the history, maximum drawdown indicates the highest drawdown value experienced by the portfolio in the history. For example, a portfolio experienced 3 drawdowns which were -40%, -50%, and -30%, and -50% is considered as the maximum drawdown in the portfolio. The lower the drawdown value, the less risky the trading strategy is. Small drawdown is important as large drawdown value will need more efforts and time to recover the losses. The longest DD days referring to the maximum time that strategy has taken to recover from the drawdown losses. Lastly, the advanced performance metrices provide even more detail and in-depth indicators that used to study the performance of portfolio in daily, monthly, and yearly.

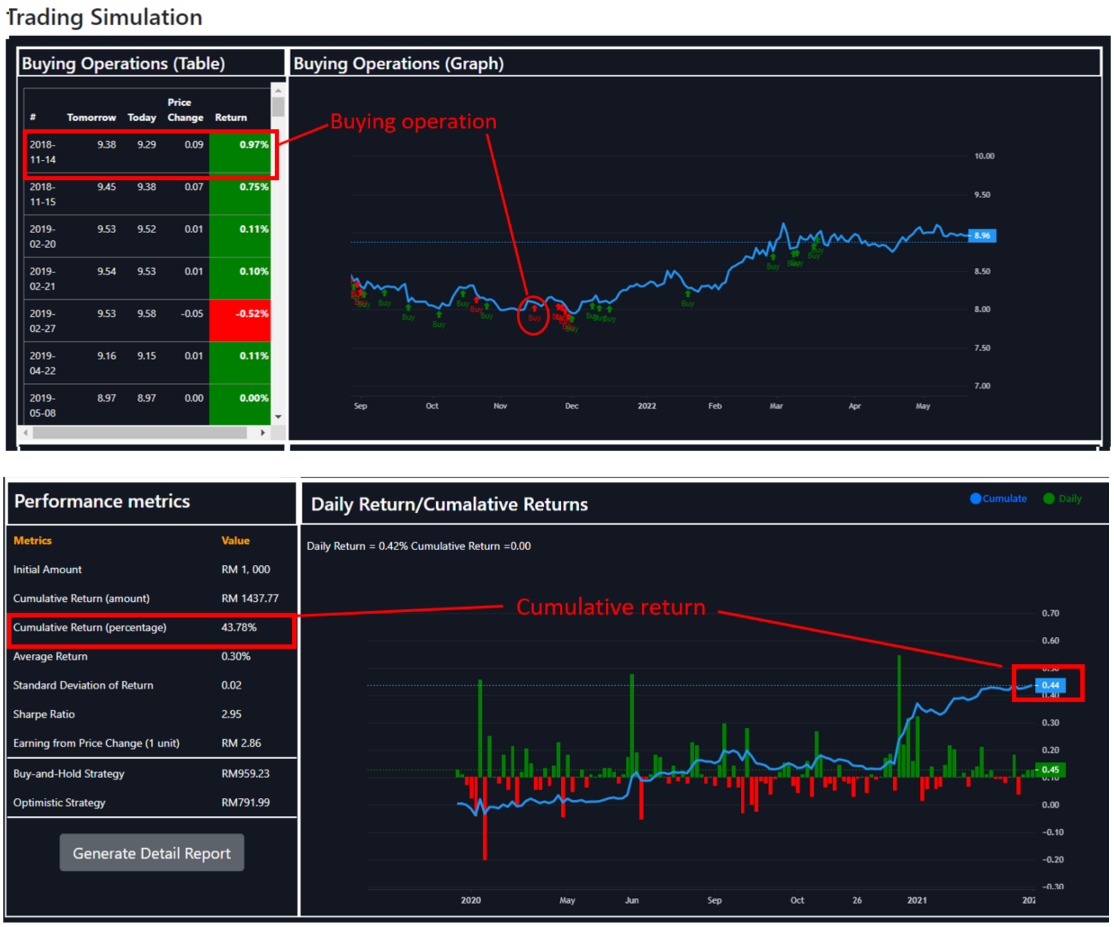
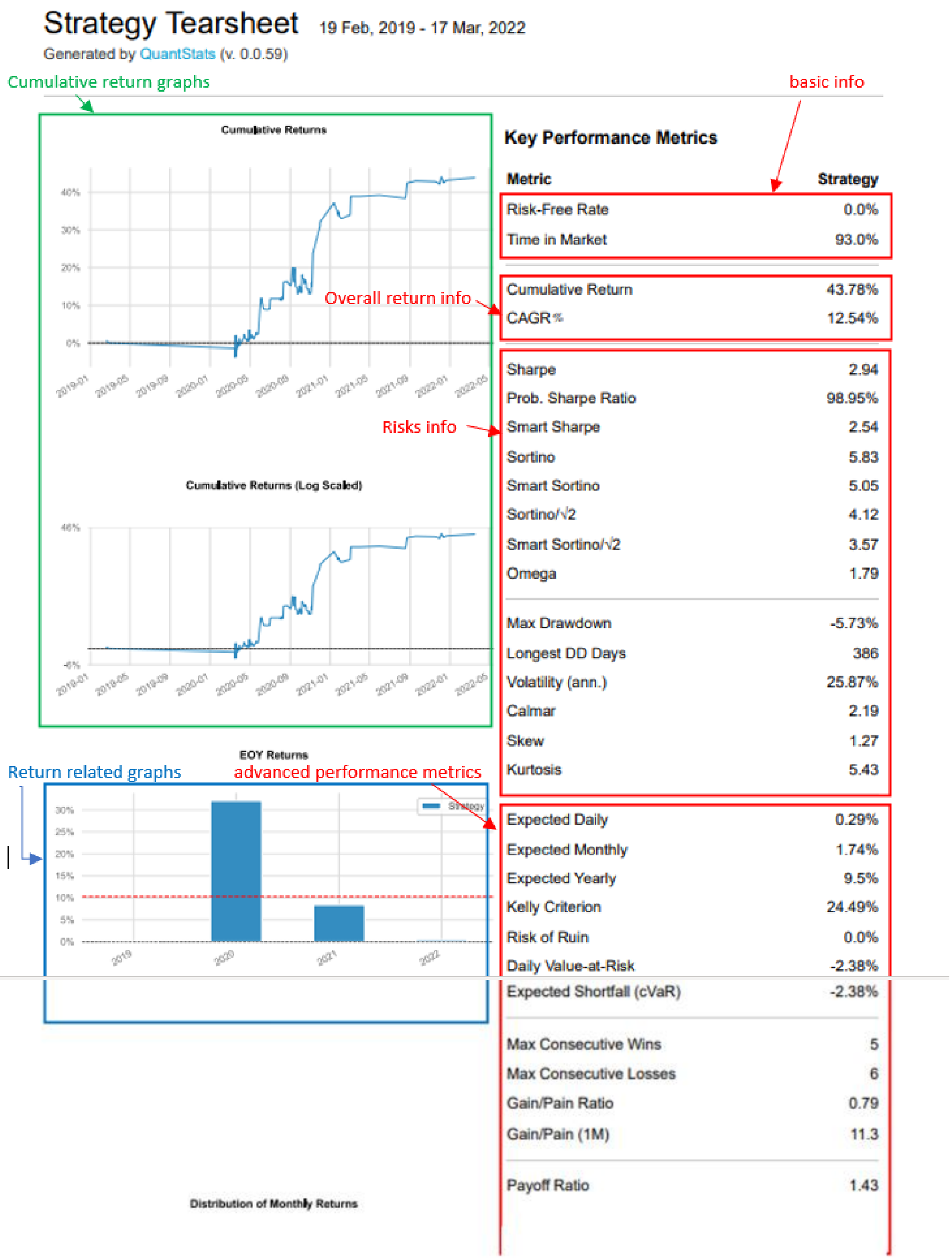
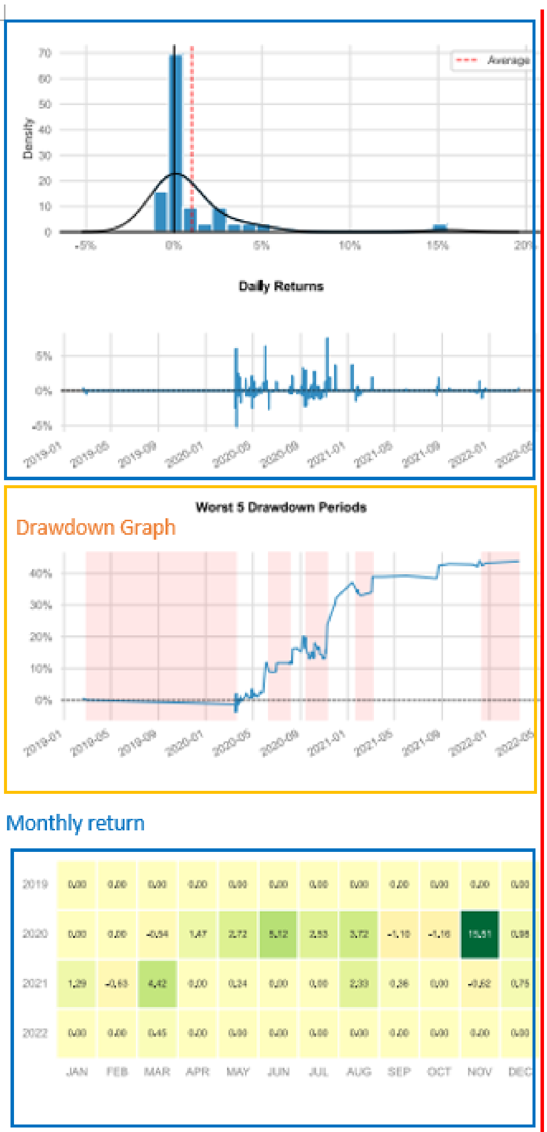



Figure 4.3.1.11 the Trading Simulation Report


Strategy Tearsheet
19 Feb, 2019 - 17 Mar, 2022
Generated by QuantStats (v. 0.0.59)
Cumulative return graphs
basic info
Cumulative Returns
Overall return info
Risks info
Cumulative Returns (Log Scaled)
Return related graphs
EOY Returns
advanced performance metrics
Strategy
Distribution of Monthly Returns
Key Performance Metrics
Metric Strategy
Risk-Free Rate 0.0%
Time in Market 93.0%
Cumulative Return 43.78%
CAGR﹪ 12.54%
Sharpe 2.94
Prob. Sharpe Ratio 98.95%
Smart Sharpe 2.54
Sortino 5.83
Smart Sortino 5.05
Sortino/√2 4.12
Smart Sortino/√2 3.57
Omega 1.79
Max Drawdown -5.73%
Longest DD Days 386
Volatility (ann.) 25.87%
Calmar 2.19
Skew 1.27
Kurtosis 5.43
Expected Daily 0.29%
Expected Monthly 1.74%
Expected Yearly 9.5%
Kelly Criterion 24.49%
Risk of Ruin 0.0%
Daily Value-at-Risk -2.38%
Expected Shortfall (cVaR) -2.38%
Max Consecutive Wins 5
Max Consecutive Losses 6
Gain/Pain Ratio 0.79
Gain/Pain (1M) 11.3
Payoff Ratio 1.43




| Metric | Strategy |
|---|---|
| Profit Factor | 1.79 |
| Common Sense Ratio | 3.05 |
| CPC Index | 1.42 |
| Tail Ratio | 1.71 |
| Outlier Win Ratio | 5.42 |
| Outlier Loss Ratio | 2.82 |
| MTD | 0.45% |
| 3M | 0.81% |
| 6M | 0.93% |
| YTD | 0.45% |
| 1Y | 3.54% |
| 3Y (ann.) | 19.91% |
| 5Y (ann.) | 12.54% |
| 10Y (ann.) | 12.54% |
| All-time (ann.) | 12.54% |
| Best Day | 7.58% |
| Worst Day | -5.19% |
| Best Month | 15.51% |
| Worst Month | -1.16% |
| Best Year | 32.12% |
| Worst Year | 0.0% |
| Avg. Drawdown | -2.18% |
| Avg. Drawdown Days | 49 |
| Recovery Factor | 7.64 |
| Ulcer Index | 0.03 |
| Serenity Index | 5.78 |
| Avg. Up Month | 2.79% |
| Avg. Down Month | -0.81% |
| Win Days | 55.56% |
| Win Month | 75.0% |
| Win Quarter | 90.0% |
| Win Year | 100.0% |

Figure 4.3.1.12 In-Depth Performance Report Provided by the Quantstats

Finally, stock indicators recommendation features will also be provided to users in the in-depth analysis section. After users have trained a model with a specific template, users might be wondering what indicators defined in the template are indeed advantageous to the model prediction, could users eliminate the use of some indicators and still be able to maintain the similar accuracy or even improve the accuracy. In-depth analysis is designed to answer the doubts of the users. In-depth analysis will perform a detail analysis on all possible combinations of indicators that defined in the template.

By doing so, users will have a clear view on how each combination perform, and what indicators can be removed without affecting the model accuracy or even improve the accuracy. The figure 4.3.1.13 shows the information about the “example” template. Based on the figure, “example” template is made up of SMA-30days, EMA-30days and RSI-14 days. These 3 indicators will result 6 possible combinations as shown in figure 4.3.1.14, and each resulted combination will be passed to the model training process. The model training process for all combinations will be carried out parallelly, and thus the whole process can be achieved in short time. The system will automatically recommend the best performing indicators combination by comparing the model accuracy gained by each combination. The best combination will be the combination that results the highest accuracy in the prediction. The figure 4.3.1.15 show the in-depth analysis function run on the “example” template. Based on the figure 4.3.1.15, model that named with “example1” earned the highest accuracy as compared to other combinations which is around 57%, thus the system will suggest “example1” as the best model.

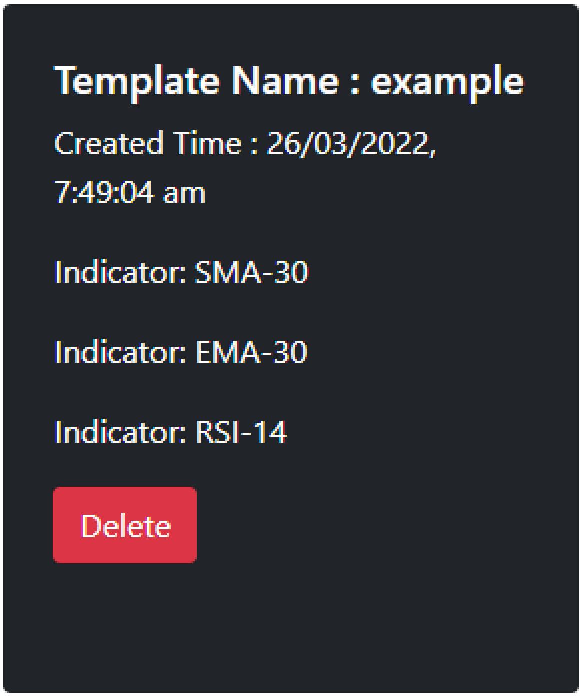



Figure 4.3.1.13 the Information about the “example” Template

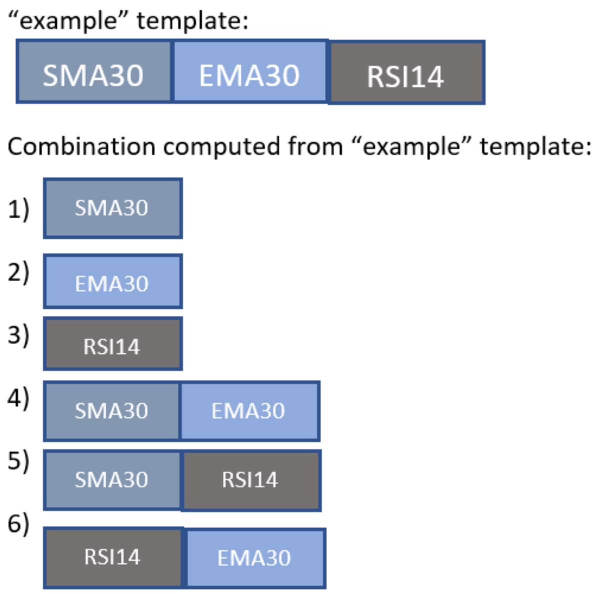
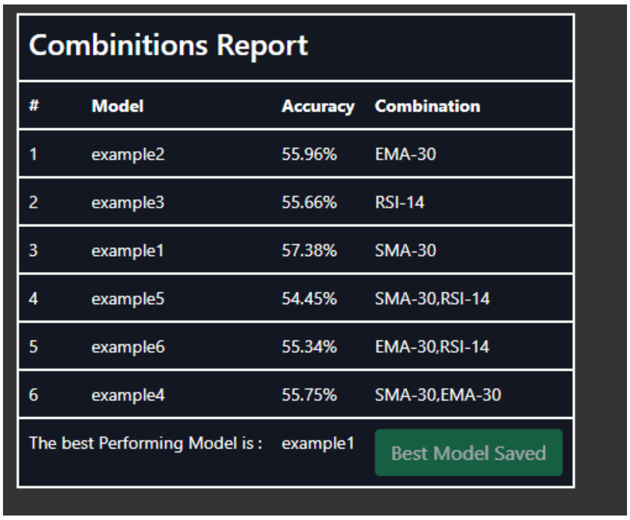



Figure 4.3.1.14 All Combinations Computed from "example" Template



**Combinitions Report**

| # | Model | Accuracy | Combination |
|---|---|---|---|
| 1 | example2 | 55.96% | EMA-30 |
| 2 | example3 | 55.66% | RSI-14 |
| 3 | example1 | 57.38% | SMA-30 |
| 4 | example5 | 54.45% | SMA-30,RSI-14 |
| 5 | example6 | 55.34% | EMA-30,RSI-14 |
| 6 | example4 | 55.75% | SMA-30,EMA-30 |
| The best Performing Model is : | | example1 | Best Model Saved |

Figure 4.3.1.15 In-Depth Analysis Function Run on the "example" Template.

In addition, the second part of the traning section, indicator suggestion section, is the section where the system will provide suggestions to the users on what indicators to be used before building the prediction model. Indicator suggestion tool is designed to help the non-technical users and users who are not familiar with the techincal indicators. For example, a user with no financial knowledge wanted to try out the prediction model provided by the system; however, during the selection of model template, he or she does not know what indicators to choose as the input to the model before building the model. Therefore, the users will probably input random indicators to the model. Randomly choosing an indicator combination as the input of the model might reduce the performance of the model. Thus, indicator suggestion tool is designed to suggest users with better options of indicator combination as the input to the model. Indicator suggestion tool will suggest users with the best combination of 1 indicator, best combination of 2 indicators, and best combination of 3 indicators as the possible inputs to the model. These suggested combinations are computed in advance from the offline training. The figure 4.3.1.16 show the process to compute the indicator combinitions suggestions for each stock in offline enviroment. Based on the figure 4.3.1.16, the system will compute all possible combinations based on the 5 popular indicators which are SMA-30days, EMA-30days, RSI-14days, OBV-2010, and ROC-14days, which will results total of 25 different combinations. Based on these 25 combinations, the system will filter out the best combination of 1 indicator, best combination of 2 indicators, and best combination of 3 indicators as the suggestion indicators for each stock, and these 3 best models will then be saved in the database so that they can be used as the suggestion options to the users later. Best combinition refers to the combination that achieved the highest prediction accuracy. The figure 4.3.1.17 show the interface of the indicators suggestion tool. Users can choose their favorite number of indicator to be included in the prediction, after that, a respective indicators suggestion will be highlighted in green boxes to the users. Since the best combination models were saved in the database in advance, users can choose to fetch a stock prediction directly from the database by using the suggested indicators. Users do not have to train the model again with the suggested indicator in the model training section, instead user can utilize the prediction function provided by the indicator suggestion tool. When the new users are getting familiar with the technical indicators, they can go ahead to build their own customized model in the model training section. The suggestions for each stock will be updated periodically by the programmers in the offline environment when the new data arrived. One point to take note is that indicator suggestion feature is completely different from the in-depth analysis indicator recommendation tool feature even though they both sound similar. Indicator suggestion tool is mainly for users who have no idea on what indicators to be used as the input of the model before the model training, whereas in-depth analysis recommendation tool is mainly for experienced users who want to study their selected indicators in detail manner. In the in-depth analysis feature, system will perform detail real-time analysis based on the indicators defined in the users selected template. For example, users selected a template that contain SMA-30days and EMA-30days, and the in-depth analysis recommendation tool will compute all possible combination based on these 2 indicators and then recommend a best combination from them, and these process are achieved in real-time. However, unlike the in-depth analysis recommendation feature, the indicator suggestion feature will not compute the suggestion options based on the dynamic input from the users in real-time, instead, it will compute the suggestion options based on the fixed 5 popular indicators in advance in the offline enviroment.

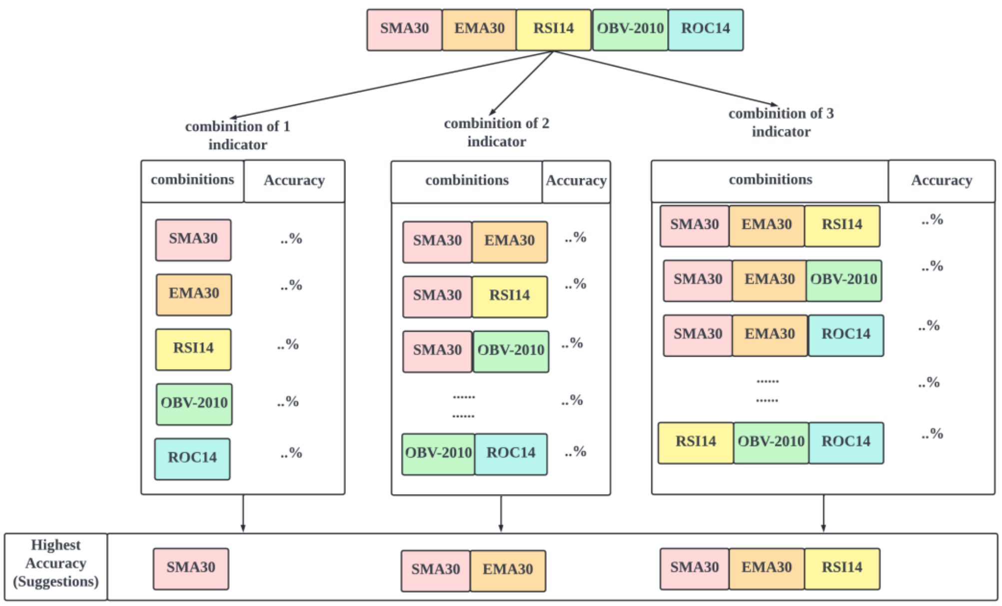



Figure 4.3.1.16 Process to Compute Indicator Combination Suggestions for Each Stock in Offline

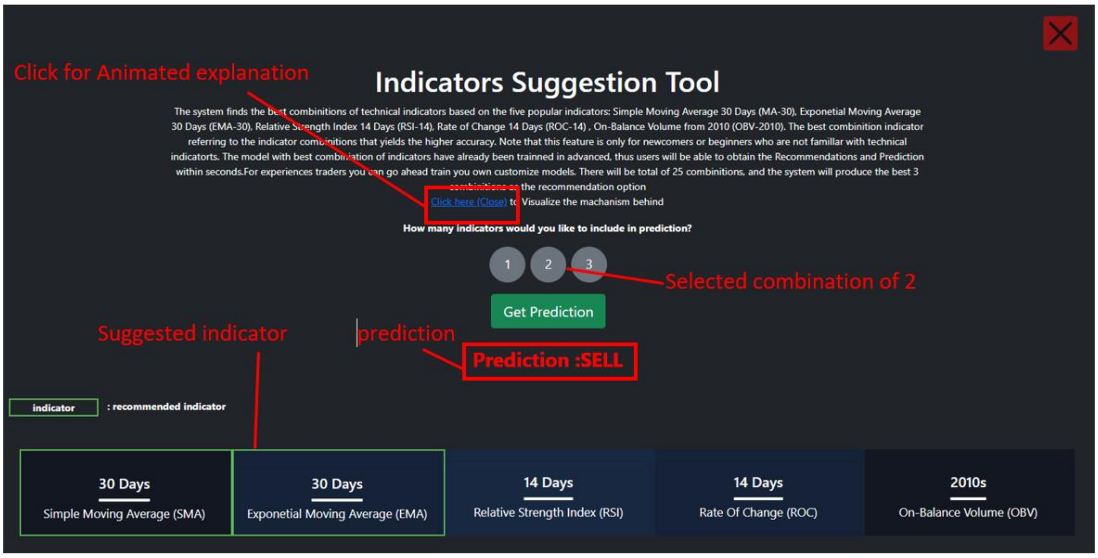



Figure 4.3.1.17 the Interface of the Indicators Suggestion Tool

### 4.3.2 Modelling Page

This page is mainly to allow customers to customize the input parameters of the model and save them as templates. So, users can use the created templates to train a model on the training section in the "Chart" page. The figure 4.3.2.1 show the GUI of modelling page. From the diagram 4.3.2.1, it is clearly showing that different model templates will contain different combinations of stock indicators. This page will be divided into 4 sections namely create template section, indicators section, animation tutorial section and saved template section. Create template section will be the place to create a new template.

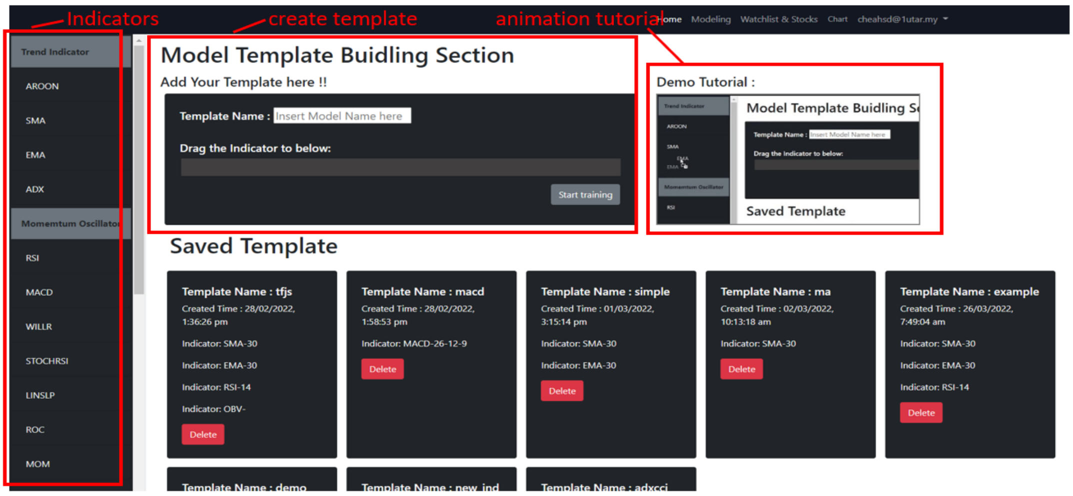



Figure 4.3.2.1 the GUI of Modelling Page

To create a new template, users must first key in the name of the template, and then drag the desired indicators from the second section, the indicator section, to the box provided. The main reason of providing this drag and drop feature is to allow user to better adapt to the functionalities and make it more user-friendly, rather than using the conventional form method to add the indicators. After users have dragged their preferred indicators to the provided box, an input window will be popped up to request the parameters of the selected indicators. After done addling the model template, the users can start utilizing the newly created template to perform model prediction in the "Chart" page. The figure 4.3.2.2 show the process of adding a model template. The third section will be the animation tutorial. This animation will illustrate the steps on how to perform drag and drop function when creating a new template. Showing animation to guide users is way better than expressing in the words form. Lastly, the saved template section will display the all the templates that have been added by the users.

