Documentation

Table of Contents

1 Introduction

Since 2021, fuel prices have risen across the world in line with global trends as the economy recovers from the COVID-19 lockdowns. This year in particular, the prices have hit record highs in many countries, including Malaysia. These soaring prices have negative impacts not only on consumers, but also on businesses. There are many uncertain and unforeseen factors that affect the petroleum fuel market. This makes the fuel price forecasting always a challenging problem.

Artificial Neural Network (ANN) is the field that has been studied by researchers since the mid-20th century. The concept of ANN is basically to develop machines that mimic the operation of biological brain neurons in the simplest form as shown in Figure 1, to recognize patterns and solve real-world problems [2]. This project aims to develop a machine learning model to predict the fuel prices in Malaysia using ANN technique.

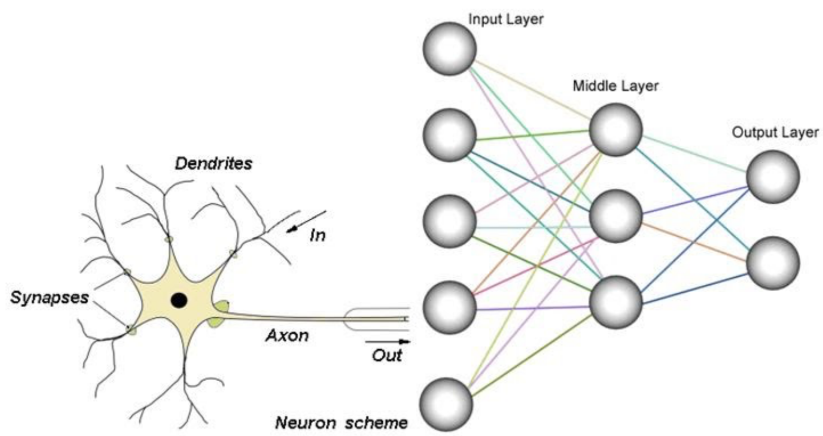

Figure 1: The brain neuron (left) and artificial neural network (right). [1]

1.1 Problem Statements

Gasoline, i.e., RON95 and RON97, as well as diesel are the fuels for vehicles in Malaysia. In 2022, the surging prices of RON97 has repeatedly break record highs, and it is still hitting new highs. Figure 2 and Figure 3 below show the prices of RON 97 from January 2020 to June 2022 and its yearly average price comparison. It is clearly shown that at the start of the coronavirus pandemic in 2020, roads were relatively empty, and fuel was cheap due to the lockdown. With the worldwide vaccination popularization, countries are ramping up business activities following the recovery of the economy. Not only because of the pandemic, there are other reasons that lead to increasing fuel prices such as soaring crude oil prices due to rising demand for crude oil and a tight market supply. And for the price of RON 95 and diesel, they are subsidized by the government with promising a ceiling price of RM 2.05 per liter and RM 2.15 per liter from 10th February 2021 onward [3].

Such price volatility caused inflation in the economy. The fuel subsidy program has shielded most households from higher fuel prices since basic cars usually can run on RON 95. However, inflation in other industries, especially the food industry, is still driving up the cost of living. Not to mention, rising global gasoline prices will affect the cost of imported materials in terms of transportation, production, etc. Thus, pushing up the price of goods and services.

Furthermore, forecasts of domestic fuel prices are rarely found on the Internet. After personal investigation, a total of two forecast system websites were found, one of which has terminated its services in September 2021, the remaining one is shown in [4].

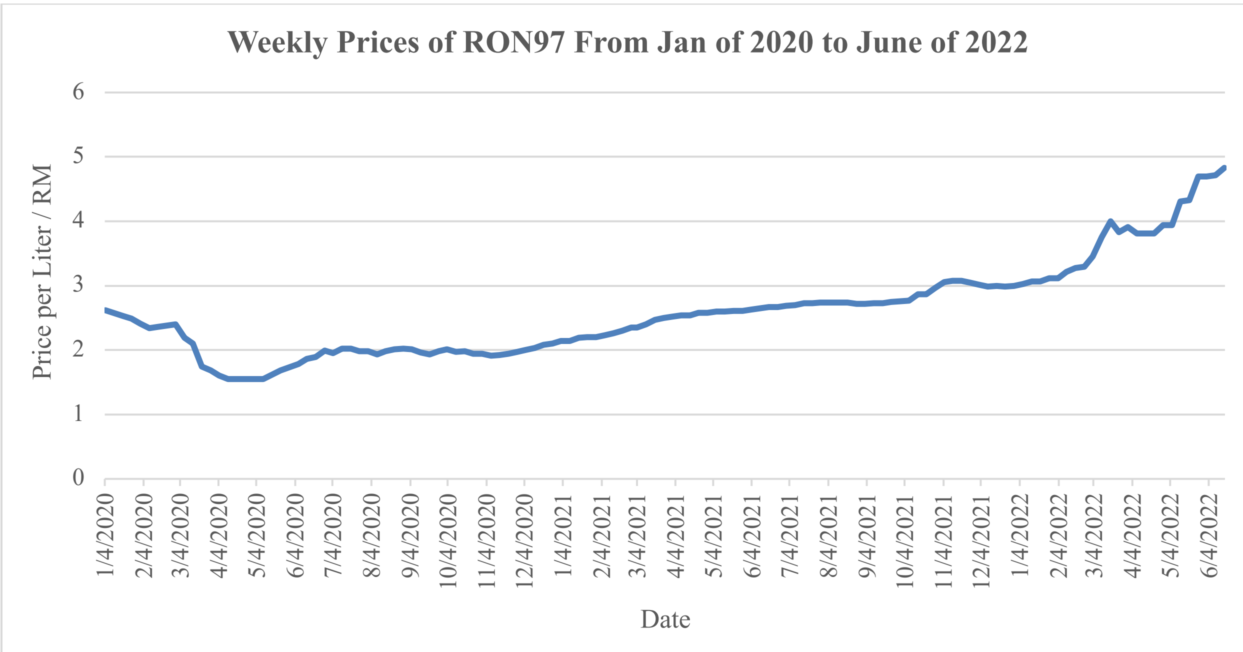

Figure 2: Weekly price of RON 97 from Jan 2020 to June 2022.

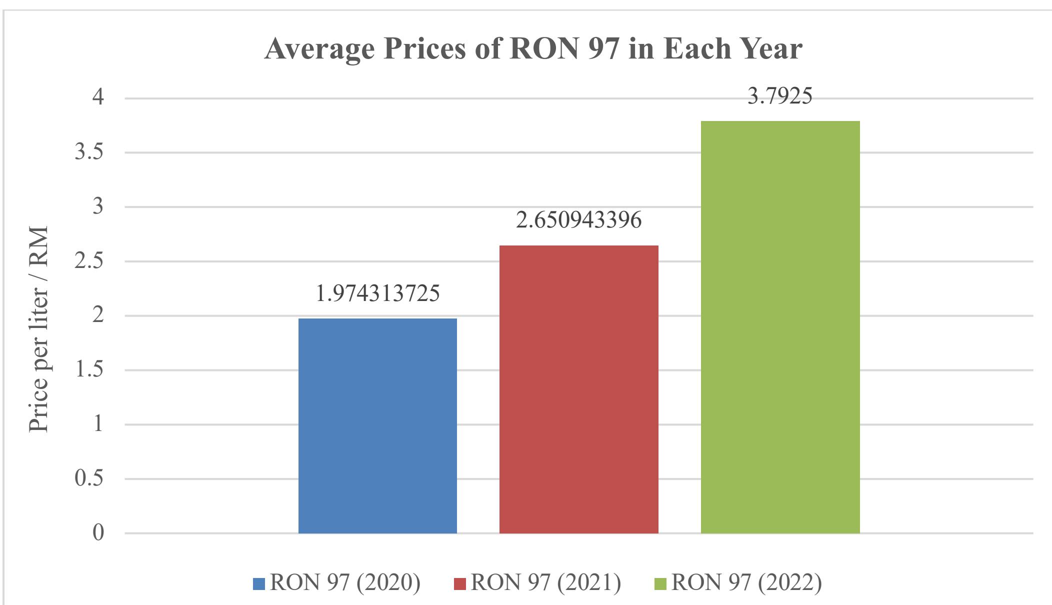

Figure 3: Average prices of RON 97 in each year.

1.2 Aims and Objectives

This project aims to develop a forecasting model using artificial neural network (ANN) to predict domestic gasoline and diesel prices, thereby providing citizens and businesses with insight into prices that may be correlated with economic inflation. The model outcomes are expected to be able to assist citizens prepare to adjust their daily spending and help businesses identify risk likelihood.

Following are the objectives of the project:

- To evaluate and develop the best suite ANN techniques in forecasting fuel prices.
- To develop an ANN model that is able to produce output with acceptable error value.
- To deploy the proposed model to an optimized platform that is able to present prediction results to audiences.

1.3 Project Scope

This project will mainly focus on the implementation of appropriate ANN techniques in model building, followed by an optimized platform.

Following are the requirements of the project:

- The proposed ANN model should be able to capture the changing patterns of actual fuel prices.
- The proposed ANN model should be able to outcome with predicted prices for the next three weeks on a weekly basis.
- The proposed ANN model should output with acceptable accuracy with not exceeding RM0.40 difference between actual and predicted values.
- The optimized platform should display the outcomes of the model in a well-defined manner.
- The optimized platform should display the record of historical fuel prices to allow audiences to view the weekly prices differences.

3 Methodology

This section will discuss the lifecycle and tools used for this project.

3.1 Project Lifecycle

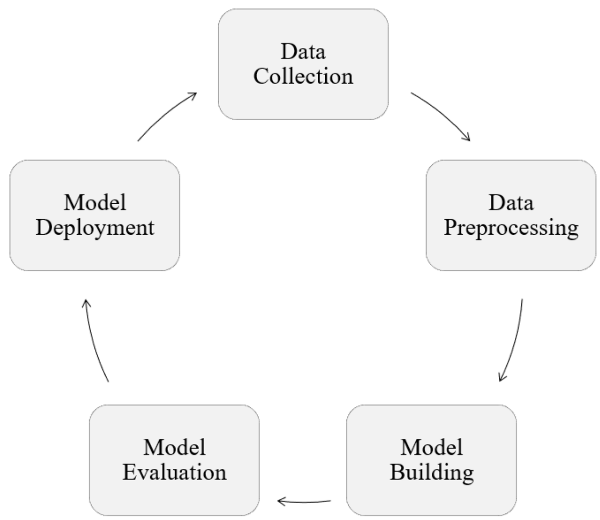

Figure 4: Stages involved in the lifecycle.

The lifecycle of this project is an iterative process where most of the steps will be repeated several times. This section aims to discuss the process done in each stage.

3.1.1 Data Collection

In machine learning, data are essential elements to be served as input and output variables for model training.

For this project, the raw data that has been collected are as follows:

- ***Gasoline Price***: Source from CompareHero [39] covered weekly prices of RON 95, RON 97 and diesel, and the timeline included from 1st January of 2019 to the present (November 2022). The data is obtained through web scraping technique using Python and is saved in CSV format.

- ***Crude Oil Price***: Source from U.S. Energy Information Administration [40-41] covered European prices of Brent and WTI, while source from OPEC [42] covered European prices of OPEC. The timeline of the crude oil data collected is on a daily-basis, which is different from the weekly-based gasoline prices. Data source [40-41] is retrieved in XLS file format, while data source [42] is in XML format.

- ***Exchange Rate from Ringgit Malaysia to U.S. Dollar***: Source from Yahoo Finance [43] covered exchange rates on open price, highest price, lowest price, and close price. Daily close price is the data collected in CSV file format.

3.1.2 Data Preprocessing

The next step is preprocessing the raw data to transform it in appropriate format for machine learning model.

The tasks of data preprocessing involved:

- ***Data Cleaning***: This is the process of correcting incomplete or inaccurate records from a dataset. For gasoline data, there are a few missing values. Backward fill is the technique used by replacing missing values with values from the next column.

- ***Data Integration***: This is the process of combining data residing in various sources into a single dataset to offer a unified view. Crude oils and exchange rates which are on a daily basis are converted to weekly-based by taking the average within one week and saved into CSV file format. The date index column from all the datasets is identified and converted into to "YYYY-MM-DD" format for merging purpose. After that, all the five datasets are combined into a unified single CSV dataset.

- ***Data Transformation***: This is the process of changing the data structure or value. The crude oils in USD per barrel unit are converted into MYR per liter unit, which is the unit same as fuel data for consistency as shown in Equation (10). Next step is normalizing the data to make model training faster. Scaling to range in (0, 1) using min-max normalization formula, as shown in Equation (11).

$$X'_d = \frac{X_d}{l} \times C_d, \tag{10}$$

where X_d is the d^{th} crude oil price in USD per barrel, C_d is the exchange rate, l is 1 barrel to liter, which is approximately 158.987295 and X'_d is the crude oil price in MYR per liter.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}, \tag{11}$$

where x_i is the i^{th} original data, z_i is the i^{th} normalized value.

3.1.3 Model Building

This is the process where the ANN model is built to be trained. The models are adjusted repeatedly to find out the best suite network architecture for this project. At the beginning of the model development, the design considerations such as hyperparameters are initialized, then modified according to the model output and metrics calculation in the next step, 3.1.4.

Before start building on the model, the relationship between the variables is investigated with the use of techniques below:

- ***Scatterplot***: It is useful to represent the relationship between two quantitative variables in graphical mode by observing the direction and form of the dots on the plots.

- ***Pearson's correlation coefficient***: It is used to evaluate the strength of the linear relationship between two variables. The nearer the output value to zero value, the weaker the linear relationship.

- ***Spearman's correlation coefficient***: It is used to evaluate the monotonic relationship between two variables which will be discussed under section 4.1.1.1.

After prior investigating the relationships, different combinations of input variables are evaluated for each output variable using an initial model architecture. The dataset is split into three sets: training, validation and testing sets with a ratio of 8:1:1. The small size for validation and testing sets is due to the fact that there are around 200 observations (data rows) only, and a larger training set is required to produce a well-trained model. Following is the initial model architecture:

- A LSTM model that is configured with:
  - dataset with number of steps of 3, which is a common number seen in most projects.
  - single hidden layer with 10 neurons numbers, which is a default number in platform like MATLAB.
  - ReLu activation function, as the minimum input values resulted will not lower than zero.
  - Adam optimizer with a default learning rate of 0.001, which has a fast convergence speed.
  - 50 number of epochs, which is a common starting number in most projects.

After evaluation (See Table 12), the sets of input data which have produced low RMSE value are chosen as list of input selection for further evaluation with different hyperparameters configuration. List of hyperparameters selection is as follows:

- ***Number of steps or timesteps***: 1, 2 and 3. As long input sequences may result in vanishing gradients, the selection is bounded at a maximum number of 3.

- ***Number of hidden neurons***: 2, 4, 6, 8, 10. This selection range is defined by referencing to the Equation (12).

$$N_h = \sqrt{N_i \times N_o}, \tag{12}$$

  where N_h is the number of hidden neurons, N_i is the number of input size, and N_o is the number of output size. N_i is defined as a number of 7, N_o is defined as a number of 3, and the result is around 4.58. Since the result is smaller than the one in initial configuration, the number that is within range of 10 and is dividable by 2 is chosen as one of selections.

- ***Activation function***: ReLu and sigmoid. Sigmoid is one of the common starting points to configure models.
- ***Learning rates***: 0.1, 0.01, 0.001. Learning rate tends to be positive values within 0 and 1. And logarithmic scale from 1e-1 to 1e-3 is chosen.
- ***Number of epochs***: 50, 100, 150. Epoch always increases in tens, hundreds and thousands. To avoid model training for long times, an upper bound of 150 is decided.

Evaluation in 3.1.4 has taken place again to reduce the configuration selection list. Then, the final validation is carried out to finalize the model configuration.

3.1.4 Model Evaluation

After model building and training, validation and evaluation are taken place to justify the model performance.

Performance measures that have been used for this project are as follows:

- MSE is used as loss function for the model as it is the default loss to use for regression problems, and it penalizes large errors.

$$MSE = \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2, \tag{13}$$

  where N is the total number of observations, y_t and $\hat{y}_t$ are the actual value and forecast value at time t.

- RMSE as MSE will yield a number with squared unit which does not make sense for the unit of price: $(RM\$)^2$, and it penalizes large errors too.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2} \tag{14}$$

Then moved to model building step to tune the hyperparameters. After the back and forth between these 2 steps and obtained a desirable result, model with finalized configuration is fed with new data or testing set to observe whether the model archives the project objectives.

