

**FACULTY OF TECHNOLOGY**

**SCHOOL OF COMPUTER SCIENCE**

**A project dissertation submitted in partial fulfilment of the**

**regulations governing the award of the degree of**

**MSc Data Science, University of Sunderland**

**July 2020**

**Prediction on hang seng index using deep learning**

**Student Name: Yeung Wang Chung**

**Project Supervisor: Dr. Ming Jiang and Marta Rudina**

**Word Count: 11189**

I agree to my dissertation and associated documentation being used as a sample for students on future cohorts

**ACKNOWLEDGEMENTS**

I would like to deeply thank Marta Rudina, who is my project supervisor. Her considerate supports and guidance are the most appreciate at this duration of dissertation. Also, thanks for all of the contributors on artificial intelligence, who inspires me to learn these valuable knowledge and technology.

Table of Contents

**ACKNOWLEDGEMENTS1**

**TABLE OF CONTENTS2**

**ABSTRACT4**

**CHAPTER 1: INTRODUCTION5**

1.1 Project Overview5

1.2 Client’s Overview5

1.3 Client Requirements6

1.4 Project Objectives6

1.5 Constraints6

1.6 Relevance to Programme7

1.7 Chapters Structure7

**CHAPTER 2: LITERATURE REVIEW8**

2.1 Introduction8

2.2 Stock Market Predictions8

2.3 Deep Learning 9

2.4 Chapter Summary14

**CHAPTER 3: RESEARCH METHODOLOGY15**

3.1 Introduction15

3.2 Data Science Product Development Lifecycle15

3.3 Methodology17

3.4 Procedures and Data Definitions33

3.5 Evaluation Methods36

3.6 Chapter Summary39

**CHAPTER 4: RESULTS AND EVALUATIONS40**

4.1 Results40

4.2 Discussions47

**CHAPTER 5: CONCLUSIONS48**

5.1 Key Findings48

5.2 Limitations48

5.3 Recommendations and Further Studies48

5.4 Social, Ethical, Professional and Legal Issues 48

5.5 Conclusions49

**REFERENCES50**

**APPENDIX – A Correlation Plots on Datasets54**

**APPENDIX – B Files of Code and Result61**

**APPENDIX – C Terms of Reference62**

ABSTRACT

This research is to use deep learning as the main method to perform stock index market prediction with various settings to provide a predictive insights for client and it is only adopted public data (Hong Kong Hang Seng Index prices of open, high, low, close and volume). This research provided the literature review for related works, and demonstrated the complete methodology and result evaluation with result visualization. Also, the research models adopted deep learning framework, which including Tensorflow.Keras, Long Short Term Memory, feature engineering, regression, and used the second input with technical indicators as the dependent variables.

All of the deep learning models with regression are dependent on features. The comparison model with single input proved that the second input (technical indicators) helped to improve deep learning models. The model results indicate that the calculation of moving average is the good features in price predictions with regression. The machine learning approach (Extreme Gradient Boosting and Support Vector Regression) on feature selection has poor performance. Also, the data generator, which including the data with 50 previous consecutive trading days, is an important technique in deep learning model.

Keywords: Deep Learning, Long Short Term Memory (LSTM), Supervised Learning, Extreme Gradient Boosting (XGBoost), Support Vector Regression (SVR), Stock Market Prediction, Feature Engineering, Feature Importance, Tensorflow.Keras, and TensorBoard

**CHAPTER 1:**

**INTRODUCTION**

* 1. **Project Overview**

At the current stage of the global economy, the financial environment is quite competitive and requires banks and financial institutions to have a leading vision to keep, maintain and protect the clients’ assets and surely increase their wealth. Therefore, banks encounter a big challenge to develop a complete intelligent system to maintain their business, such as business analysis in retail and corporate segments. A lot of leading banks currently are conducting research projects by using artificial intelligence techniques to predict the stock market trends. In 2017, Martin Chavez, Deputy Chief Financial Officer of Goldman Sachs, explained that the number of employed traders in some departments has decreased from over 600 human traders to 2 and most of the job has been done by automated trading platforms managed by software and data engineers.

One of the biggest challenges of the stock market forecasting is identifying and investigating parameters and factors which affect stock prices. There are participants, country policies, and sudden incidents. Vargas, M. R., *et al* (2017) conducted a stock index prediction by deep learning based on external information as political and economic factors, found this information affects the financial market at a certain level, not only dependent on market prices.

There are plenty of academic and commercial publications available in this field but only partly open to the public. Tsai, Y., & Zhao, Q. (2019) adopted multiple-layer perceptron and long short term memory analysis to predict the Dow Jones Index; Jiang, M., *et al* (2020) performed the three US indices prediction comparison on machine learning and deep learning approaches.

In this project, the bank (client) would like to conduct research using artificial intelligence to forecast the trend of stock market (Hang Seng Index), including uptrend or downtrend. The main data source is publicly available data, which includes date, volume, price on open, close, high, and low. This trend prediction is forecasting the index trend in the next trading day. The model would be the role performed as technical analysis, adopting technical indicators or macroeconomic indicators to predict the uptrend or downtrend of index. The main purpose of the research project is to develop an artificial intelligence system to be a trend signal/indicator.

* 1. **Client’s Overview**

HSBC (client) is an international banking group and also it provides financial services, including money deposits, exchanges, remittances and investments etc. In current stage of global economy, the environment requires banks having a leading vision to protect the assets of clients, and then increase the wealth. Therefore, banks put many resources to develop a complete intelligent system to maintain their business, such as business analysis in retail and corporate segments. There is a new trend of using machine learning and artificial intelligence to perform predictions on business, not only on operations.

The final product is a prediction model which indicates the uptrend or downtrend on Hang Seng Index (Stock Market in Hong Kong). This model of decision-making is trained and tested on Hang Seng Index open data (open, high, low, close, volume and date). The main approach is adopting deep learning and machine learning methods and using existing technical indicators to be features.

Besides, all of the documents relevant to this research belongs to the client, including model algorithm and results.

* 1. **Client Requirements**

The client purpose is to develop an innovative decision-making platform by employing the newest finance technology to serve the business in a modern progressive way. There are following requirements proposed by client, which included:

* Use Hong Kong Hang Seng Index open data as a main dataset for work on product.
* Investigate and adopt machine learning and deep learning approaches in the product..
* Design and develop an engine of indication of Hong Kong Hang Seng Index.
* Implement and support two modes for performing feature selection on deep learning model – manual selection of paraments and default mode based on usage technical indicators and/or macroeconomic indicators.
* The Model supposed to be updated when the stock market closed.
* Deliver comprehensive documentation, report and model results including limitations and future suggestions.
  1. **Project Objectives**

The aim of this research is to use deep learning as the main method to perform stock market forecasting with various model settings such as features engineering, ensemble methods, to provide predictive insights for client trading department, e.g. investment advisors and relationship managers.

The objectives of this research are listed as below:

* To conduct literature review on stock price prediction / forecasting techniques and cutting edge methodologies and frameworks.
* To investigate the state of art in data application design and development with a deep learning engine (e.g. Recurrent Neural Network, Long Short Term Memory, and Gated Recurrent Unit etc.).
* To investigate the stock market information for features, not only Open, High, Low, Close, Volume and Date, e.g. technical indicators or macroeconomic indicators.
* To develop a deep learning framework to perform prediction (specify time window, cross-validation, time period and the outcome).
* To develop a deep learning model to achieve the prediction outcome.
* To obtain client’s evaluation on delivered product.
  1. **Constraints**

For this research, few conditions would be constrained.

* No study materials and workshops provided by client.
* No existing internal researches and approaches provided by client due to sensitivity of data. Only adopted open data in Yahoo Finance.
* Limited time to train and test models to reach a certain level of accuracy.
* Limited facilities: computer with strong computation and large memory.
  1. **Relevance to Programme**

This MSc Data Science programme is mainly focused on research, design and development effective solutions for real-life data problems in different domains. This project aimed to solve the financial market forecasting problem utilizing deep learning. Deep learning as a subset of Machine learning is out-of-syllabus in this programme, the Machine Learning module (CETM 26) and the Exploratory Data Analysis (CETM 24 & 27) introduce the AI and ML concept and applicability of algorithms. Big Data (CETM 23) and Data Visualisation (CETM 25) explained the big data project design, development and best practices of data visualization. The documentation on reports, research methodology and literature review studies (CETM 11) are also adopted.

* 1. **Chapters Structure**

Chapter 1: This chapter is a brief introduction to this study and the motivation of the problem statements. It also covers client’s information, requirements, project objectives and the constraints.

Chapter 2: Literature review is conducted which involved the stock predictions, operations and methods of machine learning and deep learning. Also, the main challenge on deep learning will be discussed (gradient vanishing).

Chapter 3: This chapter introduces comprehensive view on the whole deep learning model, the methodology, assumptions, procedures, and evaluation methods.

Chapter 4: This chapter discusses about the results and evaluations on the deep learning models.

Chapter 5: This chapter summarizes the main key findings on deep learning models. Also, it provides recommendations, limitations and further research directions of this research. Also, the social ethical professional and legal issues will be discussed.

**CHAPTER 2:**

**LITERATURE REVIEW**

**2.1 Introduction**

This chapter is to introduce the related works to this research and then review the theories behind to assess whether the framework in this research is persuasive and sufficient academic support.

**2.2 Stock Market Predictions**

2.2.1 Target Predictions

The prediction value is the most important to clarify the problem statement. Song, Y. (2018) predicted the trends (within next few days) to 20 US stocks in binary classification which generated around 60% accuracy; but Liu, D., H., & Wang, J., J. (2018) got poor results (below 40% accuracy) on predicting the trends of Dow Jones Index (next day) in multiclass classification. Those prediction values of that multiclass classification have four categories, so the layer structure for the deep learning model needs to be more complicated. Fancois, C. (2018) stated that historical data of financial market is not a good dependent variables for the prediction due to very different statistical characteristics.

Besides, another problem is the accuracy acceptance. The prediction model of Pang, X., *et al* (2018) generated over 55% accuracy but some predicted values are half of the actual values. This brings out that the large differences between predicted and actual values cause a big loss, even if the accuracy is acceptable. Therefore, the evaluation method cannot only depend on accuracy. Regression to predict price is the target prediction in this research.

2.2.2 Data Generator

Kumar, *et al* (2018), Chen, Y., & Hao, Y. (2017), Tsai, Y., & Zhao, Q. (2019) and Jiang, *et al* (2020) also adopted data generator with time-series predictions. The sequential data provides days of historical data rather than the data only in previous day. It also constructs a diverse dimensionality of dataset to the deep learning model.

Also, Fancois, C. (2018) pointed out that shuffling the data which predicting the future from historical data (e.g. stock, weather) will create a temporal leak. The dataset of this research has to be in a time order.

2.2.3 Features

For most of the related works, it is found that the values of all features are normalized independently so that the larger values of indicators do not overwhelm the smaller one (Jiang, M., *et al,* 2020).

Tsai, Y., & Zhao, Q. (2019) only constructed the relative and absolute prices of open, high, low and close as the features to perform multiclass classification. Arjun, R., & Suprabha, K., R. (2020) also used high, close price-to-book, price-to-earnings, dividend yield and beta as the features. It is insufficient feature characteristics for prediction such that generated poor results. Other than that, Song, Y. (2018) suggested that importing more indicators can possibly improve the accuracy of the model. Although the public dataset only included prices of open, high, low & close, and volume, there are many finance formulas and technical indicators to build up a wide dimensionality of features. For example, Jiang, M., *et al* (2020) combined technical indicators and macroeconomic indicators; Nelson, D., *et al* (2017) adopted a python package “TA-Lib” to construct 180 features of technical indicators, which included six aspects: volume indicators, price transforms, momentum indicators, cycle indicators, overlap studies and volatility indicators.

2.2.4 Learning Method Algorithms

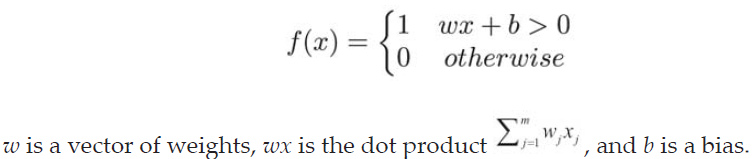
Nelson, D., *et al* (2017) adopted Google’s Tensorflow to implement deep learning model (Long Short Term Memory) and other machine learning methods. Jiang, M., *et al* (2020) adopted four tree-based ensemble algorithms and four deep learning algorithms to compare the performance with each other. Song, Y. (2018) performed comparison between methods to various US stocks, which included Long Short Term Memory, Gated Recurrent Unit, Support Vector Machine and Extreme Gradient Boosting. These comparisons are concluded that deep learning methods can perform better.

**2.3 Deep Learning**

2.3.1 Perceptron

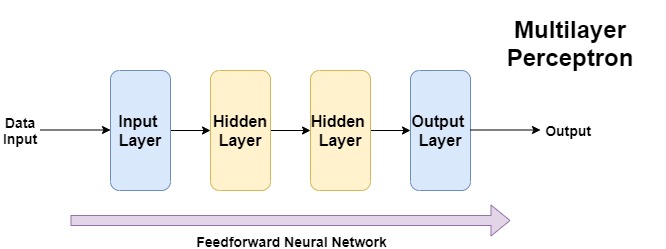
The perceptron is the basic component of a neural network. A single perceptron is a simple two-layer neuron in its simplest from, with inputs and outputs. As per definition from Antonio, G., & Sujit, P. (2017), each input value has a corresponding weight, plus an additional constant value – bias, and the sum of the dot product is sent to the activation function to return the output (Formula 2.3.1.1).

*Formula 2.3.1.1 Function of Perceptron defined by Antonio, G., & Sujit, P. (2017)*



Multiple perceptrons are combined into a single neural layer, and then multiple neural layers are connected to form a neural network. In addition to the input layer and the output layer, the middle is the hidden layer (Figure 2.3.1.2). This is an Artificial Neural Network (ANN), which extended to be other structures, including Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and Generative Adversarial Network (GAN) etc.

