Final Report

Stock Prediction Using Multivariate LSTM With Technical, Fundamental and Macroeconomic Indicators

Prepared by

Wong Ye Wern

(19032309)

BSc (Hons) Information Systems (Data Analytics)

Supervisor: Assoc. Prof. Dr Chua Hui Na

Date: 18 November, 2022

Contents

Abstract

The art of stock price prediction has been a difficult task for many researchers and analysts. In fact, stock price prediction has been a subject of interest among academicians and practitioners. Understanding the importance of technical, fundamental, and macroeconomic indicators in analysing stock performance and overall economy, this has led to the motivation of creating a multivariate LSTM model which utilises 2 fundamental, 5 technical, and 4 macroeconomic indicators as predictors to forecast the next day closing price of a chosen stock – Microsoft Corporation. The dataset constructed is the by-product of this project, it is uploaded and available to public on Kaggle. A stepwise regression model is implemented as input variable selection model to select variables that are important at predicting stock price at significance level of 95%. The regression model has rejected 2 macroeconomic indicators, Crude oil price and Gold price and 1 technical indicator, 10-days closing price standard deviation. The LSTM model consists of an input layer, a hidden LSTM layer and an output layer, hyperparameters being tuned in this project includes activation functions, number of LSTM units in a layer, optimizer's learning rate, batch size and timestep, resulting a total of 72 hyperparameter combinations. The model with 0.0081 RMSE which consists of 100 sequence length, tanh activation function for the LSTM layer, 0.001 learning rate of Adam optimizer, 200 LSTM units in the LSTM layer, batch size of 32 and dataset with variable selected by linear regression is identified to have the optimal performance in prediction.

Keywords: Long short-term memory (LSTM), artificial neural network (ANN), stock price prediction, Microsoft Corporation (MSFT), macroeconomic indicator, fundamental indicator, technical indicator

1. Introduction

Stock market is known as a public market where issuing, buying, and selling stocks that trade on a stock exchange is performed. In 2021, 56% of adults in the United States invested in the stock market [1]. Stock investment has become a norm as it not only helps in maximizing income but also protects wealth from taxes and inflation. Despite the attractive returns, there are risks associated with stock investment due to its dynamic, unpredictable, uncertain, volatile, and non-linear nature [2].

Technical analysis and fundamental analysis are the two main methods that finance practitioners subscribe to when making trading decisions and forecasting stock price. Technical analysis refers to the use of historical stock price and trading volume information in analysing and forecasting the stock trend while fundamental analysis studies and predicts the future stock price based on a detailed study of the business and the underlying factors that affects profitability, managerial, operation efficiency and many more of the company, as well as the industry and the condition of economy. Other than historical stock price, technical analysis also predicts stock price by utilizing indicators such as moving average, relative strength index and rate of change that can be derived from historical stock price. On the other hand, fundamental indicators are indicators calculated based on information obtained from company quarterly reports and annual reports. Price to earnings ratio, price to book ratio, return on assets and market capitalization are the examples of fundamental indicators.

The rise and fall of the stock price are affected by multiple factors such as geopolitics, macro and microeconomic, government policies, organization’s earnings, investor sentiment, demand, and supply etc. For instance, when the announcement of interest rate hike is released, investors would predict companies fail to meet expectations as the cost of borrowing increases which discourages firms from expanding and consumers are very likely to reduce spending and cut expenses, resulting in the market tumbling in anticipation. Changes in the factors mentioned above could make an impactful effect on the stock market. Hence, to generate profitable trading strategies, it is significant for investors to correlate the stock performance with the current market conditions and business news. Technical analysis on the stock market which identify opportunity based on stock historical price only is insufficient.

According to [3] macroeconomic indicators are a measurement that is used to judge the overall health of an economy and interpret current or future investment opportunities. Some of the common macroeconomic indicators include the Gross Domestic Product (GDP) of a country, currency index, inflation rate, interest rate, oil price, gold price and etcetera. A study conducted by researchers suggests that macroeconomic indicators can be used to predict stock market price fluctuations as bidirectional causality is seen between the macroeconomic indicators and stock market [4]. However, macroeconomic indicators such as inflation rate and interest rate are normally presented annually, or monthly and stock price is recorded daily, the difference in time interval causing them to be inappropriate for daily stock price prediction. The research conducted in [4] has suggested to use other indicators as a proxy of these macroeconomic indicators, for example, the oil price would be a good proxy of global inflation rate while the currency index is an alternative to represent consumer purchasing power of the country.

Over the past few decades, forecasting stock price has been a subject of interest among academicians and practitioners, it has driven the focus of financial institutions, organizations, individual traders and even computer scientists and data analysts who make predictions with machine learning (ML) algorithms. ML has become a powerful tool and has been widely used in the economic and financial industry for making better investment decisions. According to research work in related work section, stock prediction based on ML methods have proven to be both popular and successful. One of the ML techniques with relatively high accuracy in predicting stock price is the Long Short-Term Memory (LSTM) neural networks [5]. LSTM is one of the recurrent neural networks (RNN), it introduces the memory cell that replaces traditional artificial neurons in the hidden layer of the network. This grants LSTM the ability to remember important patterns and forget the irrelevant from the previous timestamp, resulting in it being one of the most suitable ML techniques in stock prediction [6].

Input variable selection acts as an essential part of ANN model development as a poor selection of input variable might impact negatively on the performance of ANN models. Difficulties in variable selection task might be due to the large number of variables available, correlations between potential input variables which can creates redundancy and low predictive power of the variables [69]. To overcome the difficulties of input variable selection, there are several widely used methods such as filtering variables by using genetic algorithms, recursive feature elimination, sequential feature selection like stepwise, forward, backward selection of regression. Besides, a simple chi-square or coefficient testing can also be performed to select features based on the threshold value. One of the related works of stock prediction with LSTM has applied linear regression coefficient ranking as feature selection method and the result shown a significant improvement on the model performance with variable selected by the linear regression as compared to the model using all the variables provided [59]. Hence, it is believed that use of feature selection techniques to remove features which do not contribute to the prediction of target variable will improve the accuracy of ANN models.

Problem statement: Based on the surveys performed in the related work of literature review section have shown that stock prediction with ML techniques focused on implementing technical indicators and has downplayed macroeconomic indicators and fundamental indicators. Looking at the predictors dimension of the models in the related work, it can be observed that historical stock price is the commonly used predictors. However, only one stock prediction model found to include the fundamental indicators as predictions and no stock prediction model predicts based on macroeconomic indicators. It is identified that there is a lack of studies that use macroeconomic indicators in stock prediction ML models. Understanding the importance of fundamental indicators in analysing the performance of stock and the usage of macroeconomic indicators in judging the overall health of the economy, this has then led to the motivation of creating a LSTM stock prediction model which implements fundamental, technical, and macroeconomic indicators as predictors.

Objective: This paper proposes to predict next day stock closing price by building a stock prediction model with LSTM and utilizing technical indicators, macroeconomic indicators, and fundamental indicators as predictors. Besides, a regression model will also be built to select predictors that are significant in predicting the stock price. Therefore, the same hyperparameter testing will be performed on both datasets, a dataset with all predictors and a dataset with the predictors selected by the regression model. The target and predictors of the model are listed clearly in the table below:

<table>
<tr><th colspan="3">Target</th></tr>
<tr><td colspan="3">Next day prediction on Microsoft Corporation closing price</td></tr>
<tr><th colspan="3">Time Intervals</th></tr>
<tr><td colspan="3">From 5/5/2002 until 5/5/2022 (20 years)</td></tr>
<tr><th colspan="3">Predictors</th></tr>
<tr><th>Fundamental Indicators</th><th>Technical Indicators</th><th>Macroeconomic Indicators</th></tr>
<tr><td>1. Price to Earnings Ratio (P/E)
2. Price to Earnings-to-Growth Ratio (PEG)</td><td>1. 10-days closing price moving average (MA10)
2. 10-days closing price standard deviation (STD10)
3. 10-days closing price Rate of Change (ROC)
4. 9-days closing price Relative Strength Index (RSI)
5. Yesterday’s Closing Price</td><td>1. Gold price per ounce in USD (GC=F)
2. Oil price per barrel in USD (CL=F)
3. U.S. 30-years Treasury Yield (^TYX)
4. U.S. Dollar Index (USDX Index)</td></tr>
</table>

Table 1: Target, predictors, and time interval of data of the proposed model

Proposed outcome: By using all three indicators listed above, it is believed that the stock prediction model would achieve a high performance with low Root Mean Square Error (RMSE).

Scope of work & Methodology: The scope of work for this project follows the standard methodology for data mining, CRISP-DM for building up the stock prediction model. First and foremost, the scope of work for this project will be performing literature review to acquire deep understanding on the neural network model, LSTM, fundamental, technical, macroeconomic indicators, and existing ML techniques of stock prediction. Secondly, collection of data from valid sources. Then, data exploration analysis is performed to investigate the patterns of data and identify data quality problems associated with the dataset. Treatment of the data quality will be carried out in the data preparation stage, along with the calculation of technical indicators, merging of dataset, splitting, and scaling of data to construct the final dataset for modelling. Next, stepwise regression will be performed on the dataset to select the significant predictors in forecasting the stock price. Hyperparameters combinations identified will be tested on both the datasets, namely, dataset with all predictors and dataset with predictors selected by regression. Comparison will be made on these models to identify the model with optimal performance in the modelling stage. The performance of models will be evaluated based on evaluation matrix. Interpretation of the evaluation matrix and comparison of the model result with the related work will be performed asl well. Lastly, the conclusion section would consist of summary of work, implication, limitation of the study and the future work. A final report of the project will be submitted, and presentation will be delivered at the end.

Timeline: This project was given a period of 28 weeks, the first 14 weeks focus on background research of the topic where comparison is made between several existing models of stock prediction, justification on the predictors used in the modelling, methodology and etcetera. Data collection, data preparation, data understanding, modelling and interpretation of the result will be performed in the second 14 weeks. A work plan section is created and all tasks to be performed during the project will follow strictly on the work plan to ensure tasks to be completed within the given timeframe.

2. Literature Review

In this section, literature review on ANN, RNN, and LSTM are discussed, along with fundamental indicators, technical indicators, macroeconomic indicators, evaluation matrix and lastly, related work of stock prediction with neural networks. LSTM is a variant of RNN while RNN is a variant of ANN, resulting there is a need to understanding ANN and RNN before discussing about LSTM.

2.1 Artificial Neural Network (ANN)

The Artificial Neural Network is a computational model that is inspired by the way the biological nervous system, brain processes information [7]. The first artificial neuron or node was proposed by McCulloch and Pitts in 1943 [7]. It is a subset of machine learning and the core of deep learning [8]. ANNs are broadly used in the medical field, financial trading, logistics and even space exploration. One of the applications of ANN includes the classification of brain tumour types by analysing the Magnetic Resonance image and Magnetic Resonance Spectroscopy data of the patients.

Artificial neural networks are organized in multiple layers and each layer is made up of several interconnected artificial neurons or nodes. Information is received at the input layer of ANN and weight is assigned to respective input, information is then sent to the hidden layer(s) where these signals pass through an aggregation function and activation function to produce an output at the output layer [9]. Each artificial neuron is connected to one another in an ANN and has an associated weight and threshold [8]. If the output of a node is above the specific threshold value, the node is activated, and data is sent to the next layer of the network, otherwise, no data will be passed to the next year [8].

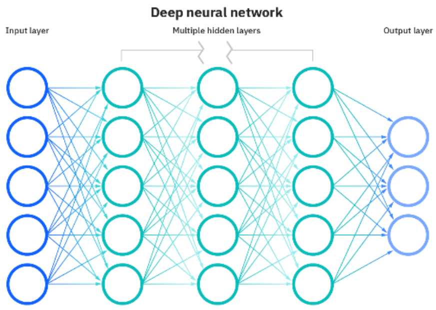

Figure 1: Artificial deep neural network
Source: [8]

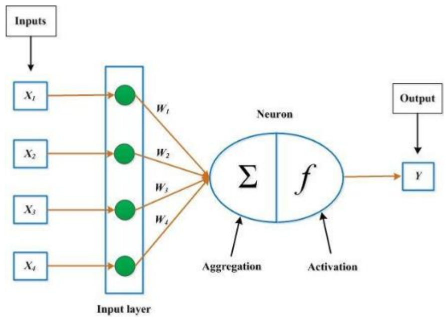

Figure 2: Artificial neural network with a hidden layer that has only one neuron
Source: [9]

The aggregation function in each artificial neuron can be treated as its own linear regression model composed of input data, weights, bias, and an output [8]. Weights determine the degree of influence of the given variable, the greater the value of weight, the greater the contribution to the output compared to other inputs [8]. All inputs are multiplied by their respective weights, the product is then summed and passed to an activation function [8].

Activation function is applied to transform the output signal. A neuron without activation function works the same as a linear regression model with limited performance [10]. Many real-world scenarios such as the stock market price have a non-linear nature and cannot be predicted with linear regression alone. Therefore, activation function which introduces non-linearity to the neural network has given the neural network ability to make sense of non-linear and complicated mappings between the nodes and further improve the performance of the model [10]. The formula of aggregation function and several commonly used activation functions are listed below.

$$Aggregation\ Function:\quad Weighted\ Sum = \sum_{i=1}^{m} w_i x_i + b$$

$$Activation\ Functions:$$

$$Sigmoid:\ f(x) = \frac{1}{1+e^{-x}}$$

$$Tanh: f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$ReLU\ (0,x): f(x) = \begin{cases} 0\ for\ x < 0 \\ x\ for\ x => 0 \end{cases}$$

Different activation functions have different usage [10, 11], which activation function to be used in the neural network depends on the range of input data, error and speed [11]. For instance, sigmoid function is suitable for classification problems but not vanishing gradient problem while ReLU function is the most widely used function and proven to perform better than other activation functions in most cases [10].

The accuracy of the model is evaluated using a cost function which is also known as loss function. Loss function tells how far away the combination of weights and biases is from the optimal solution [11]. The objective of ANN in the training phase is to minimize the cost function [11]. Some of the commonly used cost functions are mean squared error (MSE), root mean squared error (RMSE), mean absolute errors (MAE) and etcetera. The ANN model adjusts its weights and biases in every iteration or epoch by computing the derivative of cost function with respective input weights and biases [11]. Derivative measures the change of a function's output with respect to its input variable weight. The derivative of the cost function tells how much the cost function is changing given a change in the input variable weight or bias. In each epoch, ANN converges towards the local minimum of cost function where the difference between actual and predicted value is minimum using gradient descent technique [11, 7]. Model which trains by the above method is classified as back-propagation ANN where the error value is propagated backward within a network and minimizes the error [12, 11]. New weights and biases for the next epoch are calculated based on the formula below:

$$NW_i = W_i - LR * \frac{\delta L}{\delta W_i} \qquad \begin{aligned} NW_i &= New\ weight\ for\ the\ next\ epoch \\ W_i &= Weight\ of\ current\ epoch \\ LR &= Learning\ rate \\ C &= Value\ of\ Loss\ function \end{aligned}$$

$\frac{\delta L}{\delta W_i}$ can be interpreted as the change in cost function when there is a change in W_i. Learning rate determines the step size at each iteration while adjusting its weights and biases towards the local minimum of the lost function [13]. Defining the learning rate is especially significant as inappropriate learning rate can lead to model with poor performance [14].

The gradient descent technique has two main problems which is the exploding and vanishing gradient problem. Vanishing gradient occurs when $\left|\frac{\delta L}{\delta W_i}\right| \approx 0$ while exploding gradient occurs when $\left|\frac{\delta L}{\delta W_i}\right| \approx \infty$ [15]. Vanishing gradient and exploding gradient problem results the adjustment of weights and biases too small to be meaningful and too large to be precise in converging to the local minimum of cost function [15]. The vanishing gradient problem is mainly caused by the Sigmoid and Tanh activation function while the exploding gradient occurs when the value of W_i is extremely large.