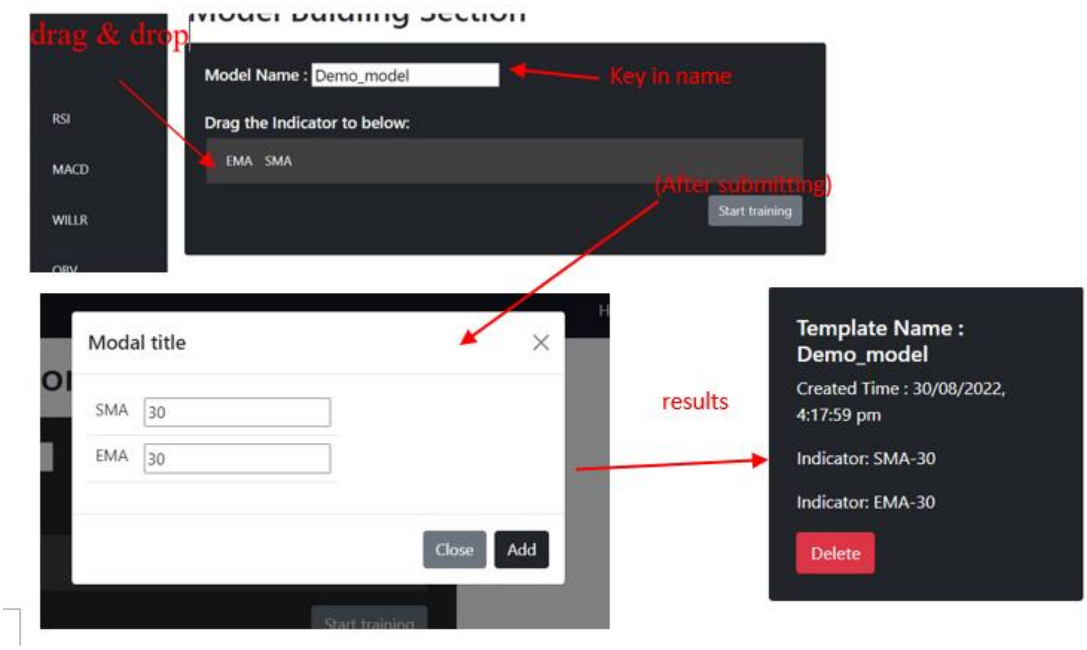



Figure 4.3.2.2 the Process of Adding a Model Template.

### 4.3.3 Watchlist & Stocks Page

This page is mainly to allow users to create watchlists. The figure 4.3.3.1 show the GUI of the Watchlist& Stocks page. Firstly, when users enter the page, an interactive and responsive table with 100 stocks information will be displayed to the users. Users can perform descending and ascending sorting on the table by clicking the header of each column except for the “change” column. In order to perform sorting on the “change” column, users have to click “Top gainers” button for descending sorting and “Top losers” bottom for ascending sorting. The reason of creating 2 extra buttons for this “change” column is that there are 2 well-known jargons in stock market can be used to describe the sorted results. When the “change” column is sorted in descending order, “Top gainers” can be used as the jargon to describe the sorted results and the same concept applies to ascending sorting which “Top losers” will be used.

Secondly, all watchlists created by the users will be displayed as the sidebar in the page, and the main watchlist will be highlighted in yellow color. Watchlists in the system will be divided into main watchlist and alternative watchlists. This is becauase the system will take main watchlist as the users’ highest priority and treated it differently in the application. For example, the system will display the status of stocks in main watchlist on most pages of the application. Therfore, if users want to closely monitor the status of some stocks in the application, users are recommeded to store the stocks in the main watchlist. In the sidebar, each watchlist will display the total number of stocks currently stored in the watchlist, and this number will be further divided into number of increasing stocks and decreasing stocks. To view the watchlist in detail, users can press the watchlist’ name. After pressing the name of watchlist, a pop-up window will be displayed to users. The figure 4.3.3.2 show the pop-up window content. In the pop-up window, the detail status information for each watchlist stock will be displayed, and because the stocks that are being stored in the watchlist are the stocks that users want to monitor closely, the offline prediction will also be included so that the users can always be notified with the prediction movement of the stock price. Moreover, users can choose to sort the content in the pop-up window by the stock’s price change, in this way, users would know what stock has the most increased/decreased in price. Figure 4.3.3.4 show the sorting function applied on the “Tech_stocks” watchlist. Thirdly, the latest updated time shown in figure 4.3.3.1 indicate the last update time of all information displayed in the page. The information will be updated to the latest changes automatically every time users refreshed the page, in this way, the information displayed in the page can almost be synchronized with the current stock market conditions. Besides that, an animated horizontal ticker will also be provided to users with the purpose to alert users on the top gainer stocks.

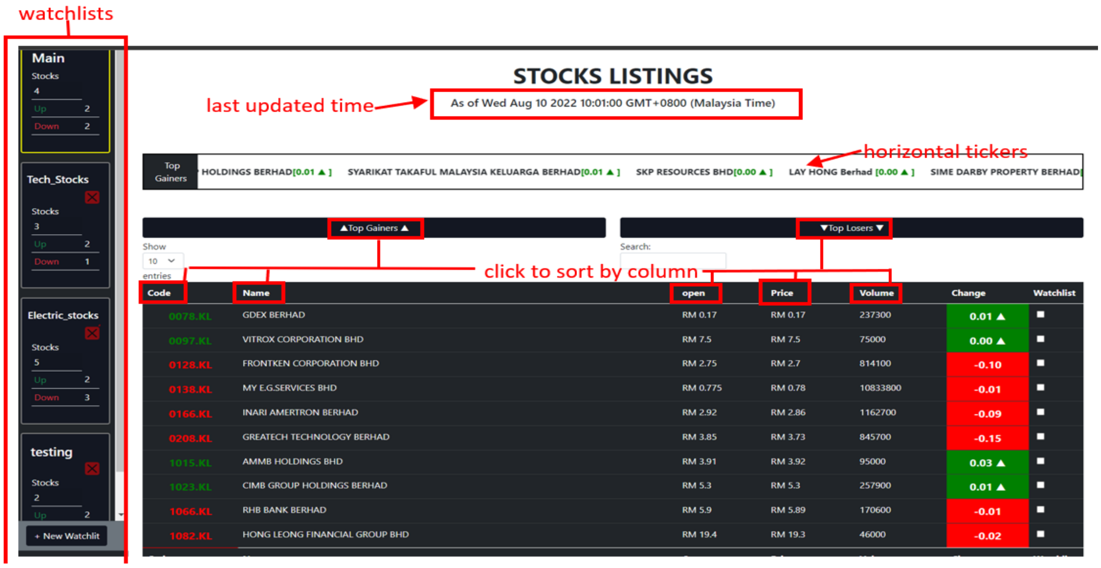



Figure 4.3.3.1 the GUI of the Watchlist& Stocks Page

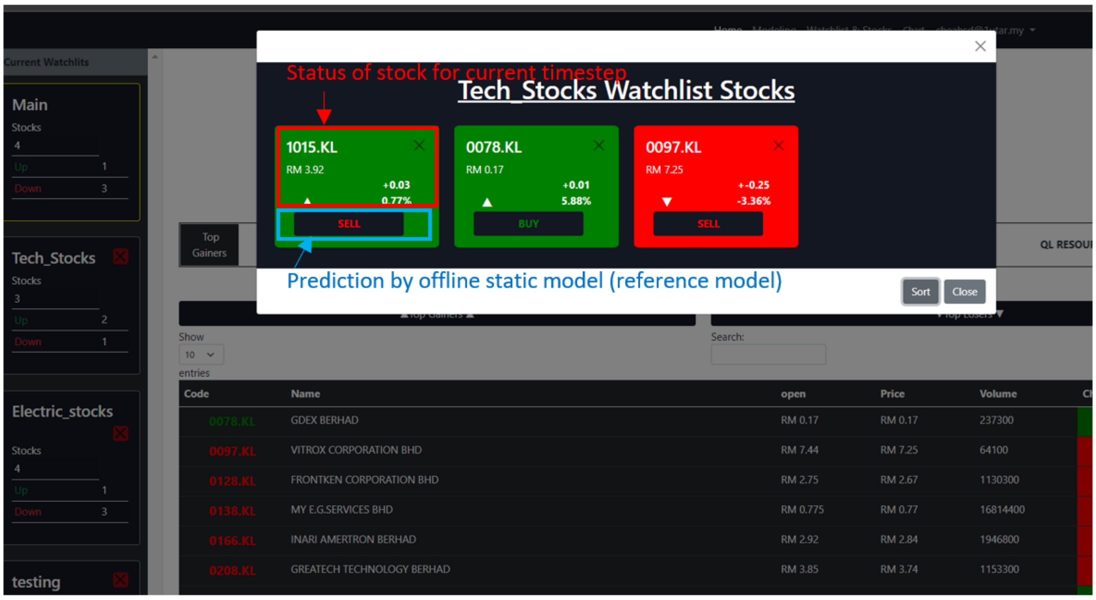



Figure 4.3.3.2 Pop-Up Window Content.

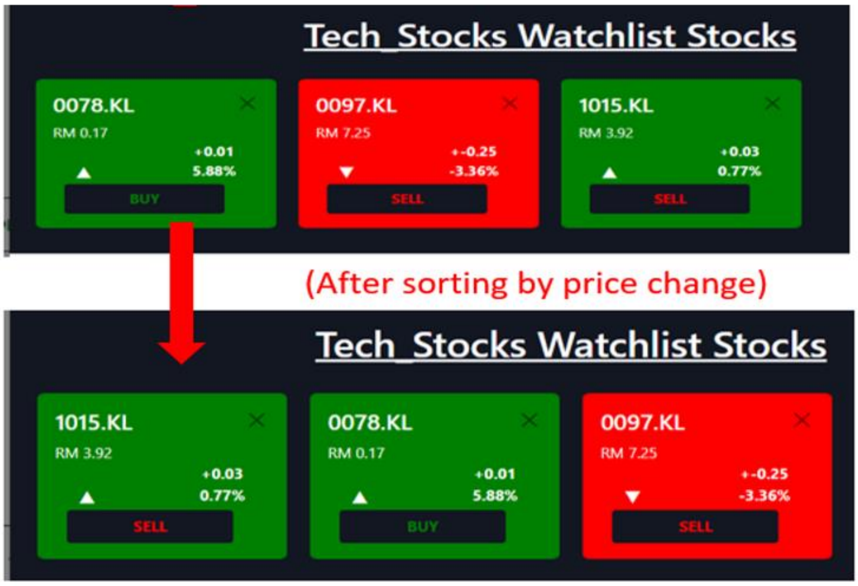



Figure 4.3.3.3 the Sorting Function for the “Tech_stocks” Watchlist

Finally, to add the stocks in the watchlist, users have to check the box of the desired stocks in the table, and after submitting the request, a window will be pop up to the users. In this pop-up window, users have to choose which watchlist they want to store in by selecting in the drop box option. Users could also delete the stocks in the pop-up window by clicking the “X” button before submitting the final request. Figure 4.3.3.4 illustrate the process of adding stocks into watchlist.

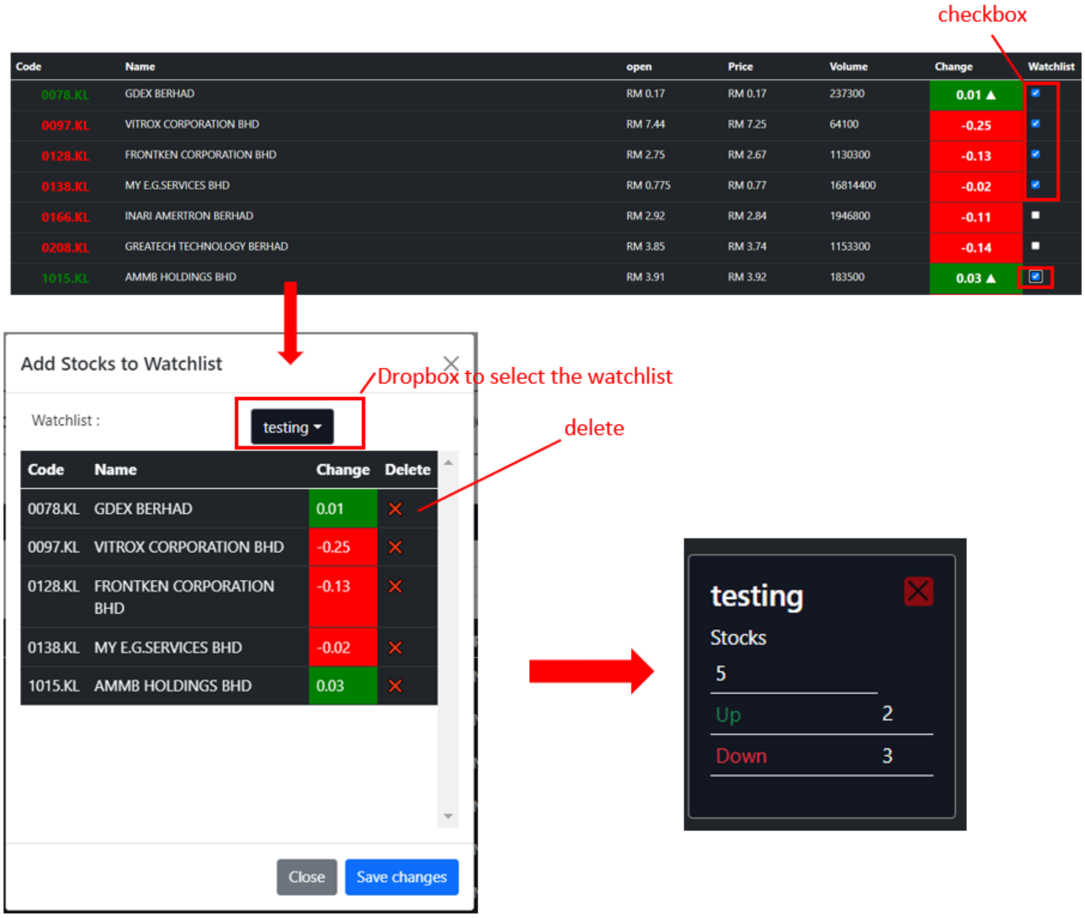



Figure 4.3.3.4 the Process of Adding Stocks into Watchlist.

## 4.4 Home Page

The main feature provided in the home page is the watchlist stocks training and prediction. This feature allows users to perform prediction and model training for all watchlist's stocks together at once. In this way, users able to monitor the prediction of all preferred stocks at a centralized place rather than monitoring them one-by-one separately in the "chart" page. For instance, if there are 5 stocks stored in a specific watchlist, having this feature, all 5 stocks can be trained and predicted by just clicking a single button. One strength about this feature is that all training process will be carried simultaneously and parallelly. Therefore, the time required to train 5 stocks will be almost similar as the time required to train 1 stock.

The figure 4.3.4.1 shows the interface of the home page. When users enter the "Home" page, the watchlist's stocks prediction and training features will first be appeared to the users. The offline prediction for main watchlist stocks will be displayed in the small absolute window. System will always utilize the reference model (offline static model) to fetch the current timestep prediction for the main watchlist stocks so that users will always be notified with the predicted movement of the stock prices. In online model section, there will be 2 modes provided namely "Pretrained" and "Train new model" which are almost similar as prediction section and training section mentioned in the "chart" page. "Pretrained" mode is used to perform prediction for watchlist stocks using the pretrained online models, whereas "Train new model" mode is used perform online model training for watchlist stocks. The figure 4.3.4.2 shows the view of "Pretrained" mode and "Train new model" mode. For new users who just joined to the system, no online pretrained models have been created in the "Pretrained" mode yet as highlighted with "empty" in figure 4.3.4.1. In order to get the predictions on the watchlist stocks, users have to build online prediction models in the "Train new model" mode.

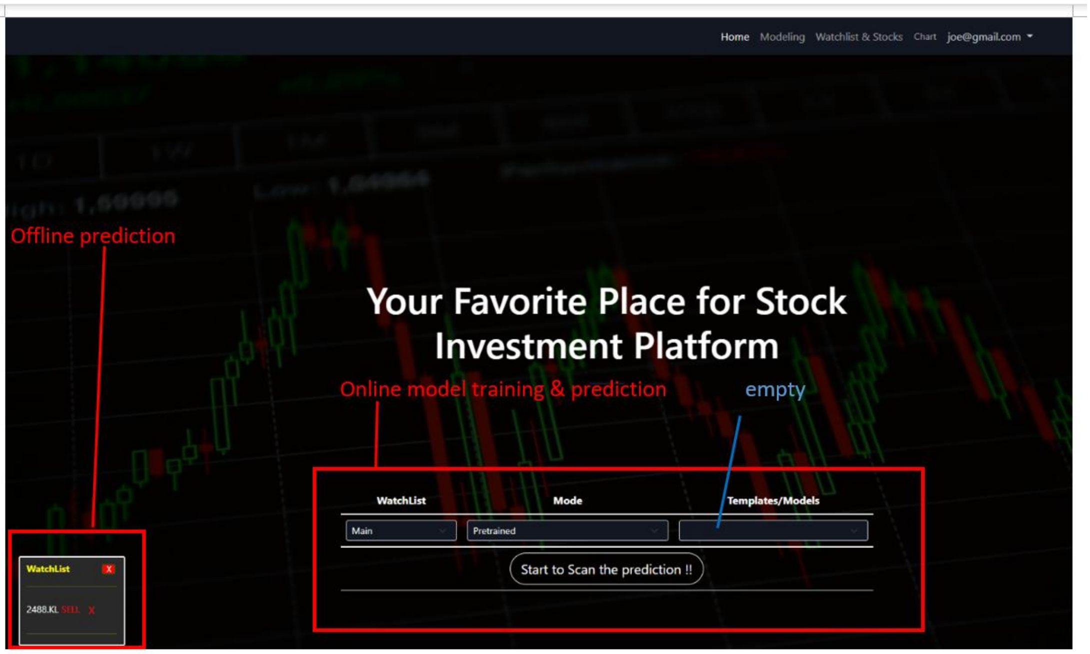



Figure 4.3.4.1 GUI of Home Page

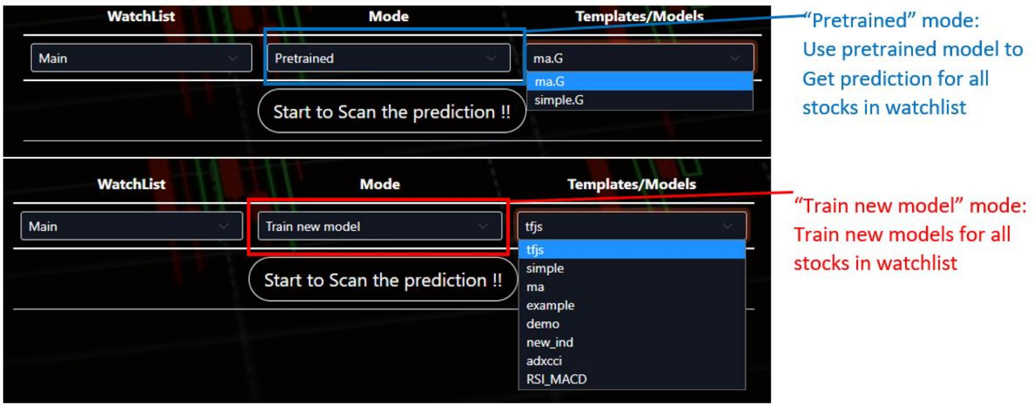



Figure 4.3.4.2 Training Section and Prediction Section of Watchlist Stocks

To build online prediction models for the stocks in a specific watchlist. The first step is to switch the mode of the online model section to "Train new model", and then select a desired watchlist and desired template for training. In this case, main watchlist and "ma" template were selected for demonstration. The selections are illustrated in figure 4.3.4.3. Based on figure 4.3.4.4, there are 4 stocks currently stored in the main watchlist, thus the system will run the model training process for all 4 stocks at once. The whole training process will be carried out parallelly and will take roughly 3 to 5 minutes. After all training process is done, the average prediction accuracy for all stocks will be displayed. The figure 4.3.4.5 shows the training process of the watchlist stocks in detail. After the 4 different models have been trained, the models will be named with the template name appended with the ".g" so that they are differentiable with the single stock models built in "Chart" page. In this case, the models will be named with "ma.g".

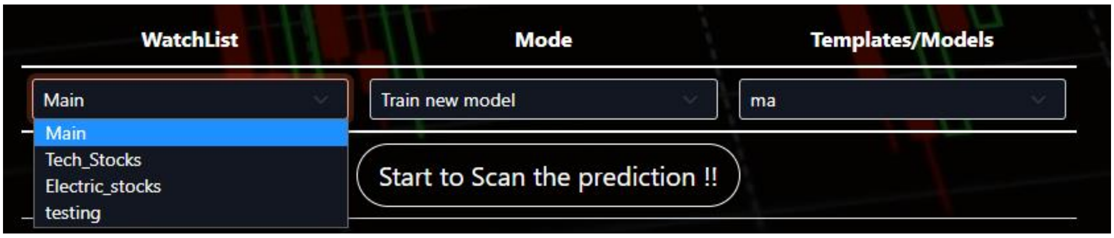



Figure 4.3.4.3 the Selections of Watchlist and Template to Train a New Model

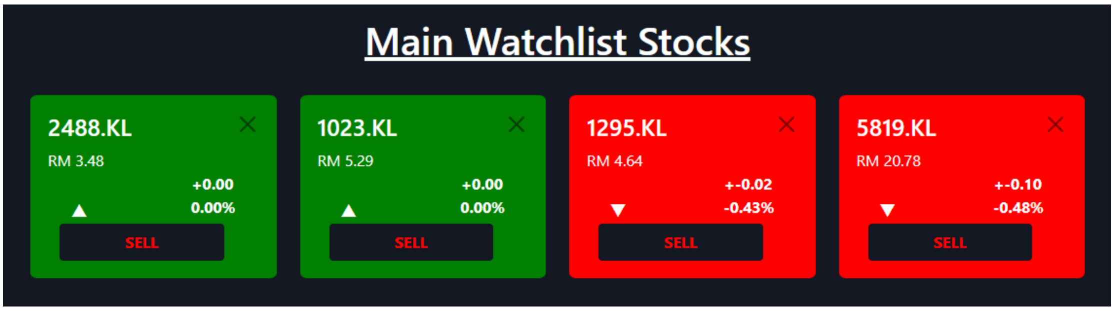



Figure 4.3.4.4 Current Stocks in Main Watchlist


Figure 4.3.4.5 the Training Process of the Main Watchlist Stocks Using “Ma” Template

After users have trained and saved the online prediction model for each stock in watchlist, users can obtain the immediate predictions for these stocks at any point in the future without the need of waiting again. To obtain the predictions, users have to switch the online model section mode to “Pretrained”, and then select the desired watchlist and desired pretrained models. The figure 4.3.4.6 shows the process of watchlist stocks prediction using the pretrained models. Based on the figure 4.3.4.6, main watchlist and “ma.g” pretrained models were selected. “ma.g” are the prediction models that have just built in figure 4.3.4.5.

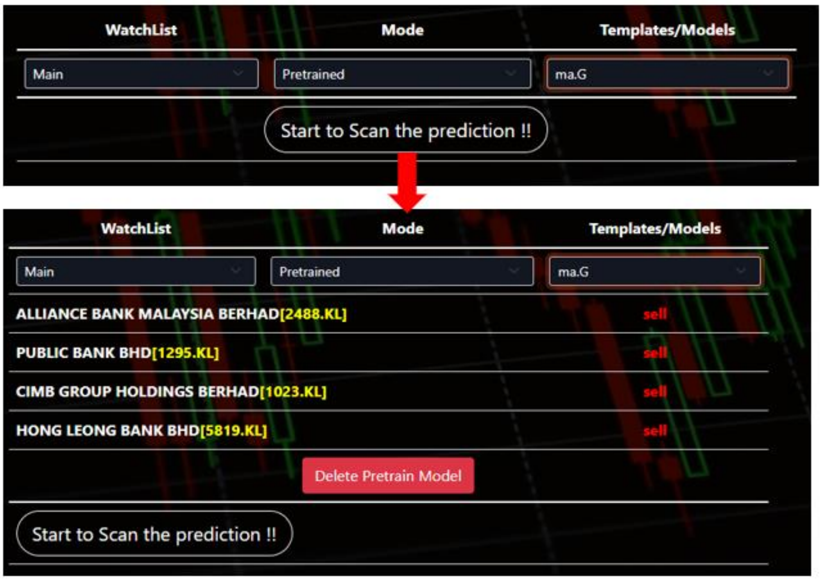



Figure 4.3.4.6 Prediction Section of the Pretrained Model

However, it would result an issue if traders just added a new stock in watchlist and directly go into the “Pretrained” mode to get the prediction of all current stocks in watchlist. The figure 4.3.4.7 shows the watchlist with one newly added stock. As we know that the newly added stock has not gone through any training yet. The system cannot get a prediction for new stock as there is no pretrained model associated to it. The solution is shown in the figure 4.3.4.8 where the system will automatically run the training process for the newly added stock in the “Pretrained” mode as well.

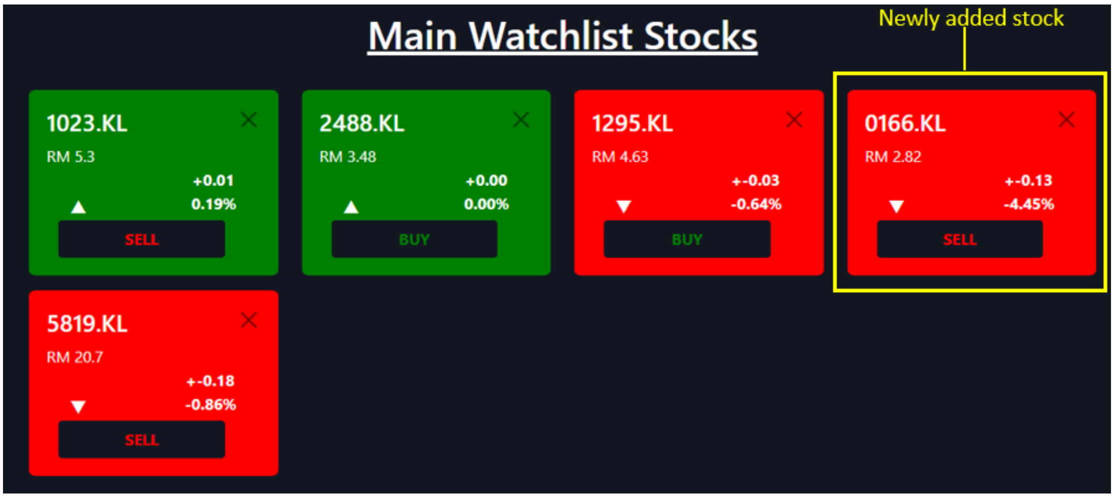



Figure 4.3.4.7 the Newly Added Stock in Main Watchlist


Figure 4.3.4.8 Solution of the Watchlist Stock Prediction Issue

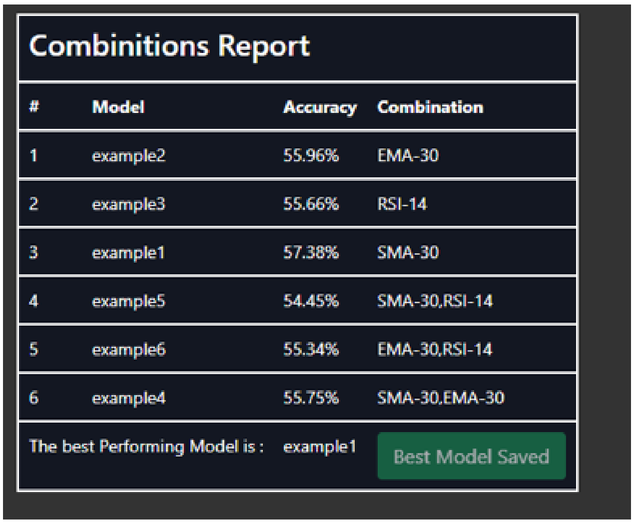
## 4.4 Parallel Processing Design

In the system, there are 2 features can be run using parallel processing approach which are in-depth analysis indicator recommendation feature, and watchlist stocks training and prediction feature. One thing these 2 features have in common is the need to train multiple stock prediction models in the system. Instead of training multiple prediction models one-by-one in the system, parallel processing is applied to train all of them simultaneously in the system. The section will describe the detail of how the parallel processing is being applied for these 2 features.