*Figure 2.3.1.2 Layers of Multilayer Perceptron (MLP)*

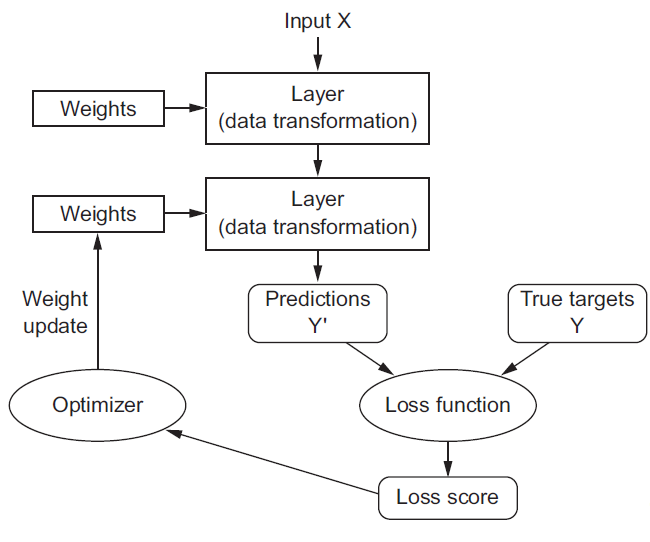


2.3.2 Basic Operation of Neural Network

The neural network adopts dataset to perform model training. This process is not only once, but is a training loop, which requires repeatedly to input data for training many times. It is called iteration. The training loop in neural network can be divided into three parts: 1) Forward Propagation, 2) Estimate the Loss and 3) Backward Propagation.

Regarding the Figure 2.3.2.1, the input data is calculated by forward propagation, and the predicted value is calculated after comparing with the actual value. Then, The back propagation is adopted to calculate the error ratio on each neural network layer. The gradient descent method is used to update the weights, where the more accurate weights are purposed to reduce the losses (the differences between actual values and predicted values).

*Figure 2.3.2.1 The Workflow of Deep Learning from Fancois, C. (2018)*



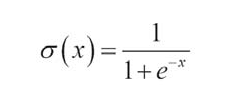
2.3.3 Model Fitting Problem

During deep learning operating, the more training loops produce better prediction model. As the number of training loops increases, the numbers and times of neural network update weights are also increased, and the learning curve of the entire neural network is started from underfitting, optimum and overfitting finally. Underfitting indicates that the prediction model is under performance in training process and overfitting indicates that the prediction model is well performance in training process but not good in testing dataset or other external recourses. Therefore, Fancois, C. (2018) advocated the ultimate milestone of the model is generalization (optimum), which indicates good performance on any related recourses.

2.3.4 Activation Function and Loss Function

Activation function in perceptron is purposed to perform non-linear data transformation, which included sigmoid (Formula 2.3.4.1) and rectified linear unit (Formula 2.3.4.2).

*Formula 2.3.4.1 Sigmoid Function*

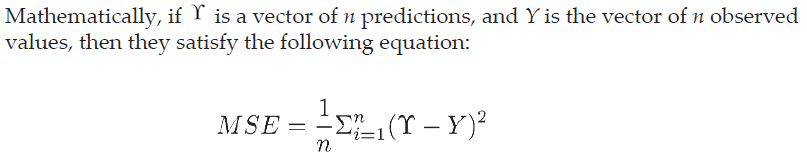


*Formula 2.3.4.2 Rectified Linear Unit (ReLU) Function*

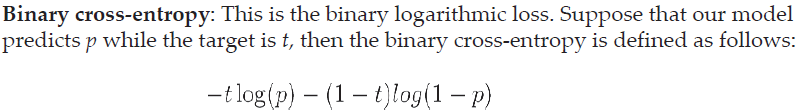


Loss function is a non-negative real number function to evaluate the differences between predicted values and actual values, i.e. defines the feedback signal for learning (Fancois, C., 2018). It is commonly adopted to mean-square-error for regression and cross-entropy for classification (Formula 2.3.4.3 and Formula 2.3.4.4).

*Formula 2.3.4.3 Mean-Square-Error (Antonio, G., & Sujit, P., 2017)*



*Formula 2.3.4.4 Binary Cross-Entropy (**Antonio, G., & Sujit, P., 2017)*



2.3.5 Gradient Vanishing Problem

When the neural network back propagation uses the linkage rate to calculate the gradient, the dot products of multiple values less than 0.25 (or less) tend to zero in only a few layers. Thus, the backpropagation would not be normally operated, i.e. all weights are not updated.

The neural network adopts the optimizer to update the weight of the neural network and the optimizer adopts back propagation to calculate the gradient that each layer of weight needs to share, and then uses the gradient descent method to update the weight of each layer of the neural network.

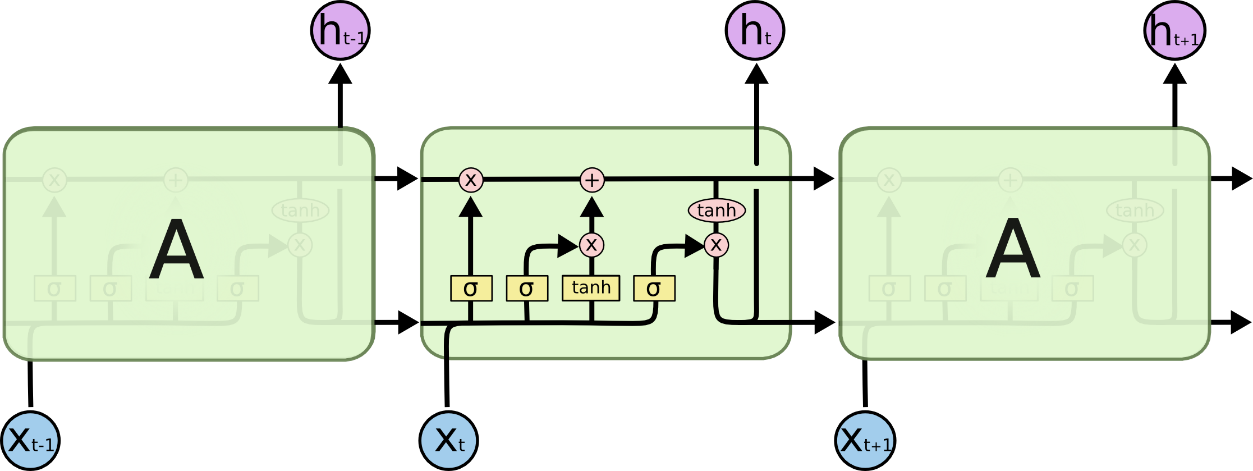
Gradient Descent

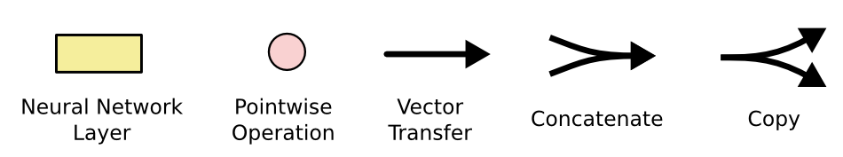
The gradient descent method is to find the local minimum value in the function, because the gradient is the direction that the function goes to the local maximum value at that point. Antonio, G., & Sujit, P. (2017) explained that gradient descent can be treated as a person who intends to climb down a mountain to a valley; while that person moves slowly, the gradient is the direction of maximum increase by calculation of partial derivative. Therefore, the opposite direction is towards the valley (minimum). The learning rate is controlling the velocity of gradient descending. However, Aurelien, G. (2019) judged that the gradient descent method depends on the activation function.

2.3.6 Long Short Term Memory

Fancois, C. (2018) explained that Long Short Term Memory is aimed at solving gradient vanishing problem as the first priority and Nelson, D. *et al* (2017) stated this method is supervised learning algorithm. This mechanism consists of three components: forget gate, input gate and output gate (Figure 2.3.6.1).

*Figure 2.3.6.1 The Workflow of Long Short Term Memory*

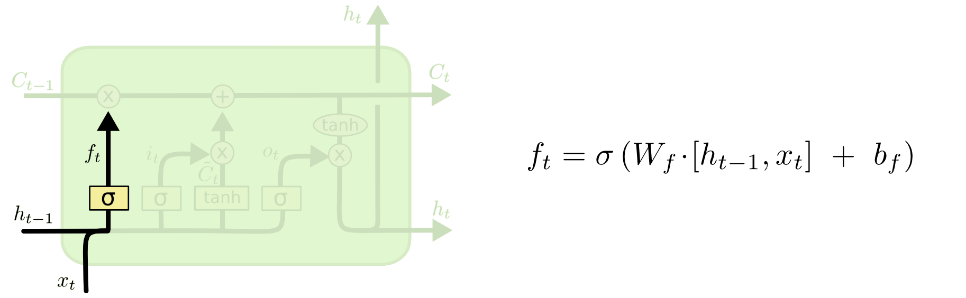




* Forget Gate

The forget gate is used to decide which data to keep and which data to forget, that is, to delete data from the long-term memory (Figure 2.3.6.2). The function is sigmoid function.

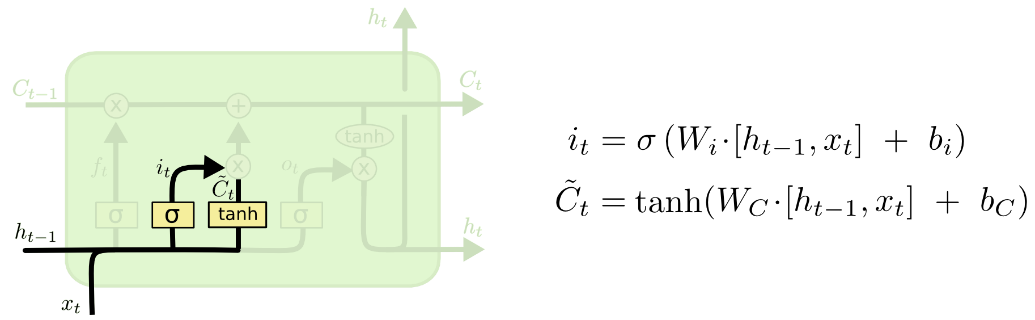
*Figure 2.3.6.2 Forget Gate*



* Input Gate

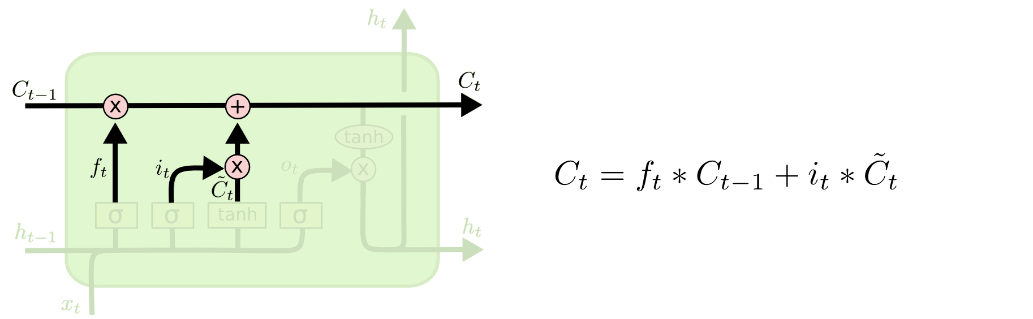
The input gate determines which data in the long-term memory needs to be updated, including new data and data to be replaced (Figure 2.3.6.3).

*Figure 2.3.6.3 Input Gate*



The cell state in previous step C(t-1) is multiplied with forget gate, which means the data will be forgotten earlier if needed. Then, the cell state in current step Ct is updated.

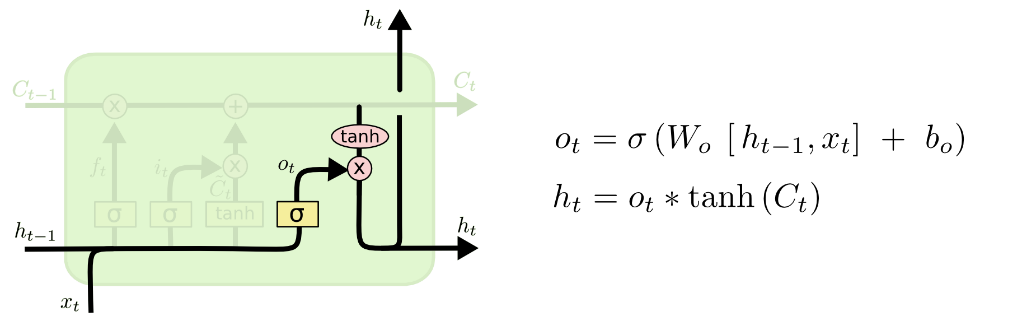
*Figure 2.3.6.4 Update Process*



* Output Gate

The output gate is used to determine what data needs to be output from the long-term memory line to the next time step t+1 (Figure 2.3.6.5).

*Figure 2.3.6.5 Output Gate*



**2.4 Chapter Summary**

The related works demonstrated that 1) classification causes accuracy acceptance problem, 2) sequential data generator, 3) more features can improve the model accuracy, and 4) Long Short Term Memory method performed better results. Besides, the principal of deep learning operation has been introduced. A challenging topic, gradient vanishing, has to be handled carefully in this research.

**CHAPTER 3:**

**RESEARCH METHODOLOGY**

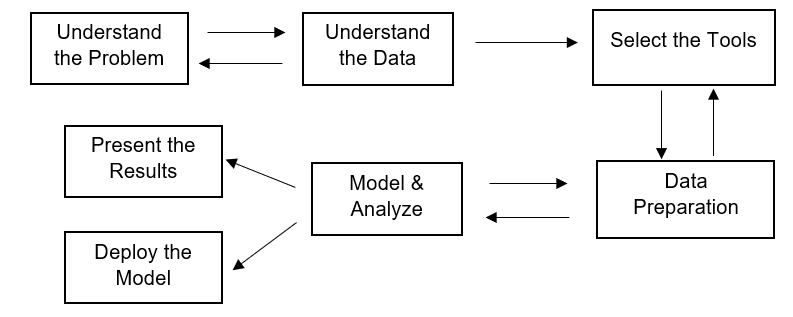
**3.1 Introduction**

This chapter is to introduce the comprehensive view on the data engine based on deep learning model of this research. The whole process obeys the lifecycle of data science: from problem statement, raw data, to data engineering, model building, evaluation and parameters tunings. At last, the result visualization and model deployment.