3.1.5 Model Deployment

The predictive model is then prepared to be deployed to an optimized platform using Django. A simple html website is built with the models and test plan is carried out to ensure all the features are functioning smoothly.

3.2 Tools

Below are the tools that have been used in this project.

3.2.1 Coding Language

3.2.1.1 Python Programming Language

Python is one of the most popular languages for machine learning and data science due to its simplicity to read and write, flexibility to suite different usages, and extensive to rich selection of libraries support.

The tasks that have been done using Python are web scraping to collect gasoline price data and data preprocessing as mentioned in above lifecycle section. Free open-source Python libraries, i.e. BeautifulSoup4, Matplotlib, Pandas, Requests, TensorFlow and Scikit-learn are installed using pip command for model development.

3.2.1.2 HyperText Markup Language (HTML)

HTML is used to create a website to be displayed in web browser.

3.2.1.3 Cascading Style Sheet (CSS)

CSS is used to specify the presentation style of HTML elements for user interface designation.

3.2.2 Application

3.2.2.1 Visual Studio Code

The only code editor used in this project as it supports hundreds of languages and speeds up productivity with built-in features like auto-indentation and keyboard shortcuts.

3.2.3 Framework

3.2.3.1 Django

A back-end Python-based web framework which is easy to learn with rich of tutorial resources available on the Internet.

4 Result and Discussion

This section will discuss the result of the model developed.

4.1 Model Evaluation and Validation

Techniques used and result retrieved are analyzed to determine an optimal configuration from the selection list mentioned in Section 3.1.3.

4.1.1 Input Data Selection

Besides the three target variables, there are four possible input variables, i.e., Brent oil, WTI oil, OPEC oil and exchange rate of USD to RM that can be served into input layer.

4.1.1.1 Relationship Between Two Variables

For each target output variable, i.e., RON 95, RON 97 and diesel, scatterplot and correlation techniques are used to describe the relationship between each possible input variable and output variable.

Scatterplot

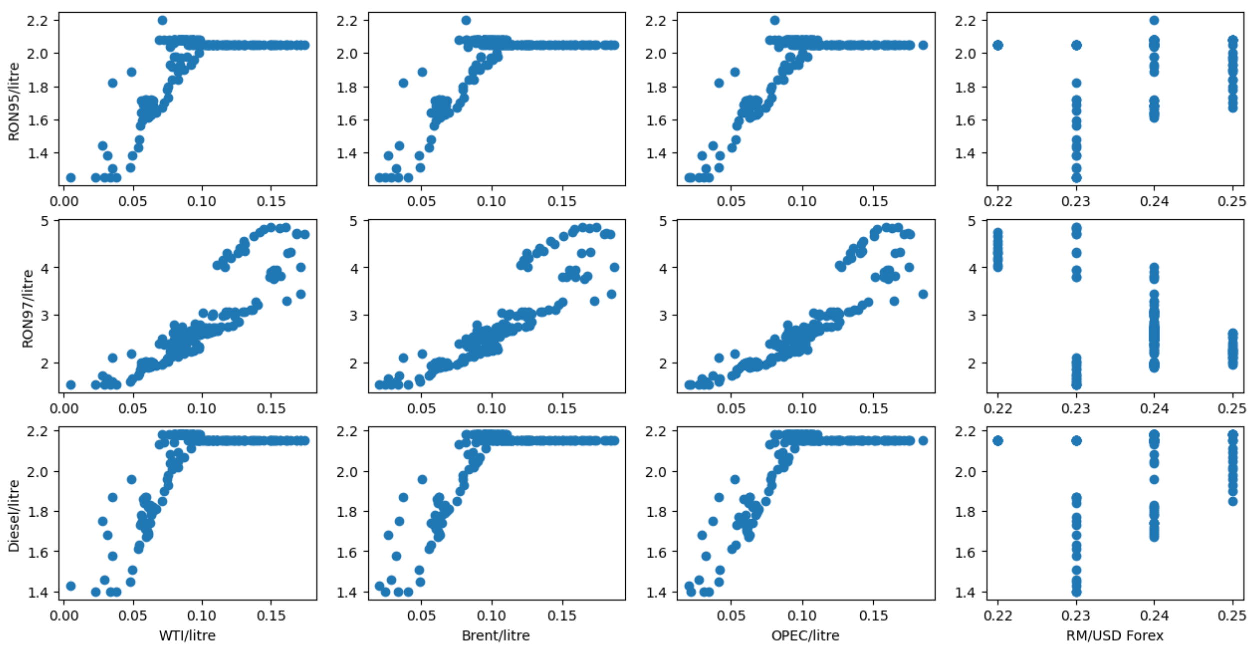

Figure 5: Scatterplot between each fuel and possible input variables.

From the above scatterplot of each fuel against each crude oil and exchange rate, it can be observed that the graphs of all fuels against all the crude oils are positive association. For RON97 with each crude oil, the plot pattern has shown there may be a strong linear relationship between each other. Whereas plots for both RON95 and diesel against each crude oil have shown an inverted Z-shaped pattern which can be observed that both fuel's linear relationships with each crude oil at the beginning becomes straight line at the middle timestamps till the end. This is due to the fuel subsidy by the government that cap both RON95 and diesel at around RM2. While plots for each fuel with exchange rates have shown multiple vertical lines which represents that there may be a weak relationship between each other. To obtain more precise evidence to understand the relationship between two variables, correlation coefficient is computed at below section to measure the strength of the relationship.

Pearson's correlation coefficient

	RON95/litre	RON97/litre	Diesel/litre	WTI/litre	Brent/litre	OPEC/litre	RM/USD Forex
RON95/litre	1.000000	0.574580	0.980877	0.672297	0.705522	0.723005	0.176937
RON97/litre	0.574580	1.000000	0.556557	0.882858	0.892034	0.902193	-0.492005
Diesel/litre	0.980877	0.556557	1.000000	0.664450	0.700753	0.717954	0.205144
WTI/litre	0.672297	0.882858	0.664450	1.000000	0.992273	0.989673	-0.169331
Brent/litre	0.705522	0.892034	0.700753	0.992273	1.000000	0.996531	-0.156182
OPEC/litre	0.723005	0.902193	0.717954	0.989673	0.996531	1.000000	-0.169437
RM/USD Forex	0.176937	-0.492005	0.205144	-0.169331	-0.156182	-0.169437	1.000000

Figure 6: Correlation coefficient using pandas Dataframe.corr(method=pearson).

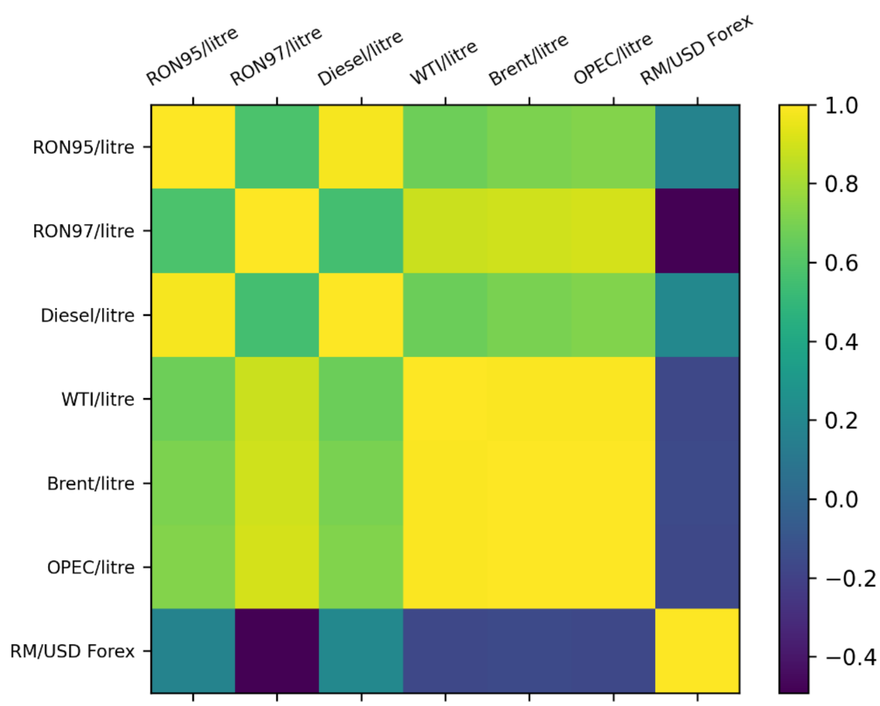

Figure 7: Heat map of Pearson's correlation matrix.

	WTI	**Brent**	**OPEC**	**RM/USD Forex**
RON 95	Moderate positive correlation	High positive correlation	High positive correlation	Negligible correlation
RON 97	High positive correlation	High positive correlation	Very high positive correlation	Low negative correlation

Diesel	Moderate positive correlation	High positive correlation	High positive correlation	Negligible correlation

*$\pm[.9\ to\ 1{:}\ Very\ high,\ .7\ to\ .9{:}\ High,\ .5\ to\ .7{:}\ Moderate,\ .3\ to\ .5{:}\ Low,\ .0\ to\ .3{:}\ Negligible]$

Table 1: Strength of Pearson's correlation coefficient

High positive correlation indicates that both variables change in the same direction. Whereas negative correlation indicates that both variables change in opposite direction (when one increases, another decreases). Besides, from the correlation, it can be found out that RON 95 and diesel are strongly linearly related to each other, and correlation between both fuels with each input variable have little differences. This indicates that the input data selection of both fuels may be the same as well as the hyperparameters selection.

Spearman's correlation coefficient

	RON95/litre	RON97/litre	Diesel/litre	WTI/litre	Brent/litre	OPEC/litre	RM/USD Forex
RON95/litre	1.000000	0.504284	0.909738	0.367815	0.401911	0.415317	0.188850
RON97/litre	0.504284	1.000000	0.457934	0.922980	0.936416	0.940888	-0.259302
Diesel/litre	0.909738	0.457934	1.000000	0.357252	0.405157	0.417631	0.266358
WTI/litre	0.367815	0.922980	0.357252	1.000000	0.988280	0.985191	-0.128474
Brent/litre	0.401911	0.936416	0.405157	0.988280	1.000000	0.997219	-0.135822
OPEC/litre	0.415317	0.940888	0.417631	0.985191	0.997219	1.000000	-0.142112
RM/USD Forex	0.188850	-0.259302	0.266358	-0.128474	-0.135822	-0.142112	1.000000

Figure 8: Correlation result using pandas Dataframe.corr(method=spearman).

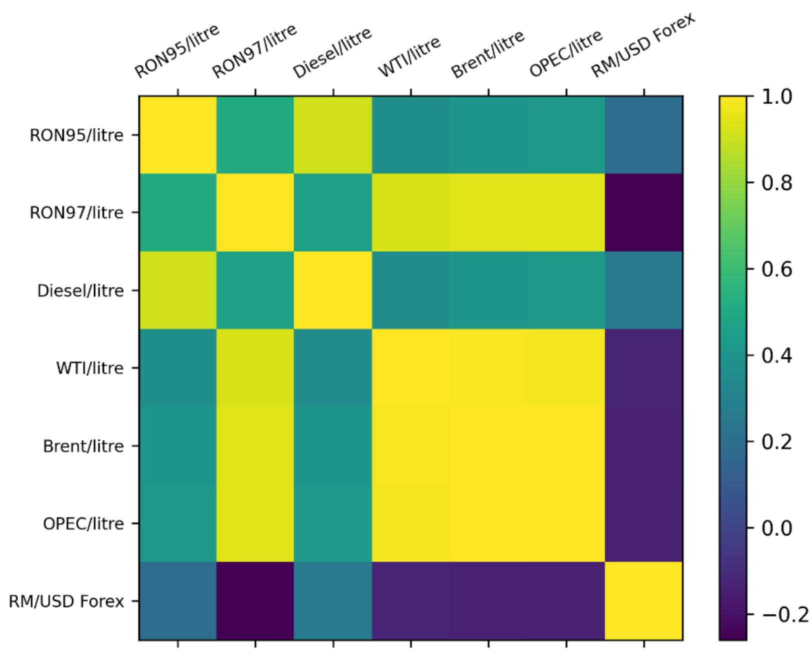

Figure 9: Heat map of Spearman's correlation matrix.

Aside from Pearson's correlation that summarizes the strength of linear relationship between two variables, Spearman's correlation is used to summarize the strength of monotonic relationship between two variables. In monotonic relationships, (1) an increasing relationship occurs when one increases, other increases too, (2) decreasing relationship occurs when one increases, other decreases, but not exactly at a constant rate. Figure 13 shows that RON 97 has a strong monotonic relationship with all the crude oil with higher correlation values compared to linear relationships as presented in Pearson's correlation.

The correlation coefficient gives insight into which variables may or may not be relevant as input for building the model. However, each variable should be experimented on the modeling process to evaluate the suitable input variable to be involved in the final training model.

4.1.1.2 Multiple Input Data Combinations

For each output variable, the input data selections are selected and tested differently to evaluate the performance with different combinations of input data on each output variable.

Firstly, for each output variable, random input variable is selected and evaluated with the initial model mentioned (See Section 3.1.3) to observe the loss function.

**Represents abbreviation of the variables.*

Output	Input Variable(s)	Runtimes	Validation RMSE: Mean
RON95	RON95	10	0.036
	RM/USD	10	0.233
	RON95, WTI	10	0.552
	RON95, Diesel	**10**	**0.028**
	RON95, RM/USD	10	0.061
	RON95, OPEC, RM/USD	10	0.424
	RON95, WTI, RM/USD	10	0.446
	Brent, WTI, OPEC	10	1.108
RON97	RON97	10	0.873
	RON97, Brent	10	1.075
	RON97, OPEC	10	0.997
	RON97, RM/USD	**10**	**0.474**
	RON97, OPEC, RM/USD	10	0.532
	RON97, Brent, OPEC, RM/USD	10	0.680
	RON97, WTI, OPEC, RM/USD	10	0.576
	RON97, Brent, WTI, OPEC, RM/USD	10	0.690
	Brent, WTI, OPEC	10	0.949
	Brent, OPEC, RM/USD	10	0.630
Diesel	**Diesel**	**10**	**0.035**
	RM/USD	10	0.220
	Diesel, RON95	10	0.037
	Diesel, RM/USD	10	0.055
	Diesel, OPEC, RM/USD	10	0.401

Table 2: Loss on multiple input data combinations.

Table 12 above shows the experiment results on different input variable combinations for each output variable. It can be seen that the simple models of RON95 and diesel performed well when taking their own historical data or combining with each other historical data as input variable. While the models of RON97 have shown higher RMSE than the other two output variables. The best performance model is with its historical data and exchange rate of RM/USD. Furthermore, it is interesting that the exchange rate of RM/USD as one of the input variables mixed with other input variables can result in a lower RMSE on all the output variables. Thus, the input data selection of RON97 considered are those combinations in the range of 0.4 to 0.6 RMSE to be evaluated with different hyperparameter. While for RON95 and diesel, multi-output model with taking both of them as output variables and their historical data together as input variables will be evaluated. Further evaluation is discussed at below sections with some sections took only the configuration with the lowest RMSE mean to increase productivity.

4.1.2 Hyperparameters

Under this section, the hyperparameters that are involved in tuning are timesteps, number of neurons in hidden layer, learning rate of Adam optimizer as well as number of epochs. The performance metrics used in the evaluation process are RMSE with its means and standard deviation from the results of running 30 times for each model.