### 4.4.1 Parallel Processing Design in In-Depth Analysis Indicators Recommendation Feature

In-depth analysis indicator recommendation feature will recommend users with the best indicator combination that achieved the highest accuracy among all combinations that generated from the original combination. Let us take the in-depth analysis feature that had run on the “example” template as shown in figure 4.3.1.15 for the demonstration. Based on the figure 4.3.1.13, “example” template is made up of SMA 30 days, EMA 30 days and RSI 14 days indicators. The feature will first compute all possible combinations that can be generated from SMA 30 days, EMA 30 days and RSI 14 days indicators. In this case, a total of 6 possible combinations will be generated. After that, the system will use each indicator combination as the inputs to train a prediction model and make prediction on the testing data. With parallel processing applied, all 6 combinations will be trained simultaneously in the system. Finally, the system will take SMA 30days combination as recommendation to the users as it achieved the highest prediction accuracy. Figure 4.4.1.1 shows the steps of process that run behind the system for the function shown in figure 4.3.1.15

In GUI:




The mechanism behind:

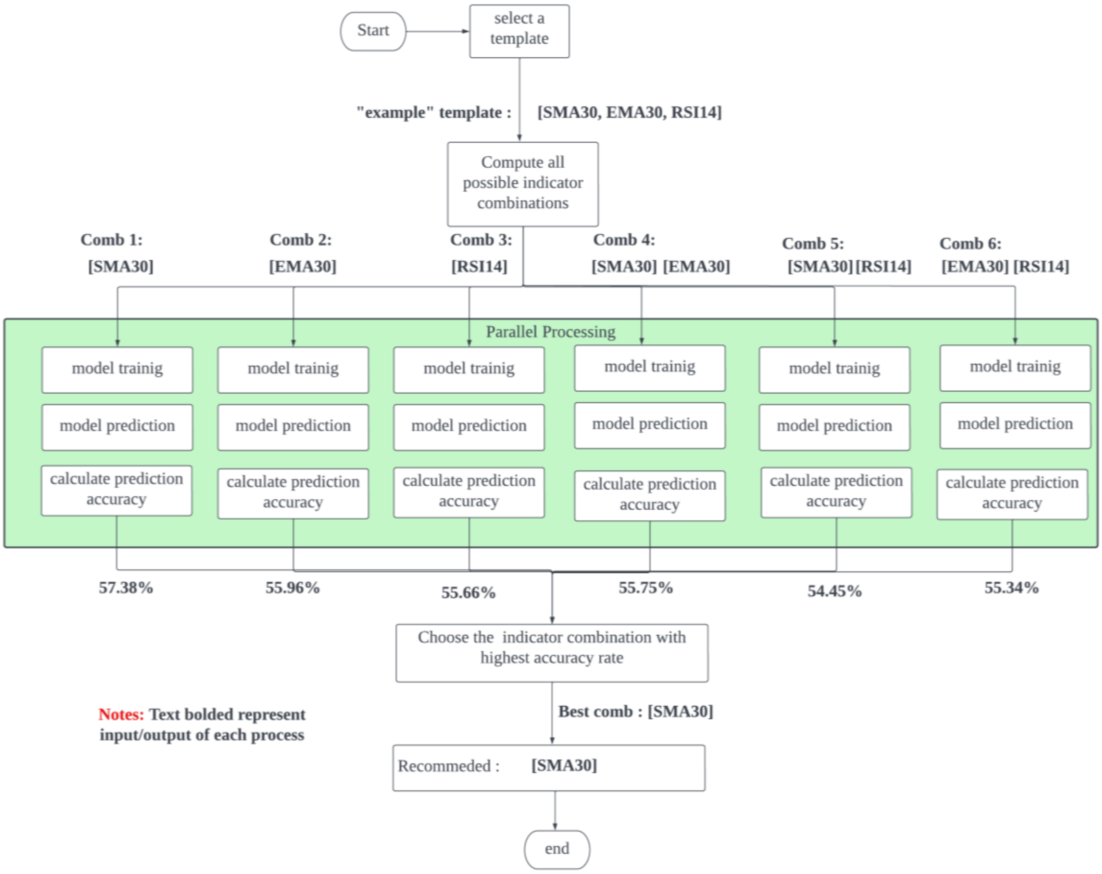



Figure 4.4.1.1 Steps of Process that Run Behind the System for the Function Shown in Figure 4.3.1.15

### 4.4.2 Parallel Processing Design in Watchlist Stocks Training and Prediction Feature

Watchlist stocks training and prediction feature allows users to perform stock prediction and model training for all watchlist's stocks together at once. Let us take the training process of the main watchlist stocks using "ma" template shown in figure 4.3.4.5 for demonstration. Based on the figure 4.3.4.4, A total of 4 stocks are currently stored in the main watchlist. Therefore, the watchlist stocks training and prediction feature will train a prediction model for each stock in the system using the "ma" template, thus 4 prediction models will be resulted at the end of the process. With parallel processing, model training process for all stocks will be performed simultaneously. After the 4 prediction models have been trained, the system will calculate the average accuracy and display it to the users. The figure 4.4.2.1 shows the steps of process that run behind the system for the function shown in figure 4.3.4.5

In GUI:


The mechanism Behind:

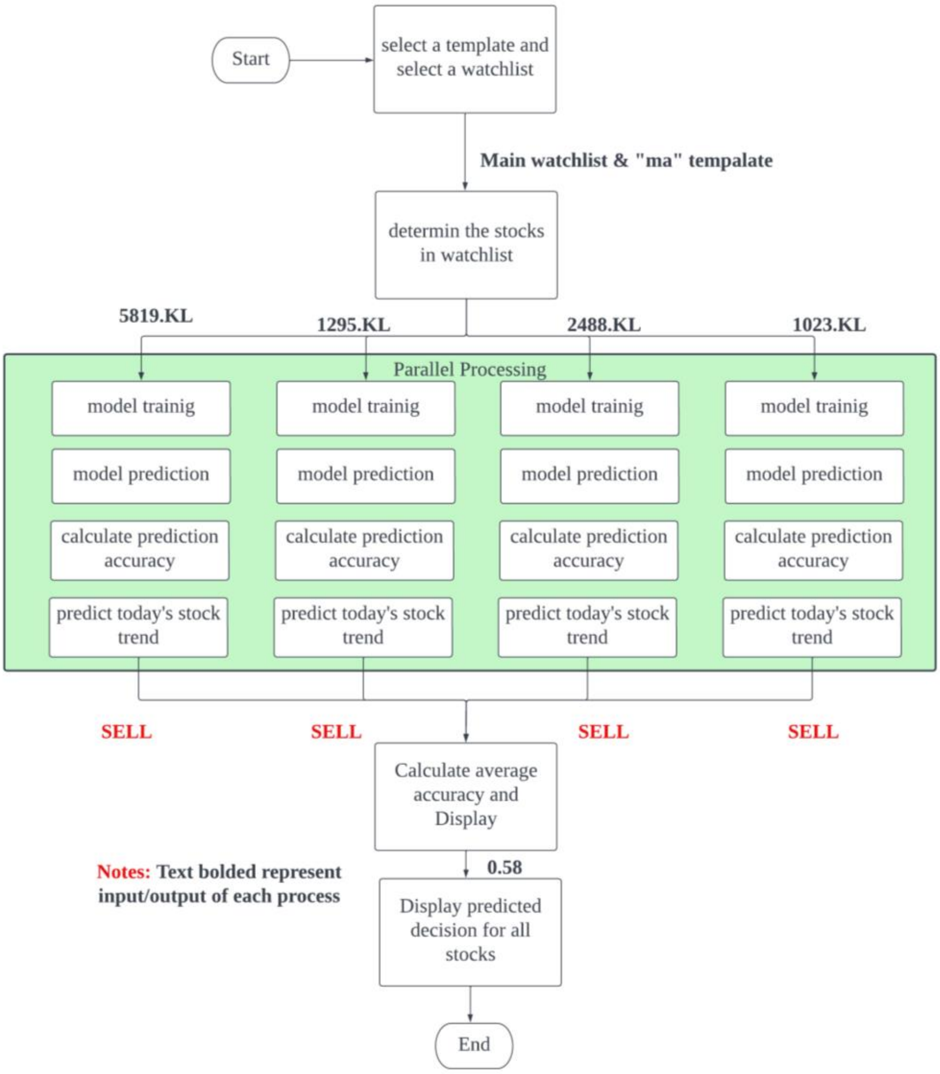



Figure 4.4.2.1 Steps of Process that Run Behind the System for the Function Shown in Figure 4.3.4.5

## 4.5 Stock Prediction Functionality Design

In this system, there will be 2 types of prediction can be performed for each of the stocks namely offline model prediction and online model prediction. The reasons of creating 2 types of prediction model and the difference between these 2 types of prediction model will be thoroughly discussed in this section.

### 4.5.1 Offline Model Prediction

Based on the name itself, we would know that the offline model prediction is generated by the offline models. Offline models are the model that have been previously trained by programmers using the fixed indicators in the offline environment. There are few reasons why offline model is provided in the system. The first reason is that it will act as reference models in the system which mean the users can get a prediction for a particular stock immediately in the system without the need to create anything on the website. This is particularly useful for new users what have not registered to the system. Since new users (non-registered users) are not given the privilege to create online customized model in the system, new users will not have the chance to experience the prediction function provided by the system. This is where offline model will solve the problem, offline models will allow any users including non-registered to perform stock prediction. In this way, new users can utilize the offline prediction to assist them in decision making even they are not registered. After having tried on prediction feature in the system, the new users might want to have more privilege in customizing the prediction models, thus the new users would possibly register an account in the system. The second reason is that offline models will be utilized by the system to perform periodical prediction for all stocks that stored in main watchlist. Since the stocks that stored in the main watchlist are the stocks that users want to monitor closely, offline model prediction for the main watchlist's stock will be displayed as the small window in most of pages in the system so that the users will always be notified with the status of the main watchlist' stocks. Moreover, the reference models (offline models) provided by the system will guarantee prediction accuracy of 50% and above across all 100 stocks. However, the accuracy of the prediction model could be pushed up further if it incorporated with experienced traders' domain knowledge. The table 4.5.1.1 shows the performance of offline model and online model for 6888.KL stock. Based on the table results, we can see that the accuracy and cumulative return of the prediction model for 6888.KL stock can be increased further by inputting another relevant indicator (RSI14) to the model. Therefore, customizing an online model still be a better choice for users to obtain higher accuracy for the prediction model.

The table 4.5.1.1 Performance of Offline Model and Online Model for 6888.Kl Stock.

| Stock | Model type | Accuracy | Cumulative return |
|---|---|---|---|
| 6888.KL | Offline model<br>[SMA30, EMA30] | 53% | 60.48% |
| 6888.KL | Online model<br>[SMA30, EMA30, **RSI14**] | 54% | 104.72% |

The figure 4.5.1.1 shows the offline model building process (red arrows) and offline model prediction process (green arrows). Firstly, the offline prediction model will be built by programmers in advance using the predetermined indicators in the offline environment (in separate platform). After offline models have been trained, they will be saved into the database. Whenever users want to perform offline prediction for a particular stock using the offline models, the system will fetch the offline models from the database and perform the prediction for that stock. After obtaining the prediction, the controller in server-side will send the predicted result to the front-end and display it to the users.

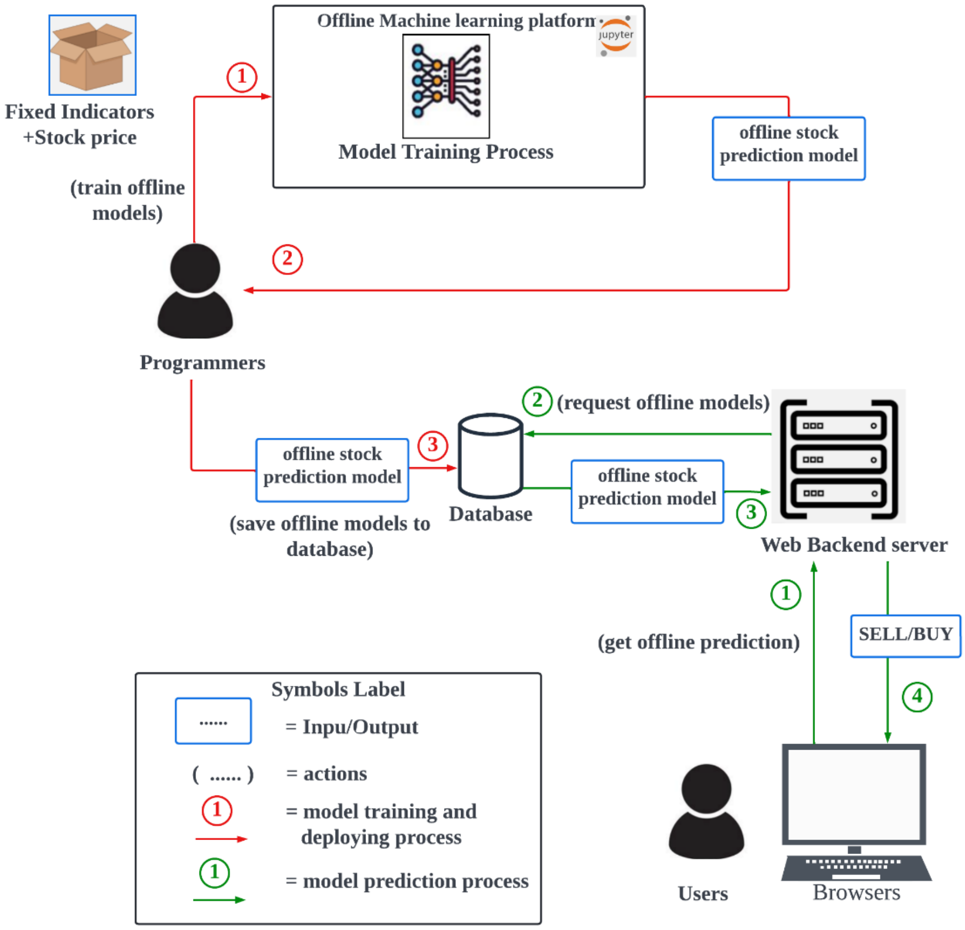



Figure 4.5.1.1 Offline Model Building Process (Red Arrows) and Offline Model Prediction Process (Green Arrows).

### 4.5.2 Online Model Prediction

Similarly, online model prediction is provided by the online models built in the system. Online models are the models that have the same architecture structure as the offline models, but the input indicators used will be depends on users' dynamic choices, and it will be built in the real-time upon customer's request. Further, providing online models is the main goal of the project as we want to provide a more flexible and customizable prediction model for users so that we can adapt the domain knowledge of the experienced traders into the machine learning algorithm to make more meaningful prediction and push up the accuracy of the prediction model.

The figure 4.5.2.1 shows the online model building process (red arrows) and online model prediction process (green arrows). Firstly, the online prediction model will be built according to the users' customized indicators in real-time. After the online customized models have been trained, it will be saved under the user's account in the database, and then the system will display the performance report to users. Whenever users want to perform prediction for a particular stock using the previously trained online customized model, the system will fetch the model from the database and perform the prediction for that stock. After obtaining the prediction, the controller in server side will send the predicted result to front-end and display it to the users.

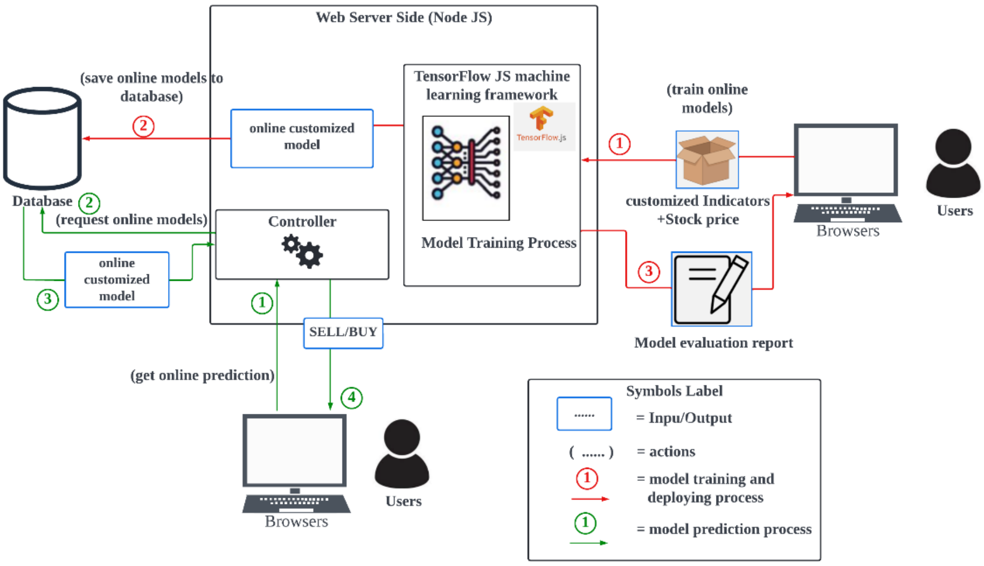



Figure 4.5.2.1 Online Model Building Process (Red Arrows) and Online Model Prediction Process (Green Arrows).

In summary, based on the figure 4.5.1.1 and figure 4.5.2.1 the main difference between offline prediction model and online prediction model is that offline prediction models will be built by the programmers, and all offline prediction models must be pretrained and saved in the database before the web application is deployed into production, whereas online prediction models will be built by system users in real-time on the web application.

# Chapter 5 System Testing

## 5.1 A Simulation on The Prediction Test for The Proposed Model, Article Model, Buy-And-Hold Strategy, and Optimistic Strategy

One of the objectives of this project is to provide stock prediction model for all 100 stocks that can achieve state-of-the-art performance which is accuracy rate of 50% and above as well as to guarantee at least 60% of the provided stocks achieve positive cumulative return when the proposed model is being used as the decision tool in the actual stock trading. A simulation of the prediction test procedure and results are presented in this section.

### 5.1.1 Test Procedure

A simulation will be carried out using the proposed model to predict the stock price movement from year 2019 to year 2022. The stock data between year 2019 and year 2022 will not be included in the model training data so that the proposed model will make prediction based the unseen data and avoid look-ahead bias. The proposed model will be evaluated using 4 performance metrics which are prediction accuracy, cumulative return, Sharpe ratio, and Maximum Drawdown. Prediction accuracy measure the correctness of the model prediction, cumulative return measure the capital gained by the model prediction, Sharpe ratio measure the amount of reward per risk, and Maximum Drawdown measure the potential risk of the trading operations. In addition, the simulation process would also compare the performance of the proposed model strategy with the Buy-and-Hold strategy, Optimistic strategy and the LSTM model that used by the article [24]. Further, A total of 3 rounds of simulation will be performed per stock. Because of the random initialization of some parameters in the LSTM algorithms such as the weight of the nodes in each gate, the model performance will be slightly different for each run. Therefore, 3 rounds of simulation will be run for each stock to obtain the average accuracy rate and average trading performance. In the simulation, SMA-30days and EMA-30days will be used as the technical indicators to train the proposed model as these are the best indicators that were filtered by the correlation tensor for most of the stocks. The figure 5.1.1.1 shows the simulation on the prediction test procedure for the proposed model.

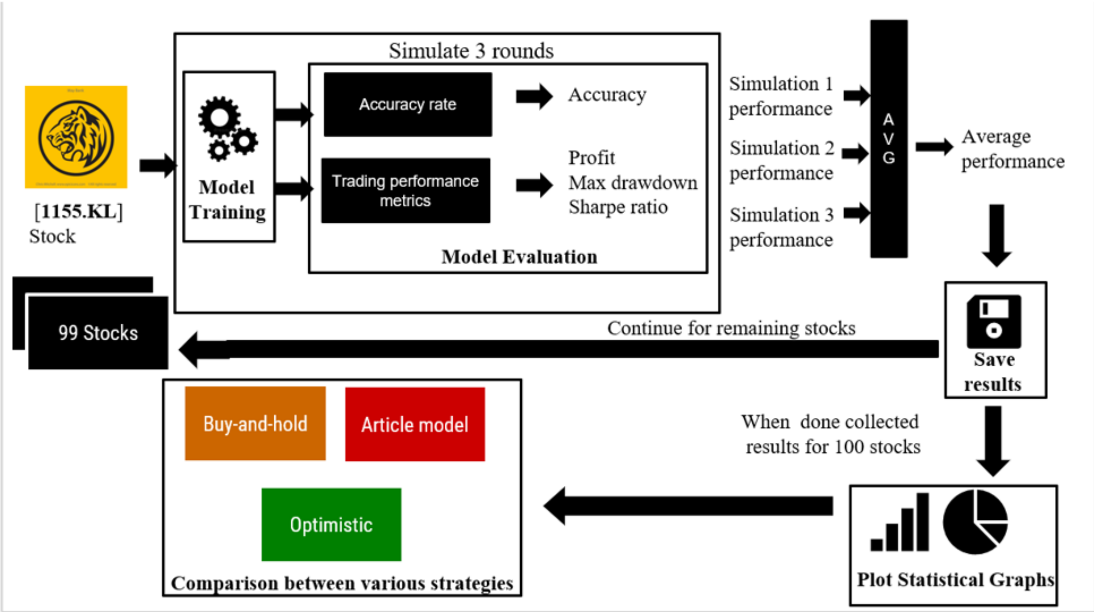
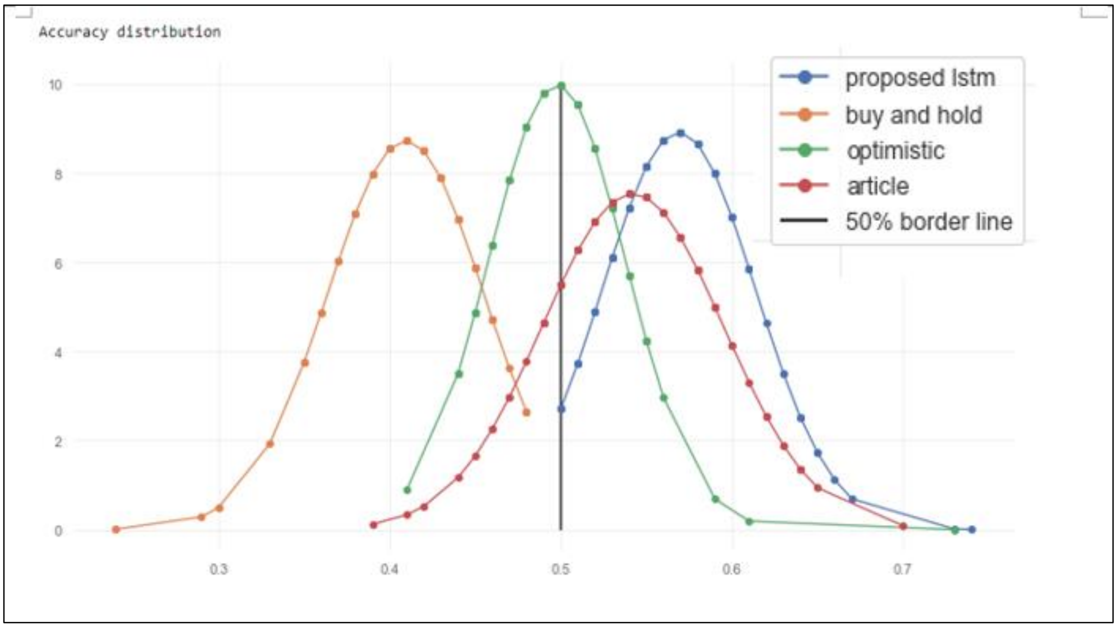



Figure 5.1.1.1 The Simulation on the Prediction Test Procedure for the Proposed Model

### 5.1.2 Average prediction accuracy rate results for the simulation

After the simulations have been done for all 100 stocks, the results will be graphed for better comparison. The figure 5.1.2.1 shows the average prediction accuracy distribution of 100 stocks for 4 different strategies along with a summary table. To understand how the distribution graph is plotted, let us take blue curve graph (proposed model) as the example. After the system has run 3 rounds of simulation for a particular stock using the proposed model, the average model accuracy for that stock will be plotted as a blue dot in the graph, combining the average accuracy for the remaining 99 stocks, it will form a distribution of 100 blue dots as shown in the figure 5.1.2.1. Based on the minimum accuracy rate of the proposed model shown in the summary table, it is evident that the proposed model (blue curve) achieved at least 50% accuracy for all 100 stocks. This indicate that the proposed models have passed the first part of the test which is to achieve accuracy of 50% and above. Another interesting point we can observe from the figure is that the proposed model is the best performing model as compared to other strategies as the average accuracy rate of 100 stocks is the highest which is around 57%. The LSTM model used by the article [24] only obtained average accuracy of 54%, and it even achieved minimum accuracy that is less than 50%, which is 39%. This prove that the proposed model is way better than the LSTM model used by the article [24]. In addition, the reason why Buy-and-hold strategy and Optimistic strategy achieved significantly worst results as compared to the proposed model and article model is that there is no statistical analysis involved in both strategies; Therefore, they are expected to obtain much lower accuracy rate than the machine learning models. The complete average accuracy results of 100 stocks for 4 different strategies is listed in appendix A.3.1

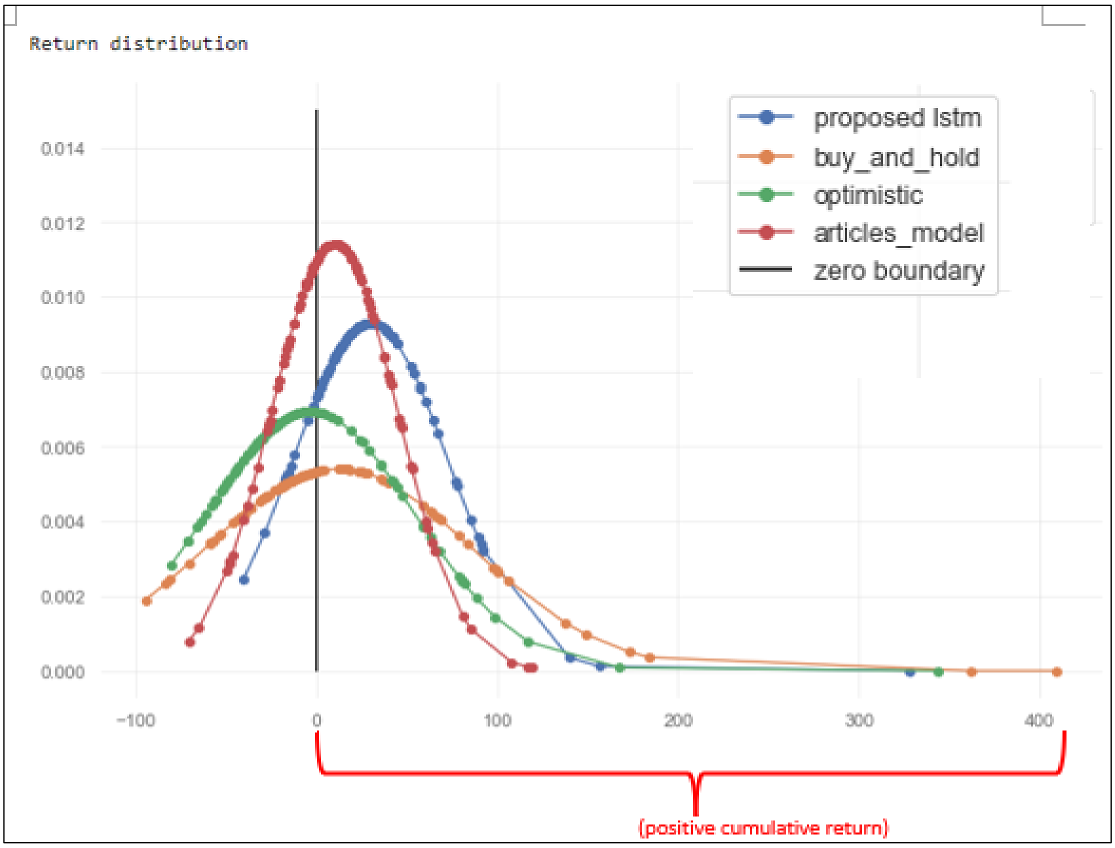



| Strategy | Average Accuracy | Max Accuracy | Min Accuracy | stdev |
|---|---|---|---|---|
| Proposed model | 0.57 | 0.74 | 0.50 | 0.045 |
| Article model | 0.54 | 0.70 | 0.39 | 0.053 |
| Buy-and-Hold | 0.41 | 0.48 | 0.23 | 0.045 |
| Optimistic | 0.49 | 0.72 | 0.41 | 0.039 |