**3.2 Data Science Product Development Lifecycle**

Larson & Chang (2016) in their work offered the lifecycle of data product design and development. The process includes main stages of data science product development starting from identifying the problem to completed product in production – deployment and insights illustration. There are six main stages including understanding the data, selecting appropriate tools and methodologies, data preparation, building a model, test it and analyze the results.(Figure 3.2.1). Each stage has own scope of tasks and its own objectives. In the following subchapters will be shown main activities and jobs accomplished in this project at all stages.

*Figure 3.2.1 Lifecycle of Data Science* *Product Development*



3.2.1 Understand the Problem

Before start to perform research and development, the research in stock market field has been conducted with determining the most actual problems and current market solution. Based on the discussion with the client and analyzing provided business scenarios and requirements, a major part of research work was dedicated to solution which utilize machine learning techniques with a deep focus on analyzing parameters, limitations and level of accuracy for different cases.

In this project, client requested to develop a deep learning model to perform stock market predictions, which is related to time series predictions. At this stage, the research is clarified to be solving stock market predictions by deep learning model. The specific problem statements have to be clarified, which is not too general. There is a brief idea that recurrent neural network as the main framework and the tools are using the APIs of Tensorflow, Keras, Theano, Pytorch and H2O etc (Nelson, D., *et al,* 2017).

3.2.2 Understand the Data

The exploratory of data is to examine the accessibility and availability of the existing collected data. This stage has to explore the dataset in analysis way with identifying the patterns. This stage sticks closely to previous stage – Understanding the Problem. The problem statement is correlated to the existing dataset; otherwise, poor insights and results may be generated in later stages.

In this project, the open and public dataset has been adopted, including daily data – Open, High, Low, Close, Adjusted Close and Volume in Hong Kong Hang Seng Index. The data exploration found the dates are in latest date order and the columns Close and Adjusted Close are quite similar. No any missing values are found in trading dates. Also, descriptive visualization are performed (Figure 3.3.1.1 and Figure 3.3.1.2).

3.2.3 Select the Tools

With a consideration on efficiency of a research, tool selection is one of the influences. The tool should be able to perform problem solving on the specific tasks.

In this research, Python and Google Colab are adopted. Python is a programming language and Google Colab is a web-based platform to run Python codes. TensorFlow.Keras is the main API on deep learning model. No setup is required in Google Colab and lots of python libraries are pre-built. The web-based platform is completely cloud computing and uses Google Drive as storage such that the specification of laptop / PC does not affect the computation time.

3.2.4 Data Preparation

Data preparation includes pre-processing data and data engineering before applying to model. Data pre-processing is related to data cleansing, which mainly handles missing values and conversion of data types. Data engineering includes many methods, e.g. feature selection, feature extraction, and data generator etc.

In this project, the dataset contains the daily prices on trading days, so only non-trading days (holiday) are filtered out. On the data engineering parts, this research adopts to create technical indicators (as client’s requirement) as feature construction. Extreme gradient boosting and support vector regression are also applied to feature importance (feature selection).

3.2.5 Model & Analyze

After the data has been well-prepared, modelling is the main part to find out the predictive insights, e.g. categorization on patterns or recognition on behaviors etc. Performing analysis on the model results can identify the correlation or significance between features, e.g. statistical testing. Besides, if adopts machine learning or deep learning related methods, the dataset is divided into three parts: training, validation and testing. This examines the level of overfitting.

In this project, the prediction engine based on deep learning model (Long Short Term Memory) as the main frame, using regression with twelve variations to predict the next day close price. Besides, this research produced a counter-example on classification method.

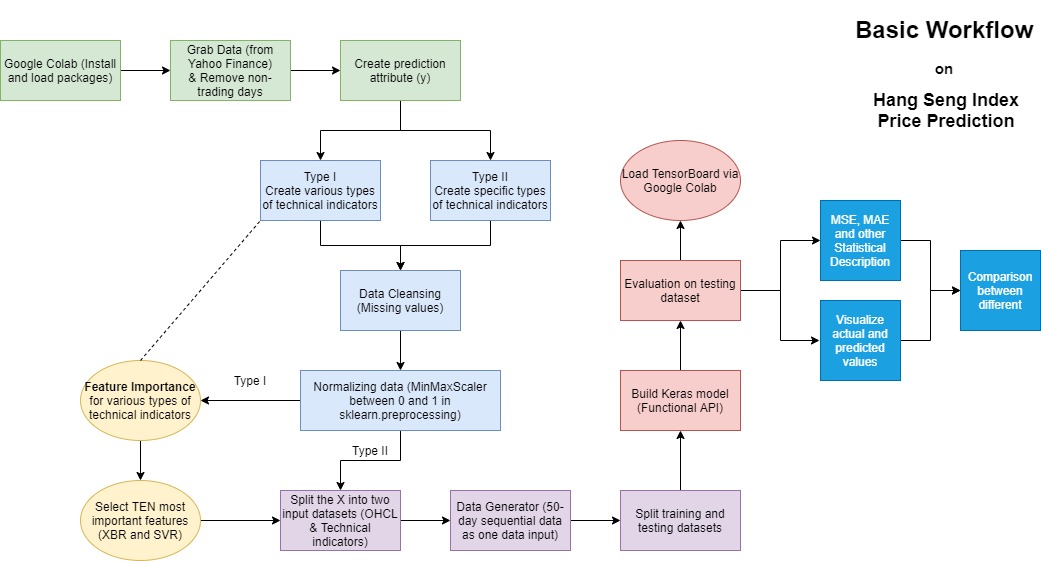
3.2.6 Result Visualization & Model Deployment

The last stages includes two parts, result visualization and model deployment. For the final product presentation, only deploying the model may confuse the audiences who are lack of the relevant knowledge. Result visualization makes audiences easier to know what the model did. Also, a successful visualization increases the persuasiveness of the model results.

In this project, the various model deployed in Google Colab and results are presented in mean square errors, mean absolute errors and other statistical numbers. The comparison between actual values and predicted values are shown in graph plots.

**3.3 Methodology**

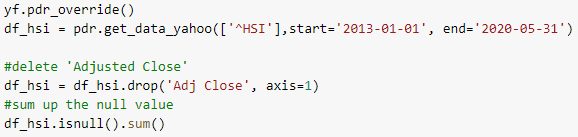
This part is to introduce the procedures of the model and clarify all codes, included model structures, data cleansing, feature selection and evaluation etc. The basic workflow diagram refers to Figure 3.3.1, which is adopted regression to perform price predictions. Various colors indicates different parts of works. An extra experiment is adopted classification and identified as counter-example (Chapter 3.5).

*Figure 3.3.1 Basic Workflow Diagram* 

3.3.1 Basic Information

This research examines the deep learning model on Hong Kong Hang Seng Index. Python is the programming language and Google Colab is the platform. Google Colab is based on Jupyter Notebook. This is an advantage that execution of deep learning model via Google Colab is using cloud computing, which is faster than laptop.

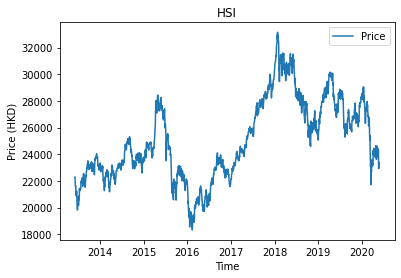
After installation and loading the packages in Google Colab, download the daily prices of Hong Kong Hang Seng Index with Open, High, Low, Close, Adjusted Close and Volume by using yfinance package. The columns “Close” and “Adjusted ” are similar prices, which are treated as duplicate columns, such that “Adjusted Close” has been deleted. The review period has been conducted between 01Jan2013 and 31May2020. Except for non-trading days, the number of rows is 1817 with 5 columns. The data type is Pandas.DataFrame.



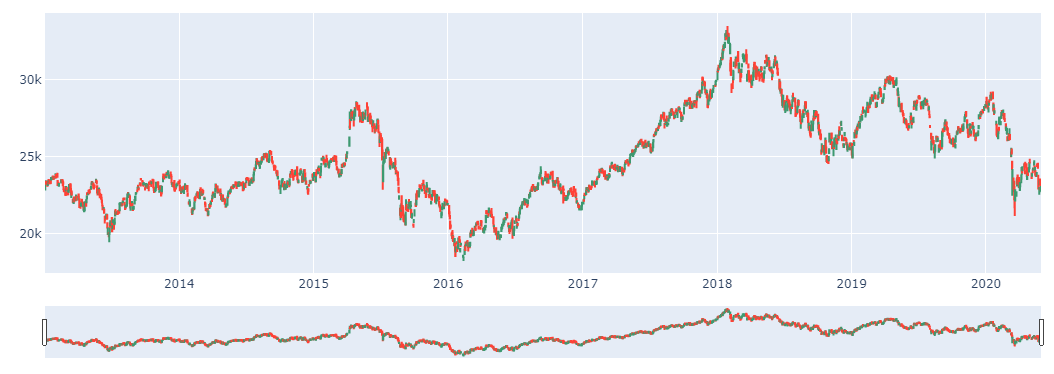


Also, the close price has been plotted in line chart (package: matplotlib) and candlesticks (package: plotly). The candlestick chart is an interactive chart, which the time frame can be manually adjusted. The corresponding plots refers to Figure 3.3.1.1 and Figure 3.3.1.2 respectively.

*Figure 3.3.1.1 Line Chart of Hang Seng Index*

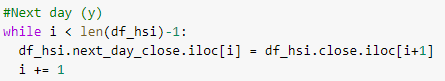


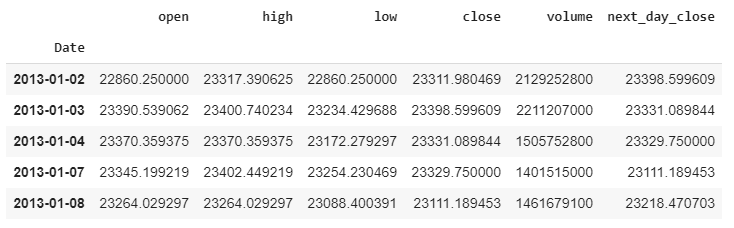
*Figure 3.3.1.2 Candlesticks of Hang Seng Index*



3.3.2 Create Y Column

The expected prediction values are the close prices in next trading day. Therefore, the creation of y column is one time-step forward of close prices (Table 3.3.2.1).



*Table 3.3.2.1 The First Five Rows of Dataset*  


3.3.3 Feature Engineering

Jiang *et al*, (2020) adopted eight technical indicators and sixteen macroeconomic indicators as the features. Sinan, O., & Divya, S. (2018) stated feature construction is to construct new features using calculations between few raw features at a time. If the number of rows on features are sufficiently large, the features interaction is captured. This research adopted technical indicators only but elaborated more types of technical indicators and various parameters.

This section is divided into two parts: 1.) various types of technical indicators and 2.) specific types of technical indicators. All of the technical indicators are adopted package named TA-Lib (John, B., 2020), which using prices open, high, low, close, and volume to create technical indicators.

* Type I: Various Types of Technical Indicators

Nelson, D., *et al* (2017) adopted Ta-Lib package to construct 180 features of technical indicators. This type of feature construction includes 58 various technical indicators with total 361 extra features, which some of them are included different time periods. These are concluded to be six aspects of technical indicators:

* Volume Indicators
* Price Transforms
* Momentum Indicators
* Cycle Indicators
* Overlap Studies
* Volatility Indicators

The indicator names, parameters and the used open, high, low, close & volume are listed in Table 3.3.3.1. Type I feature construction is purposed to create plenty of features for feature selection, which based on two machine learning methods (Extreme Gradient Boosting and Support Vector Regression). The section of feature selection refers to subchapter 3.3.5.

*Table 3.3.3.1: Type I Various Types of Technical Indicators*

\* OHLCV = Open, High, Low, Close, Volume

\*\* 1=Volume Indicators, 2=Price Transforms, 3=Momentum Indicators, 4=Cycle Indicators, 5=Overlap Studies, 6=Volatility Indicators