**Noted: Activ Func = Activation Function, Lrates = Learning Rate*

Input variable	No. of steps	No. of neurons in hidden layer	Activ Func	Lrates	No. of epochs	Runtimes	Validation RMSE: Mean (Std)
RON95 and Diesel							
RON95, Diesel	1	2	ReLu	0.001	50	30	1.879 (0.014)
RON95, Diesel	1	2	ReLu	0.01	50	30	0.012 (0.003)
RON95, Diesel	1	2	Sigmoid	0.01	50	30	0.096 (0.009)
RON95, Diesel	1	2	ReLu	0.1	50	30	0.007 (0.002)
RON95, Diesel	**1**	**2**	**ReLu**	**0.1**	**100**	**30**	**0.004 (0.002)**
RON95, Diesel	1	4	ReLu	0.01	50	30	0.035 (0.002)
RON95, Diesel	1	4	ReLu	0.1	50	30	0.007 (0.002)
RON95, Diesel	1	4	ReLu	0.1	100	30	0.009 (0.005)
RON95, Diesel	1	6	ReLu	0.1	50	30	0.007 (0.002)
RON95, Diesel	1	6	ReLu	0.1	100	30	0.005 (0.002)

RON95, Diesel	1	6	ReLu	0.1	150	30	0.009 (0.005)
RON95, Diesel	1	8	ReLu	0.1	50	30	0.006 (0.002)
RON95, Diesel	1	8	ReLu	0.1	100	30	0.007 (0.002)
RON95, Diesel	1	10	ReLu	0.1	50	30	0.007 (0.002)
RON95, Diesel	2	2	ReLu	0.001	50	30	1.922 (0.021)
RON95, Diesel	2	2	Sigmoid	0.001	50	30	1.223 (0.045)
RON95, Diesel	2	2	ReLu	0.01	50	30	0.031 (0.001)
RON95, Diesel	2	2	Sigmoid	0.01	50	30	0.053 (0.001)
RON95, Diesel	2	2	ReLu	0.1	50	30	0.008 (0.002)
RON95, Diesel	2	2	Sigmoid	0.1	50	30	0.053 (0.014)
RON95, Diesel	2	2	ReLu	0.1	100	30	0.120 (0.006)
RON95, Diesel	2	4	ReLu	0.01	50	30	0.008 (0.004)
RON95, Diesel	2	4	ReLu	0.1	50	30	0.010 (0.005)
RON95, Diesel	2	6	ReLu	0.01	50	30	0.013 (0.001)
RON95, Diesel	2	6	ReLu	0.1	50	30	0.006 (0.002)
RON95, Diesel	**2**	**8**	**ReLu**	**0.1**	**50**	**30**	**0.004 (0.001)**
RON95, Diesel	2	10	ReLu	0.1	50	30	0.007 (0.001)
RON95, Diesel	3	2	ReLu	0.1	50	30	0.007 (0.002)
RON95, Diesel	3	4	ReLu	0.1	50	30	0.005 (0.001)
RON95, Diesel	3	6	ReLu	0.1	50	30	0.006 (0.001)
RON95, Diesel	3	8	ReLu	0.1	50	30	0.007 (0.001)
RON95, Diesel	3	10	ReLu	0.1	50	30	0.005 (0.002)
RON95, Diesel	3	10	ReLu	0.1	100	30	0.009 (0.005)

RON97							
RON97, RM/USD	1	2	ReLu	0.001	50	30	2.849 (0.087)
RON97, RM/USD	1	2	ReLu	0.01	50	30	1.953 (0.068)
RON97, RM/USD	1	2	ReLu	0.1	50	30	0.145 (0.001)
RON97, RM/USD	1	2	ReLu	0.1	100	30	1.276 (0.004)
RON97, RM/USD	1	4	ReLu	0.01	50	30	1.926 (0.067)
RON97, RM/USD	1	4	ReLu	0.1	100	30	0.142 (0.002)
RON97, RM/USD	1	4	ReLu	0.1	150	30	1.273 (0.007)
RON97, RM/USD	1	6	ReLu	0.1	50	30	0.223 (0.003)
RON97, RM/USD	1	8	ReLu	0.1	50	30	0.209 (0.003)
RON97, RM/USD	1	10	ReLu	0.1	50	30	0.184 (0.015)
RON97, RM/USD	2	2	ReLu	0.001	50	30	3.162 (0.131)
RON97, RM/USD	2	2	Sigmoid	0.001	50	30	3.584 (0.021)
RON97, RM/USD	2	2	ReLu	0.01	50	30	0.231 (0.020)
RON97, RM/USD	2	2	Sigmoid	0.01	50	30	1.129 (0.004)
RON97, RM/USD	2	2	ReLu	0.01	100	30	1.363 (0.010)
RON97, RM/USD	2	2	ReLu	0.1	50	30	1.299 (0.006)
RON97, RM/USD	2	4	ReLu	0.01	50	30	0.165 (0.001)
RON97, RM/USD	2	4	ReLu	0.01	100	30	0.221 (0.002)
RON97, RM/USD	2	6	ReLu	0.01	50	30	0.194 (0.004)
RON97, RM/USD	2	6	ReLu	0.01	100	30	0.168 (0.001)
RON97, RM/USD	2	6	ReLu	0.01	150	30	0.190 (0.003)
RON97, RM/USD	2	8	ReLu	0.01	50	30	0.181 (0.003)

RON97, RM/USD	2	10	ReLu	0.01	50	30	0.192 (0.005)
RON97, RM/USD	3	2	ReLu	0.01	50	30	0.333 (0.026)
RON97, RM/USD	3	4	ReLu	0.01	50	30	0.213 (0.002)
RON97, RM/USD	3	4	ReLu	0.01	100	30	0.196 (0.003)
RON97, RM/USD	3	6	ReLu	0.01	50	30	0.236 (0.008)
RON97, RM/USD	3	8	ReLu	0.01	50	30	0.156 (0.002)
RON97, RM/USD	3	10	ReLu	0.01	50	30	0.232 (0.004)
RON97, OPEC, RM/USD	1	2	ReLu	0.001	50	30	2.466 (0.102)
RON97, OPEC, RM/USD	1	2	ReLu	0.01	50	30	0.421 (0.017)
RON97, OPEC, RM/USD	1	2	ReLu	0.1	50	30	1.275 (0.004)
RON97, OPEC, RM/USD	1	4	ReLu	0.01	50	30	0.358 (0.009)
RON97, OPEC, RM/USD	1	4	ReLu	0.01	100	30	1.278 (0.000)
RON97, OPEC, RM/USD	1	6	ReLu	0.01	50	30	0.261 (0.007)
RON97, OPEC, RM/USD	1	8	ReLu	0.01	50	30	0.276 (0.007)
RON97, OPEC, RM/USD	1	10	ReLu	0.01	50	30	0.239 (0.015)
RON97, OPEC, RM/USD	2	2	ReLu	0.001	50	30	3.464 (0.015)
RON97, OPEC, RM/USD	2	2	ReLu	0.01	50	30	0.309 (0.007)
RON97, OPEC, RM/USD	2	2	Sigmoid	0.01	50	30	1.188 (0.009)
RON97, OPEC, RM/USD	2	2	ReLu	0.01	100	30	0.192 (0.004)
RON97, OPEC, RM/USD	2	2	ReLu	0.01	150	30	1.338 (0.001)
RON97, OPEC, RM/USD	2	4	ReLu	0.01	50	30	0.339 (0.014)
RON97, OPEC, RM/USD	2	6	ReLu	0.01	50	30	0.133 (0.001)
RON97, OPEC, RM/USD	2	6	ReLu	0.01	100	30	0.127 (0.002)

RON97, OPEC, RM/USD	2	8	ReLu	0.01	50	30	0.129 (0.003)
RON97, OPEC, RM/USD	**2**	**8**	**ReLu**	**0.01**	**100**	**30**	**0.096 (0.002)**
RON97, OPEC, RM/USD	2	8	ReLu	0.01	150	30	0.132 (0.004)
RON97, OPEC, RM/USD	2	10	ReLu	0.01	50	30	0.151 (0.012)
RON97, OPEC, RM/USD	2	10	ReLu	0.01	100	30	0.135 (0.006)
RON97, OPEC, RM/USD	3	2	ReLu	0.01	50	30	0.334 (0.002)
RON97, OPEC, RM/USD	3	4	ReLu	0.01	50	30	0.239 (0.004)
RON97, OPEC, RM/USD	3	6	ReLu	0.01	50	30	0.213 (0.016)
RON97, OPEC, RM/USD	3	8	ReLu	0.01	50	30	0.193 (0.020)
RON97, OPEC, RM/USD	3	10	ReLu	0.01	50	30	0.280 (0.015)
RON97, WTI, OPEC, RM/USD	1	2	ReLu	0.01	50	30	0.426 (0.025)
RON97, WTI, OPEC, RM/USD	1	4	ReLu	0.01	50	30	0.327 (0.007)
RON97, WTI, OPEC, RM/USD	1	6	ReLu	0.01	50	30	0.192 (0.011)
RON97, WTI, OPEC, RM/USD	1	6	ReLu	0.01	100	30	0.136 (0.002)
RON97, WTI, OPEC, RM/USD	1	6	ReLu	0.01	150	30	0.119 (0.001)
RON97, WTI, OPEC, RM/USD	1	8	ReLu	0.01	50	30	0.268 (0.020)
RON97, WTI, OPEC, RM/USD	1	10	ReLu	0.01	50	30	0.307 (0.026)
RON97, WTI, OPEC, RM/USD	2	2	ReLu	0.001	50	30	1.963 (0.221)
RON97, WTI, OPEC, RM/USD	2	2	ReLu	0.01	50	30	0.209 (0.017)
RON97, WTI, OPEC, RM/USD	2	4	ReLu	0.01	50	30	0.502 (0.022)
RON97, WTI, OPEC, RM/USD	2	6	ReLu	0.01	50	30	0.238 (0.016)
RON97, WTI, OPEC, RM/USD	2	8	ReLu	0.01	50	30	0.171 (0.006)
RON97, WTI, OPEC, RM/USD	2	8	ReLu	0.01	100	30	0.298 (0.011)

RON97, WTI, OPEC, RM/USD	2	10	ReLu	0.01	50	30	0.159 (0.006)
RON97, WTI, OPEC, RM/USD	2	10	ReLu	0.01	100	30	0.146 (0.006)
RON97, WTI, OPEC, RM/USD	3	2	ReLu	0.01	50	30	0.363 (0.002)
RON97, WTI, OPEC, RM/USD	3	4	ReLu	0.01	50	30	0.245 (0.014)
RON97, WTI, OPEC, RM/USD	3	6	ReLu	0.01	50	30	0.173 (0.004)
RON97, WTI, OPEC, RM/USD	3	6	ReLu	0.01	100	30	0.169 (0.002)
RON97, WTI, OPEC, RM/USD	3	8	ReLu	0.01	50	30	0.320 (0.019)
RON97, WTI, OPEC, RM/USD	3	10	ReLu	0.01	50	30	0.135 (0.019)
RON97, WTI, OPEC, RM/USD	3	10	ReLu	0.01	100	30	0.178 (0.007)

Table 3: Hyperparameters tuning evaluation result.

Table 13 above shows the evaluation of different hyperparameters combination on different models. For the models of RON95 and diesel, there are two models produced with the same mean of RMSE, which is 0.004. However, model with timesteps of 2, 8 neurons number, ReLu activation function, Adam optimizer with learning rate of 0.1 and 50 epochs has a lower standard deviation of RMSE, which is 0.001 lower than model with a timestep of 1. While for RON97, model with timesteps of 2, 8 neurons number, ReLu activation function, Adam optimizer with learning rate of 0.01 and 100 epochs outperformed others with a RMSE mean of 0.096. Sections below will discuss the performances of each hyperparameter to decide the most appropriate hyperparameters to develop the models.

4.1.2.1 Timesteps

Timesteps selection list involve a number range of 1 to 3. This section will discuss on the importance of timesteps on the models evaluated. The performance of timesteps will be evaluated based on model with same hyperparameters excluding the number of timesteps.

RON95 and Diesel

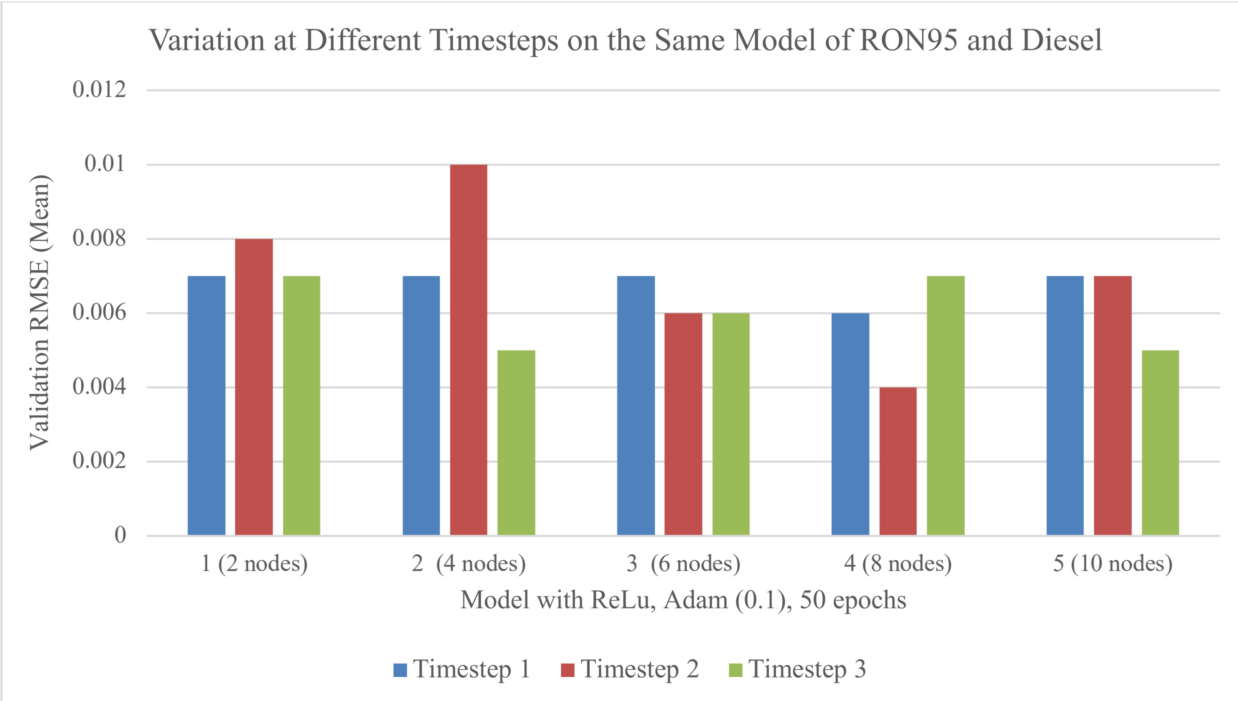

Figure 10: Different timesteps performance on each five model of RON95 and diesel.

From the above figure, it can be observed that the timesteps have little effect on the model of RON95 and diesel as each model produced slightly different RMSE means at different number of timesteps. This is due to the reason that the input data, i.e., their own historical fuel data, have constant values most of the time (See Figure 17). Either one of the timesteps can be chosen in finalizing model hyperparameter, but to reduce the work of trial and error, only timesteps of 1 and 2 will be chosen as models with these two timesteps have produced outstanding RMSE values.

Figure 11: Historical price of RON95 and diesel over the time.

RON97

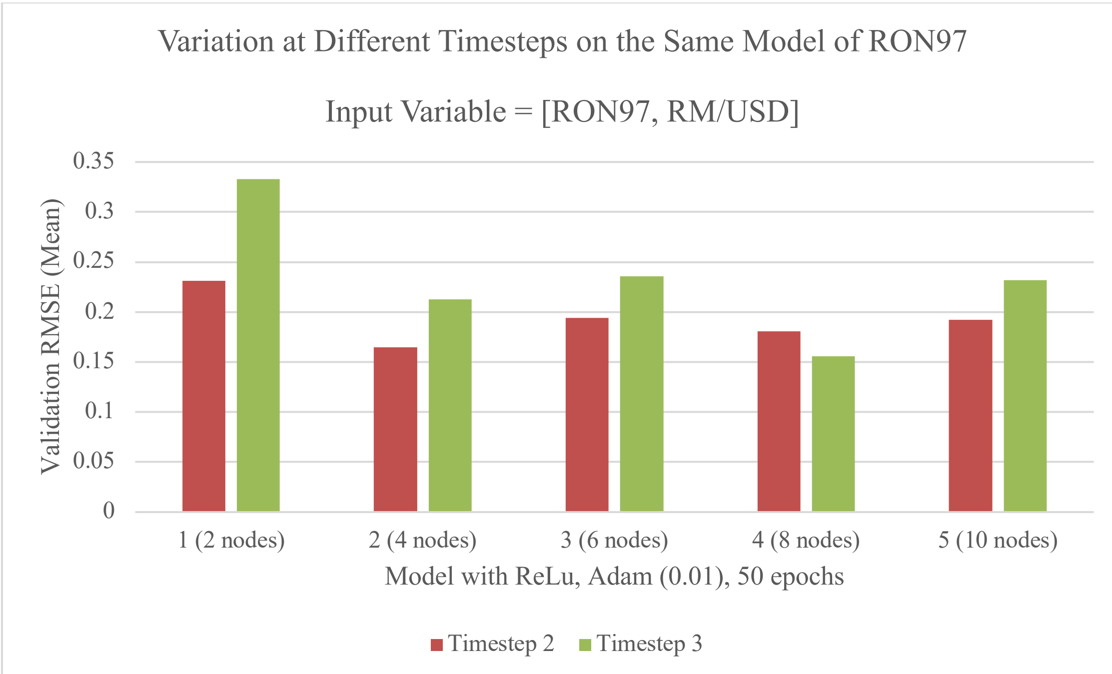

Figure 12: Different timesteps performance on each five model of RON97.