Although applications of ANN have been very successful, there is problems associated with ANN in handling time-series data. The traditional ANN assume that inputs and outputs are independent of each other and does not consider any possible temporal correlation of the input and output data [16, 8]. Consider a Natural Language Processing (NLP) application in which every word in the sentence represents input of the network. These inputs are not useful in isolation but only in the context of what occurred before them [17]. To overcome the limitation of ANN in solving time-series problems, the Recurrent Neural Network (RNN) is introduced.

2.2 Recurrent Neural Network (RNN)

RNN is one of the neural network architectures which is widely used to detect patterns in sequential data or time series data [18]. It was introduced in the 1980s and its ability to store and remember past inputs opened new problem domains to neural networks [17]. The RNN is commonly implemented for solving ordinal or temporal problems such as NLP, language translation, image captioning and speech recognition [19]. Popular applications of RNN are Google translate and Siri [19].

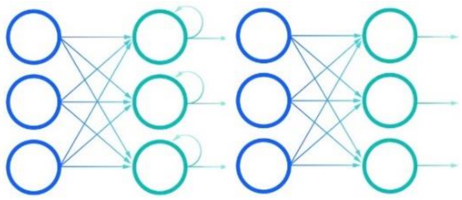

Figure 3: RNN and Feed-forward ANN
Source: [19]

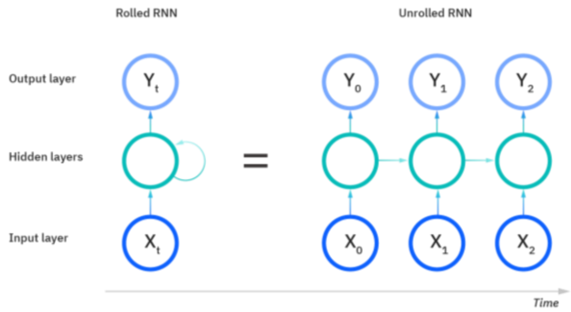

Figure 4: Rolled and Unrolled RNN
Source: [19]

The main difference between RNN and traditional ANN is that RNN architecture is a cyclic connection which allows RNN to update the current state based on the past states and current input data [20], which means information from previous timestep is used as input for next timestep. Based on figure 3, RNN on the left has a loop, it maintains internal memory with feedback by applying hidden layer output back into hidden layer while the traditional feed-forward ANN on the right passes information through the network without cycles [19, 18]. Besides, instead of using backpropagation algorithm, RNN uses backpropagation through time (BPTT) algorithm to determine gradient [19]. The principle of BPTT algorithm is similar to the backpropagation algorithm where the adjust weights and biases during the training phase by calculating the derivative of cost function to respective input variable weights and biases [19]. The main difference between BPTT and traditional backpropagation algorithm is that BPTT algorithm calculates errors at each timestep whereas the traditional backpropagation algorithm in feedforward neural network only calculates errors at the end of epoch [19].

Similarly, RNNs do face vanishing and exploding gradient problems like ANN. The vanishing and exploding gradient problem causes RNNs unable to connect relevant information from previous input when the gap between them is too large, resulting in it being unable to handle long-term dependencies [20]. This leads to the motivation of LSTM to overcome the long-term dependencies problem [20].

2.3 Long Short-Term Memory

LSTM is a type of the RNNs which focuses on solving vanishing and exploding gradient problem as discussed above. The LSTM model was created by Hochreiter and Schmidhuber in 1997, the innovation solution dubbed the LSTM model to address the problem of handling long-term dependencies. According to [20], LSTM has been the core focus of deep learning due to its powerful learning ability. It has also successfully achieved almost all existing results based on RNNs [20]. The LSTM models work tremendously well and have been widely applied in various areas such as speech recognition, acoustic modelling, sentence embedding, trajectory prediction and many more [20].

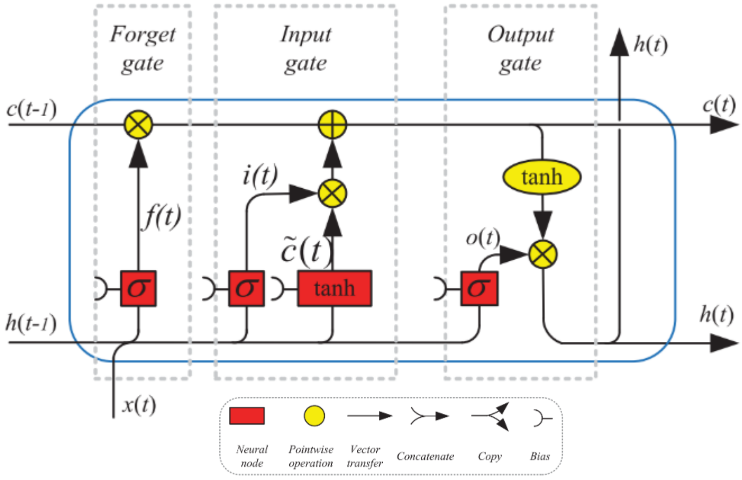

Figure 5: A comparison between RNN cell (left) and LSTM cell (right)
Source: [20]

Based on the figure 5 above, the internal composite of the memory block consists of cell state, hidden state, forget gate, input gate and output gate which are the five main components in the LSTM cell [21]. Cell state, c is the horizontal line which runs across the entire chain, and it refers to the key component in LSTM that allows the model to store longer memory of past events while the gates are important in filtering information and determining what information should be stored or removed from the cell state [22]. Hidden state, h refers to the output in a particular point of time. Output of previous hidden steps, cell state and the current input data are extremely important elements in producing the output of LSTM at the current point of time. A detailed explanation on each stage of the LSTM model will be provided in the section below.

Firstly, the memory cell receives information from the previous hidden state, $h(t-1)$ and input data of current timestep, $x(t)$ which is then passed through the forget gate [23]. The forget gate determines useful information that requires attention and redundant information that can be removed from the cell state [22]. The $h(t-1)$ *and* $x(t)$ are then passed through the Sigmoid function, σ which its value ranges from 0 to 1 where 0 indicates "remove everything" and 1 indicates "remember everything" [22]. The output of σ, $f(t)$ will then be sent to the cell state and pointwise multiplied by the previous cell state, $c(t-1)$.

Secondly, in the input gate, the same component, $h(t-1)$ *and* $x(t)$, are passed through the Hyperbolic Tangent function, *tanh* [22] to generate a 'new memory update vector'. The vector $\tilde{c}(t)$ ranges from -1 to 1. However, the vector is generated without considering the usefulness of information from $h(t-1)$ *and* $x(t)$. Hence, the $h(t-1)$ *and* $x(t)$ are passed through the second σ to generate the output $i(t)$ which ranges from 0 to 1 [23] to identify what information worth retaining and updating the 'new memory vector' [22]. The output of the input gate, which is the output of second σ, $i(t)$ and the output of tanh, $\tilde{c}(t)$ are then sent to the cell state for point-by-point multiplication [24]. At this stage, the cell state has received information from the forget gate, $f(t)$ and input gate, point-by-point multiplication of $i(t)$ and $\tilde{c}(t)$. The network has obtained enough information to update the cell state by determining what information needs to be stored and removed from the cell state. The previous cell state, $c(t\text{-}1)$ will be multiplied with the output of forget gate, $f(t)$. If the product of multiplication is 0, the values will be dropped from the cell state, else the values will be sent forward for point-by-point addition [24]. The point-by-point addition is performed between the outcome of multiplication of $c(t\text{-}1)$ and $f(t)$ and the outcome of point-by-point multiplication of $i(t)$ and $\tilde{c}(t)$. The outcome of this point-by-point addition updates the cell state and the new cell state, $c(t)$ is obtained.

Lastly, at the output gate, LSTM identifies the new value of the next hidden state $h(t)$. The third σ receives information from the previous hidden state, $h(t\text{-}1)$ and input data of current timestep, $x(t)$ to determine the useful and useless information for output [23]. Other than that, the updated cell state $c(t)$ is passed to the *tanh* function. The output of σ and *tanh* in the output gate is then multiplied point-by-point [24]. The output of the point-by-point multiplication is the new value of the next hidden state, $h(t)$ which is the information being carried over to the next time step for calculation [24].

Features of various gates [25]**:**	**Usage of activation functions** [21]**:**
• **Forget gate**: define which information to be removed from the cell state • **Input gate:** specify which information to be added into the cell state • **Output gate:** determine which information from the cell state is used as the output	• **Sigmoid function (σ)**: output of sigmoid ranges between 0 to 1 which describes how much of each component should be let through to the cell state (1: remember, 0: forget) • **Hyperbolic Tangent function (*tanh*)**: output of tanh ranges between -1 to 1. It generates a new vector which will be added to the cell state.

Table 2: A simple description on the features of LSTM gates and activation functions

2.4 Fundamental Indicators

Fundamental indicators are indicators used in fundamental analysis which can be derived from a company's financial statement [26]. Financial ratios are not only used to understand the strengths and weaknesses of companies but can also create early warning to deteriorate the company's financial condition [27]. Combination of these financial ratios provides a better view of a stock's worth than analysing a single ratio alone [28]. This section explains the financial ratios, namely Price to Earnings ratio and Price to Earnings-to-Growth ratio.

a. Price to Earnings Ratio (P/E)

$$P/E = \frac{Stock\ price}{EPS} \qquad \begin{array}{l} Where, \\ EPS = Earnings\ per\ share \end{array}$$

P/E, the ratios of share prices to earnings, is one of the financial ratios used in fundamental analysis, it can be obtained by dividing the market price by earnings per share (EPS) while EPS is the company's net earnings by the numbers of outstanding shares of the stock [28]. There are several methods of extracting EPS, this project implements the 12-month trailing EPS which is extracted from the recent 4 quarters. Based on the formula above, P/E increases when stock price increases or EPS decreases while P/E decreases when EPS increases or stock price decreases. However, when a company is not making profit, P/E will not be available as there is no earning and P/E ratio cannot be used for evaluation [29].

It is the most used valuation tool in the stock market [30] to determine if individual stock is reasonably priced, undervalued, or overvalued [31]. P/E ratios can also be used as a measure of relative value when comparing the peer companies which are in the same line of business [30]. Company with a higher P/E ratio indicates that investors are convinced to pay high for that share as investors are expecting higher growth in the coming future [28]. Besides, if stocks are perceived by investors to be less risky, demand for stocks will increase, resulting in higher stock price and higher P/E ratios [31]. Based on historical records of the last 127 years, the P/E ratio tends to rise when the P/E ratio has fallen below its long-term average and tends to fall back to its long-term average when it has risen well above its long-term average [31]. The article [31]concluded that a stock with high P/E ratio is a sign that the stock may be headed for a downturn.

b. Price to Earnings-to-Growth Ratio (PEG)

$$PEG = \frac{P/E}{Annual\ EPS\ Growth} \qquad Annual\ EPS\ Growth = \frac{Current\ EPS}{Last\ Year\ EPS} - 1$$

The Price to Earnings-to-Growth ratio is the P/E ratio of a company by the year-on-year growth rate of its earnings [28]. PEG ratio can be calculated based on past EPS growth rate or the analyst's estimation on future EPS growth rate [32]. This project focuses on PEG ratio calculated based on past EPS growth rate. Based on the formula provided above, the PEG ratio can be either positive or negative. Negative value is obtained when last year EPS is greater than current EPS.

Like P/E ratio, PEG ratio is also used for valuation, and it is widely employed to indicate a stock's possible value [32]. If PEG readings less than one, it is said that the stock is undervalued, whereas readings greater than two represent that the stock is overvalued [33]. Likewise, the stock is considered fairly valued when its PEG ratio is between one and two [33]. Based on a study conducted [33], stock with PEG ratio of 0 to 0.99 have an average return of 225.2% in 3 years' time while stocks with PEG ratio greater than 2 has only 47%. In other words, it can be said that low PEG readings promise a better stock return in future and low future stock return is expected for stocks with high PEG readings [34]. Besides, PEG ratio could also offer suggestions on whether a company's high P/E ratio reflects an excessively high stock price or is a reflection of promising growth prospects for the company [32].

2.5 Technical Indicators

Technical indicators are indicators that can be derived from the calculation of historical stock prices and trading volume information [26] .This section discusses the technical indicators used in this project which are 10-days moving average (MA10), 10-days standard deviation (STD10), 10-days Rate of Change (ROC) and 9-days Relative Strength Index (RSI).

a. 10-days moving average (MA10)

$$MA10 = \frac{\sum_{i=1}^{n} P_i}{n} \qquad \begin{aligned} &Where, \\ &n = number\ of\ days\ in\ the\ moving\ average \\ &P_i = Stock\ price\ of\ the\ day \end{aligned}$$

Above shows the formula of the MA10, where n will be 10 as the number of days in the moving average will be 10 in this project. The first MA10 value can only be obtained on the 10th trading day as it requires 9 previous days of stock price and today stock price for the calculation of MA10. A simple illustration for the calculation of 3-days moving average is shown below:

Day	Stock price	3-days moving average
1	100	N/A
2	103	N/A
3	107	(100+103+107)/3 = 103.3
4	99	(103+107+99)/3 = 103

Table 3: calculation of 3-days moving average

Moving average is one of the many techniques used to smooth and remove noise from time series data [35]. The moving average data transformation is commonly used in the finance industry, it can be used for trend forecasting [36, 37] traders buy or sell the stock depending on the direction of the moving average graph of stock market data [38]. There are different lengths of moving averages such as 7 days, 10 days, 20 days, 50 days and even up to 200 days. Different lengths of moving average is used to measure different lengths of trends [39]. The 200-days moving average is often considered a long-term indicator and is usually used to measure long term trends while the 10 or 20-days moving average is used for short term trends [39]. The shorter the length of the moving average, the more reactionary it becomes while the longer the time period of the moving average, the less sensitive it is to the price changes [28].

According to a related work which performs stock prediction using ANN integrated moving average [40], transforming stock price using moving average method is proven to increase the performance of Neural Network in stock prediction. Hence, 10-days moving average is chosen to be one of the predictors in this project.

b. 10-days standard deviation (STD10)

$$STD10 = \sqrt{\frac{\sum_{i=k-m+1}^{k}(P_i - \bar{P})^2}{m-1}}$$

$$k = 1, 2, 3, \dots, n$$

$Where,$
$P_i = Stock\ price\ of\ the\ day$
$\bar{P} = Average\ stock\ price\ of\ the\ window\ size$
$m = Window\ size$
$n = Total\ number\ of\ rows\ in\ the\ sample$

In statistics, standard deviation is a common measurement of the amount of dispersion of a set of values [41]. A low standard deviation indicates that the sample has a low dispersion, and the values are closer to the mean while a high standard deviation indicates that the sample has a high dispersion, and the values are spread out over a wider range from mean. Standard deviation is often used for data exploration to discover the structure of the sample data. Similar to the 10-days moving average discussed above, the 10-days moving standard deviation is calculated based on 10 consecutive days of stock price and it would be only available on the 10th trading day. Standard deviation calculates the dispersion of the whole sample data while the moving standard deviation only calculates the standard deviation of the window size, in this project, the window size would be 10 days.

Moving standard deviation is used as a statistical measurement of market volatility [42]. High value of moving standard deviation is obtained when the stock market price changes dramatically and low value of moving standard deviation represents a stable market [42].

c. 10-days Price Rate of Change (ROC)

$$ROC = \frac{P_i - P_n}{P_n} * 100 \qquad \begin{array}{l} Where, \\ P_i = Stock\ price\ today \\ P_n = Stock\ price\ n\ days\ ago \end{array}$$

Based on the formula for ROC above, the minimum value of ROC is 0 when the value of P_i and P_n are the same. The ROC can be either positive or negative. Positive value of ROC is obtained when P_i is larger than P_n while negative value of ROC is obtained when P_i is smaller than P_n. The stock price is constantly increasing when the ROC remains positive. Conversely, ROC would be negative when the stock price is falling [43].