\*\*\* time period = 2, 3, 5, 7, 9, 14, 20, 25, 50 & 100

|  |  |  |
| --- | --- | --- |
| **Indicator Name** | **Parameters** | **OHLCV\*** |
| 1. Chaikin A/D Line (AD) | NA | HLCV |
| 1. Chaikin A/D Oscillator (ADOSC) | fastperiod = 3, slowperiod = 10 | HLCV |
| 1. On Balance Volume (OBV) | NA | CV |
| 2. Average Price (AVGPRICE) | NA | OHLC |
| 2. Median Price (MEDPRICE) | NA | HL |
| 2. Weighted Close Price (WCLPRICE) | NA | HLC |
| 3. Average Directional Movement Index (ADX) | \*\*\* NA | HLC |
| 3. Average Directional Movement Index Rating (ADXR) | \*\*\* NA | HLC |
| 3. Absolute Price Oscillator (APO) | fastperiod = 12, slowperiod = 26, matype = 0 | C |
| 3. Aroon (AROON) | \*\*\* NA | HL |
| 3. Aroon Oscillator (AROONOSC) | \*\*\* NA | HL |
| 3. Balance of Power (BOP) | NA | OHLC |
| 3. Commodity Channel Index (CCI) | \*\*\* NA | HLC |
| 3. Chande Momentum Oscillator (CMO) | \*\*\* NA | C |
| 3. Directional Movement Index (DX) | \*\*\* NA | HLC |
| 3. Moving Average Convergence/Divergence (MACD) | \*\*\* NA | C |
| 3. MACD with controllable MA Type (MACDEXT) | fastperiod = 12, fastmatype = 0, slowperiod = 26, slowmatype = 0, signalperiod = 9, signalmatype = 0 | C |
| 3. Moving Average Convergence/Divergence Fix 12/26 (MACDFIX) | signalperiod=9 | C |
| 3. Money Flow Index (MFI) | \*\*\* NA | HLCV |
| 3. Minus Directional Indicator (MINUS\_DI) | \*\*\* NA | HLC |
| 3. Minus Directional Movement (MINUS\_DM) | \*\*\* NA | HL |
| 3. Momentum (MOM) | \*\*\* NA | C |
| 3. Plus Directional Indicator (PLUS\_DI) | \*\*\* NA | HLC |
| 3. Plus Directional Movement (PLUS\_DM) | \*\*\* NA | HL |
| 3. Percentage Price Oscillator (PPO) | fastperiod = 12, slowperiod = 26, matype = 0 | C |
| 3. Rate of Change Percentage (ROCP) | \*\*\* NA | C |
| 3. Rate of Change Ratio (ROCR) | \*\*\* NA | C |
| 3. Relative Strength Index (RSI) | \*\*\* NA | C |
| 3. Stochastic (STOCH) | fastk\_period = 5, slowk\_period=3, slowd\_period=3 | HLC |
| 3. Stochastic Fast (STOCHF) | fastk\_period = 5, fastd\_period = 3,  fastd\_matype = 0 | HLC |
| 3. Stochastic Relative Strength Index (STOCHRSI) | timeperiod = 14, fastk\_period = 5,  fastd\_period = 3, fastd\_matype = 0 | C |
| 3. One-day Rate-of-Change of a Triple Smooth EMA (TRIX) | \*\*\* NA | C |
| 3. Ultimate Oscillator (ULTOSC) | timeperiod1 = 2, timeperiod2 = 5, timeperiod3 = 7  & timeperiod1 = 7, timeperiod2 = 14, timeperiod3 = 28 | HLC |
| 3. Williams' %R (WILLR) | \*\*\* NA | HLC |
| 4. Hilbert Transform - Dominant Cycle Period (HT\_DCPERIOD) | NA | C |
| 4. Hilbert Transform - Dominant Cycle Phase (HT\_DCPHASE) | NA | C |
| 4. Hilbert Transform - Phasor Components (HT\_PHASOR) | NA | C |
| 4. Hilbert Transform - SineWave (HT\_SINE) | NA | C |
| 4. Hilbert Transform - Trend vs Cycle Mode (HT\_TRENDMODE) | NA | C |
| 5. Bollinger Bands (BBANDS) | timeperiod = 10,nbdevup = 2, nbdevdn = 2, matype = 0 | C |
| 5. Double Exponential Moving Average (DMEA) | \*\*\* NA | C |
| 5. Exponential Moving Average (EMA) | \*\*\* NA | C |
| 5. Hilbert Transform - Instantaneous Trendline (HT\_TRENDLINE) | NA | C |
| 5. Kaufman Adaptive Moving Average (KAMA) | \*\*\* NA | C |
| 5. Moving Average (MA) | \*\*\* matype = 0 | C |
| 5. MESA Adaptive Moving Average (MAMA) | fastlimit = 0.5, slowlimit = 0.05 | C |
| 5. Midpoint Over Period (MIDPOINT) | \*\*\* NA | C |
| 5. Midpint Price Over Period (MIDPRICE) | \*\*\* NA | HL |
| 5. Parabolic SAR (SAR) | Acceleration = 0.02, maximum = 0.2 | HL |
| 5. Parabolic SAR - Extended (SAREXT) | startvalue = 0, offsetonreverse = 0,  accelerationinitlong = 0.02, accelerationlong = 0.02, accelerationmaxlong = 0.2, accelerationinitshort = 0.02, accelerationshort = 0.02, accelerationmaxshort = 0.2 | HL |
| 5. Simple Moving Average (SMA) | \*\*\* NA | C |
| 5. Triple Exponential Moving Average (T3) | \*\*\* vfactor = 0 | C |
| 5. Triple Exponential Moving Average (TEMA) | \*\*\* NA | C |
| 5. Triangular Moving Average (TRIMA) | \*\*\* NA | C |
| 5. Weighted Moving Average (WMA) | \*\*\* NA | C |
| 6. One-Day True Range | NA | HLC |
| 6. Average True Range (ATR) | \*\*\* NA | HLC |
| 6. Normalized Average True Range (NATR) | \*\*\* NA | HLC |

* Type II: Specific Types of Technical Indicators

Jiang, M., *et al* (2020) adopted 8 various technical indicators. This Type II feature construction adopts 10 various technical indicators, but each type of technical indicators only applies to one model, i.e. one of the data inputs. Therefore, it will be ten more various models for the results comparison. Those ten technical indicators includes:

* Average True Range (ATR)
* Commodity Channel Index (CCI)
* Exponential Moving Average (EMA)
* Money Flow Index (MFI)
* Momentum (MOM)
* Rate of Change (ROC)
* Relative Strength Index (RSI)
* Simple Moving Average (SMA)
* Williams' %R (WILLR)
* Weighted Moving Average (WMA)

The indicator names, parameters and the used open, high, low, close & volume are listed in Table 3.3.3.2. Most of the technical indicators also adopted in Type I (except Rate of Change) and the time period has amended from 10 different ranges to be 14.

*Table 3.3.3.2 Type II Specific Types of Technical Indicators*

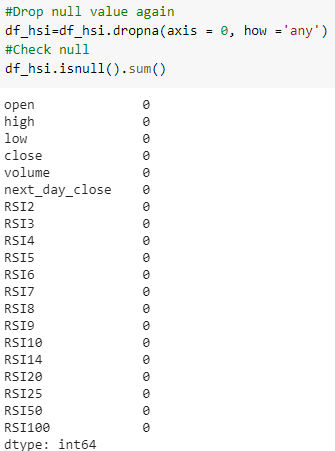
\*OHLCV = Open, High, Low, Close, Volume

\*\* 1=Volume Indicators, 2=Price Transforms, 3=Momentum Indicators, 4=Cycle Indicators, 5=Overlap Studies, 6=Volatility Indicators

|  |  |  |
| --- | --- | --- |
| **Indicator Name** | **Parameters** | **OHLCV\*** |
| 6. Average True Range (ATR) | time period = 2 - 10, 14, 20, 25, 50 & 100 | HLC |
| 3. Commodity Channel Index (CCI) | time period = 2 - 10, 14, 20, 25, 50 & 100 | HLC |
| 5. Exponential Moving Average (EMA) | time period = 2 - 10, 14, 20, 25, 50 & 100 | C |
| 3. Money Flow Index (MFI) | time period = 2 - 10, 14, 20, 25, 50 & 100 | HLCV |
| 3. Momentum (MOM) | time period = 2 - 10, 14, 20, 25, 50 & 100 | C |
| 3. Rate of Change (ROC) | time period = 2 - 10, 14, 20, 25, 50 & 100 | C |
| 3. Relative Strength Index (RSI) | time period = 2 - 10, 14, 20, 25, 50 & 100 | C |
| 5. Simple Moving Average (SMA) | time period = 2 - 10, 14, 20, 25, 50 & 100 | C |
| 3. Williams' %R (WILLR) | time period = 2 - 10, 14, 20, 25, 50 & 100 | HLC |
| 5. Weighted Moving Average (WMA) | time period = 2 - 10, 14, 20, 25, 50 & 100 | C |

3.3.4 Data Cleansing

The technical indicators (feature engineering) used a range of days of data to generate. Due to a fixed review period (between 01Jan2013 and 31May2020), there are some null (NA) values. The earliest date is different in various models.

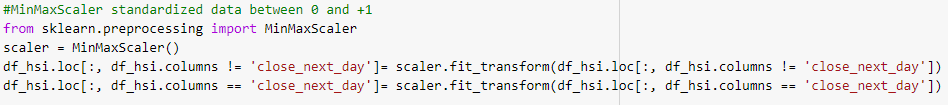


Also, the last row (latest trading day) has no close price the next day, i.e. no actual value. Therefore, that row is deleted.



3.3.5 Normalization (Scaling Data)

All of the columns in the datasets (Date is index) are normalized as within the range between zero and one (no negative values). This normalization is adopted MinMaxScaler function in scikit-learn (package name).

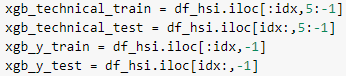
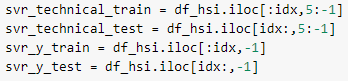


For the models of type II, the datasets are ready to transform as the inputs of deep learning models (refers to subchapter 3.3.7). Besides, one more procedures have to be executed for the models of type I – Feature Selection (subchapter 3.3.6).

3.3.6 Feature Selection

This section is only applied on the datasets of Type I methods. Sinan, O., & Divya, S. (2018) stated feature selection is to select the features which implied strong signals and to ignore the features with noise. Ten out of those additional features (technical indicators) are chosen as the highest importance by using Extreme Gradient Boosting and Support Vector Regression separately. Another words, the prices of open, high, low, close and volume are filtered out in this section.

Before performing feature importance, the dataset will be split into technical indicators, y (next date close price), training and testing with total 4 datasets on each method, which the ratio is 90:10. No shuffling split is applied such that the testing dataset is the latest trading days.

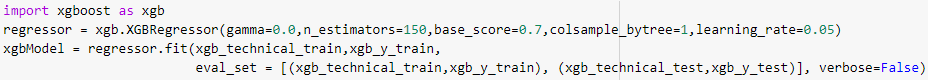
  

These two machine learning methods are adopted to be slightly different. The details will be explained in following bullets.

* Extreme Gradient Boosting

Package

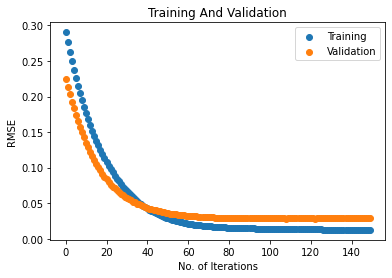
This method is adopted the package xgboost for execution. The training dataset is purposed for model fitting and the testing dataset is for evaluation purposes.



Results

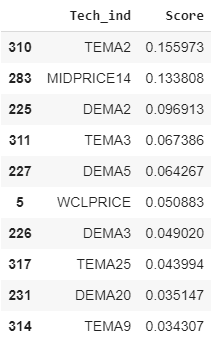
The evaluation result is acceptable for the exponentially decreasing root-mean-square error (RMSE) (Figure 3.3.6.1).

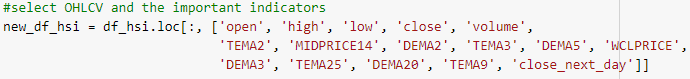
*Figure 3.3.6.1 RMSE on Training and Testing Against Iterations*



Then sorting the importance scores in descending order (Table 3.3.6.2). The new dataset has been created with open, high, low, close, volume and close price next day.

*Table 3.3.6.2 Top Ten Highest Scores of Feature Importance*





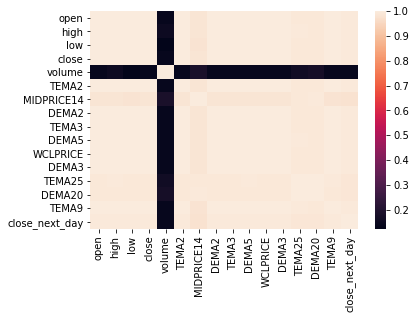
The selected features are included:

* TEMA 2, 3, 9 & 25: Triple Exponential Moving Average with time period 2, 3, 9 and 25 days
* MIDPRICE14: Midpoint Price Over Period with 14 days
* DEMA 2, 3, 5 & 20: Double Exponential Moving Average with time period 2, 3, 5 & 20 days
* WCLPRICE: Weighted Close Price

Correlation

The correlation on the new dataset found volume is negatively correlated to other attributes (Figure 3.3.6.3).

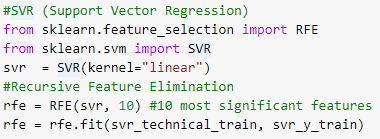
*Figure 3.3.6.3 Correlation Plot (XGB)*



* Support Vector Regression

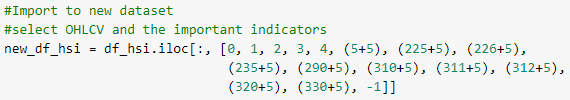
Package and Results

This method is adopted support vector regression from scikit-learn (package name) and recursive feature elimination, which is a feature selection tool in scikit-learn.





The new dataset has been created with open, high, low, close, volume and close price with next day.





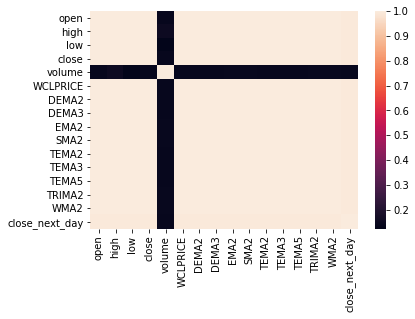
The selected features are included:

* WCLPRICE: Weighted Close Price
* DEMA 2 & 3: Double Exponential Moving Average with time period 2 and 3 days
* EMA2: Exponential Moving Average with time period 2 days
* SMA2: Simple Moving Average with time period 2 days
* TEMA 2, 3, & 5: Triple Exponential Moving Average with time period 2, 3 and 5 days
* TRIMA2: Triangular Moving Average with time period 2 days

Correlation

The correlation on the new dataset also found volume is negatively correlated to other attributes (Figure 3.3.6.4).

*Figure 3.3.6.4 Correlation Plot (SVR)*



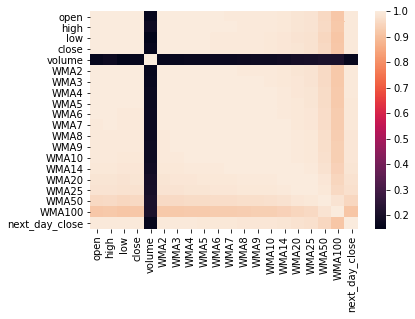
3.3.7 Training and Testing Datasets

Same as previous section (subchapter 3.3.6), the ratio of training and testing is 90:10. The deep learning model contains two inputs, one for basic prices (open, high, low, close), another for technical indicators (including Type I and Type II).

Other than that, the correlation plots of these datasets (total 12) indicates that volume is almost negatively correlated with other features (refers to the example Weighted Moving Average, Figure 3.3.7.1). Therefore, all of the models are filtered out volume feature. For the correlation plots on models, please refers to appendix A.