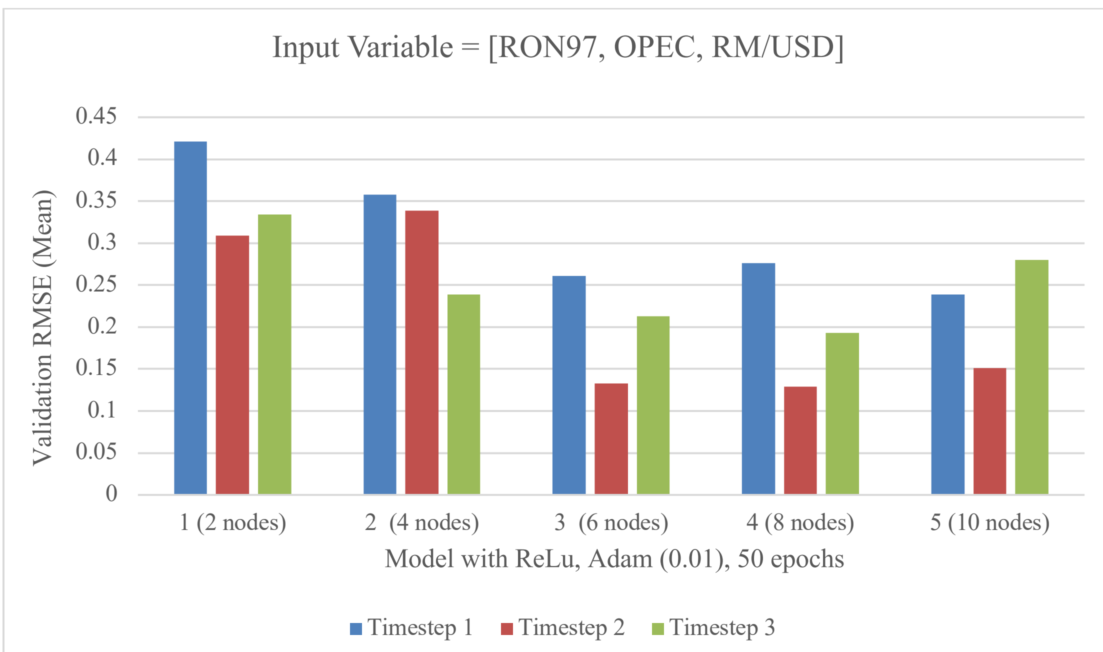

Figure 13: Different timesteps performance on each five model of RON97.

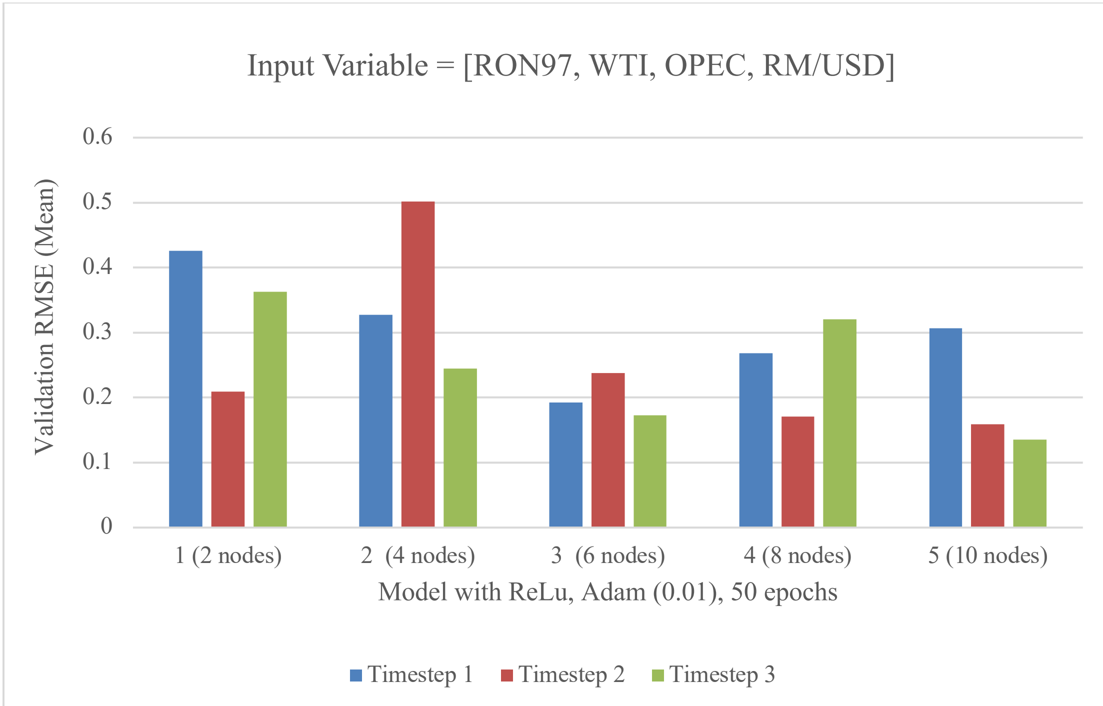

Figure 14: Different timesteps performance on the each five model of RON97.

From Figure 18 to 20, it can be seen that the models with same hyperparameters but different number of timesteps produce RMSE means that differ within 0.3. For input data in Figure 18, timestep number of 2 outperforms of 3 with lower RMSE mean in all the 5 models. Noted that timestep number of 1 is not discussed as it has no same hyperparameters evaluation as other timesteps in Table 13. For input data in Figure 19, the optimal timestep among the selection list is 2 as there are 4 models with timestep of 2 produce the lowest RMSE means when compared to other timesteps. While for input data in Figure 20, the optimal timestep is between 2 (mean of RMSE in 5 models is 0.256) and 3 (mean of RMSE is 5 models is 0.247). Both timesteps perform almost equally.

Model	Input Variables	Timesteps
RON95 and diesel	RON95, Diesel	1, 2
RON97	RON97, RM/USD	1 (not discussed), 2
	RON97, OPEC, RM/USD	2
	RON97, WTI, OPEC, RM/USD	2, 3

Table 4: Summary of appropriate number of timesteps for model evaluation.

4.1.2.2 Neurons Number

Selection list of neurons number in hidden layer involve 2, 4, 6, 8, and 10. The performance of different number of neurons will be evaluated by plotting training and validation learning curves, or in other words, training and validation losses (MSE) over epochs.

RON95 and Diesel

According to Table 14, timesteps selection can be any within the list for the model. Thus, performance of different number of neurons on the models with different timesteps and but with other constant hyperparameters are evaluated.

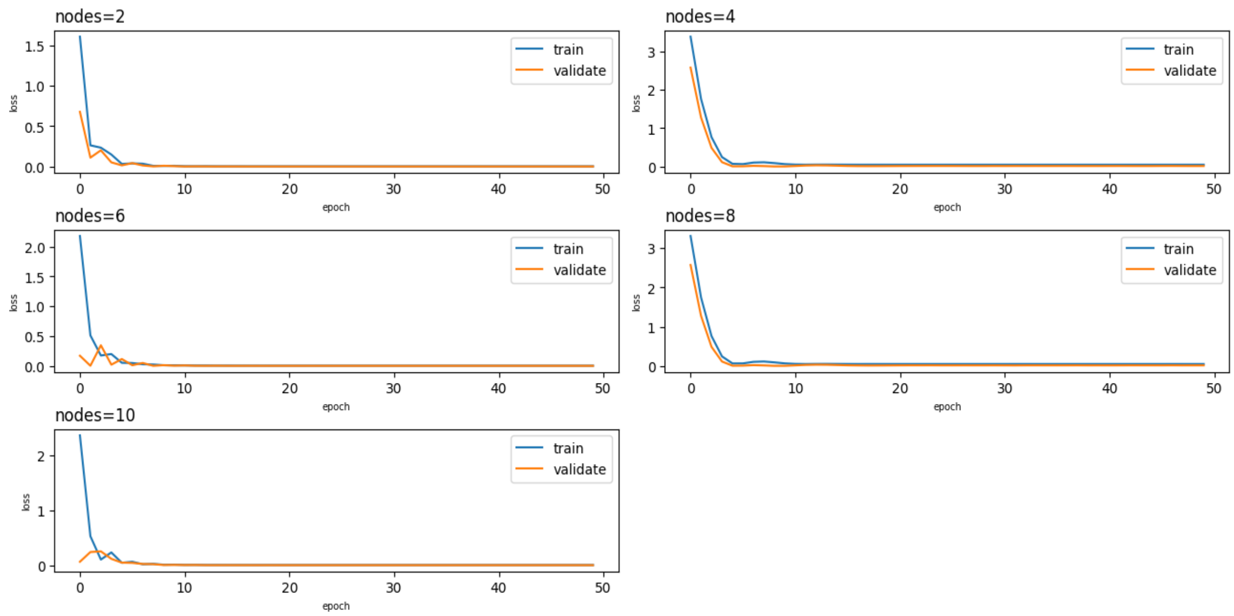

Figure 15: Loss over epochs at different number of neurons (timesteps of 1, ReLu, Adam (0.1) and 50 epochs)

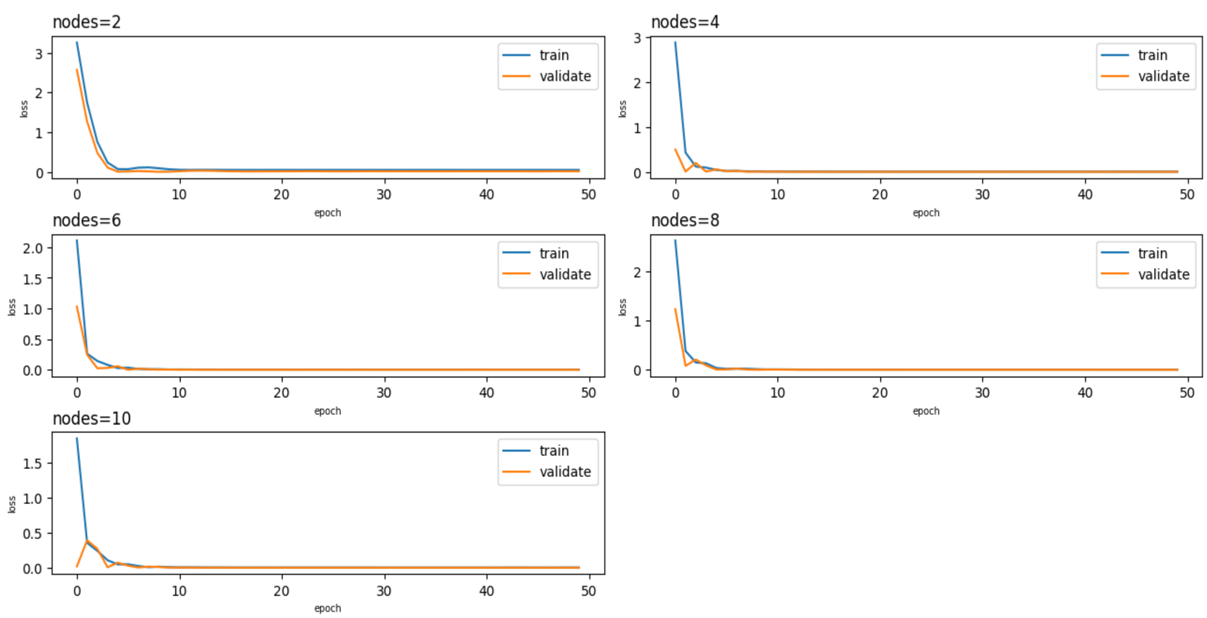

Figure 16: Loss over epochs at different number of neurons (timesteps of 2, ReLu, Adam (0.1) and 50 epochs).

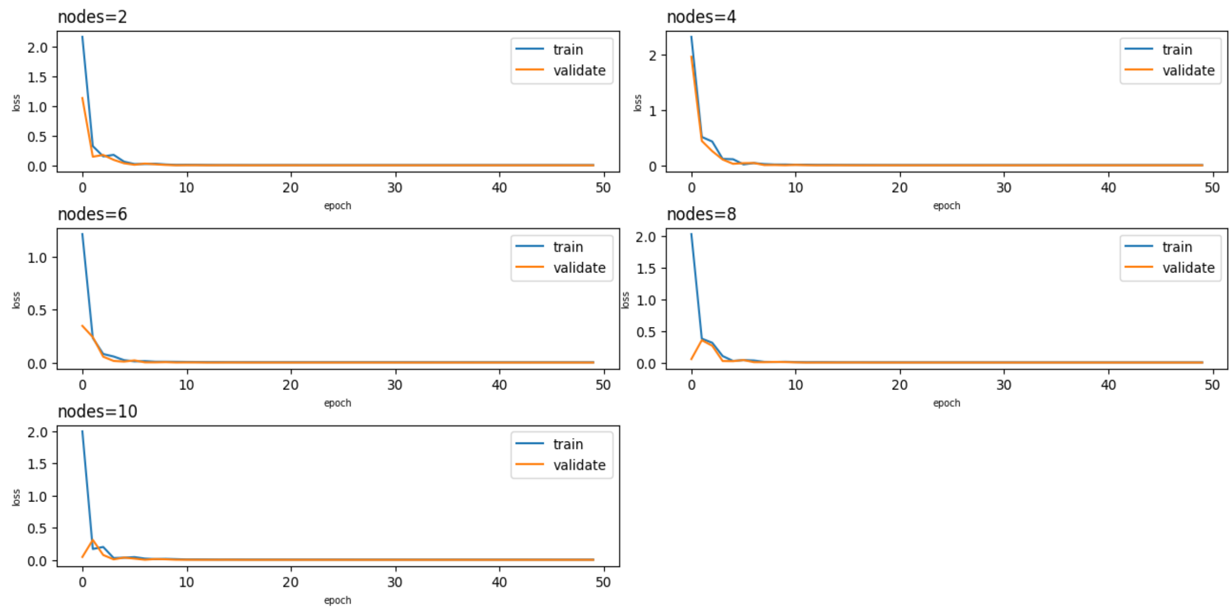

Figure 17: Loss over epochs at different number of neurons (timesteps of 3, ReLu, Adam (0.1) and 50 epochs).

From plotting results on Figure 21 to 23, all the learning curves with different number of neurons have shown their ability to convergence. Among the curves, plot with neurons number of 2, 4 and 8 on Figure 21, neurons number of 2, 6 and 8 on Figure 22, as well as neurons number of 2, 4 and 6 on Figure 23 have shown good fit curve patterns. However, when looking at the training loss and validation loss at all the models, it is observed that loss on validation dataset is almost always lower than the training dataset, which is not a sign of good fit learning curves. This is due to the fact that the models are taking the fuel's historical data, which have constant value most of the time as input variables, the models may model the training data too well. Unless changing the input variables, this situation is unavoidable. However, to develop the model that is able to output the fuel prices that will not exceed the ceiling price, the input data selection for the model will remain unchanged. To prove that the learning curves will not be overfitting in long training, an experiment with 1000 epochs is conducted (See Figure 24). And the experiment has shown that overfitting will not occur in the long run.