ROC is a pure momentum oscillator that measures the percentage of change in stock price from one period to next [43], in this project, it would be comparing today stock price and stock price 10 days ago. Like the Moving Average technique, ROC is also one of market energy indicators used in technical analysis to determine the key turning point of the stock market [44]. It is used to judge the strength of buying and selling power [44]. It can also signal overbought and oversold conditions [43]. When the ROC indicator is at high peak and begins to move down generates a selling signal while a ROC indicator at low peak and begins to increase indicates a buying signal [45].

d. 9-days Relative Strength Index (RSI)

$$RSI = 100 - \left[\frac{100}{1 + RS}\right] \qquad RS = \frac{Average\ gain\ in\ n\ days}{Average\ loss\ in\ n\ days}$$

The RSI is a momentum oscillator that measures the speed and change of price movements [46]. It is a popular method used to forecast future market trends by analysing the market's intentions and strengths by comparing the average closing gains and average closing loss over a given period to determine future market trends [44]. A simple illustration on the calculation of 3 days-RSI is shown below:

Day	Stock price	Gain	Loss
1	100	N/A	N/A
2	103	103-100 = 3	0
3	107	107-103 = 4	0
4	99	0	107-99 = 8

Average gain in 3 days: (3+4+0)/3 = 2.3
Average loss in 3 days: (0+0+8)/3 = 2.6
RS on 4th day = 2.3/2.6 = 0.884
RSI on 4th day = 100 – 100/(1+0.884) = 46.92

Table 4: Calculation of 3-days RSI

The RSI has a minimum value of 0 and maximum value of 100, with low and high levels generally marked at 30 and 70 or more extreme high and low levels marked at 80 and 20 or 90 and 10 [43]. RSI readings greater than the high level are considered in overbought territory while RSI readings lower than the low level marked are considered in oversold territory [43]. Hence, it is best to sell when the RSI value is above 70 and best to buy when it is below 30 [46]. The suggested number of periods is 14 days [46] .However, the number of periods varies for short-, medium- and long-term trading [46]. Generally, 9-days RSI are used for short term trading, 14-days RSI is used for medium term trading, 56-days, 100-days, and 200-days RSI are normally used for long term trading [46].

2.6 Macroeconomic indicators.

According to [3], macroeconomic indicators are measurements used to judge the overall health of an economy and interpret current or future investment opportunities. This section will further discuss the selected macroeconomic indicators, namely gold price, oil price, U.S. Treasury Yield and U.S. dollar index and their role in the economic environment.

a. Gold price

Throughout the centuries, gold was one of the most precious and valuable commodities. Late back in the 4000 B.C., gold was used as decoration and adornment for the wealthy ancients and have started to be used as coin by Lydians in 635 B.C. [47]. Since then, gold has been looked upon as money, as an investment and as a source of value.

Gold has been an essential component in the financial reserves of nations for centuries. Gold holdings within central banks often play a stabilizing role, especially in times of crisis [48]. It is also known as a 'safe haven' in the financial market [49] as diversifying investment portfolios with higher gold holdings may help in providing cushion against future macroeconomic shocks [48]. The demand and price of gold during times of uncertainty tend to increase as investors might use gold as havens to ride out uncertain times such as stock market crashes and especially when the confidence in the local currency is low.

b. Oil Price

Crude oil being the primary fuel source is one of the most important commodities in the world. Petroleum products can be found in everything from lip balm, fuel for transportation, fertilizer and even in the solar panels. Changes of the oil price are felt around the world, it is often considered as a significant factor for understanding inflation and fluctuations in stock prices [50].

Although study has proven that stock returns cannot be explained based on the oil price alone [50], but increase in oil price induces to higher exportation cost which will then pass-through into domestic price levels and indirectly increases inflation [51]. Fluctuation in the inflation rate or rate of increase in prices may further lead to economic changes which will affect the overall economic performance of the country [51]. Hence, crude oil price will be used as a proxy of inflation rate in this study for forecasting stock price.

c. U.S. Treasury Yield (30 years)

According to [4], the treasury yield is a useful tool to characterize risks and opportunities in the macro economy. It is also used as a proxy for interest rate in the research conducted in [4]. Since the focus of this paper is the U.S. stock market, the U.S. Treasury Yield will be discussed.

The U.S. Treasury Bonds which pay periodic interest payments based on its respective coupon rate are the U.S. government debt securities issued by the country Federal government while the U.S. Treasury yield refers to the return on investment on the

U.S. Treasury Bonds [52]. Since the periodic interest of Treasury Bonds is fixed, the treasury yield is inversely proportional to price of Treasury Bonds.

According to [53], the treasury bond prices are inversely proportional to interest rate, when Interest rate increases, bond price decreases and treasury yield increase, this phenomenon is known as interest rate risk. This scenario can be interpreted as investors are more confident in the market and believe return outside of Treasury securities will be higher than the treasury yield. Bonds with longer maturities generally present greater interest rate risks than the shorter maturities [53]. Hence, the U.S. 30-years Treasury Yield is selected to be one of the predictors in this study.

d. U.S. Dollar Index

In 1973, the U.S. Dollar Index was created by the U.S. Federal Reserve to provide an external bilateral trade-weighted average value of the U.S. dollar, which is calculated based on six currencies, namely Euro (EUR), Japanese Yen (JPY), British Pound (GBP), Canadian Dollar (CAD), Swedish Krona (SEK) and Swiss Franc (CHF) [54].

The Standard & Poor's 500 index, or S&P 500 in short track the value of top 500 corporations listed on stock exchanges in the United States and S&P 500 corporations generates about 40% of their revenue outside of the United States [55]. According to the report [56], given in the period of uneven global economic recovery from the COVID-19 pandemic, S&P500 companies which has a higher international exposure obtained a higher revenue growth than S&P500 companies which has a lower international exposure.

Exchange rate plays an important role in international business, it directly impacts the revenue of a company that trades internationally. For instance, a weak dollar increases importation and exportation costs which will then decrease the profit margins of companies in the United States [57]. A study which investigates the relationship between U.S. stock returns and the U.S. dollar found that the U.S. stocks returns were 2.5 times higher when the dollar was trending up during accommodative monetary policy as compared to when the dollar was depreciating due to the Federal Reserve tightening [57]. Strong U.S. dollar also occurred with strong economic growth due to consumer spending [57]. Therefore, the U.S. Dollar index which reflects the monetary policy of the United States should be analysed for accurate stock prediction.

2.7 Evaluation Matric, Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{Y_i} - Y_i)^2}$$

$Where,$
$n = number\ of\ observation$
$\hat{Y_i} = predicted\ value$
$Y_i = observation\ value$

In this project, RMSE will be used to evaluate the performance of models. RMSE refers to the square root of the mean of the square of all the errors, it is one of the widely used and excellent general purpose error metrics for numerical prediction [58]. It is also suggested that RMSE is relatively better than Mean Absolute Error (MAE) as RMSE amplifies and severely punishes large errors while MAE treats all errors equally [6]

2.8 Related work

A comparison table is provided in this section to compare some of the existing neural network models in stock prediction since 2017. The table includes stock or index of prediction, ML method, target, predictors, and the model's performance. Observations on the related work and the motivation on the proposed are stated in this section as well.

<table>
<tr><th></th><th>STOCK/INDEX</th><th>METHOD</th><th>TARGET</th><th>PREDICTORS</th><th>PERFORMANCE</th></tr>
<tr><td>[2]</td><td>GS, JNJ, JPM, NKE, PFE</td><td>ANN</td><td>Next day closing price</td><td>1. Stock high minus low price
2. Stock close minus open price
3. Stock 7-days moving average
4. Stock 14-days moving average
5. Stock 21-days moving average
6. Stock 7-days standard deviation</td><td>RMSE: 0.42
MAPE: 0.77
MBE: 0.013</td></tr>
<tr><td>[59]</td><td>AAPL</td><td>Multistep Output Long-Short-Term-Memory (MMLSTM)</td><td>One week prediction on the stock closing price</td><td>1. Level of interest in topic related to the company from google trend
2. Sentiment of Tweets
3. Number of comments on news
4. Sentiment of news headline</td><td>Mean MAPE%: 13.547%</td></tr>
<tr><td>[6]</td><td>NIFTY 50</td><td>LSTM</td><td>Future stock daily return</td><td>1. Stock Highest point of the day
2. Stock Lowest point of the day
3. Stock opening price of the day
4. Stock closing price of the day</td><td>RMSE: 0.00859</td></tr>
<tr><td>[40]</td><td>IDX: ANTM</td><td>Binary Sigmoid ANN</td><td>Future stock closing price</td><td>1. Stock 5-days moving average</td><td>RMSE: 0.004</td></tr>
<tr><td>[44]</td><td>SHE: 000538</td><td>Principal component analysis ANN</td><td>Future stock closing price</td><td>1. 29 technical indicators (RSI, ROC, moving average, opening price, closing price, trading volume, price change & etc.)</td><td>MSE: 6.63
MAE: 1.98</td></tr>
<tr><td>[60]</td><td>Top 10 stocks in S&P 500 index</td><td>LSTM,
Stacked LSTM,
Attention LSTM</td><td>Future stock closing price</td><td>1. Trading volume
2. Debt-to-equity Ratio
3. Return on Equity
4. Price-to-Book ratio
5. Profit margin
6. Diluted earnings per share
7. Company beta</td><td><table><tr><th></th><th>RMSE</th></tr><tr><td>LSTM</td><td>0.00012</td></tr><tr><td>Stacked LSTM</td><td>0.00020</td></tr><tr><td>Attention LSTM</td><td>0.00030</td></tr></table></td></tr>
<tr><td>[25]</td><td>KOSPI</td><td>GA-LSTM</td><td>Future stock closing price</td><td>1. Stock Highest point of the day
2. Stock Lowest point of the day
3. Stock opening price of the day
4. Stock closing price of the day
5. Traded volume
6. Stock 10-days moving average
7. Stock weighted 10-days moving average
8. Relative strength index
9. Stochastic K%
10. Stochastic D%</td><td>MSE: 181.99
MAE: 10.21
MAPE: 0.91</td></tr>
<tr><td>[61]</td><td>SSE Composite Index, SNP</td><td>Embedded layer LSTM</td><td>Future stock closing price</td><td>1. Stock Highest point of the day
2. Stock Lowest point of the day
3. Stock opening price of the day
4. Stock closing price of the day
5. Traded volume
6. Daily amplitude
7. 5-days amplitude
8. 10-days amplitude
9. Amplitude of volume fluctuation</td><td><table><tr><th></th><th>MSE</th><th>Accuracy</th></tr><tr><td>SSE</td><td>0.017</td><td>57.20%</td></tr><tr><td>SNP</td><td>0.0019</td><td>52.40%</td></tr></table></td></tr>
<tr><td>[62]</td><td>GOOGL</td><td>LSTM,
Bidirectional LSTM</td><td>Future stock closing price</td><td>1. Stock Highest point of the day
2. Stock Lowest point of the day
3. Stock opening price of the day
4. Stock closing price of the day</td><td><table><tr><th></th><th>RMSE</th></tr><tr><td>LSTM</td><td>0.0004127</td></tr><tr><td>BI-LSTM</td><td>0.0002421</td></tr></table></td></tr>
<tr><td>[63]</td><td>GOOGL, NIKE</td><td>LSTM</td><td>Future stock closing price</td><td>1. Stock opening price of the day</td><td><table><tr><th></th><th>RMSE</th></tr><tr><td>GOOGL</td><td>0.000497</td></tr><tr><td>NIKE</td><td>0.000874</td></tr></table></td></tr>
<tr><td>[64]</td><td>AAPL</td><td>LSTM, S_LSTM,
S_AM_LSTM,
S_EMDAM_LSTM</td><td>Future stock closing price</td><td>1. Stock Highest point of the day
2. Stock Lowest point of the day
3. Stock opening price of the day
4. Stock closing price of the day</td><td><table><tr><th></th><th>RMSE</th></tr><tr><td>LSTM</td><td>8.7121</td></tr><tr><td>S_LSTM</td><td>4.4826</td></tr><tr><td>S_AM_LSTM</td><td>3.4759</td></tr></table></td></tr>
</table>

		S: sentiment analysis EMB: empirical modal decomposition AM: attention mechanism		5. Traded volume 6. Sentiment of comment from stockists 7. Sentiment of comments from Yahoo Finance	S_EMDAM_LSTM: 3.1965
[65]	Shanghai Composite Index (000001)	ANN, CNN, RNN, LSTM, CNN_RNN, CNN_LSTM	Future stock closing price	1. Stock Highest point of the day 2. Stock Lowest point of the day 3. Stock opening price of the day 4. Stock closing price of the day 5. Traded volume 6. Turnover 7. Ups and downs 8. Changes in stock price (%)	R^2 ANN: 0.9442 CNN: 0.9585 RNN: 0.9593 LSTM: 0.9622 CNN_RNN: 0.9630 CNN_LSTM: 0.9646
[66]	TPE:2354 TPE:1101 TPE:2330 TPE:1301 TPE:2382	LSTM	Future 5 days stock price average	1. Stock Highest point of the day 2. Stock Lowest point of the day 3. Stock opening price of the day 4. Stock closing price of the day 5. Traded volume	MSE / RMSE TPE:2354: 0.093 / 0.030 TPE:1101: 0.019 / 0.013 TPE:2330: 0.810 / 0.090 TPE:1301: 0.730 / 0.080 TPE:2382: 0.058 / 0.024
[23]	Different sectors of India Stock Market: IT, Pharma, FMCG, Aviation, Bank	LSTM	Future stock closing price	1. Stock closing price of the day	% of error IT: 0.7826 Pharma: 0.9034 FMCG: 0.8748 Aviation: 0.9446 Bank: 0.8939

Table 5: Comparison table of related work

Looking at the predictors dimension, historical stock price (high, low, open, close, volume) is the most used predictors in stock prediction models, being present at almost all the models of related work [2, 25, 44, 59, 60 - 66]. Besides, many of the stock prediction models include technical indicators as well [2, 25, 40, 44, 61,64], the journal article [44] selected 29 daily transaction data and technical indicators such as Relative Strength Index (RSI), ROC, Moving Average as predictors to forecast future stock price with Back Propagation ANN. However, only one stock prediction found to include the fundamental indicators as predictions [60] and no stock prediction model predicts based on macroeconomic indicators. Understanding the significance of fundamental and macroeconomic indicators in valuing a company and judging the overall health of the economy, this then leads to the motivation of creating a LSTM stock prediction model with fundamental, technical, and macroeconomic indicators.

In all stock prediction models, data normalization with *MinMaxScaler* function is performed on historical stock price to prevent the output error from being large due to the differences in data magnitude. Principal Component Analysis (PCA) is also implemented in some models to reduce the dimensionality of input variables as overlarge network input affects the accuracy of prediction [44]. Besides, Research work in [40] shows that transforming stock price data using moving average transformation improved data quality and directly increased the accuracy of the ANN stock prediction model.

Looking at the perspective of hyperparameters, the best combination of hyperparameters such as optimizer, number of epochs, activation functions, learning rate, and LSTM units is normally obtained through experiment of several combinations of hyperparameters and the best combination selected based on the best score of evaluation matrix [6, 44, 59]. Moreover, correlation analysis and linear regression coefficient ranking are also performed for feature selection [59]. The result has shown that the model which includes features selected by linear regression coefficient ranking has a better performance than the model which includes all variables [59].

3. Methodology

This project adopts Cross Industry Standard Process for Data Mining (CRISP-DM) methodology as the base model. The CRISP-DM methodology provides an overview of the standard life cycle of a data mining project which consists of six phases, namely business understanding, data understanding, data preparation, modelling, evaluation, and deployment [67]. Figure 6 presents the CRISP-DM model for this project.