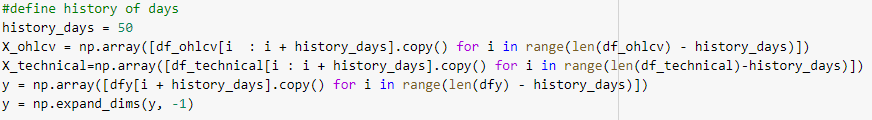


*Figure 3.3.7.1: Correlation Plot (WMA)*



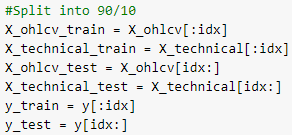
Data Generator

Regarding the literature review (Chapter 2.2.2), using sequential data (numbers of days) is better for Long Short Term Memory. This research only used fixed time frame to assure the model results not affected by different time frame. The time frame is set to be 50 days in one data frame and all are applied to two inputs and y (prediction column).



Splitting Datasets

Including two inputs (ohlc and technical indicators), prediction column (y) and training & testing, there are totaling six datasets with ratio 90 (training) :10 (testing). Since some features of technical indicators require longer time periods of data (due to formulas), the numbers of rows in various technical indicators / methods are different (Table 3.3.7.2). The dataset is sorted by date from oldest to newest, such that the testing dataset included the index price in 2020, which has a sudden drop and big v-cut rebound (Figure 3.3.7.3). It is a challenge for the deep learning model. A comparison model (OHLC) is a single input with prices of open, high, low and close.



*Table 3.3.7.2: Number of Trading Days on Training and Testing Datasets*

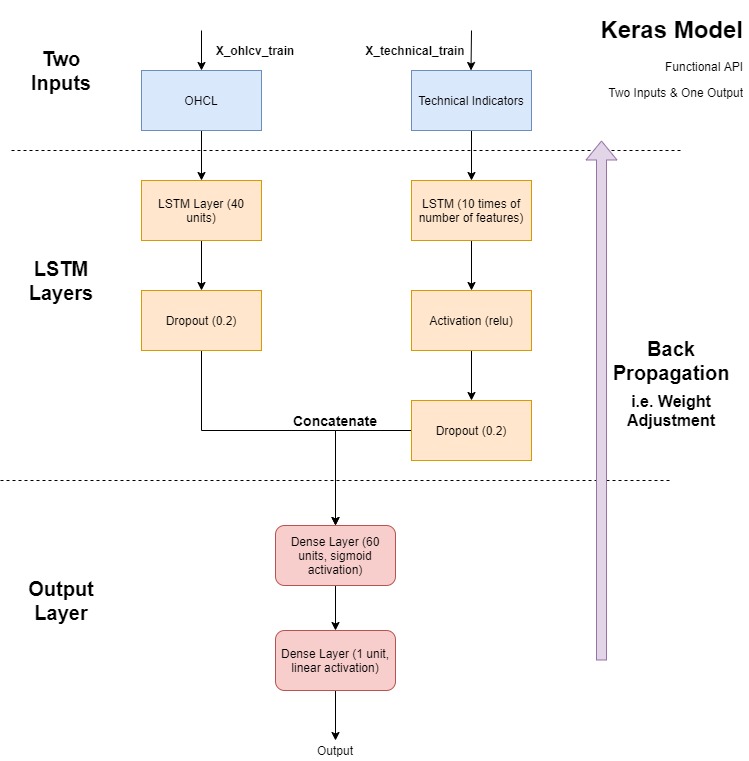
|  |  |  |
| --- | --- | --- |
| **Indicators / Methods** | **No. of Rows (Training)** | **No. of Rows (Testing)** |
| OHLC | 1591 | 175 |
| XGB | 1101 | 71 |
| SVR | 1101 | 71 |
| ATR | 1501 | 165 |
| CCI | 1502 | 165 |
| EMA | 1502 | 165 |
| MFI | 1501 | 165 |
| MOM | 1501 | 165 |
| ROC | 1501 | 165 |
| RSI | 1501 | 165 |
| SMA | 1502 | 165 |
| WILLR | 1502 | 165 |
| WMA | 1502 | 165 |

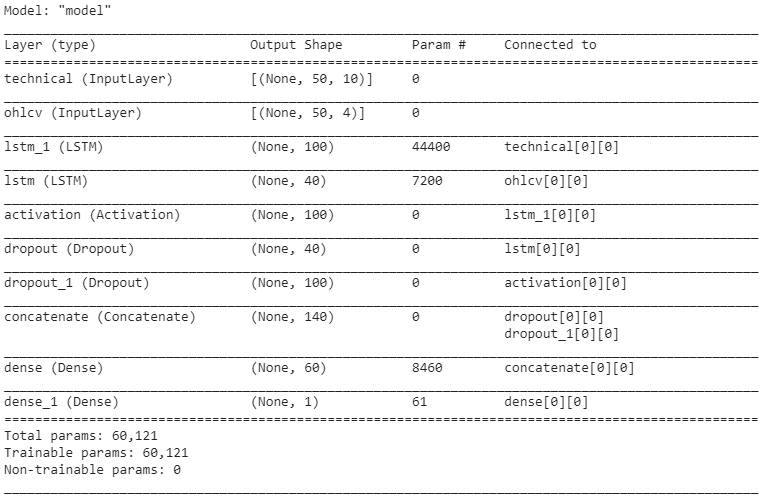
*Figure 3.3.7.3 Huge Drop of Hang Seng Index in 2020*



3.3.8 Models in Tensorflow.Keras

After the launch of Tensorflow version 2.0, the Keras API can be implemented in Tensorflow. For the deep learning model in Keras, there are several parts which have to be set up. The model structure and summary are shown in Figure 3.3.8.1 and Figure 3.3.8.2 respectively.

*Figure 3.3.8.1: Keras Model (Functional API)*  


*Figure 3.3.8.2 Model Summary (SVR)*  


* Model

Keras model provides two basic APIs: Sequential API and Functional API. This research is adopted Functional API due to more than one inputs, including dataset of Open, High, Low, & Close and dataset of technical indicators. This model type contains inputs ([ohlcv\_in and technical\_in]) and output (out).



It is noted that shape is related to the actual dimensions of the datasets. The actual shape is a 3-D array because one row includes the columns of features, and data of 50 consecutive trading days.

Output has been concatenated two outputs from two inputs ([ohlcv\_in and technical\_in]). Then it used two Dense layers to generate predicted value.

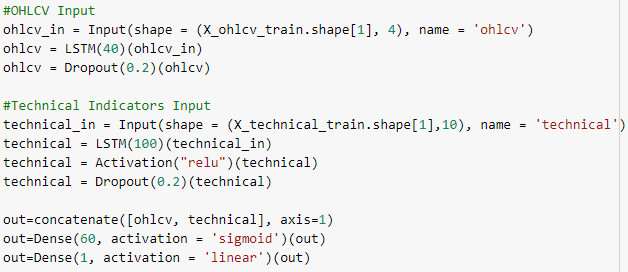


* Layers

This part only includes three components: Long Short Term Memory (LSTM), Dense, and Dropout. LSTM and Dense layers are categorized as neural networks. Dropout helps for prevention of overfitting, which randomly turn inputs as zero values at a certain level.

* Activation Functions

Activation function is the calculation method on the neurons. The research is adopted relu, sigmoid and linear functions.



* Compile

In this research, the compile function consists of three parameters: loss function, optimizer and metrics. Loss is set to be mean-square-error, which is a standard of self-adjusting weight for the neural network model. Optimizer is set to be Adam with learning rate 0.0005, where it is an algorithm to control the gradient descent issues. Metrics is set to be mean-absolute-error, which is one of the evaluation methods.

It is noted that these loss function and metrics are related to regression, which predicting numeric values. The classification is totally different settings for these two parameters. The details of classification (counter-example) refer to subchapter 3.5.



* Callbacks

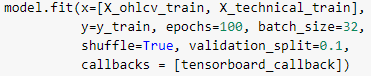
In Keras API, there are customized callbacks and standard callbacks (e.g. EarlyStooping, and modelCheckpoint etc.) This research adopted tensorboard callbacks to visualize the results during training process. It is also one of the advantages to use Google Colab as the platform.



* Fitting The Model

The deep learning model is used training datasets (X\_ohlcv\_train and X\_technical\_train) for fitting process.

* The datasets has been split out ten percent for validation purpose.
* The model has been executed 100 times (epochs=100) with batch size 32.
* The data in batch chunks is shuffled.
* Tensorboard callbacks is simultaneously activated to record the training and validation process.



3.4 Evaluation Methods

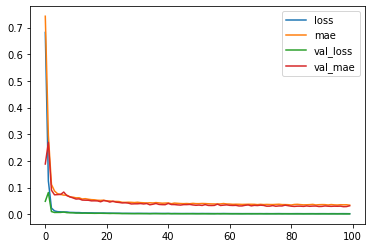
This research used three parts to perform evaluation: 1) Model Fitting Process, 2) Model Evaluation and 3) Comparison Between Actual and Predicted Values.

3.4.1 Model Fitting Process

To handle the overfitting and underfitting issues, machine learning is mainly to propose comparison between training datasets and testing datasets. Deep learning (neural networks) is slightly different to machine learning, i.e. keep tracks to loss function and metrics on each epoch via model fitting process. The model automatically takes records on each epoch (Figure 3.4.1.1), using model.history.history.

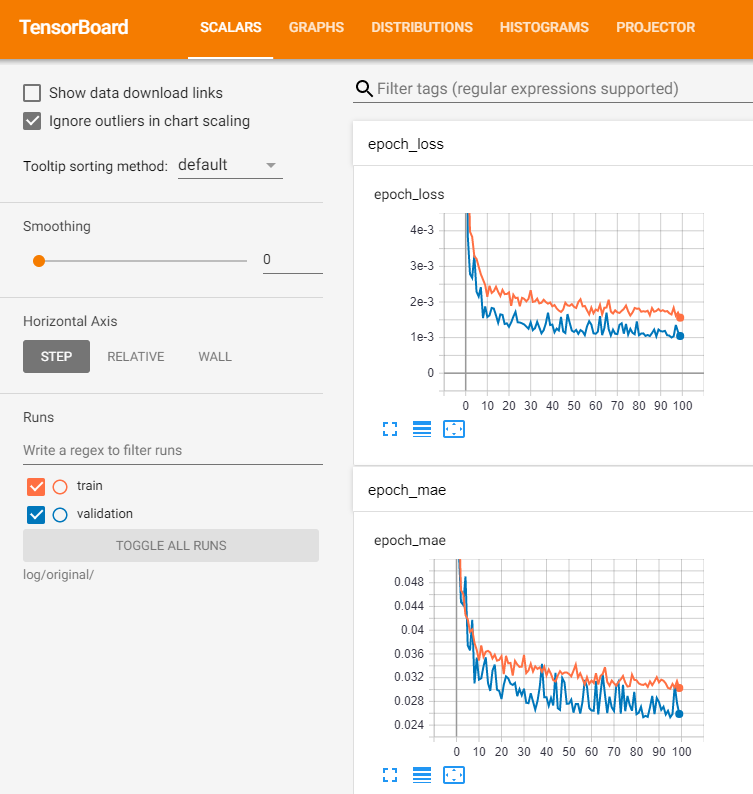


*Figure 3.4.1.1: Epoch Records on model.history.history (OHCL model)*



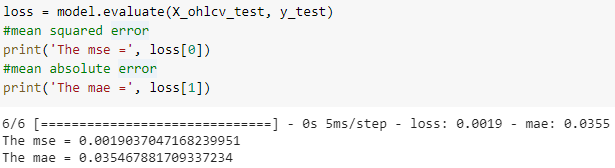
Tensorboard

Tensorboard is a tool to visualize the model fitting process and this research only adopts the records on epochs, which indicates loss function (mean-square-error) and metrics (mean-absolute-error). According to the figure 3.4.1.2, which is comparison model (OHLC, single input) the orange line is training and the blue one is validation. Against the weight adjustment (back propagation) on each epoch, both two measures are gradually decreasing, i.e. the model keeps improving by self.

*Figure 3.4.1.2 Epoch Records on Tensorboad (OHLC model)*  


3.4.2 Model Evaluation

Keras model includes an inner function evaluate() to examine testing dataset or other external datasets with same data format. Also, this evaluation generates the mean-square-error and mean-absolute error. The results are compared with the results in model fitting. The model structure will be modified if the results are obviously different.

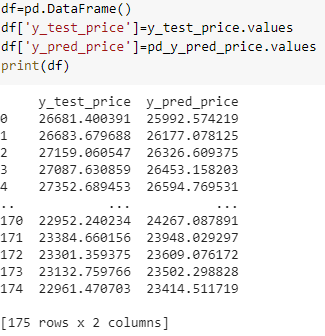


3.4.3 Comparison Between Actual and Predicted Values

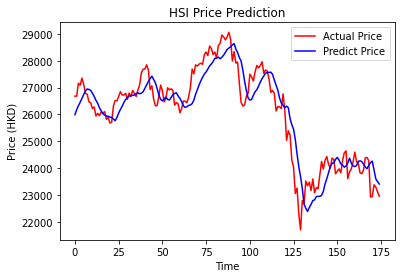
Keras model also predict values / categories using function predict() / predict\_classess(). This research uses predict() for index price prediction on testing dataset. The values predicted has been inversely transformed to be exact scale of index prices due to MinMaxScaler’s transformation in data pre-processing stage.



The actual values and predicted values are merged into one dataframe (OHLC as the example, totaling 175 trading days for testing). The plot indicates the differences between two values (Figure 3.4.3.1).