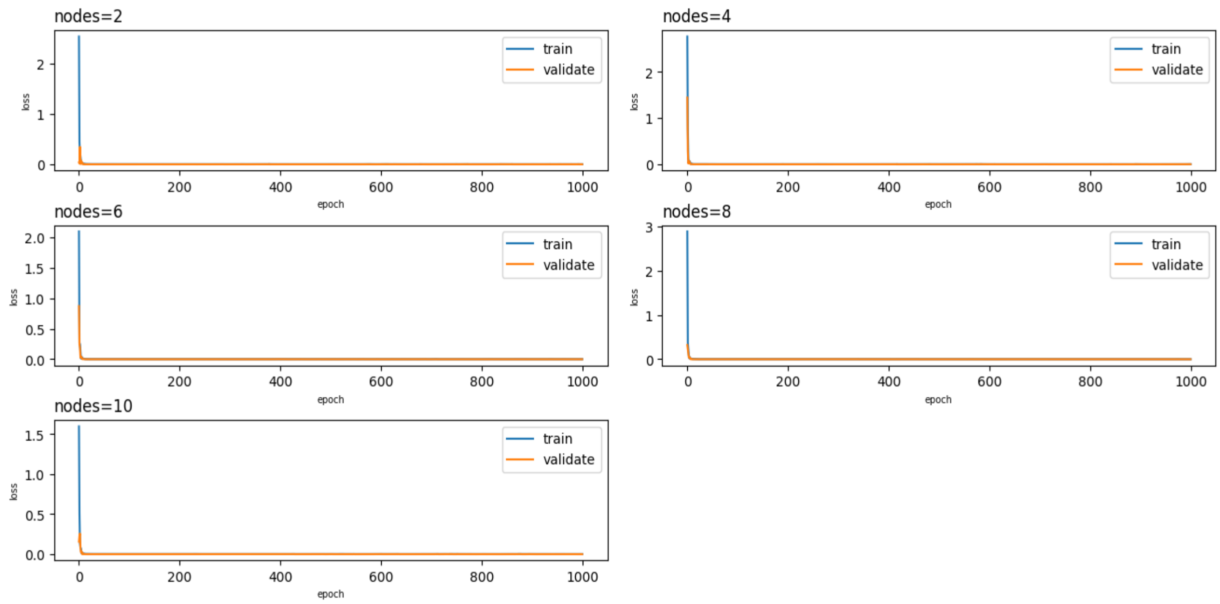

Figure 18: Loss over epochs at different number of neurons (timesteps of 1, ReLu, Adam (0.1) and 1000 epochs).

RON97

Different numbers of neurons will be evaluated with timesteps based on Table 14 on same hyperparameters model.

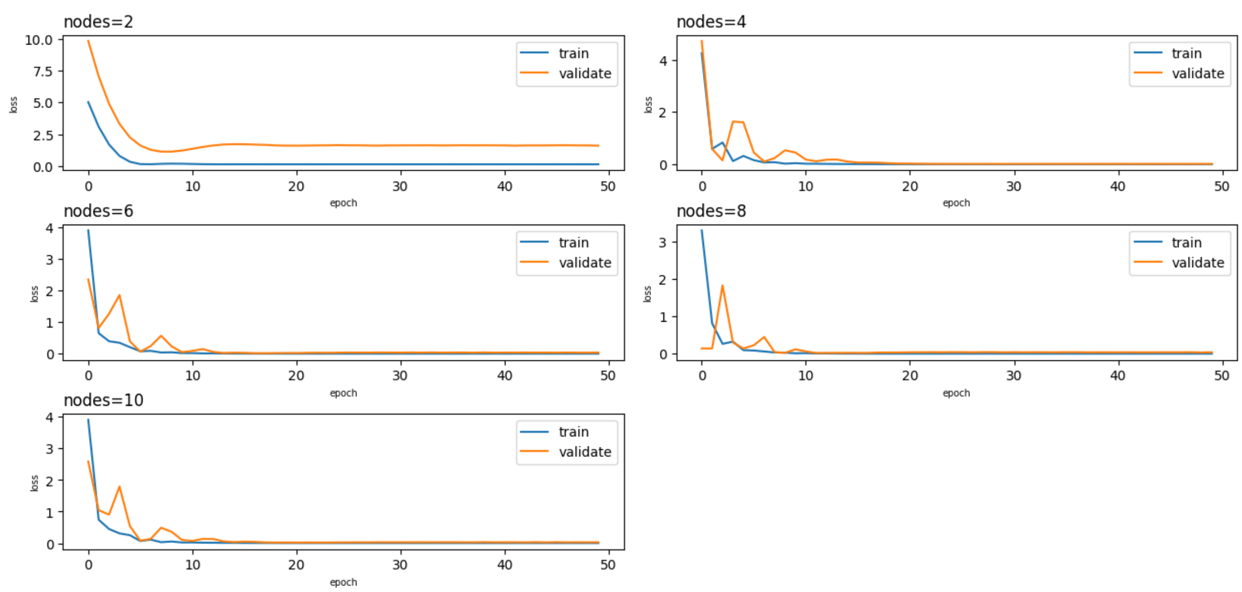

Figure 19: Loss over epochs at different number of neurons (input data = [RON97, RM/USD], timesteps of 1, ReLu, Adam (0.1) and 50 epochs).

In Figure 25, learning curves at neurons number from 4 to 10 have shown good fit curves pattern, where loss on training dataset is almost lower than validation dataset, and the two losses on all plots decrease to a point of stability. When looking at learning curves with neuron number of 2, there is always a big gap between the two loss values which does not fulfill the requirement of being good fit curves. While learning curves for other neurons have shown their two curves with small gap and stabilized after 10 epochs.

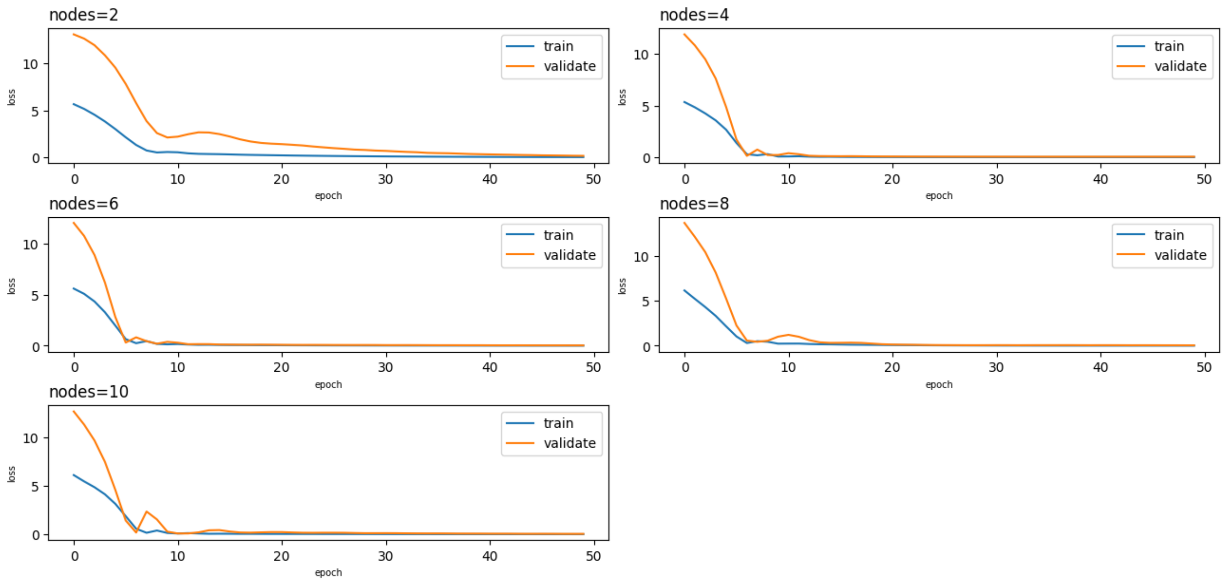

Figure 20: Loss over epochs at different number of neurons (input data = [RON97, RM/USD], timesteps of 2, ReLu, Adam (0.1) and 50 epochs).

Learning curves with neuron number from 4 to 10 on above also show good fit curve patterns. The gap between learning curves at neuron number of 2 decreases with a relatively slow speed when compared to others. Besides, it can be observed that learning curves at other neurons number has a faster convergence speed than learning curves in Figure 25.

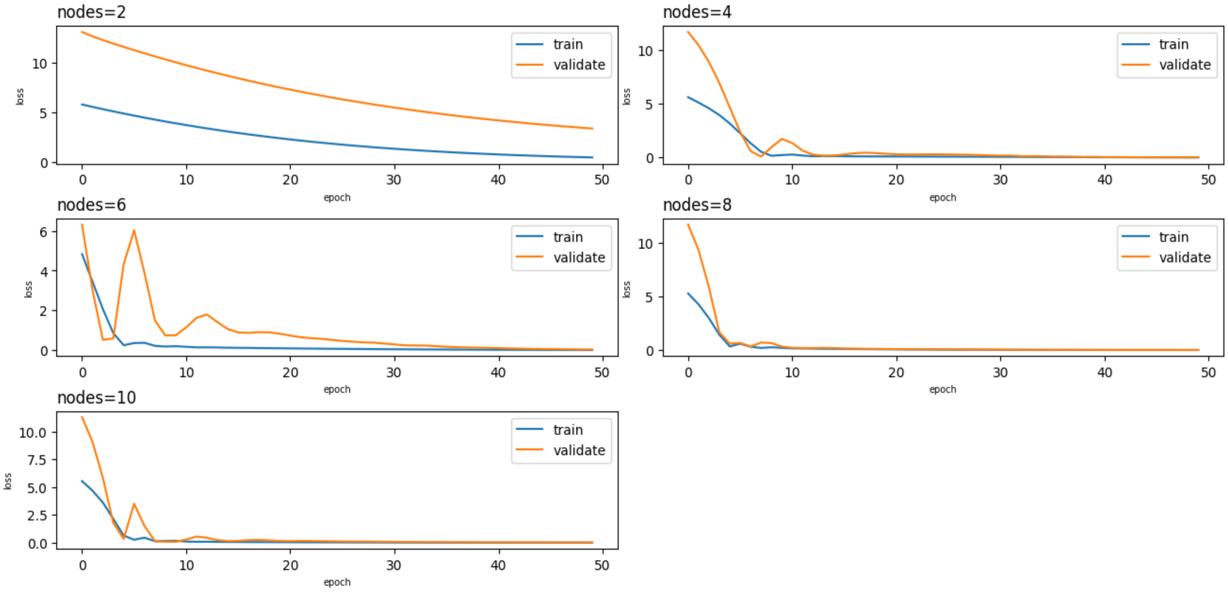

Figure 21: Loss over epochs at different number of neurons (input data = [RON97, OPEC, RM/USD], timesteps of 2, ReLu, Adam (0.01) and 50 epochs).

Underfitting learning curves are found on plot with neuron number of 2 in Figure 27, which means that the model fails to learn the training dataset. Learning curves with neuron numbers of 4, 8 and 10 show a good fit curve pattern, and learning curves with neuron number of 8 shows the fastest speed of convergence. While validation learning curves with neuron number of 6 fails to provide the ability of the model to generalize.

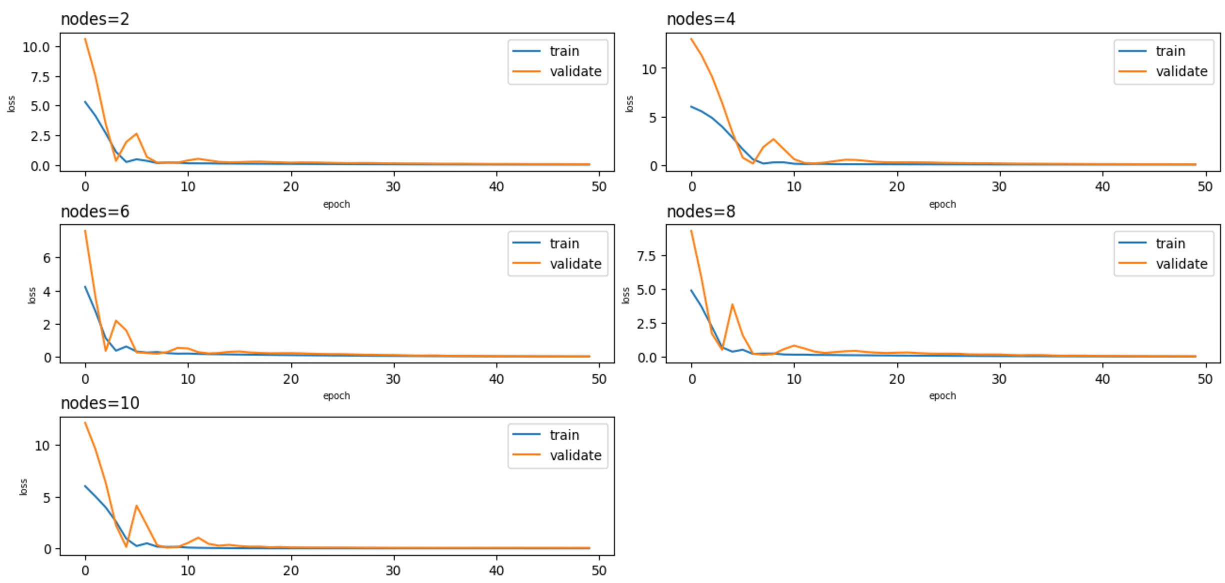

Figure 22: Loss over epochs at different number of neurons (input data = [RON97, WTI, OPEC, RM/USD], timesteps of 2, ReLu, Adam (0.01) and 50 epochs).

All the learning curves in Figure 28 show good fit curves and are able to stabilize before 20 epochs.

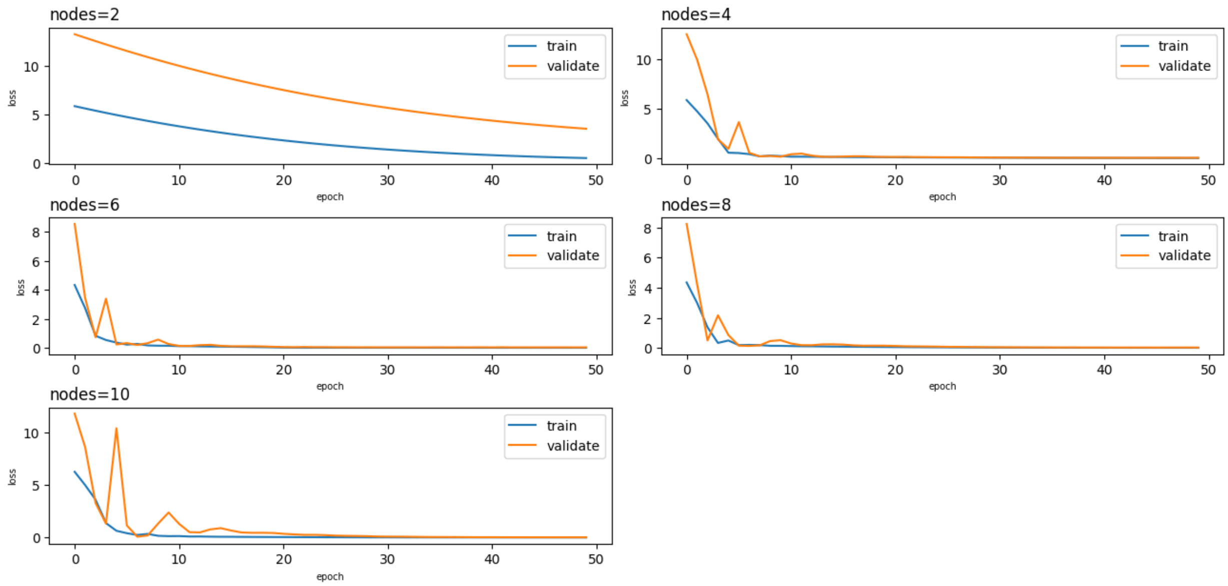

Figure 23: Loss over epochs at different number of neurons (input data = [RON97, WTI, OPEC, RM/USD], timesteps of 3, ReLu, Adam (0.01) and 50 epochs).

Plot on Figure 29 with neuron number of 2 shows underfitting learning curves, while others show good fit learning curves, with curves at neuron number of 10 converge at a relatively slow speed by showing its ability to generalize after 20 epochs.

Model	Input Variables	Number of Neurons
RON95 and diesel	RON95, Diesel	2, 4, 8 with timesteps of 1 2, 6, 8 with timesteps of 2
RON97	RON97, RM/USD	4, 6, 8, 10 with timesteps of 2
	RON97, OPEC, RM/USD	4, 8, 10 with timesteps of 2
	RON97, WTI, OPEC, RM/USD	2, 4, 6, 8, 10 with timesteps of 2 4, 6, 8 with timesteps of 3

Table 5: Summary of suitable number of neurons with timesteps.

4.1.2.3 Activation Function

Activation functions evaluated involve ReLu and sigmoid. Their performances will be evaluated using learning curves as well. For model of each target output(s), performance of each activation function at different neurons number with same hyperparameters are plotted.