Figure 5.1.2.1 Average Accuracy Results of 100 Stocks for 4 Different Strategies

### 5.1.3 Average Cumulative return results for the simulation

Other than the accuracy rate, the proposed model will also be evaluated from the trading point of view. The figure 5.1.3.1 shows the average cumulative return distribution of 100 stocks for 4 different strategies along with the summary table. The black line in the graph indicates the boundary of the zero cumulative return, any values lower than the boundary indicate a negative cumulative return. In best case, the graph is expected to skew towards the positive return direction which is skew to the right. According to the summary table shown in figure 5.1.3.1, the proposed model simulation results have 90% of the stocks with positive cumulative return. While Buy-and-Hold, Optimistic and article model have 41%, 30% and 59% of the stocks with positive return, respectively. Thus, the proposed model is better compared to other strategies. Further, the proposed model achieved an average cumulative return of 29% across 100 stocks which is the highest among the 4 strategies. The maximum capital losses experienced by the proposed model also the lowest among the strategies which is only around 40% of losses. The complete average cumulative return results of 100 stocks for 4 different strategies is listed in appendix A.3.2



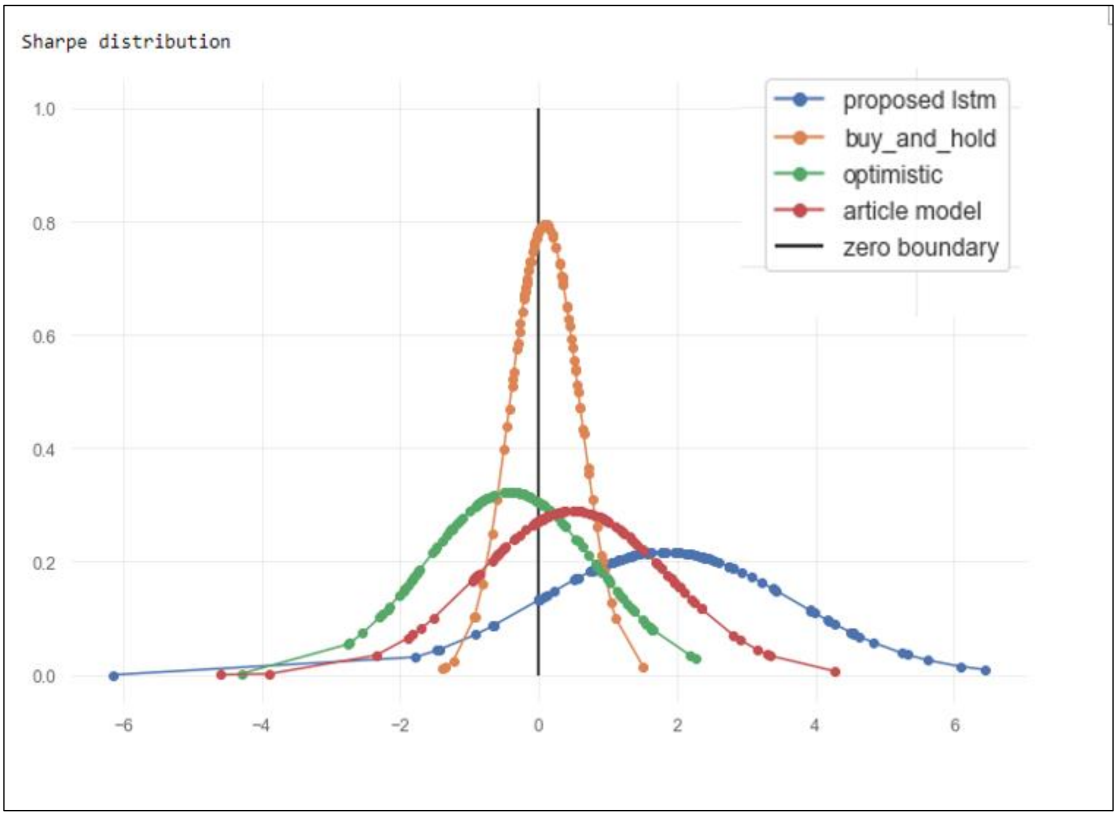

| Strategy | Average Cumulative Return (%) | Max Cumulative Return (%) | Min Cumulative Return (%) | Cumulative Return stdev | Cumulative Positive Return | Cumulative negative Return |
|---|---|---|---|---|---|---|
| Proposed model | 29 | 327 | -40 | 43.15 | 90 stocks | 10 stocks |
| Article model | 10 | 119 | -70 | 35.11 | 59 stocks | 41 stocks |
| Buy-and-Hold | 12 | 409 | -94 | 74.16 | 41 stocks | 59 stocks |
| Optimistic | -3 | 343 | -80 | 57.85 | 30 stocks | 70 stocks |

Figure 5.1.3.1 Average Cumulative Return of 100 Stocks for 4 Different Strategies

### 5.1.4 Average Sharpe ration and Maximum Drawdown results for the simulation

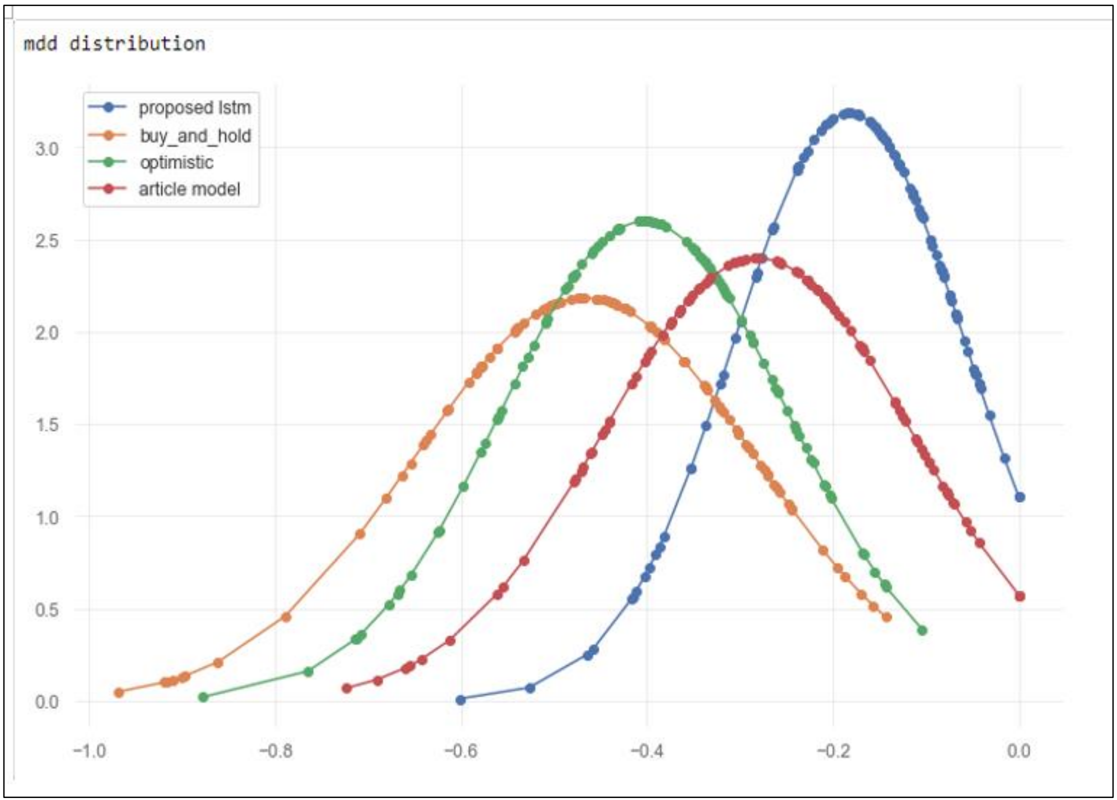
To further study the risk performance of the proposed model, Sharp ratio and Maximum Drawdown distribution will be computed. As mentioned in the above chapters, Sharpe ratio represent the amount of reward per single risk, and Maximum Drawdown represent the risk that experienced by the portfolio before it recovers to the new peak (potential risk). To be a good strategy, it must achieve Sharpe ratio at least larger than 1, and Max drawdown that is close to 0. In other words, the lower the Maximum Drawdown value and the higher the Sharpe ratio, the better the trading strategy. The figure 5.1.4.1 shows the average Sharpe ratio distribution of 100 stocks along with the summary table. Based on the Sharpe ratio distributions of the proposed model, it is evident that the proposed model is best strategy as compared to others as its distribution graph is closer to the right direction, indicating majority of stocks achieved high Sharpe ratio. Based on the average Sharpe ratio value in the summary table, the proposed model achieved 1.83 which is way higher than other strategies. Further, the figure 5.1.4.2 shows the average Maximum Drawdown distribution of 100 stocks along with the summary table. Based on the Maximum Drawdown graph, it is again evident that the proposed model is the least risky strategy as the graph is much closer to zero, and the average Maximum Drawdown value for the proposed model also the lowest which is around -0.18 (-18 %). The complete average Sharpe ratio and Maximum Drawdown results of 100 stocks for 4 different strategies is listed in appendix A.3.3 and appendix A.3.4 respectively.

In short, based on the simulation results, the proposed model has passed the test. The proposed model achieved prediction accuracy of 50% and above for all 100 stocks and achieved positive cumulative returns for 90 stocks. Based on the comparison between 4 strategies in overall, the proposed model is the best strategy as it has the highest average cumulative return, and lowest capital losses and risks. The LSTM model used by the article did not perform well in this experiment due to the model is originally designed for National Stock exchange of India (NSE) instead of Kula Lumpur Stock exchange (KLSE), but it did better than the benchmark strategy (Buy-and-hold) and Optimistic strategies.




| Strategy | Average Sharpe Ratio | Max Sharpe Ratio | Min Sharpe Ratio | Stdev Sharpe Ratio |
|---|---|---|---|---|
| Proposed model | 1.83 | 6.44 | -6.14 | 1.84 |
| Article model | 0.49 | 4.27 | -4.58 | 1.38 |
| Buy-and-Hold | 0.09 | 1.51 | -1.38 | 0.50 |
| Optimistic | -0.41 | 2.27 | -4.28 | 1.24 |

Figure 5.1.4.1 Average Sharpe Ratio of 100 Stocks for 4 Different Strategies




| Strategy | Average Maximum Drawdown | Max Maximum Drawdown | Min Maximum Drawdown | Stdev Maximum Drawdown |
|---|---|---|---|---|
| Proposed model | -0.18 | -0.60 | 0.00 | 0.12 |
| Article model | -0.28 | -0.72 | 0.00 | 0.16 |
| Buy-and-Hold | -0.46 | -0.96 | -0.14 | 0.18 |
| Optimistic | -0.40 | -0.87 | -0.10 | 0.15 |

Figure 5.1.4.2 Average Maximum Drawdown of 100 Stocks for 4 Different Strategies

## 5.2 A simulation on the prediction test for the proposed model using the recommended indicators and original indicators.

In-depth analysis indicator recommendation feature is one of the main features provided in the system. This feature will recommend the best combination of indicator to users based on the original indicators that defined in the selected template. To ensure the recommended indicators combination is indeed better than the original indicator combination in terms of model accuracy, a simulation of the prediction test will be run.

### 5.2.1 Test procedure

First, 20 stocks will be selected for the simulation. In each stock, we will choose SMA 30days, EMA 30days and RSI 14days as the original indicator combination to run the in-depth analysis indicator recommendation feature, and the feature will then recommend the best indicators combination based on the accuracy. To ensure the recommended indicators is indeed better than the original indicators, we will use the recommended indicators combination as the input feature to train a model, at the same time, we would also use the original indicator combination to train another model. Model trained with the recommended indicator combination is expected to achieve higher accuracy than the model that trained with the original indicator combination. A total of 3 rounds of simulation will be performed to measure the average accuracy of both models for each stock. The similar simulations will be carried out for the remaining 19 stocks. After we have obtained the testing results for all 20 stocks, we will visualize it in a graph form for easy comparison. The figure 5.2.1.1 shows the simulation on prediction test procedure for the proposed model using original indicators combination and recommended indicators combination

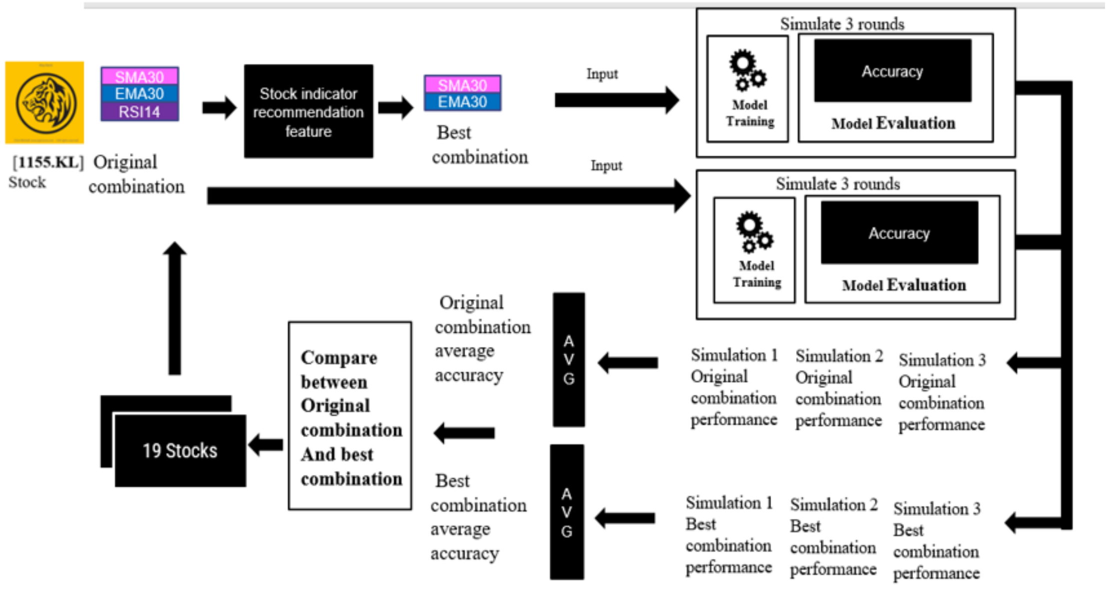



Figure 5.2.1.1 The Simulation on Prediction Test Procedure for the Proposed Model Using Original Indicators Combination and Recommended Indicators Combination.

### 5.2.2 Average accuracy rate results for the proposed model using the recommended indicators and original indicators

The table 5.2.2.1 shows the average accuracy of the model that trained with the recommended indicators combination and the model that trained with the original indicator combination for all 20 stocks. According to the data shown in the table, when the recommended indicators combination is being used as the model inputs, the proposed models achieved higher average accuracy for 18 stocks out of 20 stocks (highlighted in green). This is clear evidence to prove that the indicators combination recommended by the in-depth analysis feature is indeed reliable and better than the original indicator combination in terms of accuracy rate. To better visualize the overall results, a comparison line graph will be plotted. The figure 5.2.2.1 shows the average accuracy comparisons between model that trained with the recommended indicators combination (orange) and the model that trained with the original indicators combination (blue) for all 20 stocks. Based on the graph, the line graph in orange color is above the line graph in blue color which indicate that the model that trained with recommended indicator combinations have higher accuracy in overall.

Table 5.2.2.1 Average Accuracy of the Model that Trained with the Recommended Indicators Combination and the Model that Trained with the Original Indicators Combination for All 20 Stocks

| Stock code (.KL) | Best combinations | Best combination Average accuracy (%) | Original combination **[ SMA30, EMA30, RSI14]** Average accuracy (%) |
|---|---|---|---|
| 6888 | ['EMA30' 'RSI14'] | 54 | 52 |
| 1023 | ['SMA30' 'EMA30'] | 58 | 52 |
| 7277 | ['EMA30'] | 58 | 56 |
| 6947 | ['EMA30'] | 57 | 55 |
| 5168 | ['EMA30' 'RSI14'] | 51 | 50 |
| 5819 | ['SMA30' 'RSI14'] | 57 | 56 |
| 1082 | ['SMA30'] | 59 | 58 |
| 1961 | ['SMA30'] | 55 | 55 |
| 1155 | ['EMA30'] | 57 | 56 |
| 6012 | ['SMA30' 'EMA30'] | 54 | 53 |
| 3816 | ['EMA30' 'RSI14'] | 56 | 56 |
| 5296 | ['SMA30' 'RSI14'] | 53 | 52 |
| 4707 | ['SMA30' 'RSI14'] | 53 | 52 |
| 5183 | ['RSI14'] | 54 | 51 |
| 1066 | ['SMA30' 'EMA30'] | 53 | 51 |
| 6033 | ['SMA30' 'RSI14'] | 56 | 55 |
| 8869 | ['SMA30' 'EMA30'] | 51 | 50 |
| 1295 | ['SMA30'] | 59 | 56 |
| 4197 | ['RSI14'] | 56 | 50 |
| 5285 | ['RSI14'] | 55 | 51 |
| 5347 | ['RSI14'] | 57 | 55 |

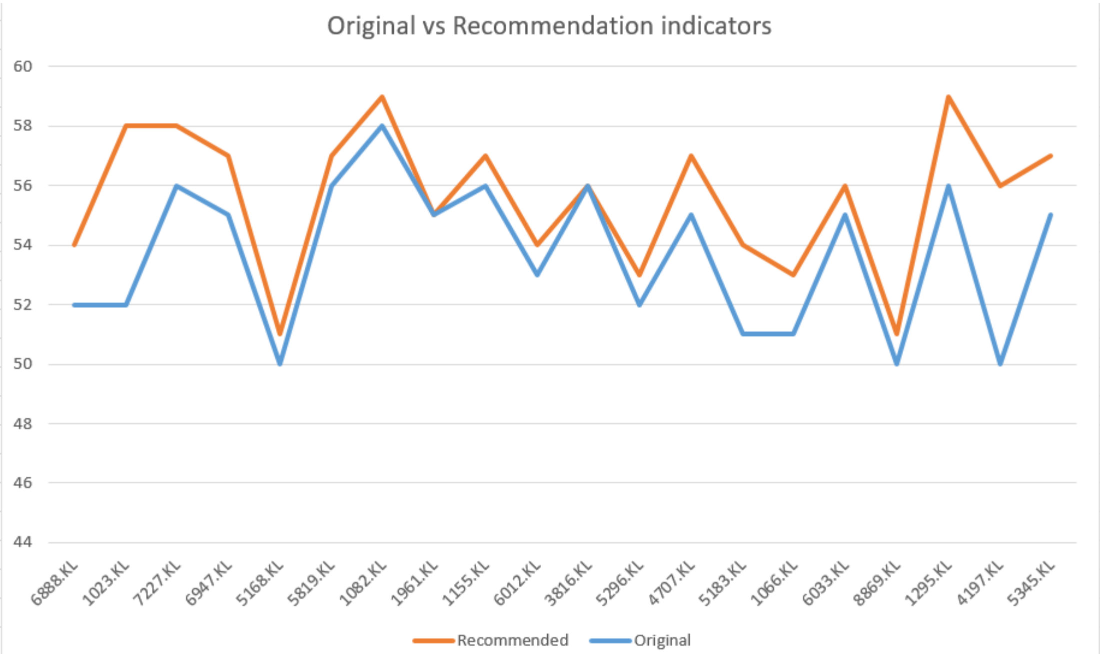



Figure 5.2.2.1 Average Accuracy of The Model that Trained with the Recommended Indicators Combination and the Model that Trained with the Original Indicators Combination for All 20 Stocks

### 5.3 Online stock prediction model building with dynamic inputs testing

The last objective of the system is to improve the flexibility of the stock prediction model in the system. This objective is achieved through 2 functions provided in the system. The first function is to provide dynamic stock prediction models that can adapt to any kinds of indicators that customized by the users. This function gives users the flexibility of customizing the input indicators of the prediction models. To ensure the first function is working as expected without errors, a simulation will be performed, and this kind of simulation is formally known as functionality testing.

#### 5.3.1 Test procedure

We will simulate a scenario to train a prediction model in real-time using customized indicators. Firstly, a model template will be created using the customized stock indicators. Next, we will use the model template to train a stock prediction model in real-time. After the model has been trained, we would verify if the input indicators used by the trained model are the same as those indicators that defined in the selected model template. If they are the same, then the function is working correctly without any errors.

#### 5.3.2 Test results

The figure 5.3.2.1 shows the simulation to train a stock prediction model using customized indicators. The “verification_test” template was used to train the prediction model in real-time, and it was made up of SMA 30days and EMA 30days indicators. Based on the “ind” column in the database record, the indicators used by the prediction model are the same as the indicators defined in the “verification_test” template. This prove that the feature is able to build stock prediction model using the dynamic inputs customized by the users. Therefore, we can conclude that the feature is working correctly as expected, and it indeed provide the flexibility to users to customize the indicators input of the stock prediction model.

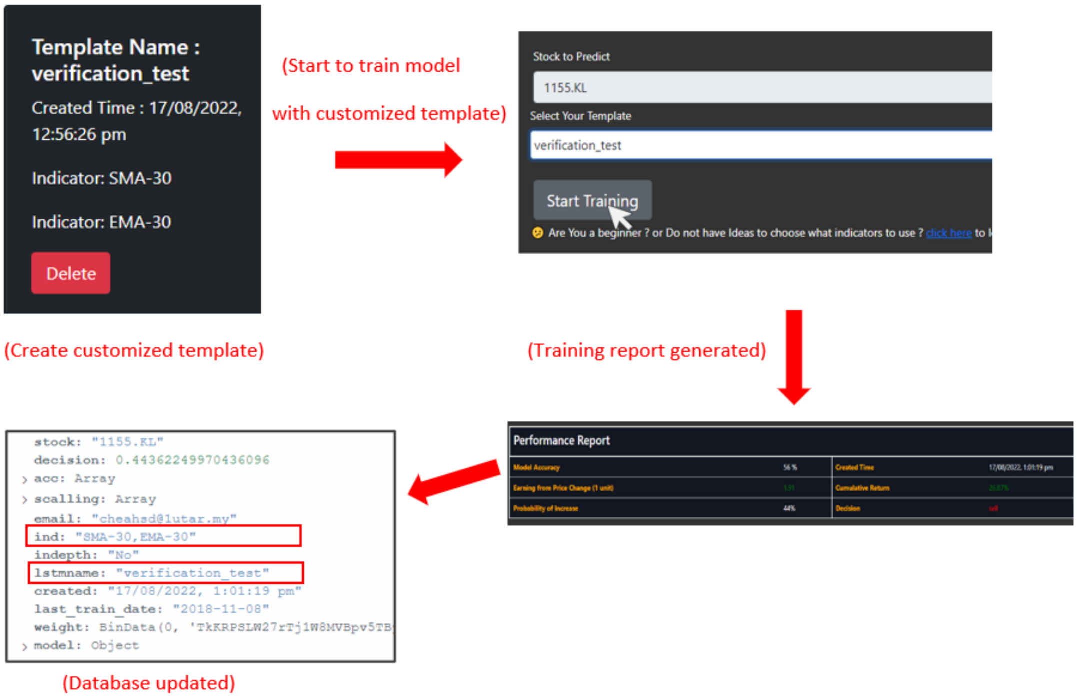



Figure 5.3.2.1 A Simulation to Train Stock Prediction Model with Customized Indicators

## 5.4 Online stock prediction with auto update capability testing

The second function is to perform auto update on the previously trained online prediction models. This function gives the flexibility to automatically update the models to the latest timesteps without programmers' intervention. To ensure the auto update function is working correctly, a simulation will be run.

### 5.4.1 Test procedure

A simulation to perform model update on the system will be carried out. Firstly, we would train an online prediction model for a particular stock, and after 3 days we would perform auto update process on this model. To verify if the model has been updated to the latest timestep, we would check if the data stored in the database has been updated.

### 5.4.2 Test results

The figure 5.4.2.1 shows the complete process of testing the auto update function on the “demo” prediction model. Based on the “last train date” column in database shown in the figure, the date was updated from “2018-11-05” to “2018-11-08” which indicate that the model has been updated with the latest 3 timesteps data. The “last train data” represent the date of the last data included in the model training data. Therefore, we can conclude that the auto updating feature is working correctly as expected.

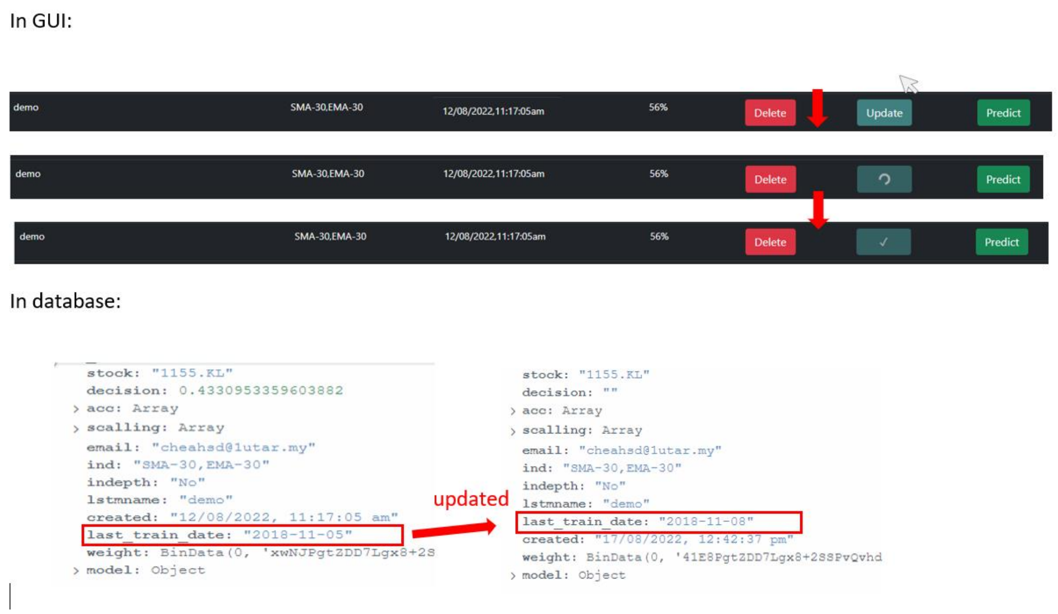



Figure 5.4.2.1 A Simulation to Perform Auto Update on the “Demo” Prediction Model

### 5.5 Time taken to train multiple watchlist stocks and to generate indicator combination recommendation with/without parallel processing

Next, parallel processing is an essential feature that must be provided in this system as this system is considered as the financial-related application, thus “time” is the most important factor. For instance, if a user were to train a stock prediction model for 10 different stocks, and the time required to complete the whole process is around 30 minutes, then users definitely will lose interest in this system as the time taken is too long. To avoid this situation happens, parallel processing will be applied to improve the system. Therefore, the following verification process is mainly to verify has the system been improved in terms of waiting time after the parallel processing feature is applied in the system

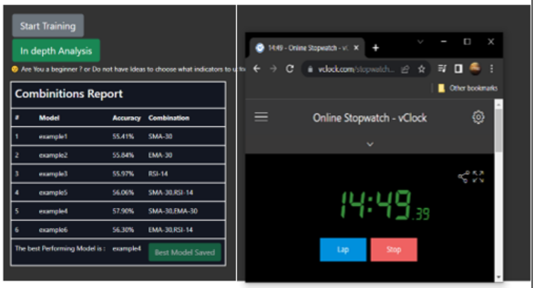
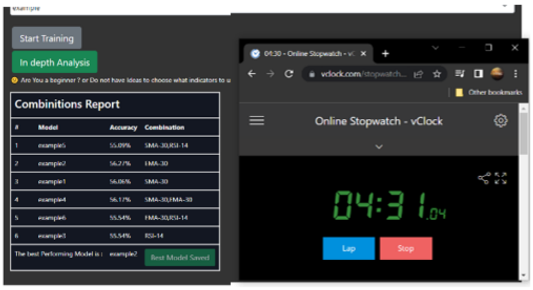
### 5.5.1 Test procedure

There are 2 features provided in the system that can be integrated with parallel processing capabilities which are the in-depth analysis indicator recommendation feature and the watchlist stocks training and prediction feature. In in-depth analysis indicator recommendation feature, the system can run the model training for all possible indicator combinations parallelly. In the watchlist stocks training and prediction feature, the system can train stock prediction models for all stocks in watchlist parallelly. The time taken with parallel processing and without parallel processing applied for these 2 features will be recorded.