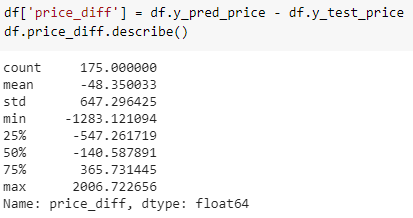


*Figure 3.4.3.1 Actual Values and Predicted Values on HSI Price (OHLC Model)*



Statistical Description

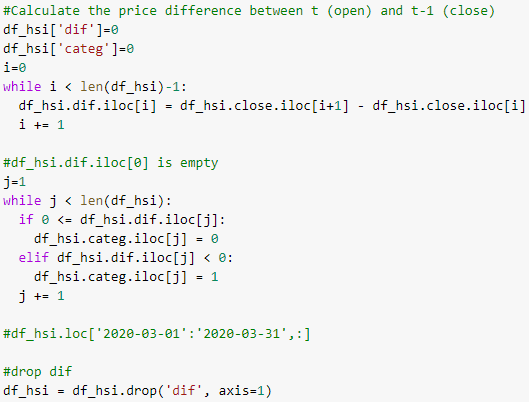
This research calculates the differences between actual values and predicted values. Despite the mean-square-error and mean-absolute error, these statistical descriptions helps to identify the accuracy of the prediction models. Also, the absolute differences between minimum and maximum are calculated as “RANGE” (refer to the table in Chapter 4).



3.5 Classification

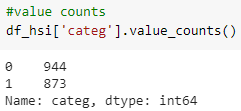
The purpose of classification model is to compare the results with regression model. This classification model adopts support vector regression methods as feature selection (refers to subchapter 3.3.6).

For the predict values, this is categorized as two classes (named ‘categ’ column), one for price drop in next day, and zero for price rise in next day. It is the model related to binary classification. It is noted that there is no feature related to next day price, except the class.

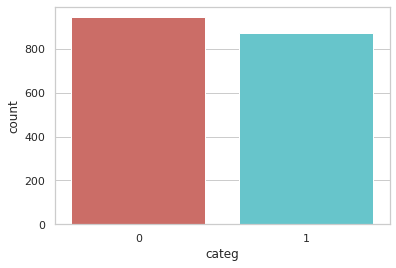


Counting

For the ‘categ’ column, it is counted the days indicating rises (944 days) rather than the days indicating drops (873 days).



*Figure 3.5.1 Counts of Next Day Trends*



Feature Importance

This classification model adopts support vector regression for feature selection. This dataset contains only one different column correspondent to same methods on regression – prediction column. The importance on features are totally different, which including:

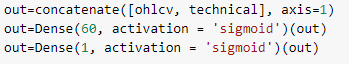
* ADXR 14 & 50: Average Directional Movement Index Rating with time period 14 & 50 days
* MFI 25 & 50: Money Flow Index with time period 25 & 50 days
* PLUS\_DI 7: Plus Directional Indicator with time period 7 days
* PLUS\_DM 100: Plus Directional Movement with time period 100 days
* HT\_DCPERIOD: Hilbert Transform - Dominant Cycle Period
* KAMA 25: Kaufman Adaptive Moving Average with time period 25 days
* NATR 20 & 25: Normalized Average True Range with time period 20 days & 25 days



Keras Model

It is noted that the prediction column ‘categ’ contains numbers of one and zero (label encoding, 1-D array), which is binary classification. Therefore, one-hot encoding method (2-D array) is not applied to prediction column.

The structure of deep learning model is basically unchanged. Only the activation functions of final outputs are changed to be two sigmoid functions because sigmoid function is more appropriate for the binary classification.



Besides, the model compile function is not applied mean-square-error and mean-absolute-error. The loss function is set to be ‘binary\_crossentropy’ for binary classification and the metrics is set to be accuracy.



Evaluation

The inner function model.evaluation() is adopted for testing dataset.



Other than that, confusion matrix and classification report are also adopted in this classification evaluation.





The results of this classification will be shown in Chapter 4 and discussed how it does a counter-example.

3.6 Chapter Summary

This chapter has explained the methodology of this research, the development from problem statements to model building and evaluation. Also, this development plan is consistent with data product development lifecycle. Especially the deep learning model produces different results compared to machine learning model, because there are more options and hyperparameters for modification. The next chapter will be discussed about the model results.

**CHAPTER 4:**

**RESULTS AND EVALUATIONS**

**4.1 Results**

As per the evaluation methods stated in Chapter 3, this research used three parts to perform evaluation: 1) Model Fitting Process, 2) Model Evaluation and 3) Comparison Between Actual and Predicted Values. The followings are visualizing the results first and then discussed on the results.

4.1.1 Model Fitting

This part of results is related to the process in model fitting on training datasets, including validations (Figure 4.1.1.1 to Figure 4.1.1.14). The model fitting process takes ten percent of training dataset purposed for validation. For the regression models, less mean-square-errors and mean-absolute-errors mean better model. On the contrary, higher accuracies and lower losses indicate better models.

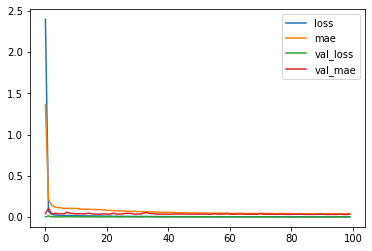
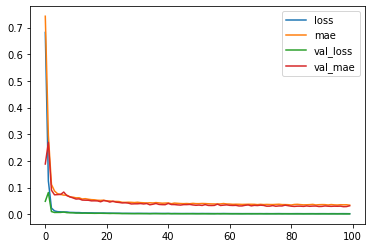
Regarding the training dataset, several findings are observed, which includes:

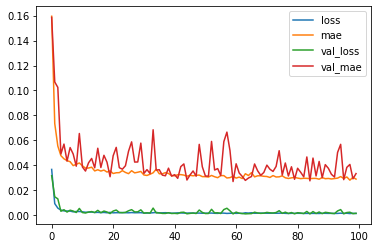
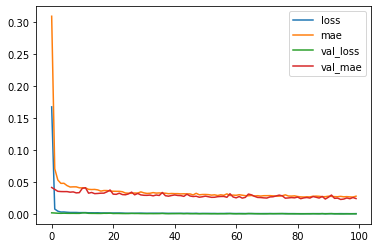
* The mean-square-errors and mean-absolute-errors were relatively large in the first few epochs.
* The model of Extreme Gradient Boosting (XGB) generated the closest values with the validation.
* The single-input model (OHLC) generated the largest values of mean-absolute-errors.
* The models of Average True Range (ATR) and Relative Strength Index (RSI) had bigger fluctuation in mean-square-errors with validation, i.e. apparently higher and lower than validation. These models might not be stable in reality.
* No obvious phenomenon indicated that the models are overfitting.

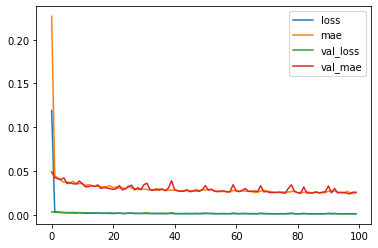
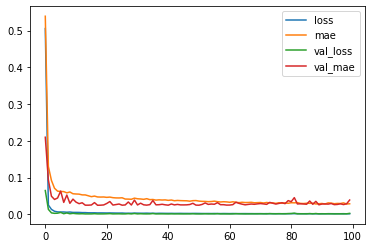
For the classification model with Support Vector Regression,

* The accuracies were stable within the range 0.5 and 0.6.
* The validation accuracies occurred sudden drops to 0.45.
* No obvious improvement was identified on the losses (binary\_crossentropy) during 100 epochs.
* An ideal model in classification is reducing the loss lower than the accuracies, but this one does not.
* No obvious phenomenon indicated that the model is overfitting.

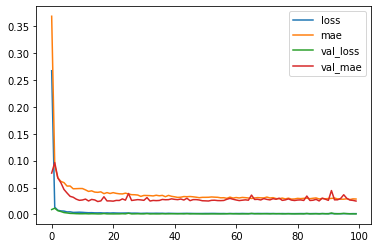
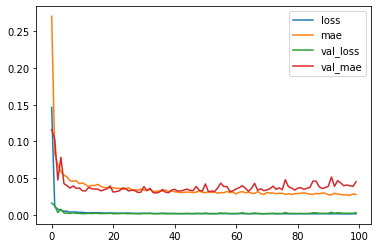
*Figure 4.1.1.1 Model Fitting (OHLC)* *Figure 4.1.1.2 Model Fitting (XGB)*

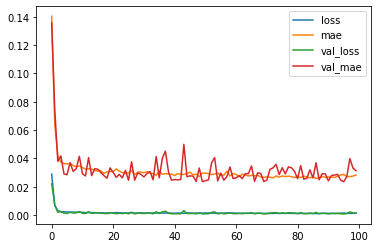
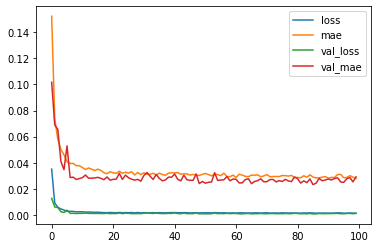


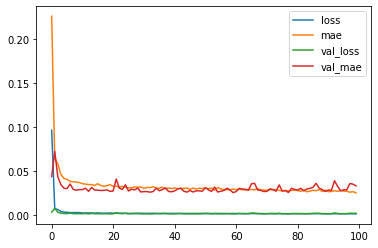
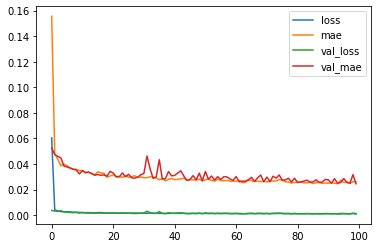
*Figure 4.1.1.3 Model Fitting (SVR)* *Figure 4.1.1.4 Model Fitting (ATR)*  


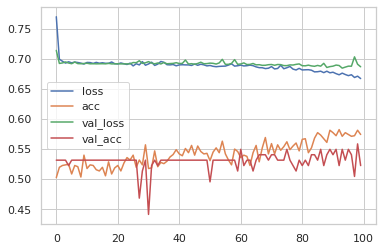
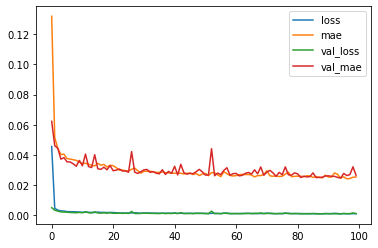
*Figure 4.1.1.5 Model Fitting (CCI)* *Figure 4.1.1.6 Model Fitting (EMA)*  


*Figure 4.1.1.7 Model Fitting (MFI)* *Figure 4.1.1.8 Model Fitting (MOM)*



*Figure 4.1.1.9 Model Fitting (ROC)* *Figure 4.1.1.10 Model Fitting (RSI)*  


*Figure 4.1.1.11 Model Fitting (SMA)* *Figure 4.1.1.12 Model Fitting (WILLR)*  


*Figure 4.1.1.13 Model Fitting (WMA)* *Figure 4.1.1.14 Model Fitting (SVR – Classification)*  


4.1.2 Evaluation Results

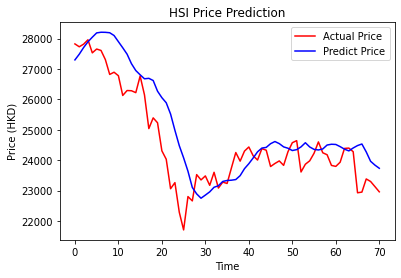
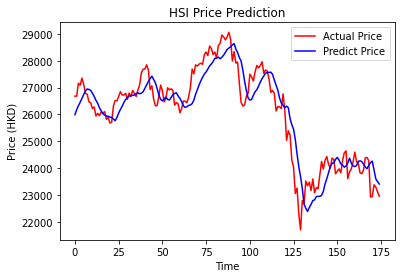
This part is to evaluate the testing dataset. The dataset is sorted by date from oldest to newest, such that the testing dataset included the index price on 2020, which has a sudden drop and big v-cut rebound. It is noted that no cross validation (without shuffling) is applied. Firstly, The prices between actual values and predicted values are visualized in line plots (Figure 4.1.2.1 – Figure 4.1.2.13). Then those two types of errors and other statistical descriptions between actual values and predicted values are demonstrated in table (Table 4.1.2.14). At last, the results indicated that the classification model with Support Vector Regression is the counter-example (Table 4.1.2.15 and Table 4.1.2.16).

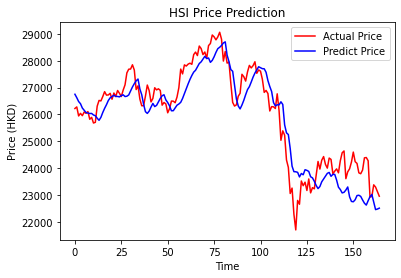
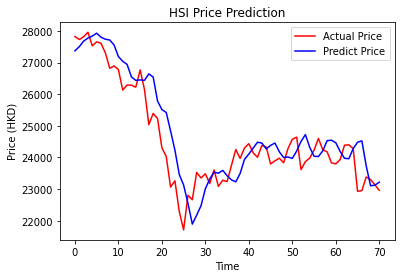
It is noted that the testing datasets of type I feature construction (XGB and SVR) contain only 71 trading days. Type II feature construction (remaining model except classification) and OHLC model contain 165 and 171 trading days.

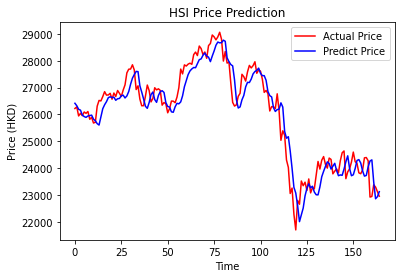
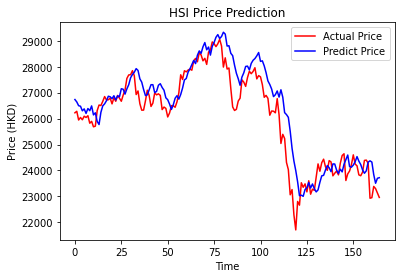
Regarding these prediction plots, several findings are observed, which includes:

* The predicted values of type I feature construction (XGB and SVR) had a huge time lags with actual values. Some predictions were opposite directions to actual values.
* The fluctuation of XGB model was smoother than the actual one, i.e. cannot predict the prices in proper trends.
* Some models were observed a huge price difference (over 2,000), e.g. ATR, MFI, ROC, RSI and WILLR.
* The models SMA, EMA and WMA were almost close to actual values.
* All models are not able to handle the big V-cut pattern.