RON95 and Diesel

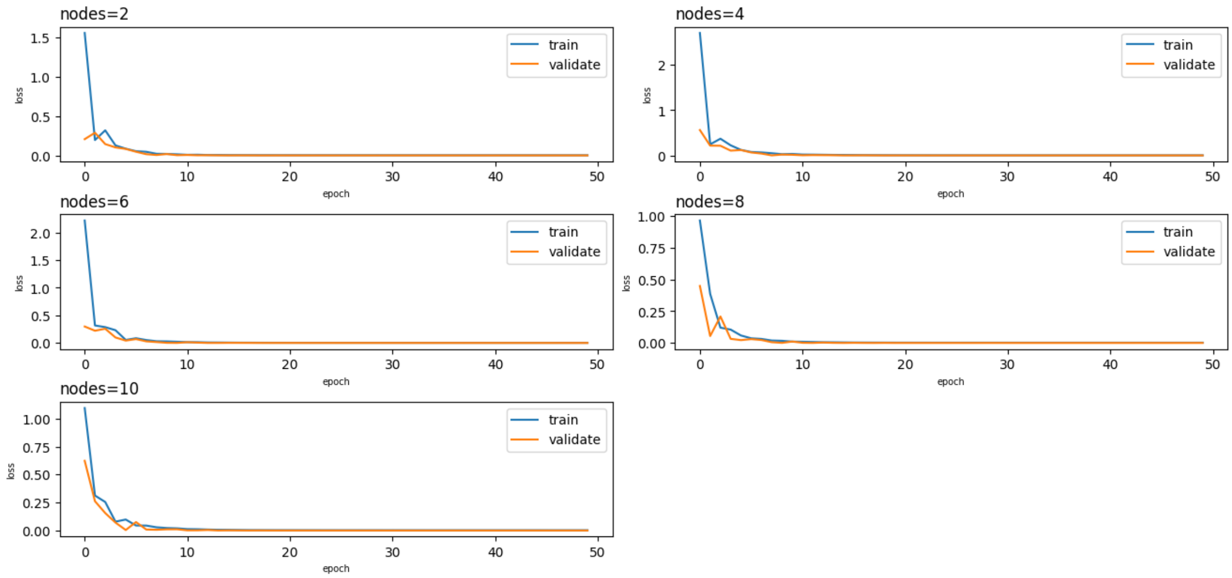

Figure 24: Loss over epochs at different number of neurons with Sigmoid activation function, timesteps of 2, Adam (0.1) and 50 epochs.

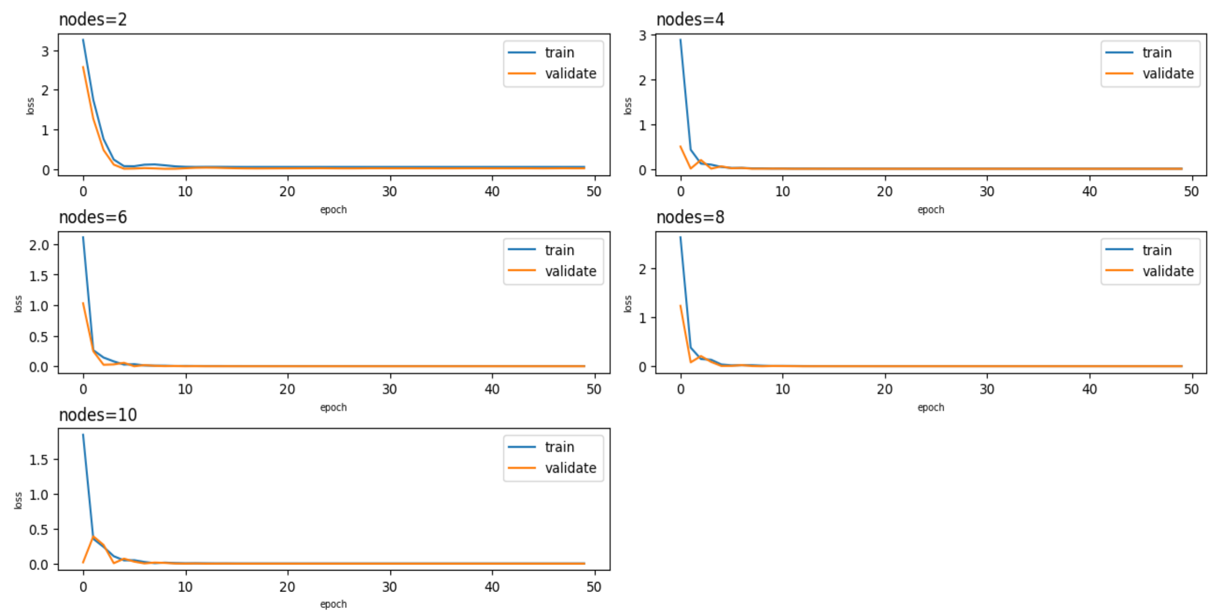

Figure 25: Loss over epochs at different number of neurons with ReLu activation function, timesteps of 2, Adam (0.1) and 50 epochs.

From Figure 30 and 31, it can be observed that there is not much difference between the learning curves at both activation functions, except there are more good fit curves at ReLu function than at sigmoid function.

RON97

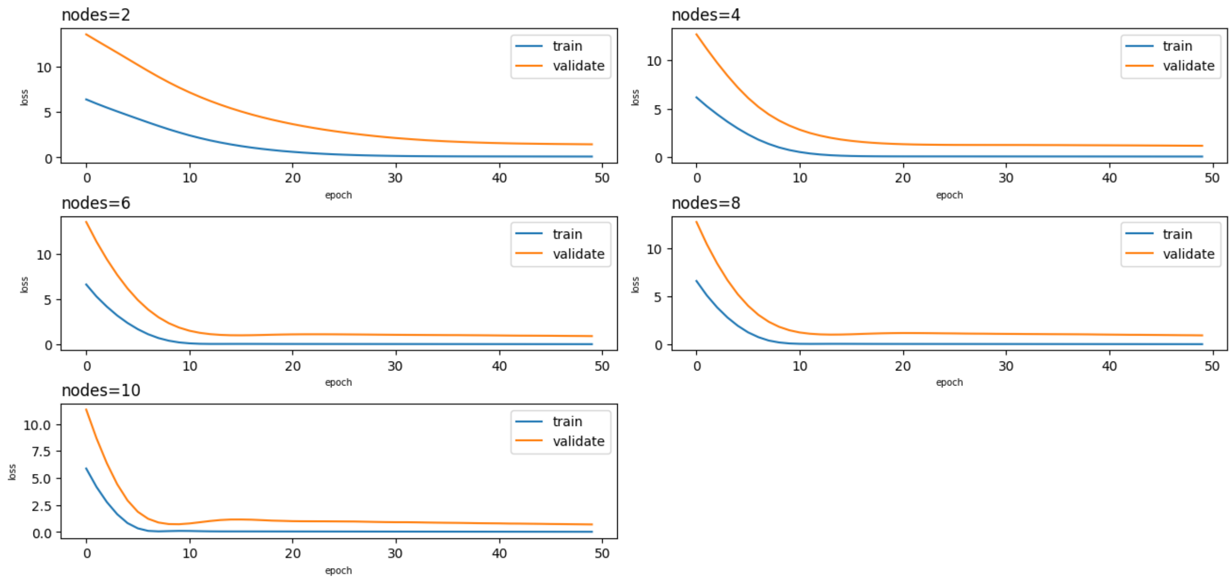

Figure 26: Loss over epochs at different number of neurons with Sigmoid activation function (input data = [RON97, WTI, OPEC, RM/USD], timesteps of 2, Adam (0.01) and 50 epochs).

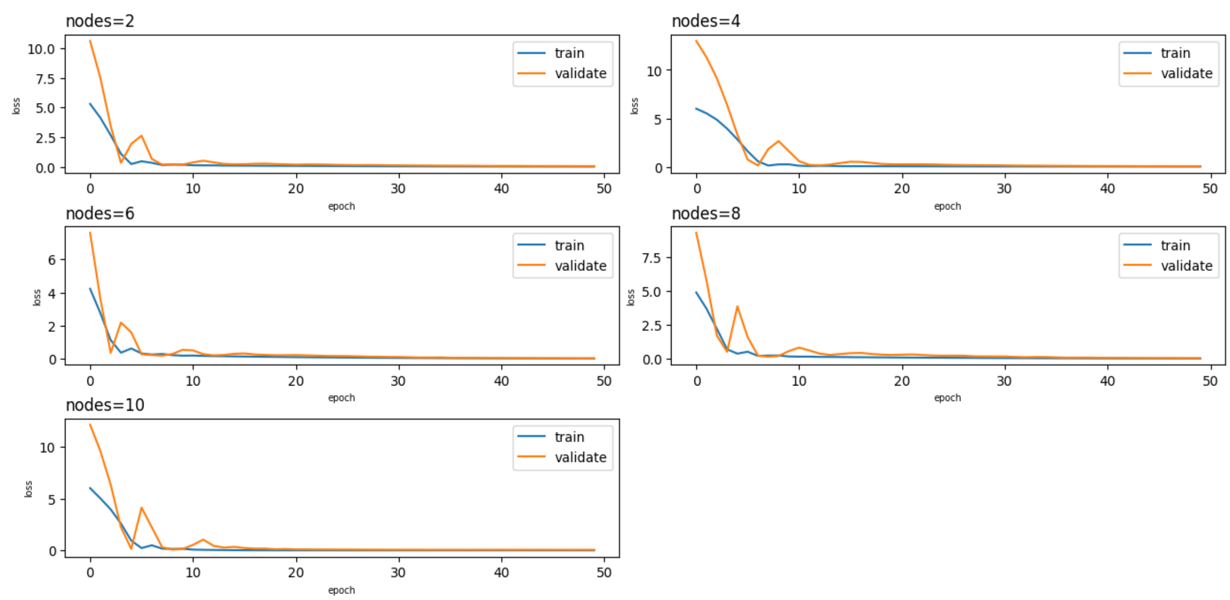

Figure 27: Loss over epochs at different number of neurons with ReLu activation function (input data = [RON97, WTI, OPEC, RM/USD], timesteps of 2, Adam (0.01) and 50 epochs).

Plotting from Figure 32 shows the training loss and validation loss decreases and stabilize over the time, there is no overfitting sign, but there is always a gap can be seen clearly between the learning curves. Compared with learning curves with ReLu function, curves with ReLu fit well than sigmoid function.

<table>
<tr><th>Model</th><th>Input Variables</th><th>Activation Function</th></tr>
<tr><td>RON95 and diesel</td><td>RON95, Diesel</td><td rowspan="2">ReLu</td></tr>
<tr><td>RON97</td><td>RON97, RM/USD</td></tr>
</table>

	RON97, OPEC, RM/USD	
	RON97, WTI, OPEC, RM/USD	

Table 6: Summary of suitable activation function for all models.

4.1.2.4 Optimization Algorithm with Learning Rates

The selection list of Adam optimizer's learning rates involves 0.001, 0.01, and 0.1. Learning curves plotting will also be used to evaluate performance of each learning rate. From Table 13, it can be observed that models with different neurons number, but same hyperparameters including learning rates result RMSE means with almost no big difference. Thus, according to Table 15, the following plotting will take only models with neuron number of 8 at different timesteps and different learning rates, but with same 50 epochs.

RON95 and Diesel

The RMSE means of all the models of RON95 and diesel with same activation function, epochs number and learning rate at different timesteps and neurons numbers has shown little differences in the values. Thus, the plotting will take only model with timesteps of 2 and neurons number of 8 as representative for all hyperparameters.

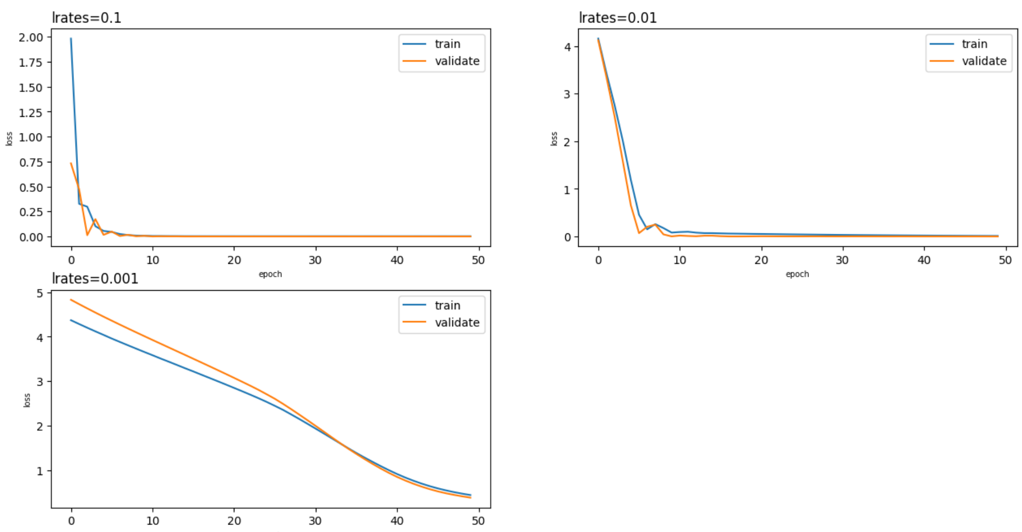

Figure 28: Loss over epochs at different learning rates (timesteps of 2, 8 neurons, ReLu and 50 epochs)

Plotting above shows models with learning rates of 0.1 and 0.01 perform better than learning rates of 0.001. Learning curves with rate of 0.001 show underfitting sign, where the model may fail to identify the true nature of data. While learning curve with learning rate of 0.01 has a slower convergence speed than rate of 0.1. Thus, Adam optimizer with learning rate of 0.1 is the optimal solution for models of RON95 and diesel.

RON97

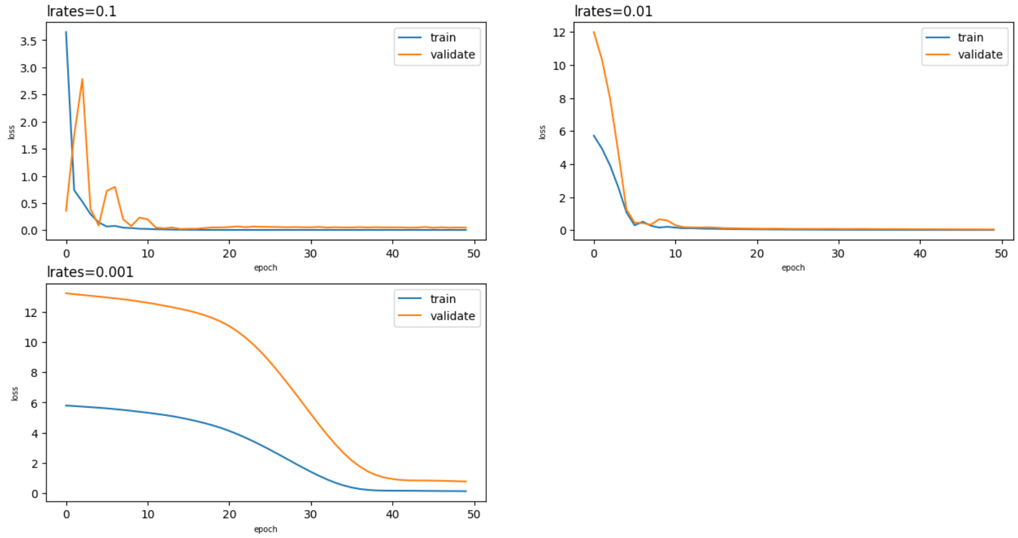

Figure 29: Loss over epochs at different learning rates (input data = [RON97, RM/USD], timesteps of 2, 8 neurons, ReLu, and 50 epochs)

Learning curves with rate of 0.01 outperform the other two learning rates by showing a relatively perfect good fit curve pattern. The validation learning curve with rate of 0.1 shows unstable loss at the beginning of the process, while validation learning curve with rate of 0.001 shows large loss at the beginning and continue decreases and stabilize at almost the end of the process.