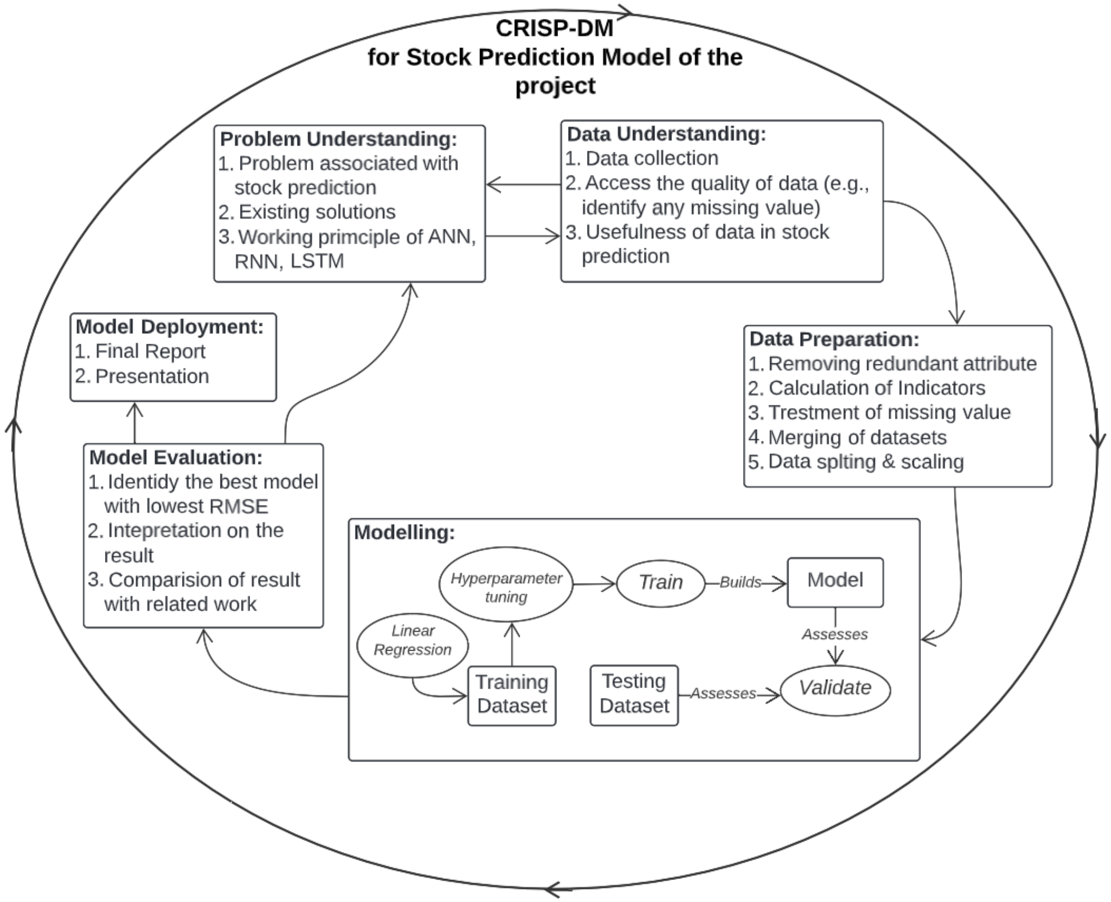

Figure 6: CRISP-DM Process Model for this project

3.1 Business Understanding

Business understanding is the initial phase of CRISP-DM methodology which focuses on the understanding of the project objective requirements and converts the knowledge into a data mining problem definition [67]. A preliminary project plan is designed in this stage to achieve the objectives of the project [67].

Step 1: Obtaining knowledge on the stock market

Perform literature review to acquire fundamental knowledge on stock market, difficulties in forecasting stock price macroeconomic, technical, and fundamental indicators to identify the problem statement of this project. The knowledge gained in this stage is also significant in providing justification on the input variables selected for modelling.

Step 2: Understanding ANN, RNN, LSTM and existing models

Understand the structure and working principle of ANN, RNN and LSTM. Explore the existing model of prediction to identify the most suitable technique for stock prediction.

Step 3: Identity project aim, objective, and plan

Based on the literature review performed, the project aim, objective and plan should be clearly defined and able to propose the predictive model and variables to be used in modelling

Step 4: Initial project plan and Gantt Chart

Design the initial project plan and produce a Gantt Chart on what to be covered in this capstone project to ensure work to be completed in a timely manner.

3.2 Data Understanding

This stage starts with an initial data collected and proceed with data exploration to identify data quality problems and discover first insights into the data [67].

Step 1: Data Collection

Firstly, data will be collected for Microsoft historical Stock price, U.S. 30-years Treasury Yield, U.D. Dollar index from *finance.yahoo.com* while Gold price per ounce in USD, Crude Oil price per barrel in USD are collected from *investing.com.* Besides, EPS is also collected from *nasdaq.com* for the calculation of P/E ratio. The data collected ranges from 5th of May 2002 until 5th of May 2022. The respective data source for each data is provided below:

a. Microsoft historical Stock price: MSFT
b. Gold price per ounce in USD: GC=F
c. Crude Oil price per barrel in USD: Cl=F
d. U.S. 30-years Treasury Yield: ^TYX
e. U.S. Dollar index: USDX - Index
f. 12 months-trailing EPS: MSFT EPS

Step 2: Data Exploration

In this stage, exploratory data analysis will be performed to generate basic statistics and access the data quality such as missing value, data format, data type, skewness and etcetera. This is an important step to get familiarized with the data set and identify treatments to overcome data quality problems found. Below shows the metadata of the datasets collected in the Data Collection step. The data exploration analysis will be performed using Python. The detailed result of data exploration will be shown in the data exploration section below.

```
RangeIndex: 5037 entries, 0 to 5036
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       5037 non-null   object
 1   Open       5037 non-null   float64
 2   High       5037 non-null   float64
 3   Low        5037 non-null   float64
 4   Close      5037 non-null   float64
 5   Adj Close  5037 non-null   float64
 6   Volume     5037 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 275.6+ KB
Dataset: MSFT
```

Figure 7: MSFT metadata

```
RangeIndex: 6126 entries, 0 to 6125
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       6126 non-null   object
 1   Open       5030 non-null   float64
 2   High       5030 non-null   float64
 3   Low        5030 non-null   float64
 4   Close      5030 non-null   float64
 5   Adj Close  5030 non-null   float64
 6   Volume     5030 non-null   float64
dtypes: float64(6), object(1)
memory usage: 335.1+ KB
Dataset: ^TYX
```

Figure 8: ^TYX metadata

```
RangeIndex: 5131 entries, 0 to 5130
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Date      5131 non-null   object
 1   Price     5131 non-null   float64
 2   Open      5131 non-null   float64
 3   High      5131 non-null   float64
 4   Low       5131 non-null   float64
 5   Vol.      5131 non-null   object
 6   Change %  5131 non-null   object
dtypes: float64(4), object(3)
memory usage: 280.7+ KB
Dataset: Cl=F
```

Figure 9: CL=F metadata

```
RangeIndex: 6126 entries, 0 to 6125
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       6126 non-null   object
 1   Open       5053 non-null   float64
 2   High       5053 non-null   float64
 3   Low        5053 non-null   float64
 4   Close      5053 non-null   float64
 5   Adj Close  5053 non-null   float64
 6   Volume     5053 non-null   float64
dtypes: float64(6), object(1)
memory usage: 335.1+ KB
Dataset: USDX
```

Figure 10: USDX metadata

```
RangeIndex: 5137 entries, 0 to 5136
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Date      5137 non-null   object
 1   Price     5137 non-null   float64
 2   Open      5137 non-null   float64
 3   High      5137 non-null   float64
 4   Low       5137 non-null   float64
 5   Vol.      5137 non-null   object
 6   Change %  5137 non-null   object
dtypes: float64(4), object(3)
memory usage: 281.1+ KB
Dataset: GC=F
```

Figure 11: GC=F metadata

```
RangeIndex: 84 entries, 0 to 83
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Dates            84 non-null     object
 1   IS_Diluted_EPS   84 non-null     float64
dtypes: float64(1), object(1)
memory usage: 1.4+ KB
Dataset: EPS
```

Figure 12: EPS metadata

3.3 Data Preparation

The data preparation phase covers activities such as data transformation, data cleaning, construction of new attributes and etcetera to produce the final dataset for modelling from initial raw data [67].

Step 1: Removing non-relevant attributes

The non-relevant attributes will be removed from the dataset. The MSFT, ^TYX and USDX dataset should only keep the ‘Date’ and ‘Close’ column while the GC=F and CL=F dataset should only keep the ‘Date’ and ‘Price’ column. This process can be performed easily on MS Excel.

Step 2: Calculation of technical and fundamental indicators

The technical indicators, MA10, STD10, ROC and RSI will be calculated based on the ‘Close’ column of the MSFT dataset. In order to calculate the P/E and PEG, Trailing Twelve Months (TTW) EPS has to be calculated first by summing the current quarter EPS and 3 previous quarter EPS. Then, P/E can be calculated by dividing the closing price of MSFT by TTW EPS calculated of the respective quarter. Annual EPS growth is also calculated based on the EPS collected for the calculation of PEG. Similarly, this process can be performed easily on MS Excel.

Step 3: Treatment of missing value

Based on the metadata generated in Data Exploration, missing values are identified in ^TYX and USDX, rows with ‘null’ value of ‘Price’ column will be removed from the dataset. The missing value is the result of including weekends and occasional holidays in the dataset when the market is not open, resulting in a ‘null’ value in ‘Price’. This process will be performed with MS Excel.

Step 4: Merging of dataset

The datasets will be merged to the MSFT dataset based on the date column and should produce a merged dataset. The output dataset should contain all the 11 explanatory variables and 1 predictor variable which is the closing price of MSFT.

It is also identified that the datasets have different number of rows although all the dataset ranges from 5th of May 2002 until 5th of May 2022. This might be due to some of the datasets include weekend and occasional holidays (when the stock market is not open). The merged dataset should include the days when the stock market is open, any rows with missing values will be removed from the dataset. This process will be performed in Ms Excel.

Step 5: Data Scaling
Data will then be transformed to the range of 0 to 1 to prevent the output error from being large due to the differences of data magnitude. This process will also be performed with python before modelling with the *MinMaxScaler* from the sklearn python library.

Step 6: Data Splitting
Data will be split into training and testing dataset. The first 18 years will be assigned to the training dataset and the last 2 years will be assigned to the testing dataset. Hence, approximately 90% of data is split into the training dataset and 10% split into the testing dataset. This process will be performed with python before modelling.

Step 7: LSTM input format
Transforming input data into LSTM data input format, which is three dimensions array of input which consists of samples, time steps and features. Please refer to the partition_dataset() function created in the system file for better understanding.

3.4 Modelling

In this phase, modelling techniques are applied [67]. The CRISP-DM methodology proposes that there is a close connection as it is often possible to construct a new dataset with different data preparation techniques when one realizes data problems during the modelling process [67]. In this project, the modelling process involves three stages, namely regression, hyperparameter tuning and the evaluation of the hyperparameter tuning result. The modelling will be carried out in python with the powerful package, Keras from TensorFlow library.

Step 1: Linear Regression model for variable selection
A linear regression model will be built with SAS Enterprise Guide to select the variable that the model suggests having a significant effect greater than 95% in predicting the next-day closing price. This will then construct a second dataset for modelling by only keeping variables suggested by the regression model.

Step 2: Hyperparameter tuning
Hyperparameters are the parameters whose value controls the learning process. The performance of all neural networks strongly relies on the hyperparameter tuning. It is significant to tune these hyperparameters so that the model can solve the machine learning problem optimally. However, the ideal hyperparameter combination is often identified through experimental methods by trying all the possible values of the hyperparameter. The hyperparameters to be tuned in this project are **activation functions, number of LSTM units in LSTM layer, Adam optimizer's learning rate, batch size and timestep**.

Activation Function: As discussed in the literature review section, activation function grants ANN the ability to learn complex and non-linear functional mappings between the predictors and target [71]. The commonly used activation functions include Sigmoid and Tanh. Sigmoid is an S-shaped curve which is smooth and ranges from 0 to 1 [72], it is the most common activation function for ANN for differentiable backpropagation. Tanh which spans from -1 to 1 is the default activation function in LSTM units. Hence, Tanh and Sigmoid are chosen for hyperparameter tuning in this project.

Number of LSTM units in LSTM layer: The LSTM unit functions as discussed in the literature review section, it receives input from the input node, perform calculation and send result to output node. The number of LSTM units determines the capacity the network. Having many hidden neurons might help in improving the network performance but when the number of neurons is excessive, the model might suffer due to potential incorrect connections [73]. In this project, LSTM model with only 100 or 200 LSTM units will be experimented to identify the best model.

Adam optimizer's learning rate: Adam is one of the algorithms that can be implemented to change the weights of variables or learning rates of a neural network to maximize performance. It Is an algorithm derived from adaptive moment estimation which is a technique for stochastic optimization which only requires first-order gradients and little memory [74], making it an optimizer which is simple to use, computationally effective, invariant to diagonal rescaling of gradients [74]. Adam has a default learning rate of 0.001, a higher learning rate allows model to learn faster and achieve the optimal performance in smaller epoch, however, too large learning might result in the model learning too fast and never converge to the local minimum of loss function. Hence, this project has chosen to select relatively small learning rate for Adam optimizer which are 0.001, 0.003 and 0.005 to tune the model.

Batch size: Batch size is the number of rows being used in each epoch in the training phase. Total batch size, mini-batch size and stochastic size are the 3 types of batch sizes. The total batch sizes utilize the whole dataset in every epoch while the mini-batch sizes use only randomly picked samples size and the stochastic size uses one sample from the dataset in each epoch during the training. Out of the 3 batch sizes, the total batch size performs the best, but it requires a large GPY memory and long training time. The report in [75], recommends using a batch size that would fit the hardware's memory. However, it is often better to use larger batch sizes so a larger learning rate can be implemented. Considering the limitation of GPU memory, this project only implements mini-batch size of 8, 16 and 32 for the hyperparameter tuning.

Timestep: which also known as sequence length is the number of days that the model would use to forecast the next predictions. However, the most appropriate number for timestep is not known and it is significant to experiment the model with different number of timestep. The most appropriate number of timestep can only be obtained through experiment. In this project, timestep of 50 days and 100 days are chosen to be tuned in the modelling process.

The same hyperparameter combinations identified will be performed on both the datasets, namely dataset with all the variables and dataset with variables selected by the regression model built.

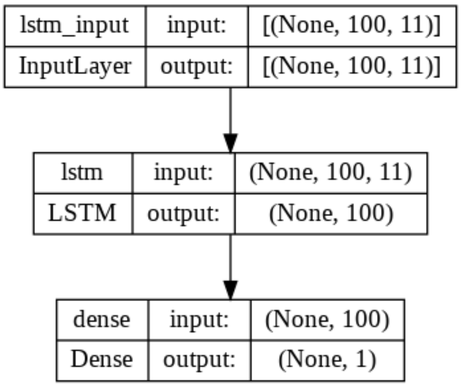

Figure 13: Architecture of the LSTM model

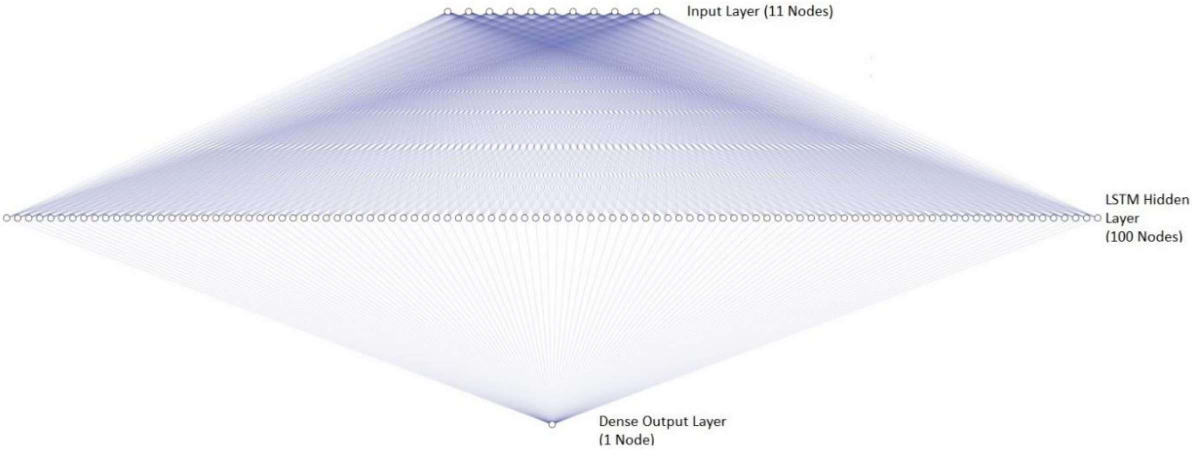

Figure 13.1: Model Representation

The figures above show the neural network architecture which consists of an input layer, followed by a LSTM layer with 100 LSTM units and a dense output layer with 1 output node. The model representation is created using NN-SVG by Alex Lenail.