### 5.5.2 Test results

The figure 5.5.2.1 shows the time taken results with/without parallel processing for both features. Based on the results, with the parallel processing applied, the time taken for both features have been reduced significantly. In the in-depth analysis recommendation feature, the time needed to run the model training process for all 6 indicators combinations is around 14 minutes; however, with parallel processing applied, the time has been reduced to around 4 minutes. In watchlist stocks training and prediction feature, the time needed to train 5 stocks is around 11 minutes; however, with parallel processing applied, the time has been reduced to around 3 minutes.

| |
|---|
| **Name:** In-depth analysis recommendation features<br>**Description:** train prediction model for all 6 combinations |

| **Without Parallel Processing** | **With Parallel Processing** |
|---|---|
|  |  |
| **Time Taken:** 14 mins 49 secs | **Time Taken:** 4 mins 31 secs |
| **Time difference** : | 10 mins 18 secs |

| |
|---|
| **Name:** Watchlist stocks training and prediction<br>**Description:** train a model for each stock in watchlist in the system, since there are 5 stocks in watchlist 5 models will be trained in the system |

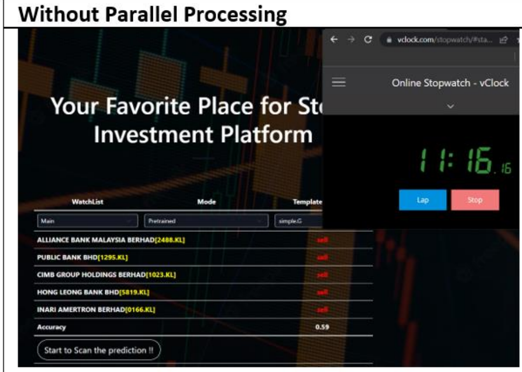
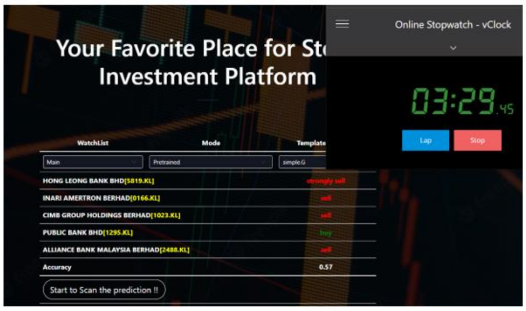
| **Without Parallel Processing** | **With Parallel Processing** |
|---|---|
|  |  |
| **Time Taken:** 11 mins 16 secs | **Time Taken:** 3 mins 29 secs |
| **Time difference** : | 8 mins 18 secs |

Figure 5.5.2.1 Time Taken Results with/without Parallel Processing for In-Depth Analysis Indicator Recommendation Feature and the Watchlist Stocks Training and Prediction Feature

# Chapter 6 Discussion

## 6.1 Project Challenges

There were some challenges faced in building the complete system. Firstly, the challenge to translate the entire machine learning code from Python to JavaScript. As mentioned in the chapter 3.1 that the process of designing the optimal architecture of LSTM model will be carried out in Python Jupyter notebook. After the optimal architecture of the LSTM model has been designed, all machine learning codes involved in constructing the model with optimal architecture design in Python will be translated and migrated to the JavaScript backend as the web application will be running in Node JS (JavaScript) platform. The machine learning codes involve many phases such as data collection, feature expansion, data labelling, data splitting and etc. These processes can be easily achieved in Python platform as Python is considered as the advanced and well-developed machine learning platform, thus there are many convenient libraries provided to ease programmers in preprocessing the input data. However, machine learning in JavaScript platform is considered as a new and recent technology, there are no such convenient libraries provided in JavaScript. Thus, all the data pre-processing operations have to be coded manually from scratch using JavaScript arrays and loops which is challenging. The figure 6.1.1 shows the difference between the Python code and JavaScript code in Min-Max normalization operation. Python able to perform the Min-Max normalization with only 3 lines of code, and it can be done perfectly without any errors. However, to achieve the similar function in JavaScript, all computations and transformation have to be coded from scratch using arrays and loops. To ensure the logic of the JavaScript code is written correctly, the output obtained from the python libraries and the output obtained from the manual JavaScript code must be the same. If the output between them is different, then the logic of the JavaScript code might be incorrect. Therefore, to ensure the entire machine learning codes are translated and migrated correctly, we must ensure the input/output for all phases must be the same in both platforms.

Python:

```python
min_max = MinMaxScaler(feature_range=(0, 1))
X_train_scaled = min_max.fit_transform(X_train)
X_test_scaled = min_max.transform(X_test)
```

JavaScript:

```javascript
var MinMaxScaler = (inputArr, callback) => {
    var outputArr = []
    let dataMax = (Math.max(...inputArr))
    let dataMin = (Math.min(...inputArr))

    inputArr.forEach(function(data, index) {
        var temp = (data - (dataMin)) / (dataMax - (dataMin));
        outputArr.push(temp);
    })

    callback(outputArr, dataMax, dataMin);
}
```

```javascript
for (let i = 0; i < x_train_new.length; i++) {
    MinMaxScaler(x_train_new[i], (resuls, max, min) => {

        var tmp = [min, max]
        min_max_scaled.push(tmp)
        x_train_new_scaled.push(resuls)

    })

}
```

Figure 6.1.1 Difference Between the Python Code and JavaScript Code in Min-Max Normalization Operation

The second challenge is to design the architecture of the LSTM model that perform well in most of the provided stocks. It is challenging to design an optimal architecture of LSTM model that can perform consistently well for all 100 stocks as different stocks will have different states. For example, architecture that only perform well in the bullish stocks, but not well in bearish stocks. However, what we aimed to find is the architecture that can generalize well in all stocks regardless of its state. Therefore, many efforts and time was spent to learn the optimal architecture design for the LSTM model that is well-suited for all 100 stocks. In this project, many experiments had been carried out before the optimal architecture design for 100 stocks was found.

## 6.2 Objective Evaluation

1. To develop a dynamic prediction model that allow users to customize the input and allow users to update the prediction model automatically when new data is reached.
2. To develop a LSTM stock prediction model design that can achieve accuracy of 50% and above as well as guarantee positive return in 60% of the provided stocks.
3. To recommend the best combination of stock indicators to the investors based on the accuracy of the models.

The first objective is achieved successfully in the system. The system is able to train a model in real-time based on the dynamic input customized by the users. Besides that, the system also able to update the previously trained model automatically to the latest timestep after user pressing the "update" button in the system. These functions were achieved perfectly in the system, and it can be proven by the functionality testing in the Chapter 5.

The second objective also achieved successfully in the system. The proposed prediction model achieved at least 50% accuracy for all 100 stocks, and the average accuracy achieved in 100 stocks was around 57%, which is the highest compared to other trading strategies like Buy-and-Hold, optimistic, and LSTM model used by the article [24]. In term of trading performance, the proposed model able to generate positive cumulative return for 90 stocks out of 100 stocks, and the average cumulative return achieved in 100 stocks is around 28%, which is the highest as compared to other trading strategies. Besides that, the proposed model also the least risky strategy. In short, the system has achieved the second objective perfectly with much better results.

The last objective also achieved successfully in the system as the system is able to recommend the best indicators combination based on the original indicators defined in the selected template, and the accuracy rate of the recommended indictors combination will usually be higher than the original indicators combination. The verification test mentioned in Chapter 5.2 has proved that the indicator combination recommended by the system perform better than the original combination of indicators in overall. Therefore, the in-depth analysis feature provided by the system is indeed reliable.

Finally, one extra improvement that has been implemented in the system is the parallel processing capabilities. The system is able to train multiple stock prediction models parallelly. Initially, the time required to train a stock prediction model is around 3 minutes, training 5 stock prediction models will require 15 minutes; however, with the parallel processing applied, the time required for training 5 stock prediction models is almost similar as the time required for training 1 stock prediction model, which is around 3 minutes. This is considered as a significant improvement, and with parallel processing, this system will become more commercialized.

### 6.3 Limitation of the System

Firstly, there are limited number of stocks provided in the system. The system only provides 100 stocks that listed in the Malaysia stock exchange (KLSE). Some popular stocks listed on foreign stock exchange centers such as Apple stock and Tesla stock are not provided in this system. Secondly, there are limited number of indicators and limited types of indicators included in the candlestick stock chart feature. The system only includes 22 types of technical indicators in the candlestick stock chart, and the system does not include any fundamental indicators in candlestick stock chart. Thiredly, only 2 kinds of data are allowed to be the input of the prediction model which are the historical stock price and the technical indicator data, the system does not consider any unstructured data such as the financial news data and top keyword searches in the browsers. Lastly, due to the hardware limitation, the parallel processing applied in the system only allows maximum of 6 processes to be run at a time.

### 6.4 Future enhancement

There are some improvements can be implemented in the future. The first improvement is to increase the number of stocks offered in the system. The number of stocks can be increased by including all the stocks listed in Kuala Lumpur stock exchange (KLSE) which is around 900 of stocks.  Further, instead of just focusing on the Malaysia stock exchange, the system can be expanded to include stock exchanges from overseas such as Hong Kong stock exchange, New York stock exchange and Tokyo stock exchange so that the target audience of the system will not only be limited to Malaysian traders but also international traders. Besides that, the system also can be expanded by providing prediction on other investment products such as gold and cryptocurrency investment so that the prediction function provided in the system is more diverse.

Other than that, the candlestick stock charting function provided in the system also can be improved. The current candlestick stock charting function is coded manually with the help of Open-source Lightweight chart Library provided by the Trading View. Lightweight chart is a free library, so some advanced functions such as drawing tools on the graph are not available. Therefore, to improve the candlestick stock charting function in the system, it is suggested to purchase for the advanced version of candlestick stock charting function API provided by the Trading View.

Moreover, the accuracy of the prediction model can be improved by involving more types of data in the model training. For example, the unstructured data such as the financial news of the companies, social feeds, and the top keywords searches in browsers can be used to improve the prediction accuracy as these kinds of data will provide human behavioral information to the model algorithm. Based on the unstructured data, the model is able to extract the investors' mood towards a particular stock and determine if it is positive or negative. If most investors have low confidence on a particular stock, then the price of the stock will probably drop in next few timesteps.

In addition, it is also recommended to integrate the system with the actual stock bidding module and payment module. This is because after model made a prediction, the users can directly bid the stock and make payment using the same system which is much more convenient. Having the actual stock bidding module and payment module integrated in the system, the system can also add a feature to automatically perform buying operations using the real money when the model predicted "Up" in the system.

Lastly, the system also can be improved in the hardware perspective. The application can be deployed into powerful high-end servers so that the time needed to train a prediction model in real-time can be reduced significantly. Other than that, the number of processes can be run at a time in parallel processing also can be increased.

# Chapter 7 Conclusion

With the advancement of machine learning algorithm, stock prediction has become more and more feasible. Many researchers have conducted research to study the usefulness of various machine learning techniques in stock prediction. LSTM algorithm seems to be the best performing algorithm as compared to traditional machine learning algorithms such as SVM and decision tree as LSTM can capture the time-series information present in the data. However, lack of flexibility in the model prediction is the major problem found in the most previous works. Since most experienced investors have formulated their own trading strategies, they will roughly know which technical indicators have some relationship with the movement of stock price and which technical indicators don't. Therefore, it is better to combine the domain knowledge of the experienced investors with the power of machine learning algorithm to form a more meaningful prediction rather than supplying random technical indicators to the prediction model. By doing so, the prediction model that trained with the investors' domain knowledge can be served as an extra decision-making factor for the investors in the stock trading.

Therefore, this project is aimed to provide a more dynamic prediction model to the investors. The system will provide the flexibility of customizing the input of the prediction model to the users before building the model in real-time. After the real-time model has been trained with the customized inputs, trading simulation will be provided to the users so that the users can evaluate the usability of model from the trading point of view. The trading simulation will generate 2 types of report which are simple interactive report and advanced portfolio report. Advanced portfolio report is mainly designed for users who are portfolio managers, and it will compare the performance of the prediction model using some advanced portfolio metrices that are well-known in the industry. Other than that, in-depth analysis feature to study the selected indicators in detail manner will also be provided to the users. This feature will tell users how each different combination of indicators perform and suggest the best combination of indicators that obtained the highest accuracy. Therefore, by this feature, the users would know what indicators are useful in prediction and what indicators are not. Besides that, with the parallel processing applied, users are also able to train multiple prediction models parallelly in real-time. With this feature, users can obtain the prediction of their favourite stocks simultaneously within short period of time. To further improve the flexibility of the stock prediction model, the system allows users to perform auto update on the pretrained customized model without programmers' intervention.

The novelty of the proposed system is providing a more flexible and customizable stock prediction model to the investors. This dynamic model will allow investors to choose their preferred indicators as the input parameters. This proposed system also allowed investors to update the model automatically when the new data is reached. This is completely different from what have been done in the previous works where they do not allow users to customize the input of model, and they will perform model update manually in the offline environment, which is cumbersome and not effective in practice. Since the stock prices will be changed frequently in real-time, the model training process and model update process should be done automatically in the application, instead of having programmers to perform these processes manually in the offline environment. Further, with the flexibility of letting users to customize input parameters of the prediction models, we can integrate the domain knowledge of the financial experts with the LSTM algorithms to form more meaningful prediction instead of just submitting random indicators to the LSTM algorithms like the previous works.

Moreover, one of the novelties applied in the system is the parallel processing approach to train multiple stock prediction models simultaneously. After applying parallel processing feature in the system, the waiting time to train multiple stock prediction models in the system has been reduced significantly. Therefore, time is no longer a big concern in the system, and users are able to train multiple stock prediction models within short amount of time.

Further, the architecture of the proposed prediction model is the combination of the LSTM layer and fully connected layers. LSTM layer is mainly to capture the time-series relationship present in the data, whereas the fully connected layer is mainly for the classification learning purposes. The proposed design of the model is able to achieve accuracy at least 50% for all 100 stocks and positive cumulative for 90% of the stocks.

# REFERENCES


[1] E. F. Fama, “Efficient Capital Markets: A review of theory and empirical work,” in *The Journal of Finance*, May. 1970. [Online]. Available: https://doi.org/10.2307/2325486

[2] A. W. Lo, “The adaptive markets hypothesis,” in *The Journal of Portfolio Management*, Oct. 2004. [Online]. Available: https://jpm.pm-research.com/content/30/5/15

[3] L. Li, Y. Wu, Y. Ou, Q. Li, Y. Zhou, and D. Chen, “Research on machine learning algorithms and feature extraction for Time Series,” in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8292668

[4] A. Moghar and M. Hamiche, “Stock market prediction using LSTM recurrent neural network,” in *Procedia Computer Science*, April. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920304865?via%3Dihub

[5] A. Hayes, “What is a growth stock?” *Investopedia*. https://www.investopedia.com/terms/g/growthstock.asp. (accessed Aug. 31, 2022).

[6] J. Chen, “Understanding the role of an institutional investor,” *Investopedia*. https://www.investopedia.com/terms/i/institutionalinvestor.asp. (accessed Aug. 31, 2022).

[7] R. Dhir, “Is the stock market efficient?,” *Investopedia*. https://www.investopedia.com/articles/basics/04/022004.asp. (accessed Aug. 31, 2022).

[8] G. S. Hiremath and J. Kumari, “Stock returns predictability and the adaptive market hypothesis in emerging markets: Evidence from India,” in *SpringerPlus*, Aug. 2014. [Online]. Available: https://springerplus.springeropen.com/articles/10.1186/2193-1801-3-428

[9] J. H. Kim, A. Shamsuddin, and K.-P. Lim, “Stock return predictability and the Adaptive Markets Hypothesis: Evidence from Century-long U.S. data,” in *Journal of Empirical Finance*, Dec. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0927539811000612

[10] I. K. Nti, A. F. Adekoya, and B. A. Weyori, “A systematic review of fundamental and technical analysis of stock market predictions,” in *Artificial Intelligence Review*, Aug. 2019. [Online]. Available: https://link.springer.com/article/10.1007/s10462-019-09754-z

[11] T. Turner, *A beginner's Guide to Day Trading Online*. Avon, Mass: Adams Media, 2007.

[12] K. Rogers, “Types of trading indicators,” *Finance*, 21-Nov-2017. https://finance.zacks.com/types-trading-indicators-9956.html. (accessed Aug. 31, 2022).

[13] T. Anbalagan and S. U. Maheswari, “Classification and prediction of stock market index based on Fuzzy Metagraph,” in *Procedia Computer Science*, March. 2015. [Online]. Available: https://www.researchgate.net/publication/277917587_Classification_and_Prediction_of_Stock_Market_Index_Based_on_Fuzzy_Metagraph

[14] Investopedia, “Technical analysis basic education,” *Investopedia*. https://www.investopedia.com/technical-analysis-basic-education-4689655. (accessed Aug. 31, 2022).

[15] G. Weckman and S. Lakshminarayanan, “Identifying Technical Indicators for Stock Market Prediction With Neural Networks,” presented at the *Proceedings of the IIE Annual Conference*, Houston, Texas, 2004.

[16] J. J. Murphy, *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Paramus, NJ: New York Institute of Finance, 1999.

[17] Z. Haider Khan, T. Sharmin Alin, and A. Hussain, "Price prediction of share market using Artificial Neural Network 'ann'," in *International Journal of Computer Applications*, May. 2011. [Online]. Available: https://www.researchgate.net/publication/226159428_Price_Prediction_of_Share_Market_Using_Artificial_Neural_Network_'ANN'

[18] A. Henriksson, J. Zhao, H. Dalianis, and H. Boström, "Ensembles of randomized trees using diverse distributed representations of clinical events," in *BMC Medical Informatics and Decision Making*, July. 2016. [Online]. Available: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-016-0309-0

[19] O. B. Sezer, M. Ozbayoglu, and E. Dogdu, "A deep neural-network based stock trading system based on evolutionary optimized technical analysis parameters," in *Procedia Computer Science*, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050917318252?via%3Dihub

[20] R. Naik et al., "Prediction of Stock Market Index Using Genetic Algorithm",in *Computer Engineering and Intelligent System*, 2012. [Online]. Available: https://core.ac.uk/download/pdf/234644513.pdf

[21] Y. Shynkevich, T. M. McGinnity, S. A. Coleman, A. Belatreche, and Y. Li, "Forecasting price movements using technical indicators: Investigating the impact of varying input window length," in *Neurocomputing*, Nov. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0925231217311074

[22] X. Zhong and D. Enke, "A comprehensive cluster and classification mining procedure for Daily Stock Market Return Forecasting," in *Neurocomputing*, Dec. 2017. [Online]. Available:

https://www.sciencedirect.com/science/article/abs/pii/S0925231217310652?via%3Dihub

[23] A. Picasso, S. Merello, Y. Ma, L. Oneto, and E. Cambria, "Technical analysis and sentiment embeddings for market trend prediction," in *Expert Systems with Applications*, Nov. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417419304142

[24] M. Agrawal, A.U. Khan, and P.K Shukla, "Stock Price Prediction using Technical Indicators: A Predictive Model using Optimal Deep Learning," in *International Journal of Recent Technology and Engineering*, July. 2019. [Online]. Available: https://www.ijrte.org/wp-content/uploads/papers/v8i2/B3048078219.pdf

[25] P. Gao, R. Zhang, and X. Yang, "The application of stock index price prediction with neural network," in *Mathematical and Computational Applications*, July. 2020. [Online]. Available: https://www.mdpi.com/2297-8747/25/3/53

[26] J. Shen and M. O. Shafiq, "Short-term stock market price trend prediction using a comprehensive deep learning system," in *Journal of Big Data*, Aug. 2020. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6

[27] "Make stock investment easy," *JStock*. https://jstock.org/. (accessed Aug. 31, 2022).

[28] "Track all markets," *TradingView*. https://www.tradingview.com/. (accessed Aug. 31, 2022).

[29] "Screener," *KLSE Screener*. https://www.klsescreener.com/v2/ (accessed Aug. 31, 2022].

[30] "Stock market quotes & financial news," *Investing.com*. https://www.investing.com/ (accessed Aug. 31, 2022).

[31] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, “Stock market's price movement prediction with LSTM neural networks,” presented at the *International Joint Conference on Neural Networks (IJCNN)*, 2017.

[32] A. Hayes, “What is the risk-free rate of return?,” *Investopedia* https://www.investopedia.com/terms/r/risk-freerate.asp (accessed Aug. 31, 2022).

[33] J. Fernando, “Compound annual growth rate (CAGR) formula and calculation,” *Investopedia*. https://www.investopedia.com/terms/c/cagr.asp (accessed Aug. 31, 2022).

# APPENDIX

## A.1 100 stocks listing

| | Code (. KL) | Company |
|---|---|---|
| **0** | 6888 | AXIATA GROUP BERHAD |
| **1** | 1023 | CIMB GROUP HOLDINGS BERHAD |
| **2** | 7277 | DIALOG GROUP BHD |
| **3** | 6947 | DIGI.COM BHD |
| **4** | 3182 | GENTING BHD |
| **5** | 4715 | GENTING MALAYSIA BERHAD |
| **6** | 3034 | HAP SENG CONSOLIDATED BHD |
| **7** | 5168 | HARTALEGA HOLDINGS BHD |
| **8** | 5819 | HONG LEONG BANK BHD |
| **9** | 1082 | HONG LEONG FINANCIAL GROUP BHD |
| **10** | 5225 | IHH HEALTHCARE BERHAD |
| **11** | 1961 | IOI CORPORATION BHD |
| **12** | 2445 | KUALA LUMPUR KEPONG BHD |
| **13** | 1155 | MALAYAN BANKING BHD |
| **14** | 6012 | MAXIS BERHAD |
| **15** | 3816 | MISC BHD |
| **16** | 5296 | MR D.I.Y. GROUP(M) BERHAD |
| **17** | 4707 | NESTLE(M) BHD |
| **18** | 5183 | PETRONAS CHEMICALS GROUP BHD |
| **19** | 5681 | PETRONAS DAGANGAN BHD |
| **20** | 6033 | PETRONAS GAS BHD |
| **21** | 4065 | PPB GROUP BHD |
| **22** | 8869 | PRESS METAL ALUMINIUM HOLDINGS BERHAD |
| **23** | 1295 | PUBLIC BANK BHD |
| **24** | 1066 | RHB BANK BERHAD |
| **25** | 4197 | SIME DARBY BHD |
| **26** | 5285 | SIME DARBY PLANTATION BERHAD |
| **27** | 4863 | TELEKOM MALAYSIA BHD |
| **28** | 5347 | TENAGA NASIONAL BHD |
| **29** | 7113 | TOP GLOVE CORPORATION BHD |
| **30** | 5139 | AEON CREDIT SERVICE(M) BHD |
| **31** | 5099 | AIRASIA GROUP BERHAD |
| **32** | 2488 | ALLIANCE BANK MALAYSIA BERHAD |
| **33** | 1015 | AMMB HOLDINGS BHD |
| **34** | 6399 | ASTRO MALAYSIA HOLDINGS BERHAD |
| **35** | 8176 | ATA IMS BERHAD |
| **36** | 5106 | AXIS REITS |
| **37** | 1562 | BERJAYA SPORTS TOTO BHD |
| **38** | 5248 | BERMAZ AUTO BERHAD |

| | | |
|---|---|---|
| **39** | 4162 | BRITISH AMERICAN TOBACCO(M) |
| **40** | 5210 | BUMI ARMADA BERHAD |
| **41** | 1818 | BURSA MALAYSIA BHD |
| **42** | 2852 | CAHYA MATA SARAWAK BHD |
| **43** | 2836 | CARLSBERG BREWERY MALAYSIA BHD |
| **44** | 7204 | D & O GREEN TECHNOLOGIES BERHAD |
| **45** | 1619 | DRB - HICOM BHD |
| **46** | 7148 | DUOPHARMA BIOTECH BERHAD |
| **47** | 5222 | FGV HOLDINGS BERHAD |
| **48** | 3689 | FRASER & NEAVE HOLDINGS BHD |
| **49** | 0128 | FRONTKEN CORPORATION BHD |
| **50** | 5398 | GAMUDA BHD |
| **51** | 0078 | GDEX BERHAD |
| **52** | 2291 | GENTING PLANTATIONS BERHAD |
| **53** | 0208 | GREATECH TECHNOLOGY BERHAD |
| **54** | 5102 | GUAN CHONG BHD |
| **55** | 3255 | HEINEKEN MALAYSIA BERHAD |
| **56** | 3301 | HONG LEONG INDUSTRIES BHD |
| **57** | 5227 | IGB REAL ESTATE INV TRUST |
| **58** | 3336 | IJM CORPORATION BHD |
| **59** | 0166 | INARI AMERTRON BERHAD |
| **60** | 7153 | KOSSAN RUBBER INDUSTRIES BHD |
| **61** | 5878 | KPJ HEALTHCARE BHD |
| **62** | 6633 | LEONG HUP INTERNATIONAL BERHAD |
| **63** | 5284 | LOTTE CHEMICAL TITAN HOLDING BERHAD |
| **64** | 3859 | MAGNUM BERHAD |
| **65** | 5264 | MALAKOFF CORPORATION BERHAD |
| **66** | 5014 | MALAYSIA AIRPORTS HOLDINGS BHD |
| **67** | 1171 | MALAYSIA BUILDING SOCIETY BHD |
| **68** | 3867 | MALAYSIAN PACIFIC INDUSTRIES |
| **69** | 1651 | MALAYSIAN RESOURCES CORPORATION BERHAD |
| **70** | 5236 | MATRIX CONCEPTS HOLDINGS BHD |
| **71** | 3069 | MEGA FIRST CORPORATION BHD |
| **72** | 5286 | MI TECHNOVATION BERHAD |
| **73** | 9385 | LAY HONG BERHAD |
| **74** | 0138 | MY E.G. SERVICES BHD |
| **75** | 5258 | Bank Islam Malaysia |
| **76** | 7160 | PENTAMASTER CORPORATION BHD |
| **77** | 7084 | QL RESOURCES BHD |
| **78** | 5218 | SAPURA ENERGY BERHAD |
| **79** | 4731 | SCIENTEX BERHAD |
| **80** | 5279 | SERBA DINAMIK HOLDINGS BERHAD |
| **81** | 5288 | SIME DARBY PROPERTY BERHAD |
| **82** | 7155 | SKP RESOURCES BHD |
| **83** | 8664 | SP SETIA BHD |

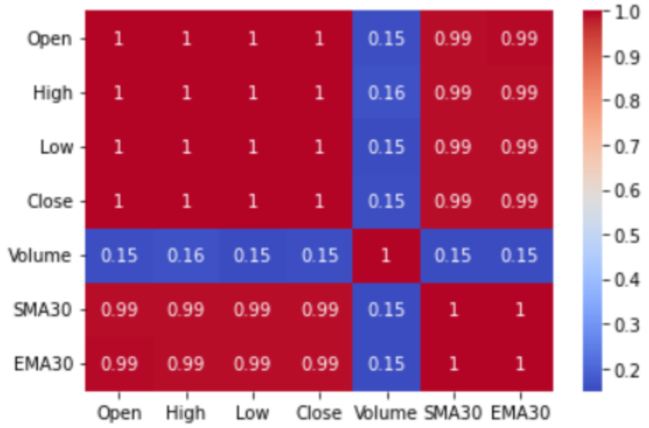
| | | |
|---|---|---|
| **84** | 5211 | SUNWAY BERHAD |
| **85** | 5176 | SUNWAY REAL ESTATE INVT TRUST |
| **86** | 7106 | SUPERMAX CORPORATION BHD |
| **87** | 6139 | SYARIKAT TAKAFUL MALAYSIA KELUARGA BERHAD |
| **88** | 5031 | TIME DOTCOM BHD |
| **89** | 5148 | UEM SUNRISE BERHAD |
| **90** | 4588 | UMW HOLDINGS BHD |
| **91** | 5005 | UNISEM(M) BHD |
| **92** | 5292 | UWC BERHAD |
| **93** | 6963 | V.S INDUSTRY BHD |
| **94** | 0097 | VITROX CORPORATION BHD |
| **95** | 5246 | WESTPORTS HOLDINGS BERHAD |
| **96** | 7293 | YINSON HOLDINGS BHD |
| **97** | 4677 | YTL CORPORATION BHD |
| **98** | 5109 | YTL HOSPITALITY REIT |
| **99** | 6742 | YTL POWER INTERNATIONAL BHD |

## A.2 LSTM model development related figures

# Correlation Study on the data

In [5]:
```
seaborn.heatmap(data_processed.corr(),annot=True,cmap="coolwarm")
```