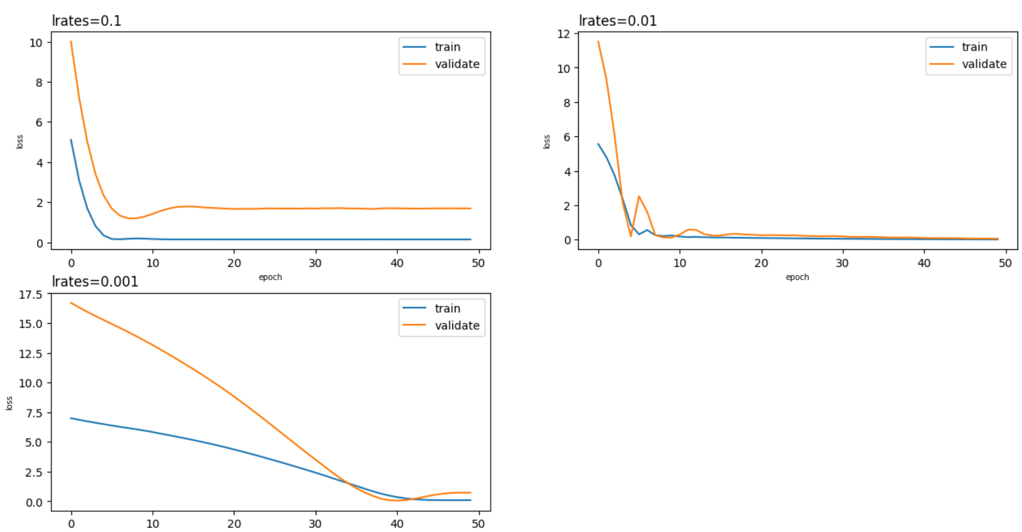

Figure 30: Loss over epochs at different learning rates (input data = [RON97, OPEC, RM/USD], timesteps of 2, 8 neurons, ReLu, and 50 epochs)

Same as plotting in Figure 35, good fit learning curves with rate of 0.01 in Figure 36 outperform the other two learning curves. While learning curves with rate of 0.1 show the sign of overfitting with validation loss decreases at the beginning, increases afterward, and stabilize when reaching the middle of the process. For learning curves with rate of 0.001, the model learnt too slow in the training dataset which result in almost a flat line in the training curve.

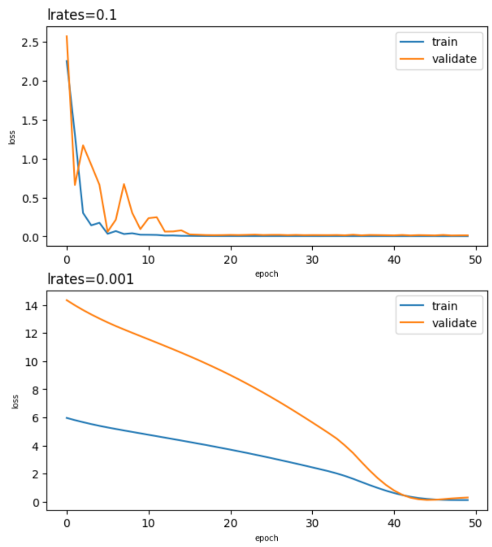

Figure 31: Loss over epochs at different learning rates (input data = [RON97, WTI, OPEC, RM/USD], timesteps of 2, 8 neurons, ReLu, and 50 epochs)

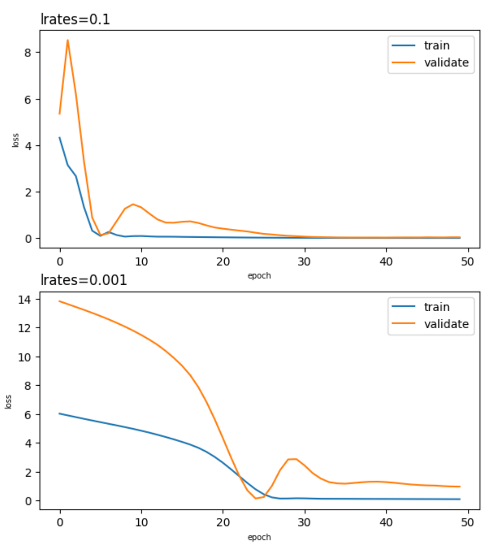

Figure 32: Loss over epochs at different learning rates (input data = [RON97, WTI, OPEC, RM/USD], timesteps of 3, 8 neurons, ReLu, and 50 epochs)

For plotting in Figure 37 and 38, the learning curves that outperform the others are with rate of 0.01. Learning curves with 0.1 rate in both figures show slow convergence speed compared to rate of 0.01. The learning curves with rate of 0.001 in Figure 37 have the same situation as in Figure 36, while validation learning curve with the 0.001 rate in Figure 38 fails to show its convergence ability.

Model	Input Variables	Learning Rate

RON95 and diesel	RON95, Diesel	0.1
RON97	RON97, RM/USD	0.01
	RON97, OPEC, RM/USD	
	RON97, WTI, OPEC, RM/USD	

Table 7: Summary of optimal learning rates for all models.

4.1.2.5 Epochs

The selection list of epochs involves a number of 50, 100 and 150. Again, learning curves will be plotted to evaluate the performance of the loss over different number of epochs.

RON95 and Diesel

The RMSE means of all the models of RON95 and diesel with same activation function, learning rate at different timesteps, neurons numbers and epochs have shown little differences in the values. Thus, the plotting will take only model with timesteps of 2 and neurons number of 8 as representative for all hyperparameters.

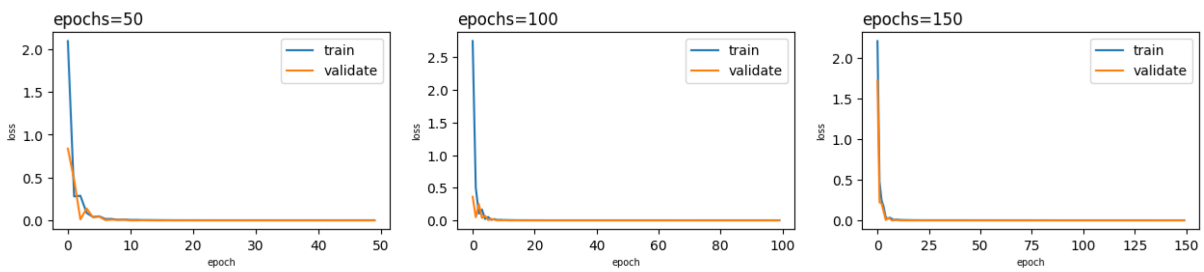

Figure 33: Loss over different epochs (timesteps of 2, 8 neurons, ReLu, Adam (0.1))

Plotting above shows there is no clear difference between the learning curves over different epochs, and the fitting pattern shows there is a room to reduce the number of epochs to below 30 to increase the model execution speed.

RON97

For model of RON97, the experiment will take only input data of RON97, OPEC and exchange rate of RM/USD as they produced the lowest RMSE mean values. Thus, plotting will be generated according to Table 15.

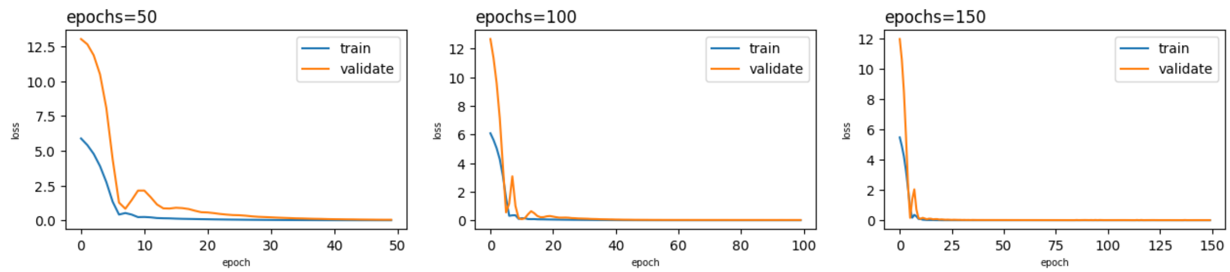

Figure 34: Loss over different epochs (input data = [RON97, OPEC, RM/USD], timesteps of 2, 4 neurons, ReLu, Adam (0.01))

Learning curves running on 50 epochs have shown that the validation learning curve reached convergence after 30 epochs. This means that epoch number of 50 is insufficient to show the stability of the model. And thus, increasing the epoch number to 100 might be an optimal solution.

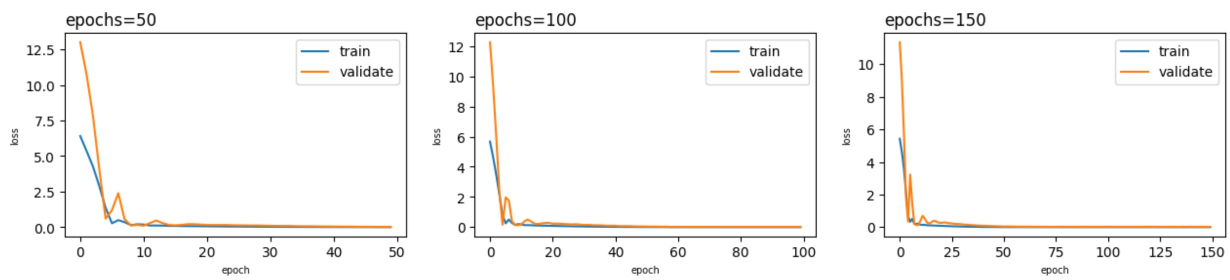

Figure 35: Loss over different epochs (input data = [RON97, OPEC, RM/USD], timesteps of 2, 8 neurons, ReLu, Adam (0.01))

Plotting above shows there is no clear difference between the learning curves over different epochs. The epoch number of 50 may be sufficient to train the model. However, according to Table 13, the model with 100 epochs results in a lower RMSE means than 50 epochs. To develop a predictive model, 100 epochs will also be considered in the next tuning.

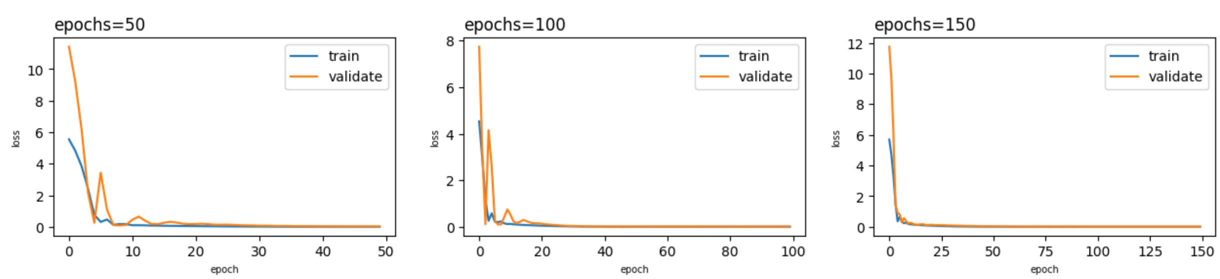

Figure 36: Loss over different epochs (input data = [RON97, OPEC, RM/USD], timesteps of 2, 10 neurons, ReLu, Adam (0.01))

Again, plotting above shows there is no clear difference between the learning curves over different epochs. The epoch number of 50 may be sufficient to train the model.

Model	Input Variables	Epochs	
RON95 and diesel	RON95, Diesel	50	
RON97	RON97, OPEC, RM/USD	4 neurons	100
		8 neurons	50, 100
		10 neurons	50

Table 8: Summary of suitable epoch numbers for all models.

4.1.2.6 Weight Assignments

In this project, the neural network model is initialized with random weights. Suitable sets of weights can be found using randomness to fit the model with different network each time the training algorithm is run. However, randomness will result in different prediction results on different runs, which increases the hardness to prepare a final model. Therefore, ensemble learning techniques are applied to reduce variance of the models.

First is model averaging ensemble. This ensemble technique tends to use the same trained model configuration, then make predictions on different runs and combine the results together as a final result. The final result is thus the average of the predictions collected from each run.

Second is weights averaging ensemble or Polyak-Ruppert averaging. This ensemble technique tends to run the model multiple times, and use the average weights seen at the end of the run. There are three methods to average the weights, which all of them will be evaluated to obtain one that is able to result a more stable model and better performance.

The three methods are as follows:

- Straightforward averaging weights with last 10 models. (Method 1)
- Linearly decreasing weighted average with last 10 models. (Method 2)
- Exponentially decreasing weighted average using decay rate with last 10 models. (Method 3)

Below will discuss performance of each method on result of RMSE and compare the standalone model which uses random weights and ensemble model.

RON95 and Diesel

Model evaluated is configured with timesteps of 2, 8 neurons, ReLu activation function, Adam with learning rate of 0.1 and 50 epochs.

Validation Result		Standalone	Method 1 Ensemble	Method 2 Ensemble	Method 3 Ensemble
RMSE	**1**	0.006	0.003	0.003	0.004
	2	0.005	0.004	0.004	0.004

	3	0.003	0.003	0.004	0.004
	4	0.003	0.003	0.004	0.004
	5	0.005	0.004	0.004	0.004
	6	0.004	0.004	0.004	0.004
	7	0.004	0.003	0.004	0.004
	8	0.004	0.003	0.003	0.004
	9	0.005	0.004	0.003	0.004
	10	0.003	0.004	0.004	0.004
RMSE: Mean (Std)		0.004 (0.001)	0.004 (0.000)	0.004 (0.000)	0.004 (0.000)

Table 9: Performance of standalone model and ensemble models with different weighting equations.

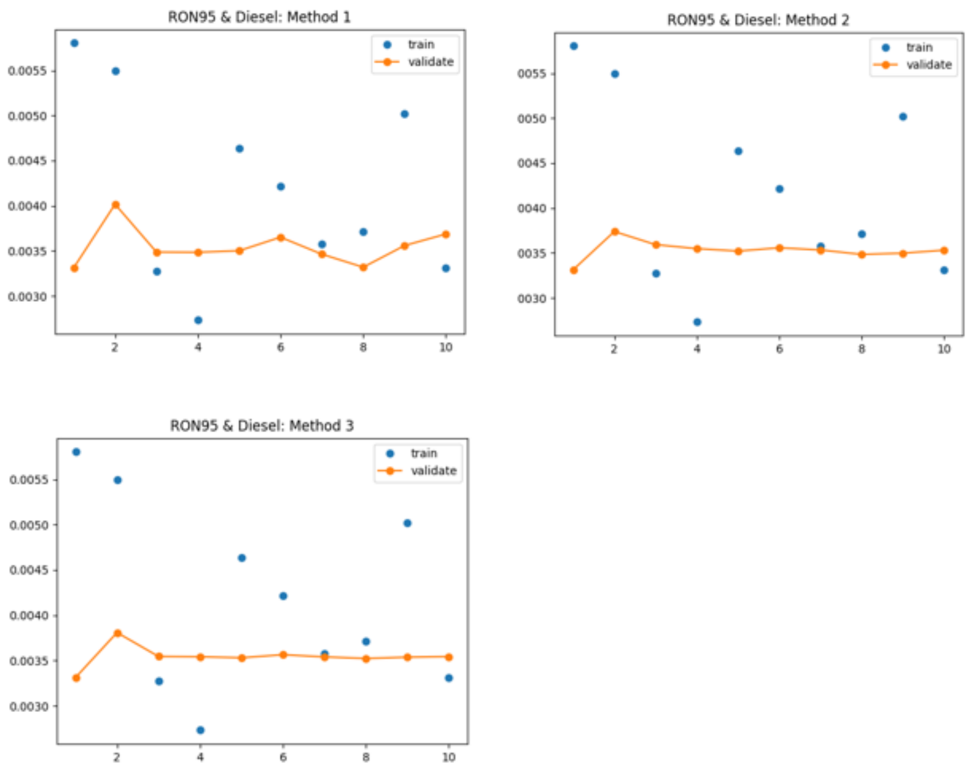

Figure 37: Comparison of different weighting equations.

From above result, it can be seen that all the ensemble models perform better than standalone model with lower standard deviation of RMSE. For different averaging weight methods, the RMSE mean and standard deviation results show the same values across all the models. Observation from plotting shows that standalone model produce the most unstable result, while ensemble models with averaging weight method 2 and 3 show stable results with method 3 has the most stable result.

RON97

The model evaluated is configured with input data of RON97, OPEC and exchange rate of RM/USD, timesteps of 2, 8 neurons, ReLu activation function, Adam with learning rate of 0.1 and 100 epochs.

Validation Result		**Standalone**	**Method 1 Ensemble**	**Method 2 Ensemble**	**Method 3 Ensemble**
RMSE	**1**	0.129	0.129	0.129	0.129
	2	0.120	0.134	0.132	0.133
	3	0.146	0.133	0.132	0.132
	4	0.130	0.129	0.131	0.131
	5	0.116	0.131	0.131	0.131
	6	0.141	0.128	0.130	0.131
	7	0.120	0.128	0.130	0.131
	8	0.130	0.130	0.130	0.131
	9	0.139	0.129	0.129	0.131
	10	0.129	0.129	0.129	0.131
RMSE: Mean (Std)		0.130 (0.009)	0.130 (0.002)	0.130 (0.001)	0.130 (0.001)

Table 10: Performance of standalone model and ensemble models with different weighting equations.