The code of this project created based on the source code from GitHub [70]. Below shows the code of the model structure. The model is using Adam optimizer and having Mean Squared Error as the loss function. The highlighted value in yellow are values to be tuned in the hyperparameter tuning process. The hyperparameter combinations tested is presented in the result section below.

Model Structure:

```
sequence_length = 100

model.add(LSTM(100,kernel_initializer=GlorotNormal(),
               activation='tanh',return_sequences=False,
               input_shape=(x_train.shape[1], x_train.shape[2])))
model.add(Dense(1,kernel_initializer=GlorotNormal(),activation='linear'))

optimizer = Adam(learning_rate=0.001)
# Compile the model
model.compile(optimizer=optimizer, loss='mse')
# Training the model
epochs = 200
batch_size = 32

history = model.fit(x_train, y_train,
                            batch_size=batch_size,
                            epochs=epochs,
                             validation_data=(x_test, y_test))
```

Code 1: LSTM Model structure

The modelling process of this project mainly is done by performing hyperparameter tuning to identify the best model with lowest RMSE. A total of 72 hyperparameter combinations are defined and these combinations will be tested on both datasets with the same number of epochs, 200. At 200th epoch, the validation MSE value will be recorded and the best validation MSE value throughout the 200 epochs will also be recorded, as well as the epoch of the best validation MSE. The RMSE will then be calculated by square rooting the validation MSE recorded. The hyperparameter tuning process will be performed with Python in Google Colab.

Step 3: Evaluation on the hyperparameter tuning result

The hyperparameter combination with the lowest RMSE will be selected as the best hyperparameter combination. A table of comparison of each hyperparameter combination will be provided to better visualize the output of the hyperparameter tuning process and the effect of hyperparameters on the performance of the model. R^2 and Mean Absolute Error (MAE) will also be derived from the best model. A graph of predicted value and actual value will also be produced to observe how close the predicted value is to the actual value. Lastly, a plot of training and validation loss values will also be produced to visualize the learning process of the model, as well as overfitting and underfitting scenario throughout the learning process.

Step 4: Applying the model to forecast the next day closing price

The model built will then be applied to forecast the next day closing price, which is the price of MSFT on 6th May 2022.

3.5 Evaluation

The model(s) built in the project are evaluated and reviewed to be certain it can archive the business objectives [67].

Step 1: Result and Discussion

In the discussion section, the result of regression and LSTM model will be interpreted. The performance of the LSTM model will be evaluated based on the RMSE, R^2 and MAE evaluation matrix. Comparison will also made based on the model built in this project and the models in related work. Graph of actual output and predicted output will be presented to better visualize the predictability of the model. Besides, a plot of model loss of training and testing phase will also be presented to visualize the training process.

Step 2: Conclusion

The conclusion section can be categorized into 3 sections, namely summary of the capstone project and the delivered model, along with the result, limitation of the capstone project, the area of improvement and future scope of work:

3.6 Deployment

Deployment is the final stage of the CRISP-DM methodology. The requirement of the deployment stage depends on the scenario, it can be generating a report to explain the model or implement a repeatable data mining process [67]. In this capstone project, the deployment stage would be the final report and presentation.

3.7 Justification on the tools used in each stage

Data Collection:

No specific tool is used for this stage, datasets are directly downloaded from the *finance.yahoo.com, investing.com* and *nasdaq.com* by selecting the desired timeframe.

Data Exploration:

Data Exploration is performed with Python by utilizing the Pandas library on Google Colab. With the Pandas library, summary statistics of datasets can be generated easily with only minimal coding needed. Besides, it provides high flexibility and customization for generating graphs and plots. For-loops are also implemented in the data exploration stage to reduce code writing for generating summary statistics and plots.

Data Preparation:

Firstly, calculation of variables such as PE, PEG, MA10, STD10, ROC, RSI are performed on Ms Excel as formulae can be created and applied more easily and quickly as compared to other tools. Other than that, the merging of datasets and deleting missing values in the merged dataset is also performed with Ms Excel which is comparatively easier to use than the other tools for simple processing in small datasets.

Apart from that, python is also applied for data processing on Google Colab. The MinMaxScaler function from the sklearn library is applied to transform all the variables to a range from 0 to 1. Other than that, a partition_dataset function is created with Python to transform the dataset to a three-dimensional array of [samples, time steps, features]. Python's user defined function allows users to create high-level and complex functions which cannot be satisfied by Ms Excel. Other than that, Python also provides a tremendous number of libraries such as Numpy, Sklearn and pandas that are useful in data manipulation and processing.

Linear Regression Modelling:

SAS Enterprise Guide is applied for generating the linear regression model. SAS Enterprise Guide provides users a simple easy-to-use menu and wizard-driven tools for data analytics and machine learning modelling. The regression model is created with SAS Enterprise Guide easily, without any coding.

LSTM Modelling:

Although SAS Enterprise Guide was used for the linear regression modelling, however, it does not provide LSTM modelling features. The modelling and hyperparameter tuning for the LSTM model was performed with the powerful package, Keras from TensorFlow, developed by Google. Keras is an ANN Application Programming Interface for Python that is integrated with TensorFlow to build deep learning models. It provides high performance and flexibility in developing the model architecture.

Furthermore, Google Colab is chosen as the platform to run all the Python codes for this project. Google Colab is a useful tool as it allows the execution of codes with GPU and TPU which reduce the processing time needed for the modelling drastically.

4. Data Exploration and Preparation

This section presents the data exploration on the data collected from various sources, data processing to construct the final dataset for modelling.

A total of 5 datasets are collected, namely the historical stock price and earning per share of Microsoft Corporation, Gold price per ounce in USD (GC=F), Oil price per barrel in USD (CL=F), U.S. 30-years Treasury Yield (^TYX) and the U.S. Dollar Index (USDX Index). Data exploration is performed on each dataset collected with Python on Anaconda Spyder; data preparation needed to be performed on each dataset are clearly stated in this section as well.

4.1 Historical Stock Price of Microsoft Corporation

Date	Open	High	Low	Close	Adj Close	Volume
2002-05-06	24.719999	25.27	24.184999	24.309999	15.280165	66299400
2002-05-07	24.59	25.145	24.174999	24.735001	15.547307	88385400
2002-05-08	25.635	27.485001	25.610001	27.485001	17.275827	101242000

Table 6: first 3 rows of observation of historical stock price of Microsoft Corporation

Attributes	Data Type	Count	Missing Count	Mean	Standard Deviation	Min	Median	Max	Skewness	Kurtosis
Date	Object	5037	0							
Open	Float	5037	0	66.8	72.5	15.2	30.5	344.6	2.1	3.6
High	Float	5037	0	67.5	73.2	15.6	30.8	349.7	2.1	3.6
Low	Float	5037	0	66.1	71.7	14.9	30.2	342.2	2.1	3.6
Close	Float	5037	0	66.8	72.5	15.2	30.5	343.1	2.1	3.6
Adj Close	Float	5037	0	60.9	74.0	11.5	24.0	341.6	2.1	3.5
Volume	Integer	5037	0	50306275	29790370	7425600	45240000	591052200	3.0	28.4

Table 7: Summary Statistics of historical stock price of Microsoft Corporation

Based on the summary statistics above, the dataset consists of 7 attributes with 5037 observations. However, only 'Date' and 'Close' attribute are needed for the modelling. Hence, the other attributes will be removed from the dataset. Besides, there are no missing value identified in the dataset.

4.2 Historical Earning per Share of Microsoft Corporation

In order to calculate the PEG and P/E, Trailing Twelve Months (TTW) EPS must be calculated first based on the historical EPS collected from each quarterly report of Microsoft Corporation.

Dates	IS_Diluted_EPS
31/3/2022	2.22
31/12/2021	2.48
30/9/2021	2.71

Table 8: first 3 rows of observation of historical Earning per Share of Microsoft Corporation

Attributes	Data Type	Count	Missing Count	Mean	Standard Deviation	Min	Median	Max	Skewness	Kurtosis
Date	Object	84	0							
IS_Diluted_EPS	Float	84	0	0.7	0.6	-0.8	0.6	2.7	1.2	2.1

Table 9: Summary Statistics of Earning per Share of Microsoft Corporation

Based on the summary statistic above, no cleaning and processing are needed to be performed on this dataset. Calculation will be performed on this dataset to derive Trailing Twelve Months (TTW) EPS

4.3 Gold price per ounce in USD (GC=F)

Date	Price	Open	High	Low	Vol.	Change %
May 05, 2022	1883	1891.3	1917.6	1879.6	28.69K	0.36%
May 04, 2022	1876.2	1876.1	1898.8	1868.8	29.35K	-0.09%
May 03, 2022	1877.8	1870.4	1885	1856.7	24.90K	0.38%

Table 10: first 3 rows of observation of Gold price per ounce in USD

Attributes	Data Type	Count	Missing Count	Mean	Standard Deviation	Min	Median	Max	Skewness	Kurtosis
Date	Object	5137	0							
Price	Object	5137	0	1125.9	473.1	302.5	1230.3	2054.6	-0.2	-1.0
Open	Float	5137	0	1121.9	470.7	301.1	1225.6	2062.4	-0.2	-1.0
High	Float	5137	0	1128.3	473.9	301.0	1232.3	2085.2	-0.2	-1.0
Low	Float	5137	0	1115.0	467.0	298.0	1218.1	2033.6	-0.2	-1.0
Vol.	Object	5137	0							
Change %	Object	5137	0							

Table 11: Summary Statistics of Gold price per ounce in USD

Based on the tables above, the dataset consists of 7 variables, but only 'Date' and 'Price' variables are needed for the modelling. Hence, the other variables will be removed from the dataset. Besides, there are no missing value identified in the dataset. Therefore, other than changing the date format of 'Date' variable and removing unneeded variables, no other processing is needed for this dataset.

4.4 Oil price per barrel in USD (CL=F)

Date	Price	Open	High	Low	Vol.	Change %
May 05, 2022	106.74	106.07	109.77	104.95	97.44K	0.49%
May 04, 2022	106.22	101.99	107	101.41	92.05K	5.27%
May 03, 2022	100.9	103.19	104.06	100.58	97.51K	-2.44%

Table 12: first 3 rows of observation of Oil price per barrel in USD

Attributes	Data Type	Count	Missing Count	Mean	Standard Deviation	Min	Median	Max	Skewness	Kurtosis
Date	Object	5131	0							
Price	Object	5131	0	64.9	23.9	24.1	58.6	145.3	0.5	-0.5
Open	Float	5131	0	65.4	24.0	-14.0	61.7	145.2	0.4	-0.5
High	Float	5131	0	66.3	24.3	13.7	62.5	147.3	0.4	-0.5
Low	Float	5131	0	64.3	23.8	-16.7	60.9	143.2	0.4	-0.5
Vol.	Object	5131	0							
Change %	Object	5131	0							

Table 13: Summary Statistics of Oil price per barrel in USD (CL=F)

Based on the summary statistics above, the dataset consists of 7 variables but only 'Date' and 'Price' are needed for modelling. Hence, the other variables will be removed from the dataset. Besides, there are no missing value identified in the dataset. Therefore, other than changing the date format of 'Date' variable and removing unneeded variables, no other processing is needed for this dataset.

4.5 U.S. 30-years Treasury Yield (^TYX)

Date	Open	High	Low	Close	Adj Close	Volume
6/5/2002	5.554	5.563	5.541	5.548	5.548	0
7/5/2002	5.539	5.566	5.532	5.549	5.549	0
8/5/2002	5.587	5.668	5.585	5.664	5.664	0

Table 14: first 3 rows of observation of U.S. 30-years Treasury Yield

Attributes	Data Type	Count	Missing Count	Mean	Standard Deviation	Min	Median	Max	Skewness	Kurtosis
Date	Object	6126	0							
Open	Float	5030	1096	3.6	1.1	0.9	3.5	5.7	-0.1	-1.1
High	Float	5030	1096	3.6	1.1	1.0	3.5	5.8	-0.1	-1.1
Low	Float	5030	1096	3.6	1.1	0.8	3.4	5.7	-0.1	-1.1
Close	Float	5030	1096	3.6	1.1	0.9	3.5	5.8	-0.1	-1.1
Adj Close	Float	5030	1096	3.6	1.1	0.9	3.5	5.8	-0.1	-1.1
Volume	Float	5030	1096	1364.3	14553.3	0.0	0.0	159700.0	10.6	110.8

Table 15: Summary Statistics of U.S. 30-years Treasury Yield

Based on the summary statistic, this dataset consists of 7 attributes but only 'Date' and 'Close' are needed for modelling. Hence, the other attributes will be removed from the dataset. Other than that, many missing values are identified in this dataset. The missing value is the result of including weekends and occasional holidays in the dataset when the market is not open. Hence, treatment must be performed to delete rows consist of missing values.

4.6 U.S. Dollar Index (USDX Index)

Date	Open	High	Low	Close	Adj Close	Volume
6/5/2002	113.85	113.96	113.45	113.66	113.66	0
7/5/2002	113.44	114.31	113.39	113.92	113.92	0
8/5/2002	114.21	115.09	113.95	114.91	114.91	0

Table 16: first 3 rows of observation of U.S. Dollar Index

Attributes	Data Type	Count	Missing Count	Mean	Standard Deviation	Min	Median	Max	Skewness	Kurtosis
Date	Object	6126	0							
Open	Float	5053	1073	88.5	8.6	71.3	88.8	115.1	0.2	-0.7
High	Float	5053	1073	88.9	8.6	71.7	89.2	115.3	0.2	-0.7
Low	Float	5053	1073	88.2	8.6	70.7	88.4	114.3	0.1	-0.8
Close	Float	5053	1073	88.5	8.6	71.3	88.8	115.1	0.2	-0.8
Adj Close	Float	5053	1073	88.5	8.6	71.3	88.8	115.1	0.2	-0.8
Volume	Float	5053	1073	4609.1	237555.2	0.0	0.0	14290000.0	53.8	2996.0

Table 17: Summary Statistics of U.S. Dollar Index

Based on the tables above, the dataset contains 7 attributes but only 'Date' and 'Close' attribute are needed for modelling, so, unneeded attributes will be removed from the dataset. Similarly, many missing values are identified in this dataset, treatment will be performed to delete rows consists of missing values.

4.7 Final Dataset for modelling.

After treating the problems identified in each dataset with MS Excel, calculation of attributes, namely PEG, P/E, MA10, STD10, ROC and RSI are performed accordingly with MS Excel as well. These datasets are merged based on the 'Dates' column to construct the final dataset for modelling. After merging, rows with missing value are removed from the dataset.