Out[5]: <AxesSubplot:>



### Data Labelling

$(n)^{th}$ represent the price of current time step, $(n+1)^{th}$ represent the price of tomorrow time step. The price of stock is increased when $(n+1)^{th}$ is greater than nth, thus positive difference will be returned. When there is increase in price, label "1" will be assigned to the current time step.

| nth | n+1th | n+1th - nth |
|---|---|---|
| 10.12 | 10.1 | -0.02 |
| 10.1 | 10.1 | 0.0 |
| 10.1 | 10.14 | 0.04 |
| 10.14 | 10.14 | 0.0 |
| 10.14 | 10.18 | 0.04 |
| 10.18 | 10.2 | 0.02 |
| 10.2 | 10.26 | 0.06 |
| 10.26 | 10.4 | 0.14 |
| 10.4 | 10.46 | 0.06 |
| 10.46 | 10.46 | 0.0 |
| 10.46 | 10.44 | -0.02 |
| 10.44 | 10.46 | 0.02 |
| 10.46 | 10.42 | -0.04 |

```
np.array(data_processed['Label'])[:20].reshape(-1,1)

array([[0],
       [0],
       [1],
       [0],
       [1],
       [1],
       [1],
       [1],
       [1],
       [0],
       [0],
       [1],
       [0]
```

**Splitting the Data into Train set (green) and Test set (red)**

```
[<matplotlib.lines.Line2D at 0x20039d1d3d0>]
```

## Train set data before and after min-max transformation

Out[13]:

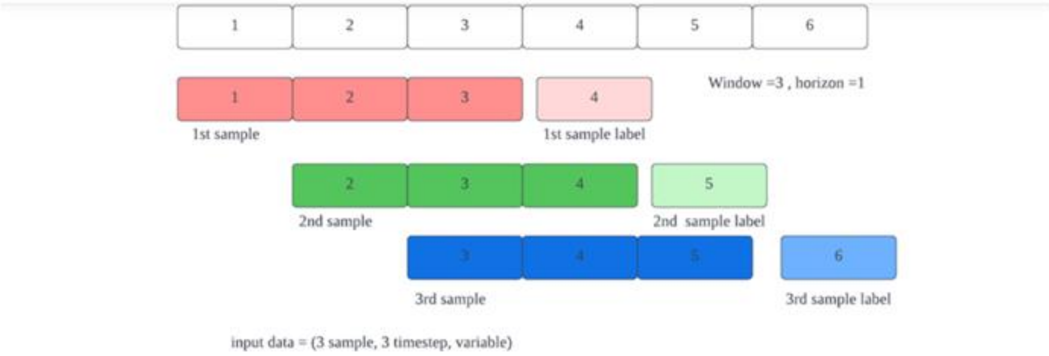
| Date | Open | High | Low | Close | Volume | OBV | EMA30 | RSI14 | One_day_change | STD |
|---|---|---|---|---|---|---|---|---|---|---|
| 2018-02-14 | 10.12 | 10.14 | 10.08 | 10.12 | 13604500 | 148834800.0 | 9.939000 | 68.404637 | 0.001980 | 0.034409 |
| 2018-02-15 | 10.10 | 10.14 | 10.10 | 10.10 | 6344400 | 142490400.0 | 9.949387 | 65.815303 | -0.001976 | 0.035777 |
| 2018-02-16 | 10.10 | 10.10 | 10.10 | 10.10 | 0 | 142490400.0 | 9.959104 | 65.815303 | 0.000000 | 0.019596 |
| 2018-02-19 | 10.12 | 10.16 | 10.08 | 10.14 | 7986200 | 150456600.0 | 9.970775 | 68.574569 | 0.003960 | 0.016000 |
| 2018-02-20 | 10.14 | 10.16 | 10.10 | 10.14 | 10243900 | 150456600.0 | 9.981693 | 68.574569 | 0.000000 | 0.017889 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2021-05-05 | 8.25 | 8.26 | 8.21 | 8.21 | 4564100 | -58357700.0 | 8.277613 | 43.796569 | -0.003641 | 0.020000 |
| 2021-05-06 | 8.22 | 8.24 | 8.16 | 8.23 | 5176100 | -53181600.0 | 8.274541 | 45.769129 | 0.002436 | 0.019596 |
| 2021-05-07 | 8.25 | 8.28 | 8.20 | 8.24 | 1872500 | -51309100.0 | 8.272313 | 46.775037 | 0.001215 | 0.019391 |
| 2021-05-10 | 8.23 | 8.26 | 8.22 | 8.24 | 3097200 | -51309100.0 | 8.270228 | 46.775037 | 0.000000 | 0.011662 |
| 2021-05-11 | 8.22 | 8.24 | 8.17 | 8.19 | 3566700 | -54875800.0 | 8.265052 | 42.232585 | -0.006068 | 0.019391 |

Transform to VVVVVV

```
]: X_train_scaled
```

```
]: array([[0.80412366, 0.76790132, 0.8082902 , ..., 0.77055368, 0.62200838,
         0.06108775],
        [0.79896915, 0.76790132, 0.81347166, ..., 0.73133235, 0.60176158,
         0.06391921],
        [0.79896915, 0.75802479, 0.81347166, ..., 0.73133235, 0.61187497,
         0.03042313],
        ...,
        [0.32216494, 0.30864188, 0.32124349, ..., 0.44292434, 0.61809321,
         0.02999898],
        [0.31701018, 0.30370373, 0.32642496, ..., 0.44292434, 0.61187497,
         0.01399928],
        [0.31443305, 0.29876534, 0.31347154, ..., 0.37411859, 0.58082205,
         0.02999866]])
```

## Min max scaling + Time Series transformation (after vs before)




| X_scaled | Y ( movement for next horizon) |
|---|---|
| $[X_1F1, X_1F2, X_1F3...... X_1Fn]$ | 1 |
| $[X_2F1, X_2F2, X_2F3...... X_2Fn]$ | 0 |
| $[X_3F1, X_3F2, X_3F3...... X_3Fn]$ | 1 |
| $[X_4F1, X_4F2, X_4F3...... X_4Fn]$ | 0 |
| $[X_5F1, X_5F2, X_5F3...... X_5Fn]$ | 1 |
| $[X_6F1, X_6F2, X_6F3...... X_6Fn]$ | 0 |
| | |

```
array([[[0.80412366, 0.76790132, 0.8082902 , 0.80412366, 0.1602779 ,
         0.6358785 , 0.79027294, 0.77055368, 0.62200838, 0.06108775],
        [0.79896915, 0.76790132, 0.81347166, 0.79896915, 0.07474491,
         0.62194548, 0.79321866, 0.73133235, 0.60176158, 0.06391921],
        [0.79896915, 0.75802479, 0.81347166, 0.79896915, 0.        ,
         0.62194548, 0.79597433, 0.73133235, 0.61187497, 0.03042313]],

       [[0.79896915, 0.76790132, 0.81347166, 0.79896915, 0.07474491,
         0.62194548, 0.79321866, 0.73133235, 0.60176158, 0.06391921],
        [0.79896915, 0.75802479, 0.81347166, 0.79896915, 0.        ,
         0.62194548, 0.79597433, 0.73133235, 0.61187497, 0.03042313],
        [0.80412366, 0.77283947, 0.8082902 , 0.80927842, 0.09385172,
         0.63944015, 0.79928407, 0.77312768, 0.63214228, 0.02297949]],

       [[0.79896915, 0.75802479, 0.81347166, 0.79896915, 0.        ,
         0.62194548, 0.79597433, 0.73133235, 0.61187497, 0.03042313],
        [0.80412366, 0.77283947, 0.8082902 , 0.80927842, 0.09385172,
         0.63944015, 0.79928407, 0.77312768, 0.63214228, 0.02297949],
        [0.80927842, 0.77283947, 0.81347166, 0.80927842, 0.12068586,
         0.63944015, 0.80238027, 0.77312768, 0.61187497, 0.02688899]]])
```

| Date | Open | High | Low | Close | Volume | OBV | EMA30 | RSI14 | One_day_change | STD |
|---|---|---|---|---|---|---|---|---|---|---|
| 2018-02-14 | 10.12 | 10.14 | 10.08 | 10.12 | 13604500 | 148834800.0 | 9.939000 | 68.404637 | 0.001980 | 0.034409 |
| 2018-02-15 | 10.10 | 10.14 | 10.10 | 10.10 | 6344400 | 142490400.0 | 9.949387 | 65.815303 | -0.001976 | 0.035777 |
| 2018-02-16 | 10.10 | 10.10 | 10.10 | 10.10 | 0 | 142490400.0 | 9.959104 | 65.815303 | 0.000000 | 0.019596 |
| 2018-02-19 | 10.12 | 10.16 | 10.08 | 10.14 | 7966200 | 150456600.0 | 9.970775 | 68.574569 | 0.003960 | 0.016000 |
| 2018-02-20 | 10.14 | 10.16 | 10.10 | 10.14 | 0243900 | 150456600.0 | 9.981693 | 68.574569 | 0.000000 | 0.017889 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2021-05-05 | 8.25 | 8.26 | 8.21 | 8.21 | 4564100 | -58357700.0 | 8.277613 | 43.796569 | -0.003641 | 0.020000 |
| 2021-05-06 | 8.22 | 8.24 | 8.16 | 8.23 | 5176100 | -53181600.0 | 8.274541 | 45.769129 | 0.002436 | 0.019596 |
| 2021-05-07 | 8.25 | 8.28 | 8.20 | 8.24 | 1872500 | -51309100.0 | 8.272313 | 46.775037 | 0.001215 | 0.019391 |
| 2021-05-10 | 8.23 | 8.26 | 8.22 | 8.24 | 3097200 | -51309100.0 | 8.270228 | 46.775037 | 0.000000 | 0.011662 |
| 2021-05-11 | 8.22 | 8.24 | 8.17 | 8.19 | 3566700 | -54875800.0 | 8.265052 | 42.232585 | -0.006068 | 0.019391 |

## Proposed model Architecture

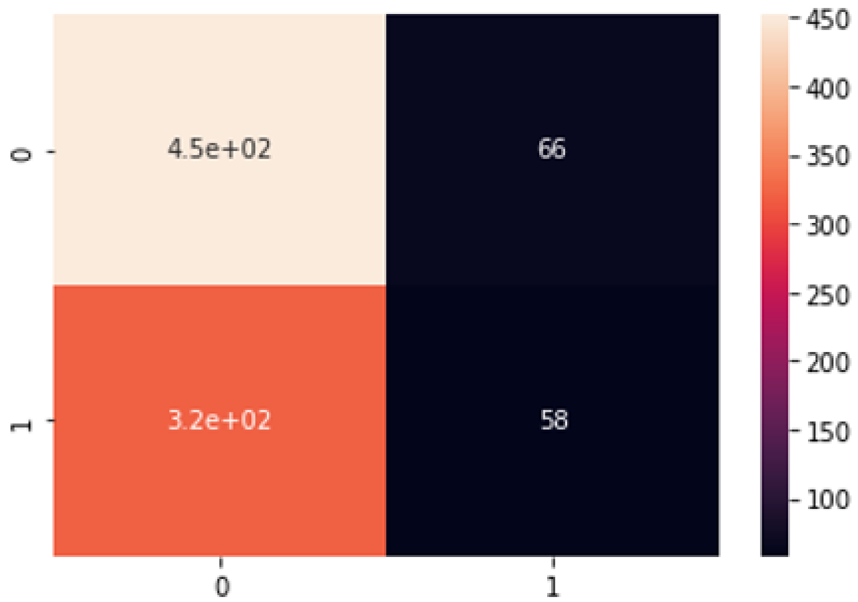
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 64)                18432

_________________________________________________________________
dropout (Dropout)            (None, 64)                0

_________________________________________________________________
dense (Dense)                (None, 32)                2080

_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 20,545
Trainable params: 20,545
Non-trainable params: 0
```

## First Stage evaluation (Machine learning evaluation )

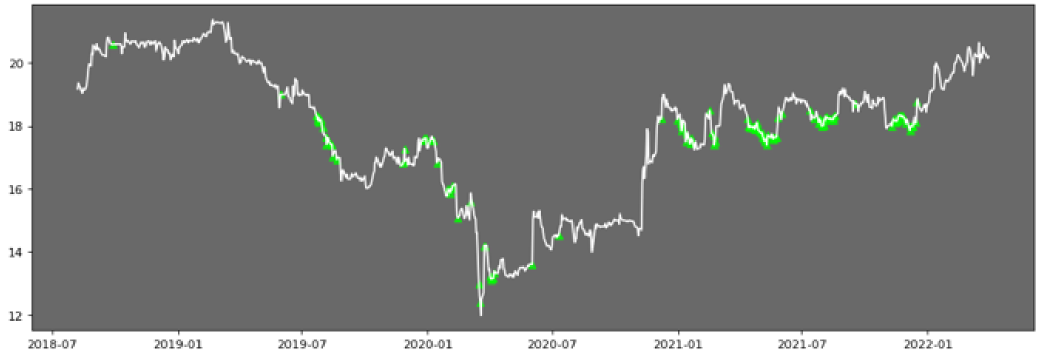



```
Precision 0.46774193548387094
f1 score 0.2310756972111537
recall 0.15343915343915343
AUC : 0.5377111805683235
```