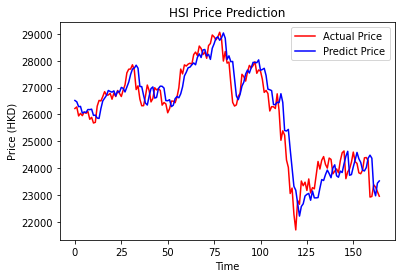
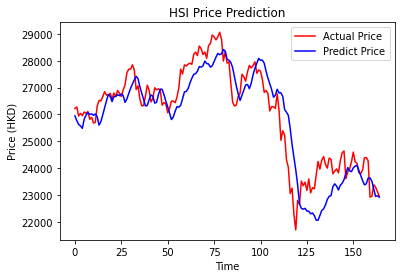
*Figure 4.1.2.1* *Prediction (OHLC)* *Figure 4.1.2.2 Prediction (XGB)*

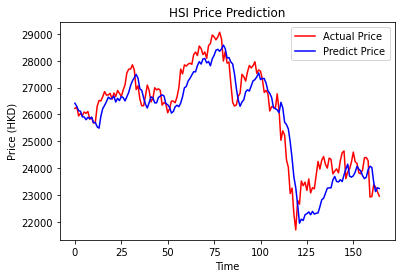
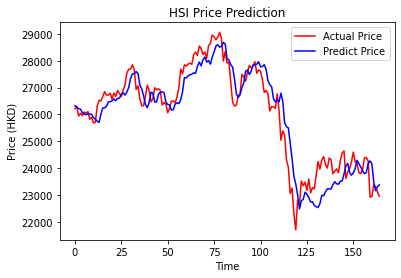


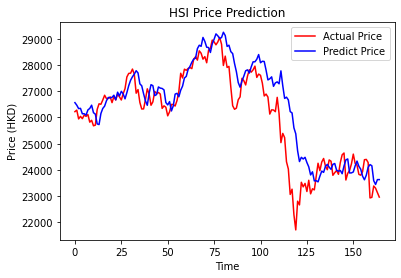
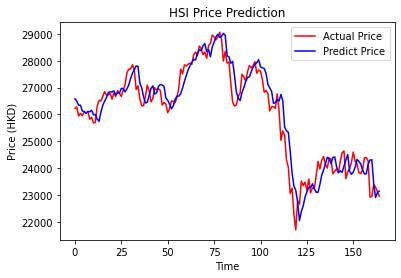
*Figure 4.1.2.3 Prediction (SVR)* *Figure 4.1.2.4 Prediction (ATR)*  


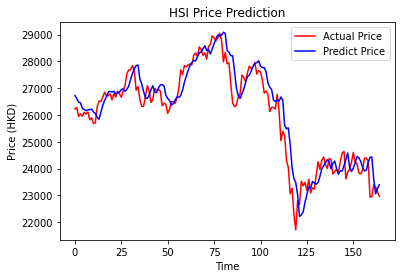
*Figure 4.1.2.5 Prediction (CCI)* *Figure 4.1.2.6 Prediction (EMA)*  


*Figure 4.1.2.7 Prediction (MFI)* *Figure 4.1.2.8 Prediction (MOM)*



*Figure 4.1.2.9 Prediction (ROC)* *Figure 4.1.2.10 Prediction (RSI)*  


*Figure 4.1.2.11 Prediction (SMA)* *Figure 4.1.2.12 Prediction (WILLR)*  


*Figure 4.1.2.13 Prediction (WMA)*  


The function model.evaluate() was adapted testing datasets without shuffling (only in training datasets). This part of evaluation is mainly to review the numerical results, such as mean-square-errors, mean-square-errors and statistical values on the differences between actual values and predicted values.

The statistical data means the minimum, maximum, standard deviation and mean on differences (predicted values minus actual values), and the absolute range (RANGE in table) between minimum and maximum. It is purposed for identifying the accuracy of predictions by these statistical descriptions.

Referred to the Table 4.1.2.14, there are some findings:

* The range of all models were over 2500.
* The model OHLC had lower mean-square-error, and mean of differences, but relatively high mean-absolute-error.
* The models SMA, WMA, EMA and MOM had low mean-square-errors and mean-absolute-error, and standard deviation & ranges on differences.
* The model WILLR had the lowest minimum and the largest maximum.
* The values of mean-square-error and mean-absolute-error on models XGB, MFI, ATR and WILLR were unsatisfied.

*Table 4.1.2.14 The Evaluation and The Statistical Descriptions*

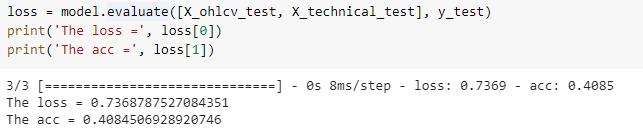
*\* Differences between predicted values minus actual values  
\*\* Absolute differences between minimum (MIN) and maximum (MAX)*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **MSE** | **MAE** | **\* MIN** | **\* MAX** | **\*\* RANGE** | **\* STD** | **\* MEAN** |
| **OHLC** | 0.0019 | 0.0355 | -1283 | 2007 | 3290 | 647.30 | -48.35 |
| **XGB** | 0.0038 | 0.0477 | -888 | 2455 | 3343 | 774.46 | 498.76 |
| **SVR** | 0.0022 | 0.0369 | -1351 | 1775 | 3126 | 660.22 | 237.30 |
| **ATR** | 0.0023 | 0.0393 | -1847 | 2161 | 4008 | 692.38 | -201.29 |
| **CCI** | 0.0024 | 0.0358 | -1002 | 2449 | 3451 | 623.81 | 364.32 |
| **EMA** | 0.0014 | 0.0289 | -1259 | 1581 | 2840 | 541.29 | -91.81 |
| **MFI** | 0.0030 | 0.0432 | -2185 | 2412 | 4597 | 787.52 | -215.83 |
| **MOM** | 0.0013 | 0.0275 | -1343 | 1731 | 3074 | 542.68 | 65.95 |
| **ROC** | 0.0018 | 0.0336 | -1714 | 1859 | 3573 | 633.24 | -83.04 |
| **RSI** | 0.0022 | 0.0374 | -1912 | 1921 | 3833 | 660.75 | -240.76 |
| **SMA** | 0.0013 | 0.0273 | -1144 | 1703 | 2847 | 531.80 | 85.91 |
| **WILLR** | 0.0036 | 0.0403 | -792 | 3668 | 4460 | 787.57 | 413.61 |
| **WMA** | 0.0015 | 0.0290 | -1259 | 1817 | 3076 | 557.85 | 153.30 |

Support Vector Regression - Classification

The classification model is the counter-example for the deep learning prediction, which is a fail trial. The reason is as follows:

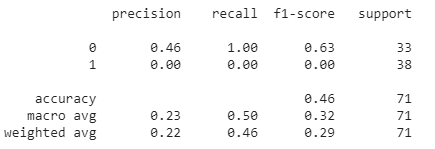
* The function evaluate() results in testing dataset was very low accuracy (0.4085), worser results in training dataset.
* In confusion matrix (Table 4.1.2.15), it is observed that the model only predicted one result (i.e. zero, uptrend) in binary classification.
* In classification report (Table 4.1.2.16), the values (precision, recall) cannot be referenced because of abnormal binary classification.



*Table 4.1.2.15 Confusion Matrix on SVR – Classification*



*Table 4.1.2.16 Classification Report on SVR – Classification*



**4.2 Discussions**

To conclude all of the results in subchapter 4.1, the research is assessed that:

* The models with the type II feature construction performed better than the type I feature construction, even type I feature construction also included the features in type I (slightly different in time periods).
* The predicted values of type I feature construction (XGB and SVR) had a huge time lags with actual values. Some predictions were opposite directions to actual values. Type I feature construction (XGB and SVR) might not be a better feature engineering method.
* The feature selection with machine learning methods had poor performance, only a slight fluctuation on index prices.
* Many models performed not accurate in sudden drop and then a big v-cut rebound.
* The results of technical indicators SMA, WMA, EMA and MOM were quite close to the actual values, even the huge drop and big v-cut rebound, so that they are the most appropriate to be features.
* Those methods of feature selection are not suitable in this prediction.
* Few kinds of technical indicators (ATR, WILLR, MFI, ROC and RSI) were not recommended for the index price prediction.
* The comparison model OHLC with single input proved that the second input (technical indicators) improved deep learning models.

Support Vector Regression – Classification

Besides, the results showed that the classification model cannot normally perform binary classification (only one predicted value). It is assessed that the problem of gradient vanishing has not been solved by Long Short Term Memory. The classification model requires a deep learning model with sophisticated layer structures and a better method of data engineering, such as auto-encoder, fourier transformation for noise reduction etc.

**CHAPTER 5:**

**CONCLUSIONS**

**5.1 Key Findings**

All deep learning models with regression depend on features. The comparison model OHLC with single input proved that the second input (technical indicators) helped to improve deep learning models. The model results indicate that the calculation of moving average is the good features in price predictions with regression. The machine learning approach (Extreme Gradient Boosting and Support Vector Regression) on feature selection has poor performance. Also, the data generator, which including the data with 50 previous consecutive trading days, is an important technique in deep learning model.

As Fancois (2018) stated that financial markets contain various statistical characteristics than natural phenomena and forecasting future price is not dependent on past performance, the classification indicates that it is not a proper method to predict stock prediction.

**5.2 Limitations**

This project approach mainly focused on data engineering, comparing with machine learning approaches on two types of feature constructions. The numbers of neurons and the layer structures in deep learning model are fixed. Also, the models and weights can be saved and re-loaded, which is one of the variants to the results. This improvement efficiency approach of save and re-loaded is not examined in this project.

**5.3 Recommendations and Further Studies**

This research only adopted public data (Open, High, Low, Close and Volume). In future studies, the dataset can use intra-day trading data. The further studies are recommended to be hybrid approaches with sentiment analysis on financial news, and macro-economy indicators; or to work on the case of sudden drop and v-cut rebound in 2020. Other than that, the section in feature engineering can adapt various methods, not only supervised learning in this research, for example, unsupervised learning (Principal Component Analysis, Auto-Encoder), or other transformation (Fourier Transformation for noise reduction).

Tensorflow.Keras has various approaches for hyperparameters tuning, such as sophisticated layer structures (not only Long Short Term Memory and Dense Layer, but also Convolutional Neural Network or more advanced techniques), different types of optimizers, customized callbacks functions for monitoring model fitting and re-load weight for the model fitting etc.

Besides, there are other trends of financial technology, algorithm trading (reinforcement learning), and stock analysis. These are also challenging topics for deep learning and machine learning.

**5.4 Social, Ethical, Professional and Legal Issues**

Client purposed to develop innovations of technology sides and kept close tracks on trendy direction of finance technology. Though this research adopted public data with no personal privacy concerns, there are several concerns on the adoption of these deep learning models for the client’s side:

* Social issues

The insight provided from the predication model is purposed for internal use. The poor performance causes reputation risk and loss market shares to the client.

* Ethical issues

The prediction model is a kind of tool for recommendation for experience user but still required some knowledges and not blindly following the forecast. Misunderstanding or misuses on the model may produce wrong investment, e.g. uptrend signal of index occurs, but some stocks may be downtrend due to significant information disclosure (profit warning).

* Professional issues

The model is only applied to Hong Kong Hang Seng Index, not universal model on various stock markets. Misuse on model may produce investment loss and reputation risk. The model users should be well-known the model usage. The prediction has no guarantee accuracy.

* Legal issues

Client must not directly provide the prediction result to its customers as the investment advices. Wrong prediction may make investment loss to client’s customers. Client should be known that investment advice should be provided by licensed person and client has to make a clear internal announcement for the usage of prediction model.

To conclude, these concerns are mainly related to the use of models. Sufficient instructions, user’s manual and user training can solve these concerns.

**5.5 Conclusions**

In summary, this research has fulfilled the client requirements (Chapter 1.3) and project objectives (Chapter 1.4). The aim of this research is to use deep learning as the main method to perform stock index market prediction with various settings to provide predictive insights for client. This research provides the literature review for related works, and demonstrates the complete data product development methodology and result evaluation with result visualization. Also, the research models adopted deep learning framework, which including Tensorflow.Keras, Long Short Term Memory, feature engineering, regression, and used the second input with technical indicators as the dependent variables.

**REFERENCES**

[1] Alice, Z., & Amanda, C. (2018). *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientist*. 1st edn. California: O’Reilly Media, Inc.

[2] Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016). *Deep learning for stock prediction using numerical and textual information*. 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS).

[3] Andreas, C., M., & Sarah, G. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. 1st edn. California: O’Reilly Media, Inc.

[4] Ankur, A., P. (2019). *Hands-On Unsupervised Learning Using Python*. 1st edn. California: O’Reilly Media, Inc.

[5] Antonio, G., & Sujit, P. (2017). *Deep Learning with Keras: Implement neural networks with Keras on Theano and Tensorflow*. 1st end. Birmingham: Packt Publishing.

[6] Arjun, R., & Suprabha, K., R. (2020). *Modeling Hybrid Indicators for Stock Index Prediction*. Advances in Intelligent Systems and Computing, vol 940. Springer, Cham

[7] Asghar, M. Z., Rahman, F., Kundi, F. M., & Ahmad, S. (2019). *Development of stock market trend prediction system using multiple regression*. Computational and Mathematical Organization Theory.

[8] Aurelien, G. (2019) *Hands-on Machine Learning with Scikit-Learn, Keras and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd edn. California: O’Reilly Media, Inc.

[9] Batra, R., & Daudpota, S. M. (2018). *Integrating StockTwits with sentiment analysis for better prediction of stock price movement*. 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET).

[10] Brett, S. (2015). *Effective Python: 59 specific ways to write better python*. 1st edn. New Jersey: Pearson Education Inc.

[11] Cao, J., & Wang, J. (2019). *Exploration of stock index change prediction model based on the combination of principal component analysis and artificial neural network*. Soft Computing.

[12] Chavez, M. (2017). *Data, Computing, and Transformation in the Financial Industry*. Symposium in Harvard’s Institute for Applied Computational Science, Cambridge, MA, United States, 7 March

[13] Chen, C., Zhao, L., Bian, J., Xing, C., & Liu, T.-Y. (2019). *Investment Behaviors Can Tell What Inside: Exploring Stock Intrinsic Properties for Stock Trend Prediction*. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD’19.