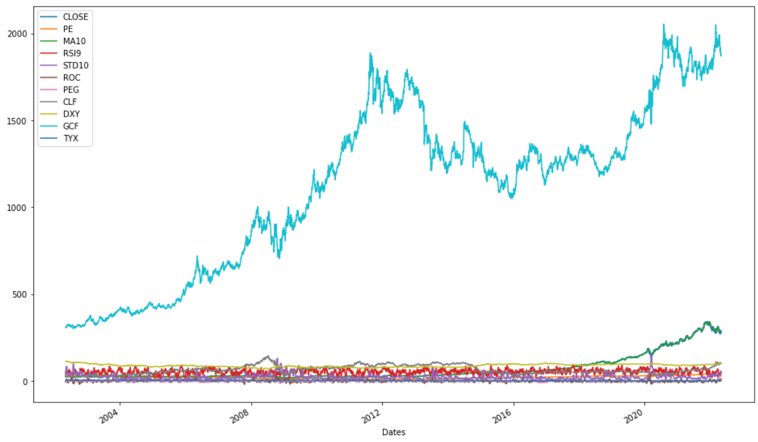

Figure 14: Combined graph for all variables before transformation

However, as shown in figure 14, the predictors are having different data range, to prevent the output error from being large due to the differences of data magnitude, the data will then be transformed to the range of 0 to 1 before modelling with the *MinMaxScaler* from the sklearn python library. Below shows the graphs of each predictor with the target variable, closing price.

Figure 15: Closing price and PE

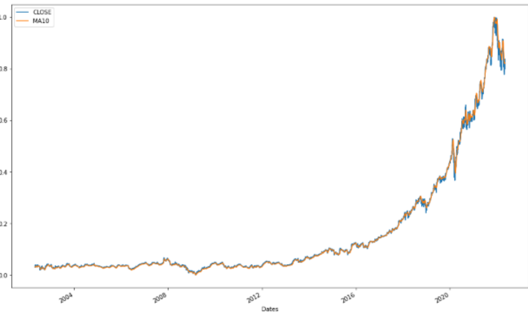

Figure 16: Closing price and MA10

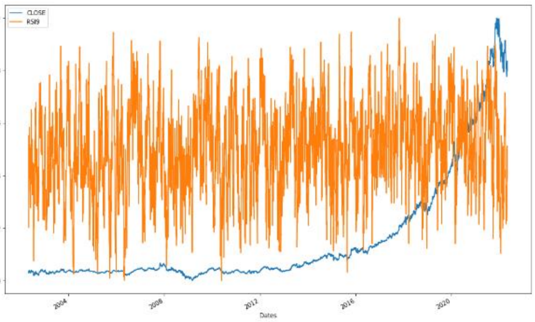

Figure 17: Closing price and RSI9

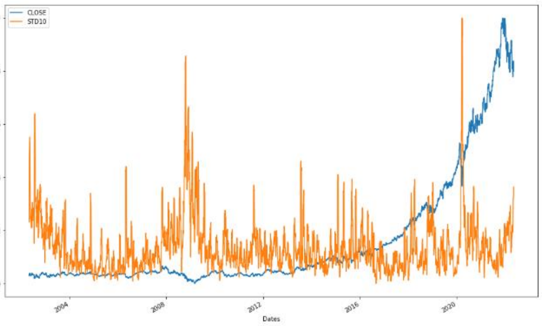

Figure 18: Closing price and STD10

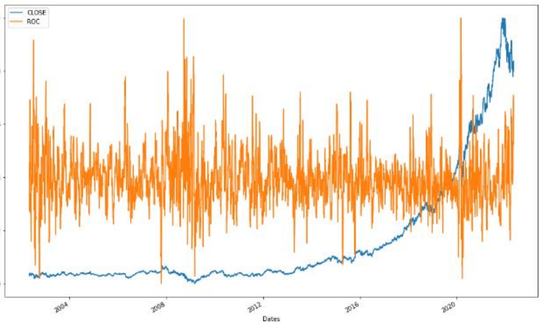

Figure 19: Closing price and ROC

Figure 20: Closing price and CLF

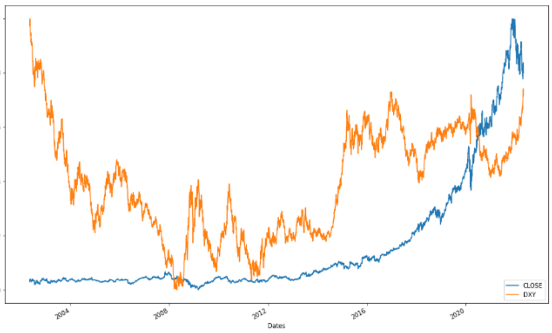

Figure 21: Closing price and DXY

Figure 22: Closing price and GCF

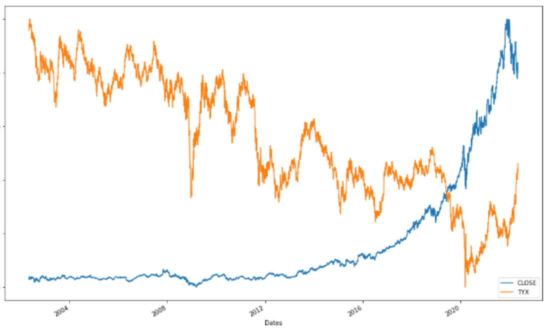

Figure 23: Closing price and TYX

Below shows the first 3 rows and summary statistics of dataset after transformation.

Dates	CLOSE	PE	MA10	RSI9	STD10	ROC	PEG	CLF	DXY	GCF	TYX
2002-05-06	0.0292261	0.557374	0.0302964	0.202066	0.231891	0.289686	0.344682	0.0165057	0.973263	0.00519377	0.957442
2002-05-07	0.0376113	0.646068	0.0305988	0.553144	0.459903	0.454143	0.342716	0.0207147	0.995887	0.00325324	0.981316
2002-05-08	0.0332663	0.600109	0.0303491	0.420263	0.476083	0.295889	0.343735	0.0307832	0.986974	0.00399521	0.976126

Table 18: first 3 rows of observation of final dataset

Attributes	Data Type	Count	Missing Count	Mean	Standard Deviation	Min	Median	Max	Skewness	Kurtosis
CLOSE	Float	5010	0	0.2	0.2	0.0	0.0	1.0	2.1	3.5
PE	Float	5010	0	0.4	0.2	0.0	0.4	1.0	0.4	-0.8
MA10	Float	5010	0	0.2	0.2	0.0	0.0	1.0	2.1	3.6
RSI9	Float	5010	0	0.5	0.2	0.0	0.5	1.0	-0.1	-0.4
STD10	Float	5010	0	0.1	0.1	0.0	0.1	1.0	2.6	11.8
ROC	Float	5010	0	0.4	0.1	0.0	0.4	1.0	0.7	2.3
PEG	Float	5010	0	0.4	0.1	0.0	0.4	1.0	1.0	6.0
CLF	Float	5010	0	0.3	0.2	0.0	0.3	1.0	0.5	-0.5
DXY	Float	5010	0	0.4	0.2	0.0	0.4	1.0	0.1	-0.8
GCF	Float	5010	0	0.5	0.3	0.0	0.5	1.0	-0.2	-1.0
TYX	Float	5010	0	0.6	0.2	0.0	0.5	1.0	-0.1	-1.1

Table 19: Summary Statistics of final dataset

Then, Data will be split into training and testing dataset. The first 18 years will be assigned to the training dataset and the last 2 years will be assigned to the testing dataset. Hence, the first 4504 rows will be in the training dataset and the last 506 rows will be in the testing dataset. This process will be performed in python before modelling.

Next, LSTM model requires three-dimensional array of input data format, the three dimensions of input are samples, time steps and features. Samples refers to the number of input rows (or days), time step (or sequence length) refers to the number of days that the model would use to forecast the next day closing price and the features refers to attributes used for prediction. The dataset will be converted to the three-dimensional array of input data format with the partition_dataset() function created as shown in the system file. With 100 sequence length, the training dataset would have a shape of (4404, 100, 11) while the testing dataset would have a shape of (406, 100, 11). Other than that, an array of the actual prediction for the training and testing datasets are also created to compare with the predicted output and calculate the training loss and validation loss.

5. Modelling & Result

In this section, firstly, a linear regression model will be built to select the significant variables in predicting the closing price of MSFT. A new dataset will be generated by only keeping the significant variables selected by the regression. Secondly, hyperparameters combinations will be identified and tested on both dataset, namely dataset with all the variables and dataset with only variables selected by the linear regression model.

5.1 Regression

A stepwise regression is performed with a significance level of 0.05 for predictors to entry and leave the model. At the 7th step, all variables left in the model are significant at the 0.05 level. No other variable met the 0.05 significance level for entry into the model anymore. Below shows the linear regression output:

Analysis of Variance					
Source	**DF**	**Sum of Square**	**Mean Square**	**F Value**	**Pr > F**
Model	7	26374592	3767799	853043	<.0001
Error	5002	22093	4.41689		
Corrected Total	5009	26396685			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-2.49576	0.52009	101.71133	23.03	<.0001
PE	0.05062	0.00827	165.64631	37.50	<.0001
MA10	0.99730	0.00093349	5041390	1141389	<.0001
RSI9	0.08998	0.00211	8024.79848	1816.84	<.0001
ROC	0.01831	0.00640	36.10616	8.17	0.0043
PEG	-0.07608	0.01252	163.08044	36.92	<.0001
DXY	-0.02499	0.00556	89.32193	20.22	<.0001
TYX	-0.20169	0.04565	86.22958	19.52	<.0001

Table 21: Linear regression result, Analysis of Variance

Based on the output above, the attributes selected by the stepwise regression model are PE, MA10, RSI9, ROC, PRG, DXY and TYX. This indicates that at significance level of 0.05, all the attributes selected are significant in predicting the closing price of MSFT. Hence, STD10, CLF and GCF are removed from the dataset, the second dataset for hyperparameter testing is contrasted as shown below:

Dates	CLOSE	PE	MA10	RSI9	ROC	PEG	DXY	TYX
2002-05-06	0.0292261	0.557374	0.0302964	0.202066	0.289686	0.344682	0.973263	0.957442
2002-05-07	0.0376113	0.646068	0.0305988	0.553144	0.454143	0.342716	0.995887	0.981316
2002-05-08	0.0332663	0.600109	0.0303491	0.420263	0.295889	0.343735	0.986974	0.976126

Table 22: Dataset with variable selected by regression model

5.2 Hyperparameter Tuning

The table below shows the result of hyperparameter tuning results:

Sequence Length	Activation Function	Adam (Learning Rate)	LSTM Unit	Batch Size	All Variables			Without STD10, CLF and GCF		
					RMSE	Lowest RMSE	Best epoch	RMSE	Lowest RMSE	Best epoch
50	**sigmoid**	**0.001**	**100**	**8**	0.0229	0.0139	103	0.0230	0.0134	62
50	**sigmoid**	**0.001**	**100**	**16**	0.0149	0.0133	185	0.0138	0.0130	184
50	**sigmoid**	**0.001**	**100**	**32**	0.0146	0.0138	198	0.0133	0.0133	200
50	**sigmoid**	**0.001**	**200**	**8**	0.0302	0.0144	56	0.0285	0.0136	64
50	**sigmoid**	**0.001**	**200**	**16**	0.0129	0.0135	166	0.0138	0.0132	166
50	**sigmoid**	**0.001**	**200**	**32**	0.0133	0.0137	199	0.0132	0.0132	193
50	**sigmoid**	**0.003**	**100**	**8**	0.0600	0.0156	11	0.0458	0.0151	18
50	**sigmoid**	**0.003**	**100**	**16**	0.0374	0.0150	27	0.0346	0.0139	100
50	**sigmoid**	**0.003**	**100**	**32**	0.0400	0.0149	71	0.0301	0.0139	100
50	**sigmoid**	**0.003**	**200**	**8**	0.0600	0.0156	11	0.0520	0.0149	26
50	**sigmoid**	**0.003**	**200**	**16**	0.0276	0.0146	84	0.0490	0.0143	56
50	**sigmoid**	**0.003**	**200**	**32**	0.0169	0.0142	154	0.0240	0.0139	128
50	**sigmoid**	**0.005**	**100**	**8**	0.0755	0.0155	9	0.0600	0.0149	8
50	**sigmoid**	**0.005**	**100**	**16**	0.0529	0.0154	22	0.0557	0.0147	43
50	**sigmoid**	**0.005**	**100**	**32**	0.0412	0.0151	42	0.0663	0.0142	58
50	**sigmoid**	**0.005**	**200**	**8**	0.0640	0.0156	13	0.0566	0.0158	13
50	**sigmoid**	**0.005**	**200**	**16**	0.0592	0.0153	33	0.0490	0.0150	39
50	**sigmoid**	**0.005**	**200**	**32**	0.0520	0.0152	63	0.0424	0.0146	84
50	**tanh**	**0.001**	**100**	**8**	0.0163	0.0112	57	0.0139	0.0099	79
50	**tanh**	**0.001**	**100**	**16**	0.0112	0.0103	128	0.0140	0.0129	182
50	**tanh**	**0.001**	**100**	**32**	0.0116	0.0097	174	0.0135	0.0132	199
50	**tanh**	**0.001**	**200**	**8**	0.0114	0.0108	171	0.0099	0.0086	183
50	**tanh**	**0.001**	**200**	**16**	0.0141	0.0105	85	0.0132	0.0129	173
50	**tanh**	**0.001**	**200**	**32**	0.0111	0.0096	116	0.0138	0.0130	190
50	**tanh**	**0.003**	**100**	**8**	0.0172	0.0145	138	0.0257	0.0130	53
50	**tanh**	**0.003**	**100**	**16**	0.0285	0.0126	61	0.1175	0.0123	21
50	**tanh**	**0.003**	**100**	**32**	0.0282	0.0119	90	0.0188	0.0099	106
50	**tanh**	**0.003**	**200**	**8**	0.0275	0.0137	9	0.0258	0.0128	9
50	**tanh**	**0.003**	**200**	**16**	0.0166	0.0122	18	0.0155	0.0119	19
50	**tanh**	**0.003**	**200**	**32**	0.0211	0.0108	84	0.0137	0.0107	103
50	**tanh**	**0.005**	**100**	**8**	0.0480	0.0145	37	0.0781	0.0132	8
50	**tanh**	**0.005**	**100**	**16**	0.0221	0.0133	47	0.0295	0.0137	41
50	**tanh**	**0.005**	**100**	**32**	0.0283	0.0116	77	0.0144	0.0101	91
50	**tanh**	**0.005**	**200**	**8**	0.0200	0.0135	16	0.0374	0.0130	29
50	**tanh**	**0.005**	**200**	**16**	0.0214	0.0129	75	0.0244	0.0120	19
50	**tanh**	**0.005**	**200**	**32**	0.0490	0.0122	88	0.0742	0.0123	90
100	**sigmoid**	**0.001**	**100**	**8**	0.0243	0.0138	115	0.0240	0.0134	74
100	**sigmoid**	**0.001**	**100**	**16**	0.0175	0.0132	74	0.0139	0.0130	196
100	**sigmoid**	**0.001**	**100**	**32**	0.0138	0.0137	198	0.0137	0.0132	198
100	**sigmoid**	**0.001**	**200**	**8**	0.0313	0.0142	48	0.0287	0.0134	74
100	**sigmoid**	**0.001**	**200**	**16**	0.0163	0.0135	181	0.0131	0.0127	186
100	**sigmoid**	**0.001**	**200**	**32**	0.0125	0.0136	198	0.0132	0.0132	200
100	**sigmoid**	**0.003**	**100**	**8**	0.0557	0.0157	13	0.0424	0.0146	31
100	**sigmoid**	**0.003**	**100**	**16**	0.0490	0.0151	35	0.0539	0.0137	51
100	**sigmoid**	**0.003**	**100**	**32**	0.0500	0.0145	96	0.0308	0.0133	125
100	**sigmoid**	**0.003**	**200**	**8**	0.0510	0.0152	19	0.0458	0.0148	23
100	**sigmoid**	**0.003**	**200**	**16**	0.0608	0.0150	41	0.0510	0.0143	46
100	**sigmoid**	**0.003**	**200**	**32**	0.0268	0.0144	101	0.0296	0.0138	115
100	**sigmoid**	**0.005**	**100**	**8**	0.0624	0.0160	6	0.0608	0.0149	16
100	**sigmoid**	**0.005**	**100**	**16**	0.0600	0.0154	17	0.0469	0.0144	40
100	**sigmoid**	**0.005**	**100**	**32**	0.0412	0.0150	43	0.0600	0.0140	74
100	**sigmoid**	**0.005**	**200**	**8**	0.0616	0.0155	11	0.0608	0.0156	8
100	**sigmoid**	**0.005**	**200**	**16**	0.0608	0.0153	26	0.0529	0.0146	44
100	**sigmoid**	**0.005**	**200**	**32**	0.0592	0.0149	57	0.0374	0.0134	125
100	**tanh**	**0.001**	**100**	**8**	0.0247	0.0109	188	0.0105	0.0105	200
100	**tanh**	**0.001**	**100**	**16**	0.0212	0.0100	180	0.0111	0.0088	196
100	**tanh**	**0.001**	**100**	**32**	0.0163	0.0164	187	0.1934	0.0084	152
100	**tanh**	**0.001**	**200**	**8**	0.0133	0.0111	29	0.0178	0.0109	80
100	**tanh**	**0.001**	**200**	**16**	0.0173	0.0108	54	0.0166	0.0096	115
100	**tanh**	**0.001**	**200**	**32**	**0.0184**	**0.0092**	**124**	**0.0081**	**0.0081**	**200**
100	**tanh**	**0.003**	**100**	**8**	0.0361	0.0146	52	0.0574	0.0122	52
100	**tanh**	**0.003**	**100**	**16**	0.0179	0.0129	163	0.0147	0.0116	70
100	**tanh**	**0.003**	**100**	**32**	0.0251	0.0123	40	0.0153	0.0115	50
100	**tanh**	**0.003**	**200**	**8**	0.0387	0.0124	68	0.0250	0.0127	52
100	**tanh**	**0.003**	**200**	**16**	0.0176	0.0126	21	0.0205	0.0117	57
100	**tanh**	**0.003**	**200**	**32**	0.0236	0.0112	69	0.0177	0.0107	124
100	**tanh**	**0.005**	**100**	**8**	0.0500	0.0168	22	0.0768	0.0141	16
100	**tanh**	**0.005**	**100**	**16**	0.0240	0.0159	23	0.0262	0.0135	150
100	**tanh**	**0.005**	**100**	**32**	0.0223	0.0122	181	0.0143	0.0119	148
100	**tanh**	**0.005**	**200**	**8**	0.0332	0.0133	36	0.0233	0.0129	47
100	**tanh**	**0.005**	**200**	**16**	0.0843	0.0127	57	0.0170	0.0131	115
100	**tanh**	**0.005**	**200**	**32**	0.0762	0.0127	57	0.0608	0.0115	83