```
Accuracy  0.5691964285714286
```

## Second stage of evaluation (Trading Evaluation)

| column 1 | column 2 |
|---|---|
| trading days | 896 |
| total action done | 124 |
| total return | 2.73999 |
| std return | 0.217223 |
| avg return per trade(action) | 0.0220967 |
| max gain | 1.42 |
| min gain | -0.440001 |
| sharp ratio | 1.61481 |
| Buy and hold startegy | 1.02 |
| Optimistic | -1.1 |



## A.3 Average Accuracy, Average Cumulative Return, Average Sharpe Ratio, and Average Max Drawdown of 100 stocks for 4 different strategies.

### A.3.1 Average Accuracy for 100 stocks

| code | Proposed_model | Article_model | Buy_and_Hold | Optimistic |
|---|---|---|---|---|
| 6888.KL | 0.534201954 | 0.506493506 | 0.433982684 | 0.464285714 |
| 1023.KL | 0.54483008 | 0.537878788 | 0.438311688 | 0.506493506 |
| 7277.KL | 0.582699964 | 0.574675325 | 0.413419913 | 0.45995671 |
| 6947.KL | 0.568946797 | 0.53030303 | 0.428571429 | 0.461038961 |
| 3182.KL | 0.523550725 | 0.512987013 | 0.423423423 | 0.520720721 |
| 4715.KL | 0.551932367 | 0.522727273 | 0.425225225 | 0.477477477 |
| 3034.KL | 0.593557727 | 0.567099567 | 0.397186147 | 0.478354978 |
| 5168.KL | 0.513210279 | 0.503246753 | 0.433982684 | 0.506493506 |
| 5819.KL | 0.558206797 | 0.50974026 | 0.42965368 | 0.483766234 |
| 1082.KL | 0.580528411 | 0.585497835 | 0.412337662 | 0.5 |
| 5225.KL | 0.553278689 | 0.565395095 | 0.455040872 | 0.482288828 |
| 1961.KL | 0.547955121 | 0.551771117 | 0.436147186 | 0.446969697 |
| 2445.KL | 0.543612016 | 0.528138528 | 0.456709957 | 0.439393939 |
| 1155.KL | 0.563517915 | 0.553030303 | 0.432900433 | 0.484848485 |
| 6012.KL | 0.543612016 | 0.518398268 | 0.426406926 | 0.451298701 |
| 3816.KL | 0.566883586 | 0.534632035 | 0.441558442 | 0.465367965 |
| 5296.KL | 0.512396694 | 0.55952381 | 0.406504065 | 0.585365854 |
| 4707.KL | 0.559562842 | 0.544715447 | 0.449180328 | 0.532786885 |
| 5183.KL | 0.537080406 | 0.549180328 | 0.453908985 | 0.470245041 |
| 5681.KL | 0.558929863 | 0.527421237 | 0.439393939 | 0.464285714 |
| 6033.KL | 0.548806941 | 0.519480519 | 0.436147186 | 0.475108225 |
| 4065.KL | 0.531814895 | 0.521645022 | 0.462121212 | 0.479437229 |
| 8869.KL | 0.501301518 | 0.53030303 | 0.483766234 | 0.484848485 |
| 1295.KL | 0.589660159 | 0.482683983 | 0.41017316 | 0.505411255 |
| 1066.KL | 0.53868402 | 0.528138528 | 0.443722944 | 0.474025974 |
| 4197.KL | 0.605676329 | 0.521645022 | 0.407207207 | 0.472072072 |
| 5285.KL | 0.505882353 | 0.488095238 | 0.425655977 | 0.413994169 |
| 4863.KL | 0.540491685 | 0.462908012 | 0.458874459 | 0.474025974 |
| 5347.KL | 0.55242227 | 0.487012987 | 0.401515152 | 0.485930736 |
| 7113.KL | 0.524584237 | 0.474025974 | 0.423160173 | 0.506493506 |
| 5139.KL | 0.587129429 | 0.568181818 | 0.418831169 | 0.510822511 |
| 5099.KL | 0.606190476 | 0.537878788 | 0.398571429 | 0.551428571 |
| 2488.KL | 0.520607375 | 0.474285714 | 0.433982684 | 0.474025974 |
| 1015.KL | 0.509057971 | 0.5 | 0.436036036 | 0.484684685 |
| 6399.KL | 0.522342995 | 0.564935065 | 0.369369369 | 0.484684685 |
| 8176.KL | 0.597900832 | 0.446304045 | 0.401950163 | 0.523293608 |
| 5106.KL | 0.625317374 | 0.575297941 | 0.369848156 | 0.47505423 |
| 1562.KL | 0.499640805 | 0.607375271 | 0.383870968 | 0.483870968 |
| 5248.KL | 0.567085954 | 0.501082251 | 0.385579937 | 0.510971787 |
| 4162.KL | 0.536231884 | 0.521943574 | 0.427027027 | 0.488288288 |
| 5210.KL | 0.602150538 | 0.438311688 | 0.379950495 | 0.507425743 |
| 1818.KL | 0.57230658 | 0.591584158 | 0.417748918 | 0.524891775 |
| 2852.KL | 0.614754098 | 0.582251082 | 0.38852459 | 0.549180328 |

| | | | | |
|---|---|---|---|---|
| 2836.KL | 0.539407086 | 0.590163934 | 0.446969697 | 0.512987013 |
| 7204.KL | 0.558097313 | 0.543290043 | 0.441304348 | 0.464130435 |
| 1619.KL | 0.57582338 | 0.55326087 | 0.416666667 | 0.467532468 |
| 7148.KL | 0.529496924 | 0.541125541 | 0.395021645 | 0.508658009 |
| 5222.KL | 0.607949413 | 0.458874459 | 0.382432432 | 0.493243243 |
| 3689.KL | 0.538183134 | 0.593243243 | 0.456121343 | 0.449620802 |
| 0128.KL | 0.512292119 | 0.486457205 | 0.479437229 | 0.470779221 |
| 5398.KL | 0.564241766 | 0.502164502 | 0.446969697 | 0.5 |
| 0078.KL | 0.673897325 | 0.554112554 | 0.293290043 | 0.541125541 |
| 2291.KL | 0.560099133 | 0.587662338 | 0.433085502 | 0.475836431 |
| 0208.KL | 0.551051051 | 0.55204461 | 0.433035714 | 0.491071429 |
| 5102.KL | 0.556641332 | 0.522321429 | 0.416666667 | 0.514069264 |
| 3255.KL | 0.554469779 | 0.579004329 | 0.454545455 | 0.488095238 |
| 3301.KL | 0.565866084 | 0.498917749 | 0.425925926 | 0.523965142 |
| 5227.KL | 0.645687646 | 0.511982571 | 0.355153203 | 0.494428969 |
| 3336.KL | 0.554831705 | 0.647632312 | 0.431818182 | 0.5 |
| 0166.KL | 0.545078577 | 0.57034632 | 0.458590853 | 0.47342398 |
| 7153.KL | 0.556037599 | 0.531520396 | 0.430735931 | 0.520562771 |
| 5878.KL | 0.644974693 | 0.535714286 | 0.354978355 | 0.510822511 |
| 6633.KL | 0.546494993 | 0.643939394 | 0.300847458 | 0.521186441 |
| 5284.KL | 0.57967033 | 0.417391304 | 0.387978142 | 0.5 |
| 3859.KL | 0.599348534 | 0.595628415 | 0.37987013 | 0.510822511 |
| 5264.KL | 0.654205607 | 0.616883117 | 0.345794393 | 0.560747664 |
| 5014.KL | 0.557003257 | 0.61682243 | 0.431818182 | 0.503246753 |
| 1171.KL | 0.496923634 | 0.564935065 | 0.348484848 | 0.515151515 |
| 3867.KL | 0.531306551 | 0.489177489 | 0.475108225 | 0.491341991 |
| 1651.KL | 0.533142585 | 0.505411255 | 0.39341917 | 0.509298999 |
| 5236.KL | 0.593641331 | 0.536480687 | 0.41307578 | 0.44576523 |
| 3069.KL | 0.603036876 | 0.615156018 | 0.397186147 | 0.544372294 |
| 5286.KL | 0.560090703 | 0.597402597 | 0.405405405 | 0.469594595 |
| 9385.KL | 0.661452514 | 0.523648649 | 0.300668151 | 0.551224944 |
| 0138.KL | 0.591106291 | 0.63363029 | 0.393939394 | 0.5 |
| 5258.KL | 0.615328995 | 0.514069264 | 0.37987013 | 0.512987013 |
| 7160.KL | 0.50253073 | 0.576839827 | 0.475108225 | 0.484848485 |
| 7084.KL | 0.578091106 | 0.507575758 | 0.412337662 | 0.471861472 |
| 5218.KL | 0.73279714 | 0.574675325 | 0.2376502 | 0.607476636 |
| 4731.KL | 0.545191612 | 0.489986649 | 0.454545455 | 0.504329004 |
| 5279.KL | 0.743459916 | 0.555194805 | 0.238693467 | 0.726130653 |
| 5288.KL | 0.574850299 | 0.701005025 | 0.367952522 | 0.504451039 |
| 7155.KL | 0.581614187 | 0.537091988 | 0.423160173 | 0.46969697 |
| 8664.KL | 0.56884058 | 0.550865801 | 0.38018018 | 0.545945946 |
| 5211.KL | 0.632457238 | 0.582251082 | 0.36329588 | 0.480649189 |
| 5176.KL | 0.644771612 | 0.631710362 | 0.35214447 | 0.534988713 |
| 7106.KL | 0.547382372 | 0.637697517 | 0.445328032 | 0.500994036 |
| 6139.KL | 0.597953216 | 0.550458716 | 0.400437637 | 0.543763676 |
| 5031.KL | 0.556399132 | 0.570021882 | 0.441558442 | 0.436147186 |
| 5148.KL | 0.571221981 | 0.545454545 | 0.334415584 | 0.503246753 |
| 4588.KL | 0.585748792 | 0.406926407 | 0.409009009 | 0.513513514 |
| 5005.KL | 0.568329718 | 0.573593074 | 0.423160173 | 0.531385281 |

| | | | | |
|---|---|---|---|---|
| 5292.KL | 0.529320988 | 0.531385281 | 0.433789954 | 0.465753425 |
| 6963.KL | 0.573391179 | 0.515981735 | 0.42965368 | 0.443722944 |
| 0097.KL | 0.527114967 | 0.566017316 | 0.46969697 | 0.497835498 |
| 5246.KL | 0.587746625 | 0.493506494 | 0.436335404 | 0.454968944 |
| 7293.KL | 0.614967462 | 0.574534161 | 0.385281385 | 0.527056277 |
| 4677.KL | 0.535287731 | 0.57034632 | 0.354978355 | 0.538961039 |
| 5109.KL | 0.63698878 | 0.393939394 | 0.362554113 | 0.520562771 |
| 6742.KL | 0.576690821 | 0.637445887 | 0.392792793 | 0.484684685 |

### A.3.2 Average Percentage Cumulative return for 100 stocks

| code | Proposed_model | Article_model | Buy_and_Hold | Optimistic |
|---|---|---|---|---|
| 6888.KL | 60.4827439 | 81.16848198 | -18.71507967 | -43.49739756 |
| 1023.KL | 32.34083835 | -18.2159349 | -2.513471077 | -3.086547833 |
| 7277.KL | 14.12841569 | 22.52273287 | -30.23952145 | -29.89256864 |
| 6947.KL | 0.653140172 | -1.480826133 | -17.97752243 | -36.56243239 |
| 3182.KL | 2.587088215 | -47.91350014 | 13.73493209 | 25.50133972 |
| 4715.KL | 24.7628634 | 6.639306105 | 24.78991117 | 3.147276022 |
| 3034.KL | 23.9507417 | 31.93140391 | -26.4974671 | -55.53688124 |
| 5168.KL | -1.87091069 | -14.7382202 | -70.47619114 | 36.01448678 |
| 5819.KL | 31.20078362 | 0.677730359 | 0.872931089 | 1.941989992 |
| 1082.KL | 14.74235506 | 18.95648181 | 4.680856935 | -27.74268169 |
| 5225.KL | 7.258352967 | 16.18000554 | 14.2605628 | -10.49507513 |
| 1961.KL | 44.89884822 | 14.79426476 | -8.771932026 | -51.11325867 |
| 2445.KL | 19.2990928 | 19.43125651 | -10.28938727 | -42.75197585 |
| 1155.KL | 21.68599976 | 40.12390418 | -4.090421055 | -21.06679882 |
| 6012.KL | 56.99619955 | 5.378171133 | -29.72476958 | -63.71140588 |
| 3816.KL | 42.53618012 | 19.75060412 | 12.4031041 | -11.96268222 |
| 5296.KL | 3.727707406 | 37.58725318 | -16.40000343 | -1.908014631 |
| 4707.KL | 53.68820702 | -5.875446199 | 83.80953211 | 63.02594617 |
| 5183.KL | 41.02417264 | 10.4792454 | -2.98343041 | -30.3337456 |
| 5681.KL | 0.076981029 | 31.23015117 | -17.55555471 | -60.85358605 |
| 6033.KL | 42.15850147 | -35.5348556 | -7.782521183 | -32.46520628 |
| 4065.KL | 17.02082893 | 0.942074417 | -1.413426252 | -37.54299107 |
| 8869.KL | 18.2657344 | 21.85739508 | 97.94237904 | 80.34887964 |
| 1295.KL | 26.9868562 | 28.72644294 | -5.668020376 | 1.660702202 |
| 1066.KL | 17.69587225 | -9.208104082 | 12.54752507 | -24.46333021 |
| 4197.KL | 29.89765145 | -2.434446935 | 18.27410608 | -35.15094525 |
| 5285.KL | 38.54718359 | 52.82536254 | -5.979380774 | -45.55197312 |
| 4863.KL | 27.49104586 | 21.46593378 | 137.2881421 | -20.6322267 |
| 5347.KL | 0.954623292 | -16.19892709 | -41.86666489 | -44.18538698 |
| 7113.KL | 33.23663335 | -26.61658505 | -58.73700953 | 59.14614123 |
| 5139.KL | 11.81971829 | 85.51549093 | -13.04348032 | 11.37687844 |
| 5099.KL | 15.96820709 | -5.157591466 | 1.063828798 | -6.076033873 |
| 2488.KL | 19.33454697 | -2.567089495 | -8.974362423 | -22.12657889 |
| 1015.KL | 26.70291814 | 5.443303141 | 36.76975607 | -9.222128887 |
| 6399.KL | 27.80784073 | 18.17827259 | -16.50485066 | -47.62892349 |
| 8176.KL | 92.27869912 | -2.069937526 | -80.90909078 | 47.59125303 |

| | | | | |
|---|---|---|---|---|
| 5106.KL | 27.16552047 | 40.90496872 | 26.31578824 | -55.96704412 |
| 1562.KL | 25.56725749 | 24.97331497 | -10.54281441 | -52.75038709 |
| 5248.KL | 67.37144005 | 6.327521004 | -13.87559683 | -26.32886448 |
| 4162.KL | 39.094477 | -27.37128444 | -22.92609283 | -2.770603532 |
| 5210.KL | -28.52201699 | -64.78239679 | 63.63637103 | -71.08763165 |
| 1818.KL | 20.32283853 | -24.62218204 | -13.51351334 | 80.15240302 |
| 2852.KL | 15.77731566 | 0 | -19.16995654 | 35.46724596 |
| 2836.KL | 25.65154687 | -9.776816813 | 23.71134264 | 41.70337438 |
| 7204.KL | 12.88523255 | 60.71478389 | 361.7977752 | -13.71208088 |
| 1619.KL | -14.14937778 | -15.59687769 | -37.56097555 | -49.37893125 |
| 7148.KL | -11.93152512 | -40.20911094 | 63.36336882 | 24.58603839 |
| 5222.KL | 85.32359375 | -49.38107718 | 39.81480838 | -17.79874287 |
| 3689.KL | 5.93467242 | 116.9528816 | -31.46935857 | -57.90458936 |
| 0128.KL | 92.42552765 | -20.62784358 | 409.3026128 | 8.718448812 |
| 5398.KL | 64.63815338 | 27.09932917 | 58.74999131 | -16.47512334 |
| 0078.KL | 139.8564847 | 15.61927918 | -53.52112351 | -71.05583503 |
| 2291.KL | 6.687681649 | 46.79566917 | 15.18737584 | 5.3218063 |
| 0208.KL | -16.86332684 | -4.769818511 | -43.64406705 | -0.688450314 |
| 5102.KL | 89.19250491 | -20.64686869 | 78.51852172 | 98.85260305 |
| 3255.KL | 38.33549156 | 65.8630418 | 24.62160012 | -7.116268566 |
| 3301.KL | 34.43058991 | -17.17055465 | -13.89414251 | 29.36495094 |
| 5227.KL | 10.97717044 | -5.283565687 | -19.02438999 | -34.2691767 |
| 3336.KL | 36.39794047 | 2.995117392 | 2.923973754 | -31.34595475 |
| 0166.KL | 23.96523158 | 37.54719973 | 68.78612972 | -5.729305113 |
| 7153.KL | 23.77788964 | -7.737210219 | -52.87081078 | 58.70222961 |
| 5878.KL | 13.45056413 | 14.13631476 | -20.00000217 | -49.10243029 |
| 6633.KL | 9.579341827 | -1.226992698 | -27.94117583 | -29.67438761 |
| 5284.KL | 25.53936627 | -12.39764974 | -27.11863962 | -21.67771375 |
| 3859.KL | 19.87198423 | 52.44634347 | -12.98461635 | -30.01277222 |
| 5264.KL | 10.24828695 | -2.07428837 | -14.90384395 | -9.630245141 |
| 5014.KL | 90.97740782 | -0.193814992 | -22.54047326 | -21.50695754 |
| 1171.KL | 31.58314468 | 46.06296686 | -39.79592222 | -40.30686786 |
| 3867.KL | 23.54814904 | 9.126716572 | 184.2105492 | 117.2383926 |
| 1651.KL | 24.49823725 | 41.35904612 | -20.56505311 | -6.463633667 |
| 5236.KL | 19.24057553 | 22.86561079 | 16.49484143 | -22.49695552 |
| 3069.KL | 4.40320477 | 29.01903361 | 105.9171539 | 88.51961142 |
| 5286.KL | -15.66537992 | -2.068321356 | -57.18654482 | -32.56213867 |
| 9385.KL | 13.28002363 | -37.94374624 | -16.17647252 | -44.90879432 |
| 0138.KL | 44.849363 | -32.36576656 | 24.60317971 | -3.008720874 |
| 5258.KL | 13.86235021 | 22.30809193 | -27.79291443 | -1.987771494 |
| 7160.KL | 57.56919969 | 64.3266842 | 172.9290387 | 82.01241401 |
| 7084.KL | 13.57499285 | 30.2200053 | 2.448977209 | -40.45353623 |
| 5218.KL | 10.53251132 | 13.25668157 | -83.33333375 | -80.19568293 |
| 4731.KL | 11.23624113 | -46.3224154 | 17.74743922 | 18.99338663 |
| 5279.KL | -40.24320344 | 19.2564559 | -94.57142864 | 68.30990314 |
| 5288.KL | -16.71237336 | -70.540199 | -27.69230522 | -20.47323732 |
| 7155.KL | 19.49626774 | -12.38860351 | 66.66666061 | -33.90626525 |
| 8664.KL | 77.16904046 | 14.23374946 | -7.361963504 | 42.0984089 |

| | | | | |
|---|---|---|---|---|
| 5211.KL | 22.05277011 | 14.37521447 | -0.836344645 | -52.23266174 |
| 5176.KL | 9.98173815 | 2.106171392 | -13.71428626 | -31.75275496 |
| 7106.KL | 327.6298838 | 12.27042437 | 149.3961746 | 343.6081668 |
| 6139.KL | 23.66955733 | 47.22950802 | -11.82795844 | 80.73298409 |
| 5031.KL | 5.951367392 | 108.1729706 | 68.15290768 | -37.96626925 |
| 5148.KL | -4.821581134 | 24.03174405 | -58.21917819 | -65.73836577 |
| 4588.KL | 52.15548182 | -47.44270128 | 35.64814585 | -15.28118086 |
| 5005.KL | 29.60803528 | 4.640636807 | 100.666666 | 167.4234013 |
| 5292.KL | -16.15871918 | -5.472159957 | -35.98055107 | -31.94256077 |
| 6963.KL | 156.5923696 | -21.52859829 | 28.66241487 | -20.62160328 |
| 0097.KL | 38.37478232 | 119.7413448 | 99.21466666 | 78.67643579 |
| 5246.KL | 78.06923976 | -3.438026263 | -17.81472668 | -55.87777038 |
| 7293.KL | 5.783831865 | 61.77895676 | -6.458796807 | 44.68429097 |
| 4677.KL | 3.730028956 | 7.036305794 | -45.94134369 | -1.750688956 |
| 5109.KL | 26.26378827 | -25.87253914 | -21.84874265 | -23.30091612 |
| 6742.KL | 14.48303296 | 11.94494862 | 13.33335523 | -10.9798174 |

### A.3.3 Average Sharpe Ratio for 100 stocks

| code | Proposed_model | Article_model | Buy_and_Hold | Optimistic |
|---|---|---|---|---|
| 6888.KL | 1.267593908 | 0.879332598 | -0.014805991 | -0.912934721 |
| 1023.KL | 0.981877165 | -0.497913991 | 0.109967218 | 0.05238395 |
| 7277.KL | 0.806225841 | 0.957874167 | -0.208535965 | -0.681670463 |
| 6947.KL | 0.842003559 | 0.060883563 | -0.08788622 | -1.179473349 |
| 3182.KL | 0.230497336 | -1.690205483 | 0.33708406 | 0.986080395 |
| 4715.KL | 2.263836282 | 0.369308185 | 0.490874767 | 0.256961127 |
| 3034.KL | 2.604676519 | 1.21929983 | -0.174008824 | -1.709992451 |
| 5168.KL | 0.073517139 | 0.061959657 | -0.454931912 | 0.633082161 |
| 5819.KL | 2.279825811 | 0.129644996 | 0.112068573 | 0.156705999 |
| 1082.KL | 2.347637067 | 1.700254146 | 0.172775486 | -0.802796859 |
| 5225.KL | 1.461306694 | 1.311427262 | 0.307120898 | -0.152611296 |
| 1961.KL | 1.166624366 | 3.312294487 | 0.001622321 | -1.765373959 |
| 2445.KL | 2.411734695 | 0.552154038 | -0.017888684 | -1.238205633 |
| 1155.KL | 1.565195694 | 1.911658862 | 0.018308305 | -0.888551972 |
| 6012.KL | 1.199860317 | 0.340601088 | -0.261399586 | -2.536045194 |
| 3816.KL | 6.08922044 | 0.750243298 | 0.254199144 | -0.215527619 |
| 5296.KL | 1.29004406 | 1.195984362 | -1.34104576 | -0.265452367 |
| 4707.KL | 1.969926813 | -1.512702428 | 1.515370695 | 2.180605027 |
| 5183.KL | 1.999957847 | 2.211201522 | 0.135810625 | -0.644894427 |
| 5681.KL | 0 | 0.949604477 | -0.029098062 | -1.952639345 |
| 6033.KL | 2.393545696 | -1.879171876 | -0.001359787 | -1.003919031 |
| 4065.KL | 4.547711096 | 0.124411405 | 0.072422678 | -1.527789876 |
| 8869.KL | 1.137579655 | 1.863263252 | 0.715755189 | 1.219691524 |
| 1295.KL | 1.792389706 | 0.564166678 | 0.053827648 | 0.158419058 |
| 1066.KL | 0.776626066 | -0.194401063 | 0.253130564 | -0.732412896 |
| 4197.KL | 5.244570874 | 0.053453205 | 0.434974724 | -1.844010552 |
| 5285.KL | 1.185180716 | 0.745371873 | 0.0861302 | -2.249700172 |

| | | | | |
|---|---|---|---|---|
| 4863.KL | 2.803705988 | 0.616903248 | 0.907601139 | -0.365904927 |
| 5347.KL | 0.129849856 | -0.475901875 | -0.596223804 | -1.883298252 |
| 7113.KL | 0.566125407 | -0.600286056 | -0.194974464 | 0.821921225 |
| 5139.KL | 1.338631879 | 1.383215212 | -0.000633215 | 0.394137878 |
| 5099.KL | 4.637431856 | 0.02805431 | 0.19604752 | 0.005681611 |
| 2488.KL | 0.835894466 | 0.151963622 | 0.04626757 | -0.494911148 |
| 1015.KL | 1.062357047 | 0.246532719 | 0.652169282 | -0.226473375 |
| 6399.KL | 1.15696494 | 0.919036504 | -0.141017126 | -2.180873338 |
| 8176.KL | 2.137695613 | 0.14366014 | -0.355962406 | 0.728327876 |
| 5106.KL | 3.083429986 | 0.990518503 | 0.412822497 | -2.746729004 |
| 1562.KL | 0.791983867 | 2.103060119 | -0.043742273 | -2.721805907 |
| 5248.KL | 1.31932594 | 0.315959172 | 0.021076604 | -0.787518046 |
| 4162.KL | 2.405386834 | -0.488853737 | -0.367667893 | 0.002472138 |
| 5210.KL | -1.77845321 | -0.954827074 | 0.575769471 | -1.24549904 |
| 1818.KL | 1.820128052 | -0.58629459 | -0.032011946 | 1.504641347 |
| 2852.KL | 1.38831537 | 0 | -0.199048274 | 1.286899817 |
| 2836.KL | 1.363434858 | -0.92491206 | 0.341518619 | 0.845813022 |
| 7204.KL | 2.751708423 | 1.018703378 | 1.049983435 | 0.0615495 |
| 1619.KL | -1.433743426 | -3.881714734 | -0.163915858 | -1.14737614 |
| 7148.KL | 0.008497392 | -0.918916179 | 0.515947918 | 0.544210451 |
| 5222.KL | 3.432346044 | -0.359709816 | 0.475173087 | -0.18297526 |
| 3689.KL | 1.777931415 | 2.921743335 | -0.278429052 | -2.004714285 |
| 0128.KL | 0 | -0.279745822 | 1.109790577 | 0.32776028 |
| 5398.KL | 4.185874006 | 1.173393881 | 0.536342702 | -0.133367111 |
| 0078.KL | 2.043877429 | 1.431925985 | -0.165446776 | -1.491580639 |
| 2291.KL | 4.273571876 | 0.801154203 | 0.451077531 | 0.359746551 |
| 0208.KL | -0.662563911 | -2.340490445 | -0.926449088 | 0.171941418 |
| 5102.KL | 2.805538063 | -0.671576222 | 0.604403718 | 1.347257827 |
| 3255.KL | 5.321821116 | 2.02787198 | 0.355308364 | -0.041249678 |
| 3301.KL | 2.923099 | -0.885158895 | -0.070818469 | 0.91365097 |
| 5227.KL | 4.536608833 | -0.02160215 | -0.238921455 | -1.939657657 |
| 3336.KL | 1.472655171 | 3.350043525 | 0.211951 | -0.445652288 |
| 0166.KL | 2.329786852 | 2.249306372 | 0.604575148 | 0.072432415 |
| 7153.KL | 0.794533977 | 0.012930975 | -0.185110663 | 0.85166917 |
| 5878.KL | 6.441554368 | 0.535483836 | -0.115948823 | -1.818387614 |
| 6633.KL | 1.06453681 | 0 | -1.229900796 | -4.280414126 |
| 5284.KL | 2.174849555 | -0.573081176 | -0.424195923 | -0.876853232 |
| 3859.KL | 2.209236743 | 3.167219121 | -0.055164737 | -1.09366253 |
| 5264.KL | 0 | -0.056687255 | -0.499570107 | -0.885047465 |
| 5014.KL | 2.375657254 | 0.056797609 | 0.002364397 | -0.283827997 |
| 1171.KL | 0.543746272 | 2.033990196 | -0.299818285 | -1.323120052 |
| 3867.KL | 2.452106722 | 0.276974812 | 0.92339359 | 1.373695118 |
| 1651.KL | 0.557536572 | 1.779933828 | 0.038711261 | 0.034895754 |
| 5236.KL | 1.844854138 | 0.550237212 | 0.359058242 | -0.849986618 |
| 3069.KL | 1.043851857 | 2.344963658 | 0.837531245 | 1.642301422 |
| 5286.KL | -0.912370616 | 0.064237204 | -1.385025785 | -1.731403736 |
| 9385.KL | 0.871610603 | -1.815493202 | 0.123310833 | -0.806422502 |
| 0138.KL | 1.591632805 | -0.858831212 | 0.359044794 | 0.182284888 |
| 5258.KL | 3.989179286 | 0.485393086 | -0.171872327 | 0.143148458 |

| | | | | |
|---|---|---|---|---|
| 7160.KL | 0.74870124 | 2.812212611 | 0.818127555 | 1.029236648 |
| 7084.KL | 1.985179666 | 0.526113534 | 0.147446362 | -1.277086467 |
| 5218.KL | 1.620567237 | 1.965130631 | -0.139485799 | -1.253513225 |
| 4731.KL | 0.51302272 | -0.024772608 | 0.304751913 | 0.566827072 |
| 5279.KL | -6.141359152 | 1.959834305 | -0.921728791 | 1.637711124 |
| 5288.KL | -1.461709163 | -4.588987235 | -0.672991468 | -1.460944602 |
| 7155.KL | 3.918216914 | -0.544782495 | 0.533392767 | -0.373767463 |
| 8664.KL | 2.118775651 | 0.658057697 | 0.122116548 | 1.134274802 |
| 5211.KL | 3.219470264 | 0.582115806 | 0.118057548 | -2.29620411 |
| 5176.KL | 2.541844415 | 0.968745637 | -0.117330587 | -1.395104512 |
| 7106.KL | 3.928839225 | 1.510514997 | 0.951355275 | 2.270338795 |
| 6139.KL | 3.385887134 | 1.123662364 | 0.031591165 | 1.557315814 |
| 5031.KL | -0.63986437 | 2.351816005 | 0.779813452 | -1.257392946 |
| 5148.KL | 0.972465501 | 1.704313525 | -0.380751175 | -1.922130603 |
| 4588.KL | 4.840731131 | -0.339518716 | 0.560914952 | -0.257200744 |
| 5005.KL | 3.410997763 | 0.386646347 | 0.643222256 | 1.607002387 |
| 5292.KL | -0.64126237 | -0.05399202 | -0.807790431 | -2.145892665 |
| 6963.KL | 2.503612324 | -0.618046549 | 0.407591468 | -0.102119751 |
| 0097.KL | 4.497814994 | 1.246104722 | 0.726842239 | 1.171070369 |
| 5246.KL | 5.618265064 | -0.043477418 | -0.03082263 | -1.796013407 |
| 7293.KL | 2.744909417 | 4.271507062 | 0.094885842 | 1.00050356 |
| 4677.KL | 0.772290958 | 0.376638365 | -0.312798244 | 0.151807629 |
| 5109.KL | 4.177687782 | -0.087817716 | -0.116796294 | -0.752313505 |
| 6742.KL | 0.956505904 | 1.346228466 | 0.343396289 | -0.36794269 |

## A.3.4 Average Max Drawdown for 100 stocks

| code | Proposed_model | Article_model | Buy_and_Hold | Optimistic |
|---|---|---|---|---|
| 6888.KL | -0.236451375 | -0.356676581 | -0.501901184 | -0.528211579 |
| 1023.KL | -0.321478193 | -0.351066121 | -0.508474568 | -0.402352113 |
| 7277.KL | -0.171083435 | -0.223410241 | -0.503759395 | -0.383982018 |
| 6947.KL | -0.046730414 | -0.210135663 | -0.389763789 | -0.480237356 |
| 3182.KL | -0.173017257 | -0.533073957 | -0.38155135 | -0.154990162 |
| 4715.KL | -0.114128486 | -0.227520338 | -0.270072995 | -0.223642771 |
| 3034.KL | -0.12792946 | -0.081578933 | -0.301301283 | -0.559837442 |
| 5168.KL | -0.464334652 | -0.643384304 | -0.909268292 | -0.665867677 |
| 5819.KL | -0.089159329 | -0.216048948 | -0.439663236 | -0.203842378 |
| 1082.KL | -0.083587319 | -0.100685715 | -0.434393656 | -0.393423262 |
| 5225.KL | -0.072162078 | -0.169000228 | -0.210792586 | -0.239440448 |
| 1961.KL | -0.156812179 | -0.06963252 | -0.263485513 | -0.522170985 |
| 2445.KL | -0.104336038 | -0.255546051 | -0.320434451 | -0.479709843 |
| 1155.KL | -0.095022694 | -0.110759427 | -0.277605748 | -0.249065798 |
| 6012.KL | -0.201096633 | -0.109298273 | -0.454388966 | -0.677431 |
| 3816.KL | -0.046444211 | -0.217947886 | -0.303738347 | -0.325839171 |
| 5296.KL | -0.067847095 | -0.16872107 | -0.244063116 | -0.142856731 |
| 4707.KL | -0.160333325 | -0.095935349 | -0.187420543 | -0.104193104 |
| 5183.KL | -0.175247905 | -0.042808367 | -0.560897406 | -0.470008495 |

| | | | | |
|---|---|---|---|---|
| 5681.KL | 0 | -0.294446817 | -0.397887302 | -0.624010529 |
| 6033.KL | -0.08353361 | -0.398941775 | -0.257932481 | -0.347634075 |
| 4065.KL | -0.031833444 | -0.237078755 | -0.24849696 | -0.431660531 |
| 8869.KL | -0.172801285 | -0.096608226 | -0.45297507 | -0.351776644 |
| 1295.KL | -0.133208376 | -0.345263264 | -0.497219981 | -0.290063953 |
| 1066.KL | -0.128535909 | -0.334526094 | -0.290268458 | -0.430744681 |
| 4197.KL | -0.058480897 | -0.240641668 | -0.169291367 | -0.457466343 |
| 5285.KL | -0.238775827 | -0.188084299 | -0.322381924 | -0.480134272 |
| 4863.KL | -0.111166264 | -0.306652232 | -0.301492543 | -0.337767415 |
| 5347.KL | -0.139572048 | -0.298552968 | -0.470469794 | -0.542716438 |
| 7113.KL | -0.601943624 | -0.364623297 | -0.917187502 | -0.707785224 |
| 5139.KL | -0.134128141 | -0.337241107 | -0.543023268 | -0.259060622 |
| 5099.KL | -0.042974796 | -0.471460168 | -0.48260866 | -0.259487641 |
| 2488.KL | -0.227795786 | -0.445803913 | -0.637413376 | -0.50870695 |
| 1015.KL | -0.160490093 | -0.459432091 | -0.245283039 | -0.32904402 |
| 6399.KL | -0.146961075 | -0.181466846 | -0.311999989 | -0.506914602 |
| 8176.KL | -0.336339174 | -0.402808834 | -0.9198813 | -0.35718675 |
| 5106.KL | -0.074331996 | -0.476562475 | -0.195555581 | -0.556734914 |
| 1562.KL | -0.139193134 | -0.057334059 | -0.360213466 | -0.578747058 |
| 5248.KL | -0.41209501 | -0.168927352 | -0.538834944 | -0.393496614 |
| 4162.KL | -0.114831782 | -0.554705167 | -0.359649108 | -0.236230033 |
| 5210.KL | -0.397677616 | -0.660225086 | -0.788990828 | -0.764785324 |
| 1818.KL | -0.124407905 | -0.374295056 | -0.41886796 | -0.201657219 |
| 2852.KL | -0.205298398 | 0 | -0.396261668 | -0.144999593 |
| 2836.KL | -0.212957085 | -0.257754334 | -0.540832055 | -0.320296897 |
| 7204.KL | -0.184402941 | -0.35502252 | -0.510869587 | -0.460253705 |
| 1619.KL | -0.299371583 | -0.227966508 | -0.640677991 | -0.574280798 |
| 7148.KL | -0.526744334 | -0.46932185 | -0.654560122 | -0.478827456 |
| 5222.KL | -0.116965381 | -0.655267965 | -0.538461503 | -0.339715593 |
| 3689.KL | -0.066850624 | -0.124708662 | -0.474567084 | -0.668845831 |