[14] Chen, Y., & Hao, Y. (2017). *A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction*. Expert Systems with Applications, 80, pp. 340-355.

[15] Chong, E., Han, C., & Park, F. C. (2017). *Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies*. Expert Systems with Applications, 83, pp. 187–205.

[16] Chris, A. (2018). *Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning*. 1st edn. California: O’Reilly Media, Inc.

[17] David, K. (2019). *Classic Computer Science Problems in Python*. 1st edn. New York: Manning Publications.

[18] Deng, S., Zhang, N., Zhang, W., Chen, J., Pan, J. Z., & Chen, H. (2019). *Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network*. Companion Proceedings of The 2019 World Wide Web Conference.

[19] Fancois, C. (2018). *Deep Learning with Python*. 1st edn. New York: Manning Publications.

[20] Gavin, H. (2017). *Mastering Machine Learning with Scikit-Learn: Learn to implement and evaluate machine learning solutions with scikit-learn*. 2nd edn. Birmingham: Packt Publishing.

[21] Giuseppe, B. (2019). *Hands-On Unsupervised Learning with Python: Implement machine learning and deep learning models using Scikit-Learn, TensorFlow and more*. 1st edn. Birmingham: Packt Publishing.

[22] Ivan, V., Daniel, S., Gianmario, S., Peter, R., & Valentino, Z. (2019). *Python Deep Learning: Exploring deep learning techniques and neural network architectures with PyTorch, Keras, and Tensorflow*. 2nd edn. Birmingham: Packt Publishing.

[23] Jake, V. (2017). *Python Data Science Handbook: Essential Tools for Working with Data*. 1st edn. New York: O’Reilly, Inc.

[24] Jiang, M., Liu, J., Zhang, L., & Liu, C. (2020). *An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms*. Physica A: Statistical Mechanics and its Applications, 541, 122272.

[25] Jojo, M. (2019). *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python*. 1st edn. New York: Springer Science+Business Media & Apress.

[26] Julian, A., Trent, H. (2017). *Scikit-Learn Cookbook: Over 80 recipes for machine learning in Python with scikit-learn*. 2nd edn. Birmingham: Packt Publishing.

[27] Kamble, R. A. (2017). *Short and long term stock trend prediction using decision tree*. 2017 International Conference on Intelligent Computing and Control Systems (ICICCS).

[28] Khattak, A. M., Ullah, H., Khalid, H. A., Habib, A., Asghar, M. Z., & Kundi, F. M. (2019). *Stock Market Trend Prediction using Supervised Learning*. Proceedings of the Tenth International Symposium on Information and Communication Technology - SoICT 2019.

[29] Kumar, I., Dogra, K., Utreja, C., & Yadav, P. (2018). *A comparative study of supervised machine learning algorithms for stock market trend prediction*. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 1003-1007). IEEE.

[30] Larson, D., & Chang, V. (2016). *A review and future direction of agile, business intelligence, analytics and data science*. Elsevier, International Journal of Information Management, 36(5), pp. 700–710.

[31] Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F., … Deng, X. (2014). *Empirical analysis: stock market prediction via extreme learning machine*. Neural Computing and Applications, 27(1), pp. 67–78.

[32] Liu, D. H., & Wang, J. J. (2018). *A pca-lstm model for stock index prediction*. DEStech Transactions on Engineering and Technology Research, (ecar).

[33] Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). *Stock market index prediction using artificial neural network*. Journal of Economics, Finance and Administrative Science, 21(41), pp. 89–93.

[34] Nelson, D. M. Q., Pereira, A. C. M., & de Oliveira, R. A. (2017). *Stock market’s price movement prediction with LSTM neural networks*. 2017 International Joint Conference on Neural Networks (IJCNN).

[35] Newman, R., Chang, V., Walters, R. J., & Wills, G. B. (2016). *Model and experimental development for Business Data Science*. Elsevier, International Journal of Information Management, 36(4), pp. 607–617.

[36] Pang, X., Zhou, Y., Wang, P., Lin, W., & Chang, V. (2018). *An innovative neural network approach for stock market prediction*. The Journal of Supercomputing, pp. 1-21.

[37] Qi, Z., Bu, Z., Xiong, X., Sun, H., Cao, J., & Zhang, C. (2019). *A Stock Index Prediction Framework: Integrating Technical and Topological Mesoscale Indicators*. 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI).

[38] Qiu, M., Song, Y., & Akagi, F. (2016). *Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market*. Chaos, Solitons & Fractals, 85, pp. 1–7.

[39] Sebastian, R. (2015). *Python Machine Learning: Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics*. 1st edn. Birmingham: Packt Publishing.

[40] Sebastian, R., & Vahid, M. (2017). *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn and TensorFlow*. 2nd edn. Birmingham: Packt Publishing.

[41] Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). *Stock price prediction using LSTM, RNN and CNN-sliding window model*. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).

[42] Sinan, O., & Divya, S. (2018). *Feature Engineering Made Easy: Identify unique features from your dataset in order to build to build powerful machine learning systems*. 1st edn. Birmingham: Packt Publishing.

[43] Singh, R., & Srivastava, S. (2016). *Stock prediction using deep learning*. Multimedia Tools and Applications, 76(18), pp. 18569–18584.

[44] Song, Y. (2018). *Stock trend prediction: Based on machine learning methods* (Doctoral dissertation, UCLA).

[45] John, B. (mrjbq7) (2020). TA-Lib. [Online] Available at: <https://github.com/mrjbq7/ta-lib>

[46] Tsai, Y., & Zhao, Q. (2019). *An Experimental Study on the Effectiveness of Artificial Neural Network-Based Stock Index Prediction*. In 2019 International Conference on Machine Learning and Cybernetics (ICMLC) (pp.1-6). IEEE.

[47] Vargas, M. R., de Lima, B. S. L. P., & Evsukoff, A. G. (2017). *Deep learning for stock market prediction from financial news articles*. 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA).

[48] Wes, M. (2012). *Python for Data Analysis*. 1st edn. California: O’Reilly, Inc.

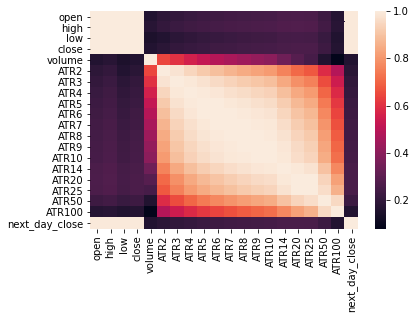
[49] Yves, H. (2018). *Python for Finance*. 2nd edn. California: O’Reilly, Inc.

[50] Zhang, L., Aggarwal, C., & Qi, G.-J. (2017). *Stock Price Prediction via Discovering Multi-Frequency Trading Patterns*. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD’17.

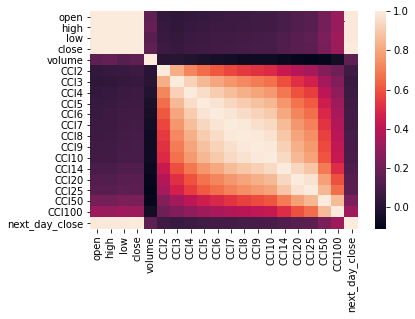
[51] Zhang, X., Li, A., & Pan, R. (2016). *Stock trend prediction based on a new status box method and AdaBoost probabilistic support vector machine*. Applied Soft Computing, 49, pp. 385–398.

**APPENDIX A – Correlation Plots on Datasets**

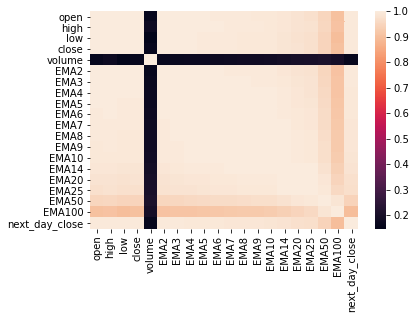
*Correlation Plots on (ATR)*



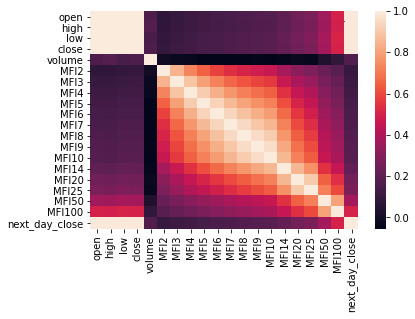
*Correlation Plots on (CCI)*



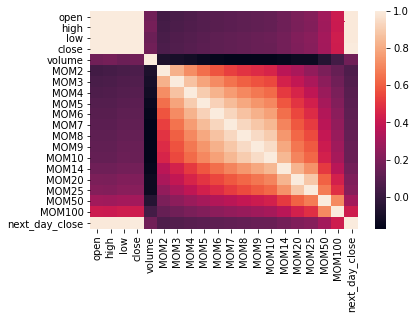
*Correlation Plots on (EMA)*



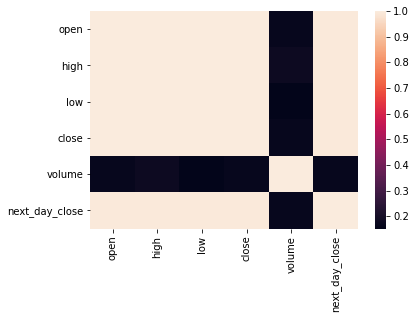
*Correlation Plots on (MFI)*



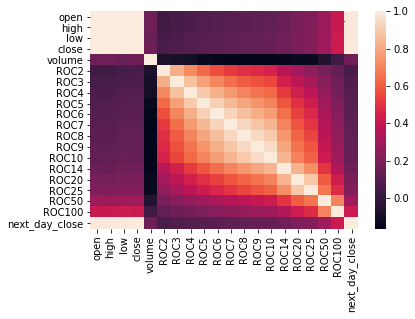
*Correlation Plots on (MOM)*



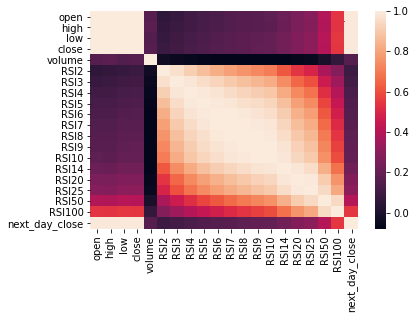
*Correlation Plots on (OHLC)*



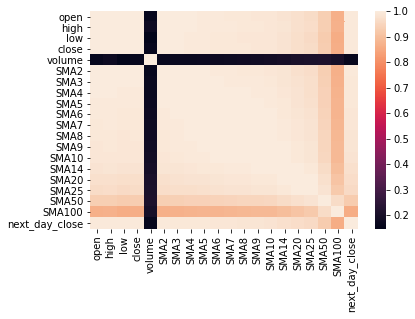
*Correlation Plots on (ROC)*



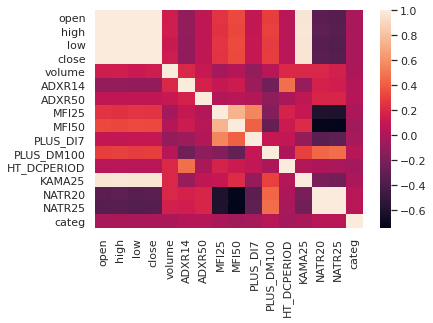
*Correlation Plots on (RSI)*



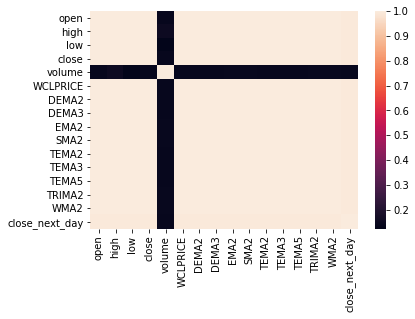
*Correlation Plots on (SMA)*



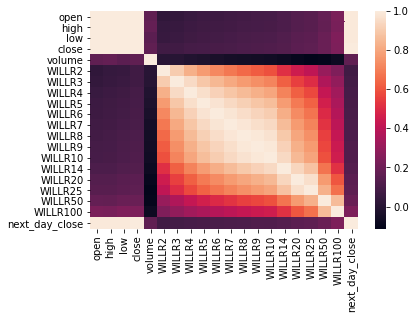
*Correlation Plots on (SVR - Classification)*



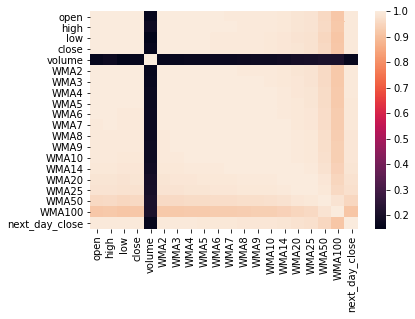
*Correlation Plots on (SVR)*



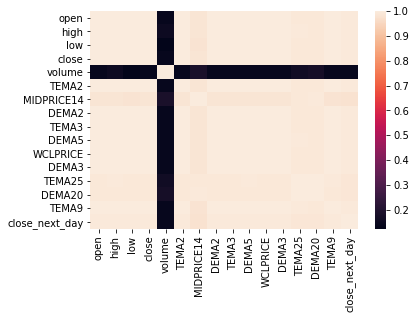
*Correlation Plots on (WILLR)*



*Correlation Plots on (WMA)*



*Correlation Plots on (XGB)*



**Appendix B – Files of Code and Results**

It is noted that those codes are preferred to be opened with Google Colab using the hyperlink (The Link of the Codes), due to the Tensorboard visualization. The code results also demonstrated the relevant visualizations.

**The Link of the Codes**

<https://drive.google.com/drive/folders/1MBZuPuobmKKXQ_kTiKgVEjyD8WNyo6_Y?usp=sharing>

**The Link of the Result Screens**

<https://drive.google.com/drive/folders/115BVmCahGTcXCalLtBHng3BuflhR8aIo?usp=sharing>

**Appendix C – Terms of Reference**

Terms of Reference File