Table 23: Result of hyperparameter tuning of all combinations.

6. Discussion

As shown in the table above, the model with 0.0081 RMSE is identified as the best model, the best model is having input variable selected by linear regression model which has excluded STD10, CLF and GCF as predictors, it has a hypepramaeter combination which consists of 100 sequence length, tanh activation function for the LSTM layer, 0.001 learning rate of Adam optimizer, 200 LSTM units in the LSTM layer and batch size of 32. It reaches its optimal performance at 200th epoch. Although the best hyperparameter combination indentified in this project achieved a good performance in predictability but different datasets have differnet distribution, resulting the best hyperparameter combinations identified in this prediction would not be applicable in other datasets.

	RMSE	MAE	R^2
Best Model	0.0081	0.0062	0.9953

Table 24: Best model performance

Other than the RMSE that is used as the evaluation matrix for selecting the best model, the Coefficient of Determination (R^2) and Mean Absolute Error (MAE) are also derieved from the model. The RMSE value derived is in the same unit as the target value, scaled stock price which has the range of 0 to 1. The RMSE of 0.0081 indicates that the prediction made by the model built to be 0.0081 out on average. Similarly, the MAE is also in the same unit as the targte value, it shows that the average error between prediction and actual stock price is 0.0062. Apart from that, the R^2 value reveals that 99.53% of variability observed in the target variable is explained by the LSTM model built. Overall, it can be concluded that the model is having a good performance as the average error is small and it can explain a high variability of the observed variable in the dataset.

Based on the related work of literature review, Table 5, the prediction model built in this study has outperformed the related work of [2,6,25,44,61,64,65] which only included technical indicators as predictors as the model. Showing that the model built in this project which utilizes technical, fundamental and macroeconomic indicators has better predictability than models which only uses technical indicators of the related work. However, the model built did not outperform the studies of [40,60,62,63, 66]. The related work of [60] included many more fundamental indicators derived from the company financials such as Debt-to-equity Ratio, Profit margin, Company beta and etcetera, providing the model with more comprehensive and quality data related to the company performance. Besides, although the related work of [40,62,63,66] only included technical indicators for prediction, the model structure and architecture are more complex as compared to the model built in this project. For example, the study in [62] uses a bidirectional LSTM and the study in [63] built a dense LSTM model with 4 LSTM hidden layers which is more complex than the LSTM model built in this project. Therefore, other than the predictors, model architecture also makes significant contributions to the model performance.

Based on the regression result, it can be observed that the STD10, CLF and GCF are excluded from the stepwise regression model built for input selection. This indicates that these variables might not be significant in forecasting the price of MSFT. Although prior study in the economic conditions stated that crude oil price could be a proxy of inflation [50,51], the effect of the crude oil price on the MSFT stock price is minimal due to several reasons. Based on the revenue breakdown of Microsoft Corporation, 23% of it is from its Office product, 34% from cloud services, 12% from Windows and 7% from LinkedIn [76]. It can be observed that most of the revenue of Microsoft Corporation is contributed from its software application and cloud service which only require minimal exportation and importation. Other than that, Microsoft Corporation is also shifting to renewable energy sources like solar and wind to reduce greenhouse gas emissions across its energy value chain [77]. Hence, the factors mentioned above cause the changes in crude oil price to only have a minimal impact on the profitability of Microsoft Corporation, as well as its stock price. Apart from that, the gold price might not be significant in forecasting the stock price due to several reasons. A research study in [78] has shown that during pre financial crisis, a strong correlation between the stock price and gold price is identified, however, the positive correlation becomes weak and low and eventually insignificant during the financial crisis, while during the post financial crisis, a negative correlation is found between stock price and gold price. Hence, the correlation and significance of gold price in forecasting stock price varies across different periods and this study uses 18 years of historical data where several financial crises are included, resulting in it not being an appropriate predictor of stock prediction anymore. Lastly, the STD10 only acts as an indicator of volatility as it measures how widely values are dispersed from the average stock price during the period, it only acts as an indication of volatility but it does not have a prediction ability on the stock price [42], resulting it being insignificant in forecasting the price.

Furthermore, it can be observed that combinations with higher learning rate (0.003 & 0.005) reaches its optimal performance in a smaller epoch while combinations with smaller learning rate (0.001) reaches its optimal performance with a larger epoch. Apart from that, in overall, it also be seen that combinations with lower learning rate would yield a lower RMSE as compared to combinations with higher learning rate. learning rate that is too large reduces the generalization accuracy, it often learns too fast, making the model difficult to converge towards the local minimum of cost function. On the other hand, small learning rate learns slower, although it requires more computational resources, but the model will converge slowly towards the local minimum of cost function. Hence, it is recommended to train neural network models with small learning rates with greater number of epochs.

Figure 24: Actual stock price vs predicted stock price of MSFT without STD10, CLF and GCF (testing result)

As shown in figures above, represented graph shows the actual closing stock price with respect to the predicted closing stock price of MSFT in validation phase, it is observed that most of the predicted closing stock price follows closely to the line of actual closing stock price, this indicates that the prediction has lower variances and fewer errors. This may infer that the LSTM model which has a good fitting effect and a good performance in predicting the actual closing stock price of MSFT.

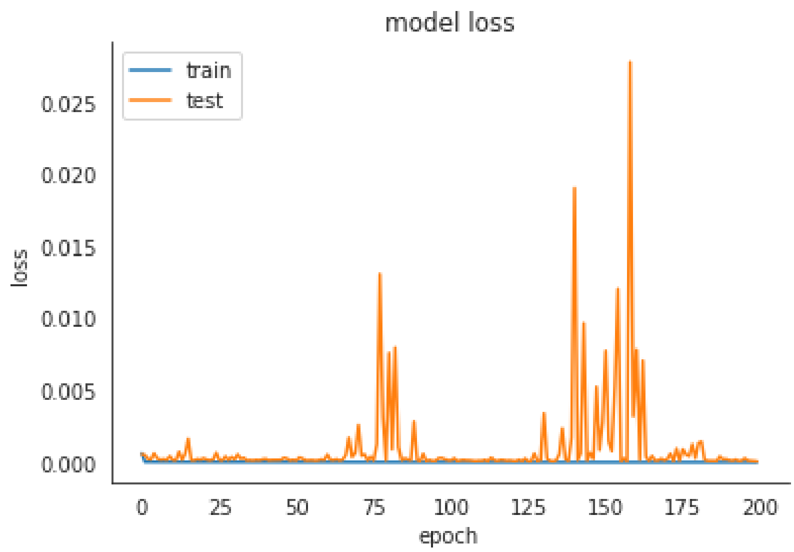

Figure 25: Validation loss and training loss of the best model

In this study, 200 epochs are set as this is the minimum threshold for the model to train. Training loss assesses the error of the model on the training dataset while the validation loss assesses the error of the model on the testing dataset. A plot shown above represents the validation loss and training loss of the best model to diagnose the model's learning performance. Based on the figure above, there are several peaks in the validation loss where it is greater than training loss, this may indicate the model is overfitting and the model is unable to generalize on the testing dataset. In other words, the model performs well on the training dataset but poorly on the testing dataset, resulting the validation loss increasing drastically.

It is suggested to set the number of epochs as higher as possible, and the training should be terminated when validation loss starts increasing because this indicates an overfitting issue. However, different datasets exhibit different behaviour, the most appropriate number of epochs depends on a variety of factors, it is important to monitor the value of the validation and training losses to avoid underfitting and overfitting thereby ensuring the generalisability of the model, thus optimally train the networks [68].

Finally, the model produced is applied to forecast the next day (6^{th} of May 2022) closing price of MSFT. The closing price for MSFT on 5^{th} of May 2022 was $277.35 and the predicted closing price for 6^{th} of May 2022 is $274.68, -0.97% change based on yesterday price.

7. Conclusion

Summary of work: The objective of this project to deliver a multivariate LSTM model is built to forecast the next day closing price of Microsoft Corporation achieved. The model built included 11 predictors which consists of 2 fundamental indicators, 5 technical indicators, and 4 macroeconomic indicators. A stepwise regression input selection model suggests that STD10, CLF and GCF are insignificant in predicting the price of Microsoft Corporation at significance level of 95%. The architecture of the LSTM model consists of an input layer, followed by a LSTM hidden layer and an output layer with linear activation function. The best model identified has a RMSE of 0.0081, it is built with the dataset without STD10, CLF and GCF.

Implication: Data from various sources are collected and processed to construct this dataset for modelling. The dataset constructed is the by-product of this project and it is available on Kaggle. The best hyperparameter combination found on this dataset for predicting Microsoft Corporation closing price is identified as 100 sequence length, tanh activation function for the LSTM layer, 0.001 learning rate of Adam optimizer, 200 LSTM units in the LSTM layer and batch size of 32. This could be a reference of hyperparameter tuning for readers who wishes to build LSTM model with the dataset constructed in this project. However, different datasets have different distribution, resulting the best hyperparameter combinations identified in this prediction would not be applicable in other datasets. The stepwise regression built as an input selection model has shown that STD10, CLF and GCF are insignificant in forecasting the stock price. In selecting macroeconomic indicators, it is significant to study on the effect of it the specific industry or organization. For instance, although crude oil price is commonly used as a proxy of inflation rate but Microsoft Corporation being a software company which involves comparatively lesser importation and exploration would have a minimal effect of crude oil price in its profitability.

Limitation: Although the project has achieved a satisfactory result, there are several limitations associated with the model built. Firstly, there are many more hyperparameters such as number of LSTM layers, dropout rate, bias regulariser and etcetera that could be experimented to improve the model performance. Secondly, the selected variables are not comprehensive enough. This project only selects 11 predictions, but in fact, there are many more stock price evaluation indicators that could analyse the stock performance. Thirdly, unlike regression, artificial neural network model does not show the contribution of variable in forecasting stock price, the result of the LSTM model does not indicate the variable importance in prediction. Lastly, the timeframe for the model to remain relevant is not known. Time series models requires historical data to train, the training dataset might need to be updated from time to time to include significant changes in the stock price. However, the appropriate timeframe to update and re-train the model is not known.

Future work: There are several further improvements and in-depth research can be carried out to improve the performance of the model. Firstly, other hyperparameters and network architectures can be further explored and identify more hyperparameter combinations to be experimented to improve the model performance. Secondly, with more hyperparameter combinations to be tested, there is a need to implement hyperparameter algorithms such as Grid Search, Random Search or Bayesian Optimization to automate the tuning process. Thirdly, other input variable selection methods such as principal component analysis and genetic algorithms can also be tested to identify the best set of prediction variables. Fourthly, future study is suggested to explore more indicators to be included in the stock prediction model. Lastly, the model has shown good predictability and efficiency in forecasting the closing price of Microsoft Corporation. The scope of the model could also be extended to numerous use cases, including predicting the price of other stocks.

8. References

[1] "Share of Americans investing in stocks 2021 | Statista." https://www.statista.com/statistics/270034/percentage-of-us-adults-to-have-money-invested-in-the-stock-market/ (accessed May 17, 2022).

[2] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," *Procedia Computer Science*, vol. 167, pp. 599–606, Jan. 2020, doi: 10.1016/J.PROCS.2020.03.326.

[3] D. Mügge, "Studying Macroeconomic Indicators as Powerful Ideas," *Journal of European Public Policy*, vol. 23, no. 3, pp. 410–427, 2016, doi: 10.1080/13501763.2015.1115537.

[4] N. Tripathy, "Causal Relationship between Macro-Economic Indicators and Stock Market in India," *Asian Journal of Finance & Accounting*, vol. 3, no. 1, pp. 208–226, 2011, doi: 10.5296/ajfa.v3i1.633.

[5] M. Obthong, N. Tantisantiwong, W. Jeamwatthanachai, and G. Wills, "A Survey on Machine Learning for Stock Price Prediction: Algorithms and Techniques," *FEMIB*, pp. 63–71, 2020, doi: 10.5220/0009340700630071.

[6] M. Roondiwala, H. Patel, and S. Varma, "Predicting Stock Prices Using LSTM," *Article in International Journal of Science and Research*, vol. 6, no. 4, pp. 1754–1756, 2017, doi: 10.21275/ART20172755.

[7] M. Zakaria, M. Al-Shebany, and S. Sarhan, "Artificial Neural Network : A Brief Overview," *Journal of Engineering Research and Applications www.ijera.com*, vol. 4, no. 2, pp. 7–12, 2014, Accessed: May 26, 2022. [Online]. Available: www.ijera.com

[8] IBM Cloud Education, "What are Neural Networks? ," *IBM*, Aug. 17, 2020. https://www.ibm.com/cloud/learn/neural-networks (accessed May 26, 2022).

[9] N. Kumar, G. Upadhyay, and P. Kumar, "Comparative Performance of Multiple Linear Regression and Artificial Neural Network Based Models in Estimation of Evaporation," *Advances in Research*, vol. 11, no. 5, pp. 1–11, Jan. 2017, doi: 10.9734/AIR/2017/36473.