| 0128.KL | -0.265502258 | -0.364058464 | -0.577358713 | -0.387371373 |
| 5398.KL | -0.105651048 | -0.257235198 | -0.395631049 | -0.485720365 |
| 0078.KL | -0.38697392 | -0.133858236 | -0.663265299 | -0.713507159 |
| 2291.KL | -0.041643467 | -0.412484224 | -0.143269214 | -0.168454024 |
| 0208.KL | -0.382339085 | -0.092118365 | -0.613333321 | -0.26585016 |
| 5102.KL | -0.305102973 | -0.366395381 | -0.512500025 | -0.332730333 |
| 3255.KL | -0.073955823 | -0.395953109 | -0.423324772 | -0.322934914 |
| 3301.KL | -0.144323195 | -0.203442452 | -0.431818161 | -0.221600925 |
| 5227.KL | -0.048236559 | -0.312636757 | -0.326829255 | -0.342691767 |
| 3336.KL | -0.154637515 | 0 | -0.519999981 | -0.453011456 |
| 0166.KL | -0.184209537 | -0.128142462 | -0.5 | -0.409082377 |
| 7153.KL | -0.352529563 | -0.373900485 | -0.897288843 | -0.597277503 |
| 5878.KL | -0.043064584 | -0.375732398 | -0.338842986 | -0.48773097 |
| 6633.KL | -0.080980561 | 0 | -0.286764692 | -0.286168606 |
| 5284.KL | -0.147313416 | -0.198347109 | -0.50857142 | -0.409148761 |
| 3859.KL | -0.081317084 | -0.160871651 | -0.43733562 | -0.379548574 |
| 5264.KL | 0 | -0.169828942 | -0.260869529 | -0.20940016 |
| 5014.KL | -0.189289379 | -0.122162845 | -0.540816289 | -0.261419144 |
| 1171.KL | -0.264094982 | -0.167333985 | -0.560747686 | -0.448756878 |

| | | | | |
|---|---|---|---|---|
| 3867.KL | -0.092720676 | -0.305588902 | -0.473333321 | -0.334199247 |
| 1651.KL | -0.402674951 | -0.228085506 | -0.633039571 | -0.432104794 |
| 5236.KL | -0.129434032 | -0.440805439 | -0.270408204 | -0.313995009 |
| 3069.KL | -0.199577122 | -0.207650266 | -0.424460417 | -0.208449447 |
| 5286.KL | -0.352986134 | -0.206920115 | -0.680459766 | -0.397233375 |
| 9385.KL | -0.318345819 | -0.4610109 | -0.592233011 | -0.625104932 |
| 0138.KL | -0.237728912 | -0.383522776 | -0.532352944 | -0.430035609 |
| 5258.KL | -0.054533445 | -0.299567367 | -0.442553193 | -0.337657736 |
| 7160.KL | -0.415378993 | -0.075240374 | -0.58395246 | -0.316309555 |
| 7084.KL | -0.066279576 | -0.44833434 | -0.337450554 | -0.440967229 |
| 5218.KL | -0.113535683 | -0.071092003 | -0.900000006 | -0.87757533 |
| 4731.KL | -0.207993177 | -0.69042562 | -0.381097564 | -0.167499361 |
| 5279.KL | -0.416429213 | -0.051370141 | -0.968911917 | -0.393639894 |
| 5288.KL | -0.276668699 | -0.723453328 | -0.447368431 | -0.241710977 |
| 7155.KL | -0.22063208 | -0.331649598 | -0.578616327 | -0.47717937 |
| 8664.KL | -0.18068736 | -0.284220608 | -0.583333333 | -0.299118487 |
| 5211.KL | -0.067766315 | -0.343599286 | -0.335135133 | -0.533632495 |
| 5176.KL | -0.085428814 | -0.077828659 | -0.317948713 | -0.34895667 |
| 7106.KL | -0.39086158 | -0.133601216 | -0.861583053 | -0.653986973 |
| 6139.KL | -0.10350875 | -0.25841808 | -0.561281313 | -0.274189312 |
| 5031.KL | -0.015635341 | -0.193844345 | -0.156767783 | -0.403767117 |
| 5148.KL | -0.231567792 | -0.104650467 | -0.709183683 | -0.713182946 |
| 4588.KL | -0.150248519 | -0.612250296 | -0.293548347 | -0.404519657 |
| 5005.KL | -0.153795158 | -0.300008125 | -0.542483696 | -0.402659825 |
| 5292.KL | -0.458012119 | -0.261283545 | -0.569841277 | -0.408471881 |
| 6963.KL | -0.281941941 | -0.440891868 | -0.615151515 | -0.508988378 |
| 0097.KL | -0.107362347 | -0.478139157 | -0.467509013 | -0.317942853 |
| 5246.KL | -0.105044465 | -0.27896087 | -0.291139228 | -0.561885089 |
| 7293.KL | -0.095108967 | -0.077540097 | -0.49447511 | -0.229572961 |
| 4677.KL | -0.283340802 | -0.416746305 | -0.579166687 | -0.326024567 |
| 5109.KL | -0.082997171 | -0.561998312 | -0.49275363 | -0.285769147 |
| 6742.KL | -0.143017075 | -0.171270692 | -0.273885384 | -0.311949037 |

Weekly Report

# FINAL YEAR PROJECT WEEKLY REPORT

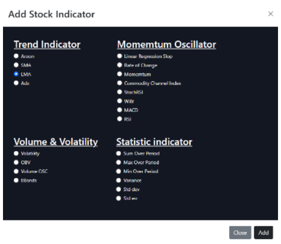
*(Project II)*

| | |
|---|---|
| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:**1 |
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

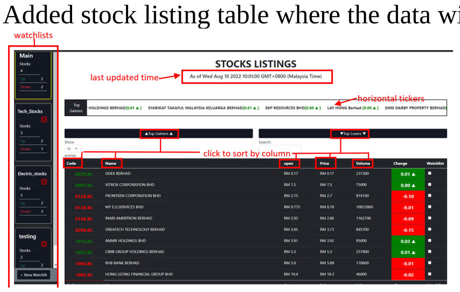
## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- have increased the number of indicators in the system from 9 to 22




- Added feature to allows user to add multiple watchlists
- Added stock listing table where the data will almost be synchronized with the real-time




-

## 2. WORK TO BE DONE

- Improve the stock trading simulation report by incorporating interactive graphs and adding some common portfolio metrices
- Integrate the system with Quanstats API to generate more advanced and in-depth analysis portfolio report

## 3. PROBLEMS ENCOUNTERED

## 4. SELF EVALUATION OF THE PROGRESS

- Keep the progress in timeline

Supervisor's signature

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

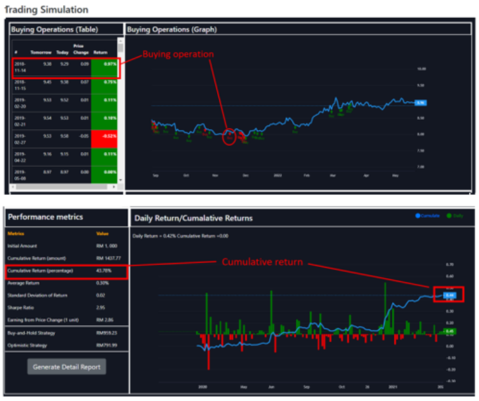
*(Project II)*

| | |
|---|---|
| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:**2 |
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Successfully improved the trading simulation feature with interactive graphs and integrated the Quanstats API to generate advanced portfolio report in real-time using dynamic data.





## 2. WORK TO BE DONE

- Apply parallel processing approach to improve the system in term of waiting time.

## 3. PROBLEMS ENCOUNTERED

- It is challenging to incorporate the Quanstats API on the system due to limited documentation provided in Quanstas API

## 4. SELF EVALUATION OF THE PROGRESS

- Even there's limited documentation in the Quanstats API, I was able to integrate the API in the system and use it generate advanced portfolio report in real-time based on dynamic data.

Supervisor's signature

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:**3 |
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Successfully integrated parallel processing in model training and prediction for watchlist stocks feature and in-depth analysis indicator recommendation feature.
- Multiple training process can be carried out parallelly in the system. Waiting time for both features was reduced significantly after applying parallel processing.

**2. WORK TO BE DONE**

- Supervisor suggests adding indicator suggestion feature in the system. This feature will find out the best combination of indicators for a stock in advance in the offline environment so that they can be suggested to the users in the system.

**3. PROBLEMS ENCOUNTERED**

- The code to implement parallel processing is complicated and challenging. More time and effort are needed to figure out the code logic to make the function work as expected.

**4. SELF EVALUATION OF THE PROGRESS**

- Manage to implement parallel processing perfectly in the system.

Supervisor's signature

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

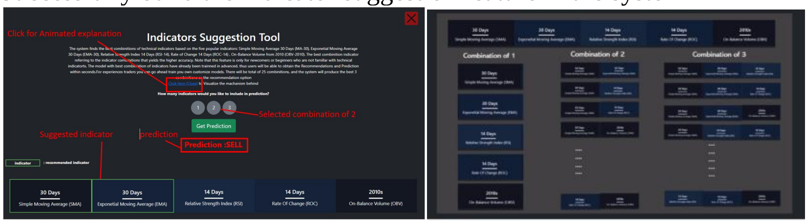
*(Project II)*

| | |
|---|---|
| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:**4 |
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Successfully build the indicator suggestion feature in the system




- 
- 
- Have pretrained the suggested models for each stock and store in the database so that the users can use the suggested model to make prediction in real-time
- Started the report writing

## 2. WORK TO BE DONE

- Revise the architecture design of the prediction model and make improvement to it to ensure it fits well across all 100 stocks.
- Fine-tunning and experiments will be carried out to improve the architecture design of the prediction model.
- Continue the report writing process.

## 3. PROBLEMS ENCOUNTERED

## 4. SELF EVALUATION OF THE PROGRESS

- Keep the progress in timeline.

Supervisor's signature

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:** 5 |
|---|---|
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Still in the progress of designing the optimal architecture of the prediction model for 100 stocks
- Many experiments have been run to figure out the optimal architecture

| code | Proposed_model | Article_model | Buy_and_Hold | Optimistic |
|---|---|---|---|---|
| 6888.KL, | 60.4827439 | 81.16848198 | -18.71507967 | -43.49739756 |
| 1023.KL, | 32.34083835 | -18.2159349 | -2.513471077 | -3.086547833 |
| 7277.KL, | 14.12841569 | 22.52273287 | -30.23952145 | -29.89256864 |
| 6947.KL, | 0.653140172 | -1.480826133 | -17.97752243 | -36.56243239 |
| 3182.KL, | 2.587088215 | -47.91350014 | 13.73493209 | 25.50133972 |
| 4715.KL, | 24.7628634 | 6.639306105 | 24.78991117 | 3.147276022 |
| 3034.KL, | 23.9507417 | 31.93140391 | -26.4974671 | -55.53688124 |
| 5168.KL, | -1.87091069 | -14.7382202 | -70.47619114 | 36.01448678 |
| 5819.KL, | 31.20078362 | 0.677730359 | 0.872931089 | 1.941989992 |
| 1082.KL, | 14.74235506 | 18.95648181 | 4.680856935 | -27.74268169 |
| 5225.KL, | 7.258352967 | 16.18000554 | 14.2605628 | -10.49507513 |
| 1961.KL, | 44.89884822 | 14.79426476 | -8.771932026 | -51.11325867 |
| 2445.KL, | 19.2990928 | 19.43125651 | -10.28938727 | -42.75197585 |
| 1155.KL, | 21.68599976 | 40.12390418 | -4.090421055 | -21.06679882 |
| 6012.KL, | 56.99619955 | 5.378171133 | -29.72476958 | -63.71140588 |
| 3816.KL, | 42.53618012 | 19.75060412 | 12.4031041 | -11.96268222 |
| 5296.KL, | 3.727707406 | 37.58725318 | -16.40000343 | -1.908014631 |
| 4707.KL, | 53.68820702 | -5.875446199 | 83.80953211 | 63.02594617 |
| 5183.KL, | 41.02417264 | 10.4792454 | -2.98343041 | -30.3337456 |
| 5681.KL, | 0.076981029 | 31.23015117 | -17.55555471 | -60.85358605 |
| 6033.KL, | 42.15850147 | -35.5348556 | -7.782521183 | -32.46520628 |
| 4065.KL, | 17.02082893 | 0.942074417 | -1.413426252 | -37.54299107 |
| 8869.KL, | 18.2657344 | 21.85739508 | 97.94237904 | 80.34887964 |

(Performance report from one of the experiments)

- Have done some research on the articles to improve the model accuracy and model's profit

## 2. WORK TO BE DONE

- Continue the optimal architecture designing process
- Continue writing the report

## 3. PROBLEMS ENCOUNTERED

- It is challenging to design the optimal model architecture that can achieved good accuracy for all 100 stocks as different stocks has different distribution of data and different status (bull market/bear market)

## 4. SELF EVALUATION OF THE PROGRESS

- Keep the progress in timeline.
-

Supervisor's signature　　　　　　　　　　Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

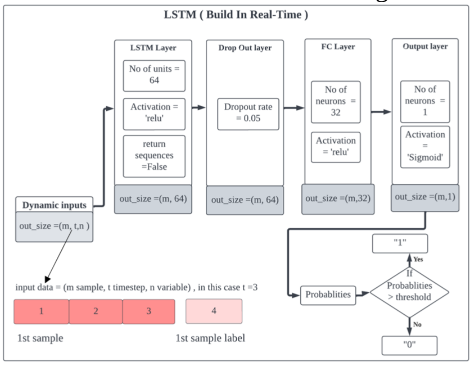
*(Project II)*

| | |
|---|---|
| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:** 6 |
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Have finalized the architecture design of the prediction model




## 2. WORK TO BE DONE

- Start to incorporate the optimal architecture of the prediction model into the web application backend using TensorFlow JS.
- Continue writing the report
- Fixing some bugs and errors

## 3. PROBLEMS ENCOUNTERED

## 4. SELF EVALUATION OF THE PROGRESS

- Optimal architecture model that is well-suited for 100 stocks has been design after many runs of experiment and fine-tuning process.
- Keep the progress in timeline.

Supervisor's signature

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:** 7 |
|---|---|
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Have incorporated the optimal architecture design into the web application.
- The proposed system is almost completed
- Make some changes on the candlestick charts to make it looks more professional
- 


## 2. WORK TO BE DONE

- Start to conduct system testing to test out the project objectives
- Continue the report writing process
- Fixing bugs and errors in the system

## 3. PROBLEMS ENCOUNTERED

## 4. SELF EVALUATION OF THE PROGRESS

- Keep the progress in timeline.

Supervisor's signature

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:** 8 |
|---|---|
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Conducted testing on the performance of the proposed model design
- Conducted the simulation testing on all the functionalities provided in the system
- Recorded the testing data results

**A.3.2 Average Percentage Cumulative return for 100 stocks**

| code | Proposed_model | Article_model | Buy_and_Hold | Optimistic |
|---|---|---|---|---|
| 6888.KL, | 60.4827439 | 81.16848198 | -18.71507967 | -43.49739756 |
| 1023.KL, | 32.34083835 | -18.2159349 | -2.513471077 | -3.086547833 |
| 7277.KL, | 14.12841569 | 22.52273287 | -30.23952145 | -29.89256864 |
| 6947.KL, | 0.653140172 | -1.480826133 | -17.97752243 | -36.56243239 |
| 3182.KL, | 2.587088215 | -47.91350014 | 13.73493209 | 25.50133972 |
| 4715.KL, | 24.7628634 | 6.639306105 | 24.78991117 | 3.147276022 |
| 3034.KL, | 23.9507417 | 31.93140391 | -26.4974671 | -55.53688124 |
| 5168.KL, | -1.87091069 | -14.7382202 | -70.47619114 | 36.01448678 |
| 5819.KL, | 31.20078362 | 0.677730359 | 0.872931089 | 1.941989992 |
| 1082.KL, | 14.74235506 | 18.95648181 | 4.680856935 | -27.74268169 |
| 5225.KL, | 7.258352967 | 16.18000554 | 14.2605628 | -10.49507513 |
| 1961.KL, | 44.89884822 | 14.79426476 | -8.771932026 | -51.11325867 |
| 2445.KL, | 19.2990928 | 19.43125651 | -10.28938727 | -42.75197585 |
| 1155.KL, | 21.68599976 | 40.12390418 | -4.090421055 | -21.06679882 |
| 6012.KL. | 56.99619955 | 5.378171133 | -29.72476958 | -63.71140588 |

**A.3.1 Average Accuracy for 100 stocks**

| code | Proposed_model | Article_model | Buy_and_Hold | Optimistic |
|---|---|---|---|---|
| 6888.KL, | 0.534201954 | 0.506493506 | 0.433982684 | 0.464285714 |
| 1023.KL, | 0.54483008 | 0.537878788 | 0.438311688 | 0.506493506 |
| 7277.KL, | 0.582699964 | 0.574675325 | 0.413419913 | 0.45995671 |
| 6947.KL, | 0.568946797 | 0.53030303 | 0.428571429 | 0.461038961 |
| 3182.KL, | 0.523550725 | 0.512987013 | 0.423423423 | 0.520720721 |
| 4715.KL, | 0.551932367 | 0.522727273 | 0.425225225 | 0.477477477 |
| 3034.KL, | 0.593557727 | 0.567099567 | 0.397186147 | 0.478354978 |
| 5168.KL, | 0.513210279 | 0.503246753 | 0.433982684 | 0.506493506 |
| 5819.KL, | 0.558206797 | 0.50974026 | 0.42965368 | 0.483766234 |

## 2. WORK TO BE DONE

- start to construct the graphical representation of the system testing results for better explanation in the report.
- Continue the report writing process

## 3. PROBLEMS ENCOUNTERED

## 4. SELF EVALUATION OF THE PROGRESS

- Keep the progress in timeline.

Supervisor's signature

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

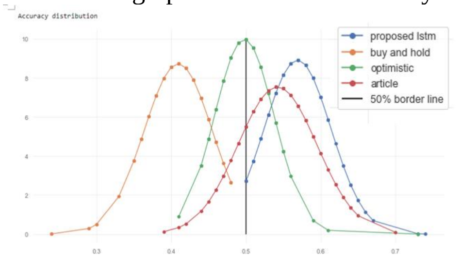
*(Project II)*

| | |
|---|---|
| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:** 9 |
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Plotted few graphs to visualize on the system testing result




| Strategy | Average Accuracy | Max Accuracy | Min Accuracy | stdev |
|---|---|---|---|---|
| Proposed model | 0.57 | 0.74 | 0.50 | 0.045 |
| Article model | 0.54 | 0.70 | 0.39 | 0.053 |
| Buy-and-Hold | 0.41 | 0.48 | 0.23 | 0.045 |
| Optimistic | 0.49 | 0.72 | 0.41 | 0.039 |

- Finalized the GUI design of the system.
- Started to include the system testing and object evaluation discussion on the report
- Revise the chapter 1 and chapter 2 content in the report

## 2. WORK TO BE DONE

- Polishing and finalizing the system code
- Continue the report writing process

## 3. PROBLEMS ENCOUNTERED

## 4. SELF EVALUATION OF THE PROGRESS

- Keep the progress in timeline.

Supervisor's signature

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:** 11 |
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Completed 90% in the report writing
- Finalized the complete web application
- Have sent the draft report to supervisor for checking

**2. WORK TO BE DONE**

- Polishing and finalizing the report

**3. PROBLEMS ENCOUNTERED**

**4. SELF EVALUATION OF THE PROGRESS**

- Keep the progress in timeline.

Supervisor's signature

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| **Trimester, Year:** Trimester 3, Year 3 | **Study week no.:** 12 |
|---|---|
| **Student Name & ID:** Cheah Shing Dhee 18ACB05004 | |
| **Supervisor:** Ts Dr Ku Chin Soon | |
| **Project Title:** Stock Indicator Scanner Customization Tool | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Completed 100% of report writing
- Finalized the complete web application
- Made corrections according to the supervisor's comments

**2. WORK TO BE DONE**

- formatting the report
- Design a poster for the proposed system
- Prepare for project submission.

**3. PROBLEMS ENCOUNTERED**

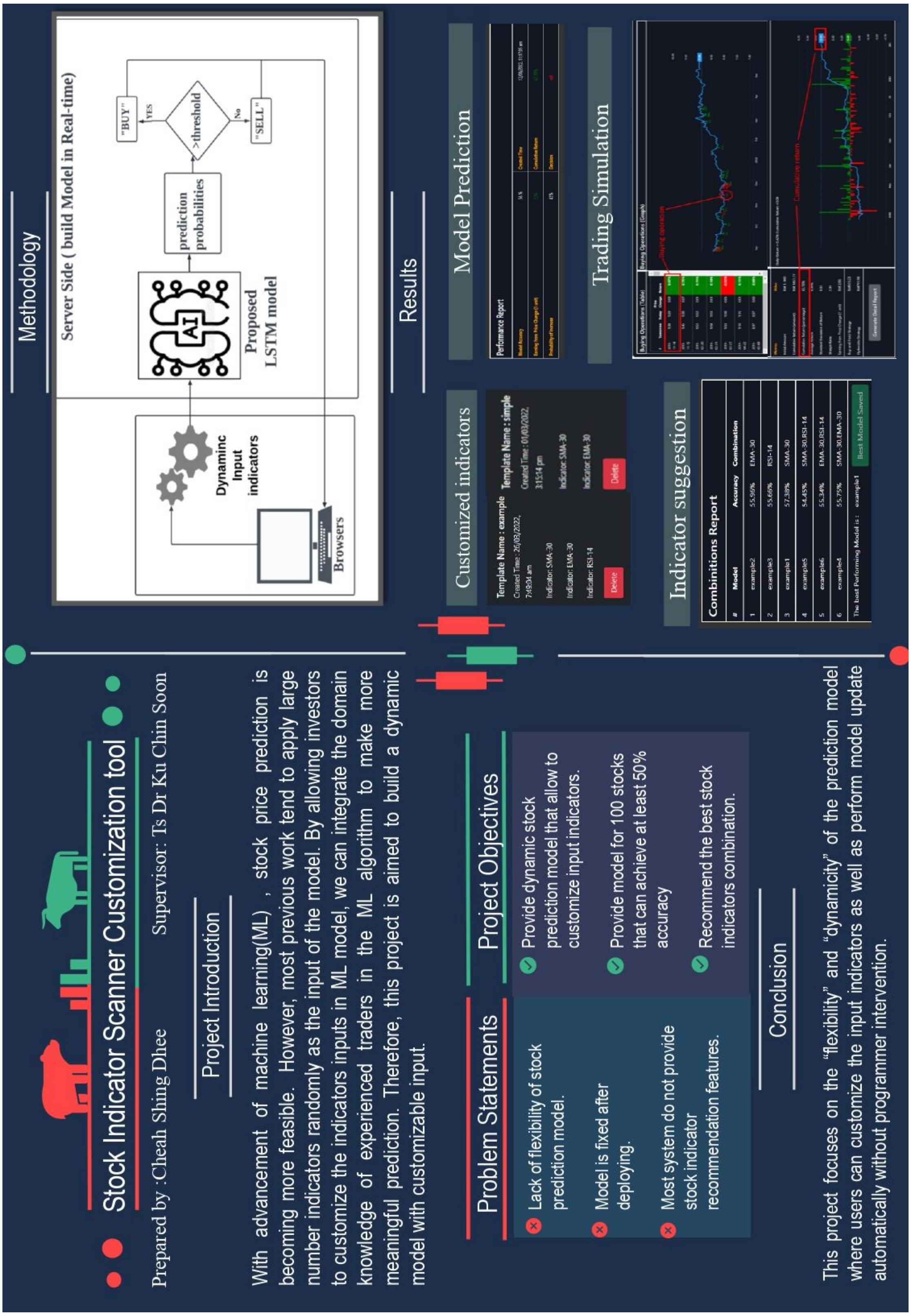
**4. SELF EVALUATION OF THE PROGRESS**

- Keep the progress in timeline.

Supervisor's signature

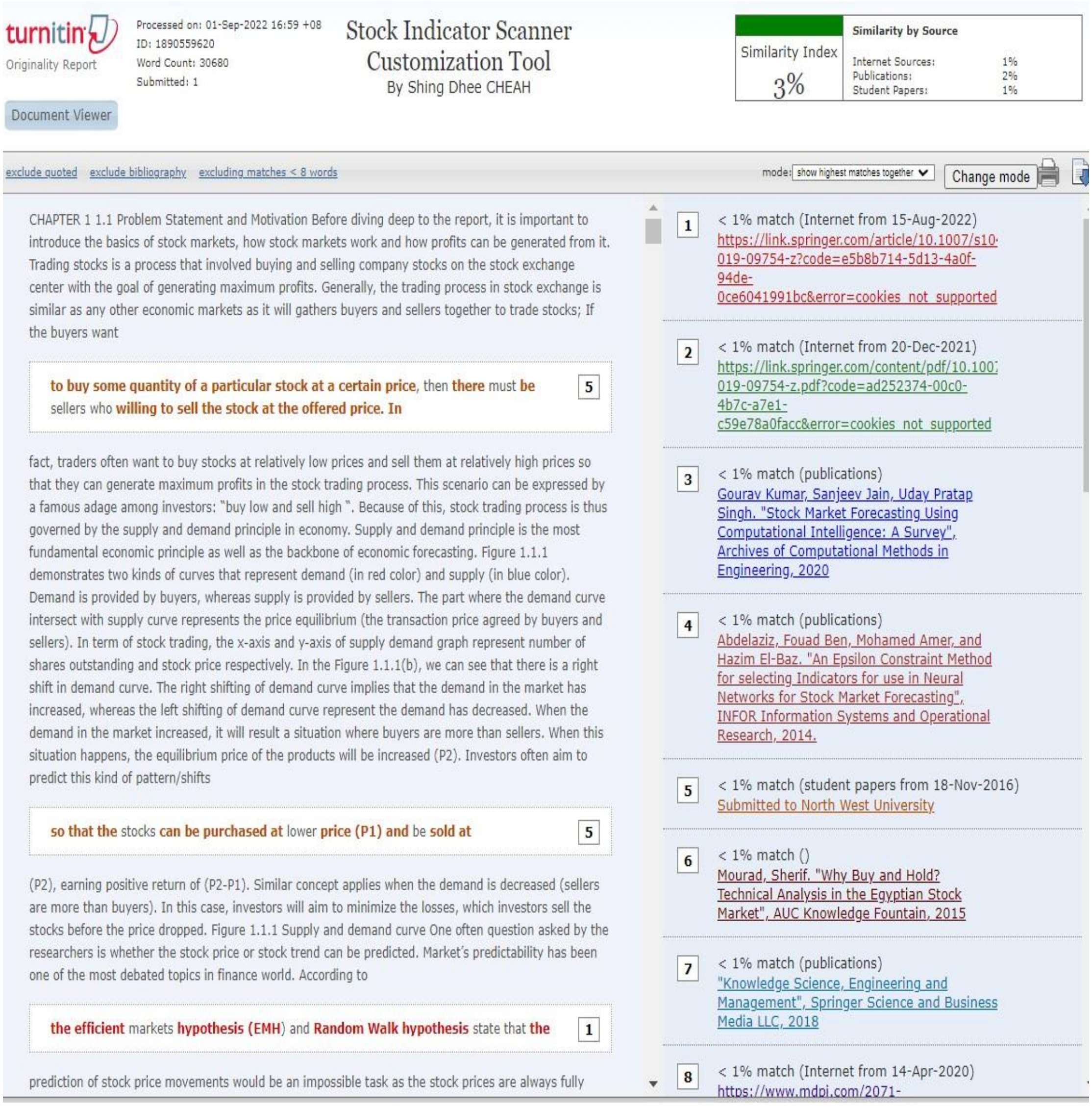
Student's signature

**POSTER**



Stock Indicator Scanner Customization tool
Prepared by :Cheah Shing Dhee
Supervisor: Ts Dr Ku Chin Soon
Project Introduction
With advancement of machine learning(ML) , stock price prediction is becoming more feasible. However, most previous work tend to apply large number indicators randomly as the input of the model. By allowing investors to customize the indicators inputs in ML model, we can integrate the domain knowledge of experienced traders in the ML algorithm to make more meaningful prediction. Therefore, this project is aimed to build a dynamic model with customizable input.
Problem Statements
Lack of flexibility of stock prediction model.
Model is fixed after deploying.
Most system do not provide stock indicator recommendation features.
Project Objectives
Provide dynamic stock prediction model that allow to customize input indicators.
Provide model for 100 stocks that can achieve at least 50% accuracy
Recommend the best stock indicators combination.
Conclusion
This project focuses on the "flexibility" and "dynamicity" of the prediction model where users can customize the input indicators as well as perform model update automatically without programmer intervention.
Methodology
Server Side ( build Model in Real-time)
Browsers
Dynaminc Input indicators
AI
Proposed LSTM model
prediction probabilities
>threshold
YES
No
"BUY"
"SELL"
Results
Customized indicators
Template Name : example
Template Name : simple
Delete
Indicator suggestion
Combinations Report
Model Prediction
Performance Report
Trading Simulation

# PLAGIARISM CHECK RESULT


turnitin Originality Report

Processed on: 01-Sep-2022 16:59 +08
ID: 1890559620
Word Count: 30680
Submitted: 1

Stock Indicator Scanner Customization Tool
By Shing Dhee CHEAH

Similarity Index 3%

Similarity by Source
Internet Sources: 1%
Publications: 2%
Student Papers: 1%

Document Viewer

exclude quoted exclude bibliography excluding matches < 8 words

mode: show highest matches together Change mode

CHAPTER 1 1.1 Problem Statement and Motivation Before diving deep to the report, it is important to introduce the basics of stock markets, how stock markets work and how profits can be generated from it. Trading stocks is a process that involved buying and selling company stocks on the stock exchange center with the goal of generating maximum profits. Generally, the trading process in stock exchange is similar as any other economic markets as it will gathers buyers and sellers together to trade stocks; If the buyers want

to buy some quantity of a particular stock at a certain price, then there must be sellers who willing to sell the stock at the offered price. In [5]

fact, traders often want to buy stocks at relatively low prices and sell them at relatively high prices so that they can generate maximum profits in the stock trading process. This scenario can be expressed by a famous adage among investors: "buy low and sell high ". Because of this, stock trading process is thus governed by the supply and demand principle in economy. Supply and demand principle is the most fundamental economic principle as well as the backbone of economic forecasting. Figure 1.1.1 demonstrates two kinds of curves that represent demand (in red color) and supply (in blue color). Demand is provided by buyers, whereas supply is provided by sellers. The part where the demand curve intersect with supply curve represents the price equilibrium (the transaction price agreed by buyers and sellers). In term of stock trading, the x-axis and y-axis of supply demand graph represent number of shares outstanding and stock price respectively. In the Figure 1.1.1(b), we can see that there is a right shift in demand curve. The right shifting of demand curve implies that the demand in the market has increased, whereas the left shifting of demand curve represent the demand has decreased. When the demand in the market increased, it will result a situation where buyers are more than sellers. When this situation happens, the equilibrium price of the products will be increased (P2). Investors often aim to predict this kind of pattern/shifts

so that the stocks can be purchased at lower price (P1) and be sold at [5]

(P2), earning positive return of (P2-P1). Similar concept applies when the demand is decreased (sellers are more than buyers). In this case, investors will aim to minimize the losses, which investors sell the stocks before the price dropped. Figure 1.1.1 Supply and demand curve One often question asked by the researchers is whether the stock price or stock trend can be predicted. Market's predictability has been one of the most debated topics in finance world. According to

the efficient markets hypothesis (EMH) and Random Walk hypothesis state that the [1]

prediction of stock price movements would be an impossible task as the stock prices are always fully

1 < 1% match (Internet from 15-Aug-2022)
https://link.springer.com/article/10.1007/s10-019-09754-z?code=e5b8b714-5d13-4a0f-94de-0ce6041991bc&error=cookies_not_supported

2 < 1% match (Internet from 20-Dec-2021)
https://link.springer.com/content/pdf/10.1007-019-09754-z.pdf?code=ad252374-00c0-4b7c-a7e1-c59e78a0facc&error=cookies_not_supported

3 < 1% match (publications)
Gourav Kumar, Sanjeev Jain, Uday Pratap Singh. "Stock Market Forecasting Using Computational Intelligence: A Survey", Archives of Computational Methods in Engineering, 2020

4 < 1% match (publications)
Abdelaziz, Fouad Ben, Mohamed Amer, and Hazim El-Baz. "An Epsilon Constraint Method for selecting Indicators for use in Neural Networks for Stock Market Forecasting", INFOR Information Systems and Operational Research, 2014.

5 < 1% match (student papers from 18-Nov-2016)
Submitted to North West University

6 < 1% match ()
Mourad, Sherif. "Why Buy and Hold? Technical Analysis in the Egyptian Stock Market", AUC Knowledge Fountain, 2015

7 < 1% match (publications)
"Knowledge Science, Engineering and Management", Springer Science and Business Media LLC, 2018

8 < 1% match (Internet from 14-Apr-2020)
https://www.mdpi.com/2071-

Plagiarism Check Result

| Universiti Tunku Abdul Rahman | | | |
|---|---|---|---|
| **Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)** | | | |
| Form Number: FM-IAD-005 | Rev No.: 0 | Effective Date: 01/10/2013 | Page No.: 1of 1 |

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

| | |
|---|---|
| **Full Name(s) of Candidate(s)** | CHEAH SHING DHEE |
| **ID Number(s)** | 18ACB05004 |
| **Programme / Course** | Bachelor of Computer Science (HONOURS) |
| **Title of Final Year Project** | Stock Indicator Scanner Customization Tool |

| **Similarity** | **Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)** |
|---|---|
| **Overall similarity index:** __3__ **%**<br><br>**Similarity by source**<br>Internet Sources: __1__ %<br>Publications: __2__ %<br>Student Papers: __1__ % | |
| **Number of individual sources listed** of more than 3% similarity: _0_ | |
| **Parameters of originality required, and limits approved by UTAR are as Follows:**<br>**(i) Overall similarity index is 20% and below, and**<br>**(ii) Matching of individual sources listed must be less than 3% each, and**<br>**(iii) Matching texts in continuous block must not exceed 8 words**<br>***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.*** | |

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

***Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.***

| | |
|---|---|
| Signature of Supervisor | Signature of Co-Supervisor |
| Name: __ Ts Dr Ku Chin Soon ____ | Name: __________________________ |
| Date: __5_September 2022_________ | Date: ____________________________ |

# UNIVERSITI TUNKU ABDUL RAHMAN

## FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

### CHECKLIST FOR FYP2 THESIS SUBMISSION

| | |
|---|---|
| Student Id | 18ACB05004 |
| Student Name | Cheah Shing Dhee |
| Supervisor Name | Ts Dr Ku Chin Soon |

| TICK (√) | DOCUMENT ITEMS<br>Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item. |
|---|---|
| | Front Plastic Cover (for hardcopy) |
| √ | Title Page |
| √ | Signed Report Status Declaration Form |
| √ | Signed FYP Thesis Submission Form |
| √ | Signed form of the Declaration of Originality |
| √ | Acknowledgement |
| √ | Abstract |
| √ | Table of Contents |
| √ | List of Figures (if applicable) |
| √ | List of Tables (if applicable) |
| √ | List of Symbols (if applicable) |
| √ | List of Abbreviations (if applicable) |
| √ | Chapters / Content |
| √ | Bibliography (or References) |
| √ | All references in bibliography are cited in the thesis, especially in the chapter of literature review |
| √ | Appendices (if applicable) |
| √ | Weekly Log |
| √ | Poster |
| √ | Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005) |
| √ | I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report. |

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

______________________________
(Signature of Student)
Date: 5 September 2022