[10] S. Sharma, S. Sharma, and A. Athaiya, "Activation Functions in Neural Networks," *International Journal of Engineering Applied Sciences and Technology*, vol. 4, pp. 310–316, 2020, Accessed: May 26, 2022. [Online]. Available: http://www.ijeast.com

[11] A. D. Lopez and C.-K. Li, "Neural Networks: The Backpropagation Algorithm," *College of William and Mary*.

[12] "A neural networks deep dive - IBM Developer," *IBM*, Jul. 23, 2017. https://developer.ibm.com/articles/cc-cognitive-neural-networks-deep-dive/ (accessed Jul. 06, 2022).

[13] K. P. Murphy, "Machine Learning: A Probabilistic Perspective," *Machine Learning: A Probabilistic Perspective*, p. 247, 1991, Accessed: Jun. 02, 2022. [Online]. Available: http://link.springer.com/chapter/10.1007/978-94-011-3532-0_2

[14] T. Takase, S. Oyama, and M. Kurihara, "Effective Neural Network Training with Adaptive Learning Rate based on Training Loss," *Neural Networks 101*, pp. 68–78, 2018.

[15] B. Hanin, “Which Neural Net Architectures Give Rise to Exploding and Vanishing Gradients?,” *Adv Neural Inf Process Syst*, vol. 31, 2018.

[16] A. C. Tsoi, “Recurrent neural network architectures: An overview,” pp. 1–26, 1998, doi: 10.1007/BFB0053993.

[17] “Recurrent neural networks deep dive ,” *IBM Developer*, Aug. 16, 2017. https://developer.ibm.com/articles/cc-cognitive-recurrent-neural-networks/ (accessed Jul. 06, 2022).

[18] R. M. Schmidt, “Recurrent Neural Networks (RNNs): A gentle Introduction and Overview,” *arXiv preprint arXiv:1912.05911*, 2019.

[19] “What are Recurrent Neural Networks? | IBM.” https://www.ibm.com/cloud/learn/recurrent-neural-networks (accessed May 31, 2022).

[20] Y. Yu, X. Si, C. Hu, and J. Zhang, “Communicated by Terrence Sejnowski A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” 2019, doi: 10.1162/neco_a_01199.

[21] S. Selvin, V. Ravi, V. Krishna Menon KeepFlying, and V. Krishna Menon, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” *international conference on advances in computing, communications and informatics (icacci)* , pp. 1643–1647, 2017, doi: 10.1109/ICACCI.2017.8126078.

[22] D. Shah, W. Campbell, and F. H. Zulkernine, “A Comparative Study of LSTM and DNN for Stock Market Forecasting,” *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, pp. 4148–4155, Jan. 2019, doi: 10.1109/BIGDATA.2018.8622462.

[23] G. Maji, S. Sen, A. Ghosh, S. Bose, N. C. Debnath, and A. K. Chowdhury, “Stock Price Prediction Using LSTM on Indian Share Market,” *Proceedings of 32nd international conference*, vol. 63, pp. 101–110, 2019, doi: 10.29007/qgcz.

[24] G. Singhal, “Introduction to LSTM Units in RNN | Pluralsight,” Sep. 09, 2020. https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn (accessed Jun. 09, 2022).

[25] H. Chung and Kyung-shik Shin, “Genetic algorithm-optimized long short-term memory network for stock market prediction,” *Sustainability*, vol. 10, no. 10, p. 3765, Aug. 2018.

[26] E. Beyaz, F. Tekiner, X.-J. Zeng, and J. A. Keane, “Comparing Technical and Fundamental indicators in stock price forecasting,” *IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems*, pp. 1607–1613, 2018, doi: 10.1109/HPCC/SmartCity/DSS.2018.00262.

[27] Y. A. Abdullah, D. Kusmayadi, R. Rahman, and Y. Abdullah, “ANALYSIS OF THE EFFECT OF NET PROFIT MARGIN, PRICE TO BOOK VALUE, AND DEBT TO EQUITY RATIO ON STOCK RETURN,” 2018, doi: 10.24327/ijrsr.2018.0907.2392.

[28] J. G. Agrawal, V. S. Chourasia, and A. K. Mittra, “State-of-the-Art in Stock Prediction Techniques,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, pp. 2278–8875, 2013, Accessed: May 23, 2022. [Online]. Available: www.ijareeie.com

[29] M. R. Ghaeli, “Price-to-earnings ratio: A state-of-art review,” 2016, doi: 10.5267/j.ac.2016.7.002.

[30] Radim Gottwald, “The Use of the P/E Ratio to Stock Valuation,” *European Grants Projects Journals*, pp. 21–24, 2012, Accessed: May 23, 2022. [Online]. Available: https://www.grantjournal.com/issue/0102/PDF/0102.pdf#page=21

[31] P. Shen, “The P/E Ratio and Stock Market Performance,” *Economic review-Federal reserve bank of Kansas city*, vol. 85, no. 4, pp. 23–36, 2000, Accessed: Jun. 01, 2022. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.196.7397&rep=rep1&type=pdf

[32] B. K. Meher and S. Sharma, “Is PEG Ratio a Better Tool for Valuing the Companies as Compared to P/E Ratio? (A case study on selected Automobile Companies),” *International Journal of Banking, Risk and Insurance*, vol. 3, no. 2, pp. 48–52, 2015, Accessed: May 26, 2022. [Online]. Available: http://www.publishingindia.com

[33] L. H. Chan and L. Hung Chan Associate Professor of Finance, “A More Intuitive Formula for PEG Ratio Preliminary Draft Oil Futures Market Speculations View project A More Intuitive Formula for PEG Ratio Preliminary Draft,” 2019, doi: 10.13140/RG.2.2.14244.12161.

[34] P. D. Easton, “PE Ratios, PEG Ratios, and Estimating the Implied Expected Rate of Return on Equity Capital,” *The Accounting Review*, vol. 79, no. 1, pp. 73–95, Jan. 2004, doi: 10.2308/ACCR.2004.79.1.73.

[35] P. Fiedor, “Sector strength and efficiency on developed and emerging financial markets,” *Physica A: Statistical Mechanics and its Applications*, vol. 413, pp. 180–188, Nov. 2014, doi: 10.1016/J.PHYSA.2014.06.066.

[36] A. Kononovicius and V. Gontis, “Three-state herding model of the financial markets,” *Europhysics Letters*, vol. 101, no. 2, p. 28001, Feb. 2013, doi: 10.1209/0295-5075/101/28001.

[37] A. Kononovicius and V. Gontis, “Herding interactions as an opportunity to prevent extreme events in financial markets,” *The European Physical Journal B 2015 88:7*, vol. 88, no. 7, pp. 1–6, Jul. 2015, doi: 10.1140/EPJB/E2015-60160-0.

[38] A. Raudys, V. Lenčiauskas, and E. Malčius, “Moving Averages for Financial Data Smoothing,” *Communications in Computer and Information Science*, vol. 403, pp. 34–45, 2013, doi: 10.1007/978-3-642-41947-8_4.

[39] Nathan Narendra, “All you need to know about moving averages ,” *The Economic Times*, Oct. 18, 2019. Accessed: May 23, 2022. [Online]. Available: https://economictimes.indiatimes.com/markets/stocks/news/moving-averages/articleshow/71642611.cms

[40] I. Suryani and D. C. P. C. P. Buani, “Stock price prediction using artificial neural network integrated moving average,” *Journal of Physics: Conference Series*, vol. 1641, no. 1, p. 12028, 2020, doi: 10.1088/1742-6596/1641/1/012028.

[41] J. M. Bland and D. G. Altman, “Statistics notes: Measurement error,” *BMJ*, vol. 312, no. 7047, p. 1654, Jun. 1996, doi: 10.1136/BMJ.312.7047.1654.

[42] “Moving Standard Deviation,” *StoneX Financial Inc.* . https://www.danielstrading.com/education/technical-analysis-learning-center/moving-standard-deviation (accessed May 24, 2022).

[43] S. Xu and Y. Yang, "Fractional black-scholes model and technical analysis of stock price," *Journal of Applied Mathematics*, vol. 2013, 2013, doi: 10.1155/2013/631795.

[44] H. Zhang, "The Forecasting Model of Stock Price Based on PCA and BP Neural Network," *Journal of Financial Risk Management*, vol. 7, no. 4, pp. 369–385, Nov. 2018, doi: 10.4236/JFRM.2018.74021.

[45] M. Kordos and A. Cwiok, "A New Approach to Neural Network Based Stock Trading Strategy," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6936 LNCS, pp. 429–436, 2011, doi: 10.1007/978-3-642-23878-9_51.

[46] R. Bhargavi, B. R. Associate Professor, and S. Gumparthi Professor, "Relative strength index for developing effective trading strategies in constructing optimal portfolio," *Article in International Journal of Applied Engineering Research*, vol. 12, pp. 8926–8936, 2017, Accessed: May 25, 2022. [Online]. Available: http://www.ripublication.com

[47] P. L. Bernstein, "The power of gold: the history of an obsession," *John Wiley & Sons*, 2012.

[48] S. Rathi, S. Mohapatra, and A. Sahay, "Central bank gold reserves and sovereign credit risk," *Finance Research Letters*, vol. 45, 2022.

[49] D. G. Baur and B. M. Lucey, "Is Gold a Hedge or a Safe Haven? An Analysis of Stocks, Bonds and Gold," *Financial Review*, vol. 45, no. 2, pp. 217–229, May 2010, doi: 10.1111/J.1540-6288.2010.00244.X.

[50] L. Kilian and C. Park, "The impact of oil price shocks on the US stock market," *International Economic Review*, vol. 50, no. 4, pp. 1267–1287, 2009, Accessed: May 17, 2022. [Online]. Available: www.cepr.org

[51] S. K. Sek, X. Q. Teo, and Y. N. Wong, "A Comparative Study on the Effects of Oil Price Changes on Inflation," *Procedia Economics and Finance*, vol. 26, pp. 630–636, Jan. 2015, doi: 10.1016/S2212-5671(15)00800-X.

[52] U.S. Securities And Exchange Commission, "Bonds." Accessed: May 17, 2022. [Online]. Available: https://www.investor.gov/introduction-investing/investing-basics/investment-products/bonds-or-fixed-income-products/bonds

[53] U.S. Securities And Exchange Commission, "Interest rate risk — When Interest rates Go up, Prices of Fixed-rate Bonds Fall ." Accessed: May 17, 2022. [Online]. Available: https://www.sec.gov/files/ib_interestraterisk.pdf

[54] ICE Futures U.S., "U.S. Dollar Index ® Contracts FAQ," 2015. Accessed: May 17, 2022. [Online]. Available: www.theice.com

[55] Hulbert H, "Companies Like to Use This Excuse for Weak Earnings. Don't Believe It.," *Barron's*, 2021. Accessed: May 17, 2022. [Online]. Available: https://www.barrons.com/articles/stronger-dollar-weak-earnings-51638220967

[56] J. Butter, "S&P 500 Companies With More International Revenues See Higher Revenue Growth in Q1," *Factset*, 2021. Accessed: May 17, 2022. [Online]. Available: https://insight.factset.com/sp-500-companies-with-more-international-revenues-see-higher-revenue-growth-in-q1

[57] J. C. Hughen and S. Beyer, “Stock returns and the US dollar: the importance of monetary policy,” *Managerial Finance*, vol. 41, no. 10, pp. 1046–1058, Oct. 2015, doi: 10.1108/MF-09-2014-0234/FULL/PDF.

[58] D. Christie and S. P. Neill, “Measuring and Observing the Ocean Renewable Energy Resource,” *Comprehensive Renewable Energy*, vol. 8, pp. 149–175, Jan. 2022, doi: 10.1016/B978-0-12-819727-1.00083-2.

[59] B. Aasi, S. A. Imtiaz, H. A. Qadeer, M. Singarajah, and R. Kashef, “Stock price prediction using a multivariate multistep LSTM: A sentiment and public engagement analysis model,” *2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 - Proceedings*, Apr. 2021, doi: 10.1109/IEMTRONICS52119.2021.9422526.

[60] Z. Zou and Z. Qu, “Using LSTM in Stock prediction and Quantitative Trading”.

[61] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, “Stock Market Prediction based on Deep Long Short Term Memory Neural Network,” *COMPLEXIS*, pp. 102–108, 2018, doi: 10.5220/0006749901020108.

[62] M. A. Istiake Sunny, M. M. S. Maswood, and A. G. Alharbi, “Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model,” *2nd Novel Intelligent and Leading Emerging Sciences Conference, NILES 2020*, pp. 87–92, Oct. 2020, doi: 10.1109/NILES50944.2020.9257950.

[63] A. Moghar and M. Hamiche, “Stock Market Prediction Using LSTM Recurrent Neural Network,” *Procedia Computer Science*, vol. 170, pp. 1168–1173, Jan. 2020, doi: 10.1016/J.PROCS.2020.03.049.

[64] Z. Jin, Yang Yang, and Yuhong Liu, “Stock closing price prediction based on sentiment analysis and LSTM,” *Neural Computing and Applications*, vol. 32, no. 13, pp. 9713–9729, 2020, Accessed: Jun. 08, 2022. [Online]. Available: https://web.p.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=0&sid=fb7febc9-08f7-4f6e-814f-ad8443faad8f%40redis

[65] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, “A CNN-LSTM-based model to forecast stock prices,” *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/6622927.

[66] C. Y. Lai, R. C. Chen, and R. E. Caraka, “Prediction Stock Price Based on Different Index Factors Using LSTM,” *Proceedings - International Conference on Machine Learning and Cybernetics*, vol. 2019-July, Jul. 2019, doi: 10.1109/ICMLC48188.2019.8949162.

[67] R. Wirth and J. Hipp, “CRISP-DM: Towards a Standard Process Model for Data Mining,” *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining.* , vol. 1, pp. 29–40, 2000.

[68] U. Vijay, “What is epoch and How to choose the correct number of epoch,” Medium, Sep. 06, 2019. https://medium.com/@upendravijay2/what-is-epoch-and-how-to-choose-the-correct-number-of-epoch-d170656adaaf (accessed Jul. 03, 2022).

[69] R. May, G. Dandy, and H. Maier, “Review of Input Variable Selection Methods for Artificial Neural Networks,” *Artificial neural networks-methodological advances and biomedical applications.* , vol 10, no. 1, pp19-45, 2011.

[70] F. Follonier, “007 Time Series Forecasting - Multivariate Time Series Models.ipynb,” GitHub, May. 16, 2022. https://github.com/flo7up/relataly-public-python-tutorials/blob/master/007%20Time%20Series%20Forecasting%20-%20Multivariate%20Time%20Series%20Models.ipynb , (accessed May. 05, 2022)

[71] B. Ding, H. Qian, and J. Zhou, “Activation functions and their characteristics in deep neural networks,” *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, pp. 1836–1841, Jul. 2018, doi: 10.1109/CCDC.2018.8407425.

[72] K. Vijayaprabakaran and K. Sathiyamurthy, “Towards activation function search for long short-term model network: A differential evolution based approach,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2637–2650, Jun. 2022, doi: 10.1016/J.JKSUCI.2020.04.015.

[73] M. I. C. Rachmatullah, J. Santoso, and K. Surendro, “Determining the number of hidden layer and hidden neuron of neural network for wind speed prediction,” *PeerJ Computer Science*, vol. 7, pp. 1–19, Sep. 2021, doi: 10.7717/PEERJ-CS.724/SUPP-3.

[74] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2014, doi: 10.48550/arxiv.1412.6980.

[75] L. N. Smith, “A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay,” Mar. 2018, doi: 10.48550/arxiv.1803.09820.

[76] F. Kamil, "Microsoft Revenue Breakdown by Product, Segment and Country", 2022, Available: https://www.kamilfranek.com/microsoft-revenue-breakdown/

[77] Microsoft, "Sustainability", 2022, Available: https://www.microsoft.com/en-us/sustainability/energy

[78] Al-Ameer, M., Hammad, W., Ismail, A. and Hamdan, A., 2018. The relationship of gold price with the stock market: The case of frankfurt stock exchange. International Journal of Energy Economics and Policy, 8(5), p.357.