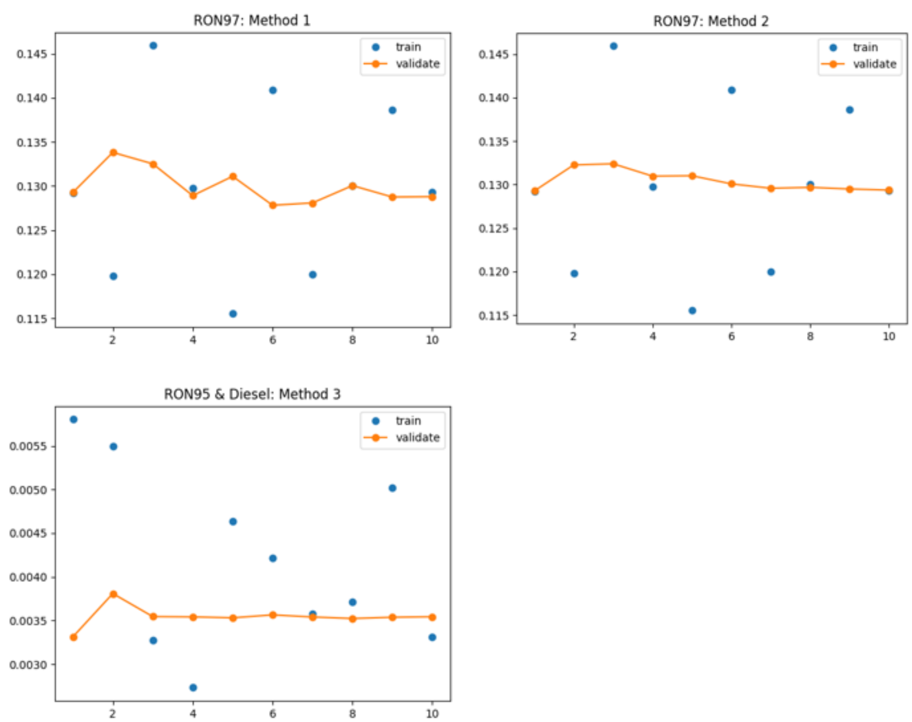

Figure 38: Comparison of different weighting equations.

From above result, it can be seen that all the ensemble models perform better than standalone model with lower standard deviation of RMSE. For different averaging weight methods, the RMSE standard deviation result of method 1 is higher than other methods. Observation from plotting shows that standalone model produce the most unstable result, while ensemble models with averaging weight method 2 and 3 show stable results with method 3 has the most stable result.

Model	Input Variables	Ensemble Model
RON95 and diesel	RON95, Diesel	Method 3
RON97	RON97, OPEC, RM/USD	Method 3

Table 11: Summary of suitable weighting average method for all models.

4.1.2.7 Hyperparameters Finalization

Model validation and evaluation will be conducted again to finalize the hyperparameters according to the result from the discussion above.

**Noted: Activ Func = Activation Function, Lrates = Learning Rate*

Input variable	No. of steps	No. of neurons in hidden layer	Activ Func	Lrates	No. of epochs	Weight Averaging Method	Runtimes	Validation RMSE: Mean (Std)
RON95 and Diesel								
RON95, Diesel	1	2	ReLu	0.1	50	Method 3	30	0.117 (0.001)
RON95, Diesel	1	4	ReLu	0.1	50	Method 3	30	0.115 (0.001)
RON95, Diesel	**1**	**8**	**ReLu**	**0.1**	**50**	**Method 3**	**30**	**0.006 (0.001)**
RON95, Diesel	2	2	ReLu	0.1	50	Method 3	30	0.014 (0.002)
RON95, Diesel	2	6	ReLu	0.1	50	Method 3	30	0.009 (0.001)
RON95, Diesel	2	8	ReLu	0.1	50	Method 3	30	0.0010 (0.002)
RON97								
RON97, OPEC, RM/USD	2	4	ReLu	0.01	50	Method 3	30	0.192 (0.002)
RON97, OPEC, RM/USD	2	8	ReLu	0.01	50	Method 3	30	0.126 (0.000)
RON97, OPEC, RM/USD	**2**	**8**	**ReLu**	**0.01**	**100**	**Method 3**	**30**	**0.107 (0.000)**
RON97, OPEC, RM/USD	2	10	ReLu	0.01	50	Method 3	30	0.164 (0.000)

Table 12: Hyperparameters tuning result.

Table 22 showed the models with the lowest RMSE mean values are the models with best selection hyperparameters. The finalized hyperparameters for the models are as follows:

- ***RON95 and Diesel:*** input data with their own historical data, timesteps of 1, 8 neurons number, ReLu activation function, Adam optimizer with learning rate of 0.1 and 50 epochs.
- ***RON97:*** input data with its historical data, OPEC oil data and exchange rates, timesteps of 2, 8 neurons number, ReLu activation function, Adam optimizer with learning rate of 0.01 and 100 epochs.

After completing model evaluation and validation, the trained or predictive models are ready to be deployed in Django.

4.2 Model Deployment

4.2.1 Prediction Result

Before deploying the models to Django, a testing set is fed into the trained models to observe the performance of the models and provide results as the past predictions to be displayed on the platform.

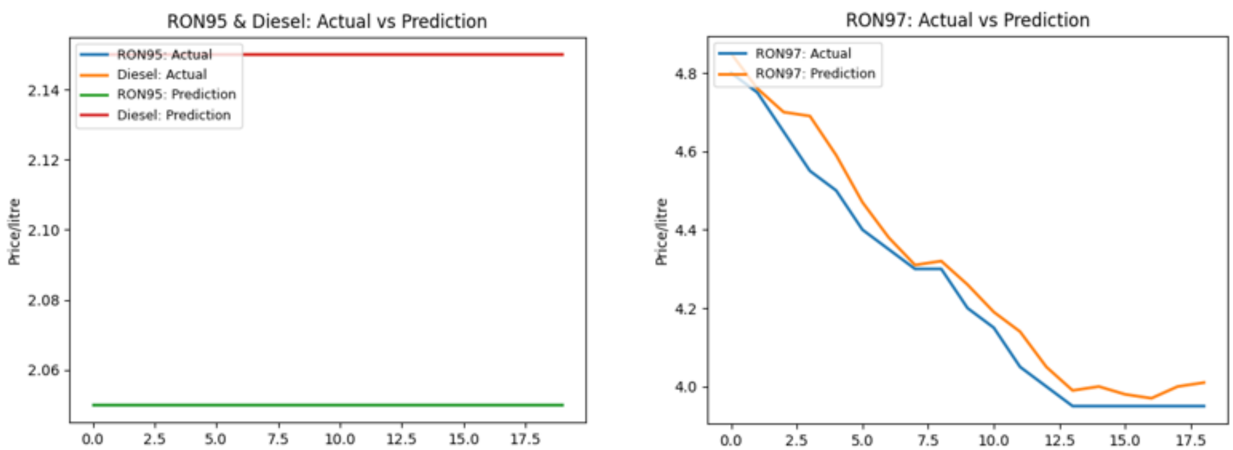

Figure 39: Actual versus prediction fuel prices for all models.

Plotting at the left showed the outputs for both fuels have perfectly fitted with actual values with a 0.003 RMSE mean score, while plotting at the right showed the outputs are capable of catching the changing patterns of the actual prices and resulted in a RMSE mean score of 0.069. Figure 46 and 47 below showed the price differences between the actual and prediction have not exceeded RM 0.40. Besides, it is allowed to make predictions for the following three weeks based on existing input data. It can be concluded that the models have succeeded in achieving the main objective of this project.

```
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
RON 95: Actual= 2.05, Prediction= 2.05 (0.00)
Diesel: Actual= 2.15, Prediction= 2.15 (-0.00)
```

Figure 40: Actual price versus prediction for RON95 and diesel.

```
Actual= 4.80, Prediction= 4.70 (0.10)
Actual= 4.75, Prediction= 4.69 (0.06)
Actual= 4.65, Prediction= 4.67 (-0.02)
Actual= 4.55, Prediction= 4.57 (-0.02)
Actual= 4.50, Prediction= 4.54 (-0.04)
Actual= 4.40, Prediction= 4.55 (-0.15)
Actual= 4.35, Prediction= 4.44 (-0.09)
Actual= 4.30, Prediction= 4.34 (-0.04)
Actual= 4.30, Prediction= 4.25 (0.05)
Actual= 4.20, Prediction= 4.21 (-0.01)
Actual= 4.15, Prediction= 4.23 (-0.08)
Actual= 4.05, Prediction= 4.15 (-0.10)
Actual= 4.00, Prediction= 4.10 (-0.10)
Actual= 3.95, Prediction= 4.05 (-0.10)
Actual= 3.95, Prediction= 3.97 (-0.02)
Actual= 3.95, Prediction= 3.93 (0.02)
Actual= 3.95, Prediction= 3.96 (-0.01)
Actual= 3.95, Prediction= 3.93 (0.02)
Actual= 3.95, Prediction= 3.93 (0.02)
```

Figure 41: Actual price versus prediction for RON97.

4.2.2 Test Plan

The models that have deployed on Django required a test plan to ensure all the features can function well.

Test Case	Expected Result	Actual Result	Pass/Fail	Corrective Action
Webpage accessing	Able to access to the webpage without error.	Same as expected	Pass	-
Dataset update	Able to update the dataset every time user clicks on the webpage	Same as expected	Pass	-
Model running	Able to run the model to make prediction every time user clicks on the webpage	Same as expected	Pass	-
Data displaying	Able to retrieve data from file and display it on webpage	Same as expected	Pass	-
Webpage responsiveness	Able to display HTML elements in proper way with different screen size	Same as expected	Pass	-

Table 13: Test plan.

5 Conclusion

In a nutshell, Artificial Neural Network (ANN) is a powerful adaptive system that are capable of predicting fuel prices with appropriate configurations. Literature review of others' works provided insight on the methods to carry out the lifecycle process such as the differences between validation and testing datasets, and the useful techniques that can be applied to improvise the model such as normalizing dataset and hyperparameters configurations using some thumb of rule. In the model developing process, trial and error is important to find out the best solution for this project. At the end of the project, the model built succeeded in achieving the objectives.

However, there is still improvement space for the model. For future directions, here are some recommendations that can be considered to improvise the model. Firstly, the topic for this project is a non-stationary time series problem. This statement is proven by conducting Augmented Dicky-Fuller test using Statsmodels Python library. The technique to transform non-stationary to stationary time series is differencing transformation to remove trends and seasonality from the data. There are a lot of differencing methods, in-depth investigation is required to develop suitable differencing to the data. Secondly, there is a big flaw existing in the current model, which is the trained model no longer getting updates on the new data to adjust the configurations. This will cause the model predictability to decrease over time. The techniques that can be applied to overcome this defect include (1) update model on new data only by fitting new data only, discarding the old model, (2) update model and both old and new data by simply fit a new model with both data, (3) ensemble of existing model and new model with approach (1), and (4) ensemble of existing model and new model with approach (2). All these strategies are recommended to be conducted to find the best fit solution. Lastly, the current model is deployed on back-end framework only. In the future, the model can be deployed to web hosting service like Heroku, which is a popular easy-to-use cloud platform.

References

[1] Arojas. “Multilayer Networks.” cs.csubak. https://www.cs.csubak.edu/~arojas/cs2680/FinalProject/multi_content.html (accessed May 1, 2022).

[2] J. Mahanta. “Introduction to Neural Networks, Advantages and Applications.” towards data science. https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207 (accessed May 1, 2022).

[3] “Finance Ministry: RON95 Petrol Capped At RM2.05, Diesel At RM2.15.” mof.gov.my https://www.mof.gov.my/portal/en/news/press-citations/finance-ministry-ron95-petrol-capped-at-rm2-05-diesel-at-rm2-15 (accessed May 31, 2022).

[4] “Malaysia Gasoline Prices.” tradingeconomics.com. https://tradingeconomics.com/malaysia/gasoline-prices (accessed Jun. 1, 2022).

[5] G. Zhang, B. E. Patuwo, and M. Y. Hu, “Forecasting with artificial neural networks: The state of the art,” *Int. J. Forecasting*, vol. 14, no. 1, pp. 35-62, Mar. 1998, doi: https://doi.org/10.1016/S0169-2070(97)00044-7.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533-536, Oct. 1986, doi: https://doi.org/10.1038/323533a0.

[7] C. Olah. “Understanding LSTM Networks,” colah.github.io. https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed Jun. 22, 2022).

[8] S. Hochreiter and J. Schmidhuber, “Long Short-term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997, doi: https://doi.org/10.1162/neco.1997.9.8.1735.

[9] G. Rishikesh and M. Boopathi, “Neural network-based fuel price prediction,” *Int. J. Scientific Technol. Res.*, vol. 7, no. 6, pp. 6-11, Jun. 2018. [Online]. Available: https://research.vit.ac.in/publication/neural-network-based-fuel-price-prediction.

[10] J. Sivaprakash and K. S. Manu, “Forecasting Crude Oil price using Artificial Neural Network model,” *Asian J. Manage.*, vol. 12, no. 3, pp. 321-326. 2021, doi: https://doi.org/10.52711/2321-5763.2021.00049.

[11] N. Gupta and S. Nigam, “Crude Oil Price Prediction using Artificial Neural Network,” *Procedia Comput. Sci.*, vol. 170, pp. 642-647, Apr. 2020, doi: https://doi.org/10.1016/j.procs.2020.03.136.

[12] S. Kulkarni and I. Haidar, "Forecasting Model for Crude Oil Price Using Artificial Neural Networks and Commodity Future Prices," *Int. J. Comput. Sci. Inf. Secur.*, vol. 2, no. 1, Jun. 2009, doi: https://doi.org/10.48550/arXiv.0906.4838.

[13] P. Ramani and P.D. Murarka, "Stock Price Prediction using Multi-Layer Feed Forward Neural Network," *JIMS 8i-Int. J. Inf. Commun. Comput. Technol.*, vol. 1, no. 1, pp. 1-3, Jul. 2013. [Online]. Available: https://www.jimsindia.org/8i_journal/stock_price_prediction_revisedpaper.pdf.

[14] Z. Aman, L. Ezzine, Y. F. E. Bahi, and H. E. Moussami, "Improving the modeling and forecasting of fuel selling price using the radial basis function technique: A case study," *J. Algorithms Comput. Technol.*, vol. 13, Oct. 2019, doi: https://doi.org/10.1177/1748302619881120.

[15] J. Fattah, L. Ezzine, Z. Aman, H. E. Moussami, and A. Lachhab, "Forecasting of demand using ARIMA model," *Int. J. Eng. Bus. Manage.*, vol. 10, Oct. 2018, doi: https://doi.org/10.1177/1847979018808673.

[16] H. Li and S. Huang, "Research on the Prediction Method of Stock Price Based on RBF Neural Network Optimization Algorithm," in *2020 Int. Conf. New Energy Technol. Ind. Develop. (NETID 2020)*, G. Kierans, H. Liu and E. H. K. Ng, Eds. Feb. 2021, vol. 235, Art. no. 03088, doi: https://doi.org/10.1051/e3sconf/202123503088.

[17] N. Moitra, S. Saxena, P. Raj, R. Kumar, and M. V. Kumar, "Crude Oil Price Prediction Using LSTM," *J. Eng. Sci*., vol. 11, no. 6, pp. 736-741, Jun. 2020. [Online]. Available: https://jespublication.com/upload/2020-1106107.pdf.

[18] V. Gupta and A. Pankey, "Crude Oil Price Prediction Using LSTM Networks," *Int. J. Comput. Inf. Eng.*," vol. 12, no. 3, pp. 226-230, 2018. [Online]. Available: https://publications.waset.org/10008992/pdf.

[19] A. Moghar and M. Hamiche, "Stock Market Prediction Using LSTM Recurrent Neural Network," *Procedia Comput. Sci*., vol. 170, pp. 1168-1173, 2020, doi: https://doi.org/10.1016/j.procs.2020.03.049.

[20] R. Sokkalingam, R. M. N. Y. Sarpong-Streetor, M. Othman, H. Daud, and D. A. Owusu, "Forecasting Petroleum Fuel Price in Malaysia by ARIMA Model," in *Proc. 6th Int. Conf. Fundam. Appl. Sci.*, 2022, pp. 671-678, doi: http://dx.doi.org/10.1007/978-981-16-4513-6_58.

[21] R. M. N. Y. Sarpong-Streetor, R. Sokkalingam, M. Othman, H. Daud, and D. A. Owusu, "ARIMAX Modelling of RON 97 Price with Crude Oil Price as an Exogenous Variable in Malaysian", in *Proc. 6th Int. Conf. Fundam. Appl. Sci.*, 2022, pp. 679-691, doi: http://dx.doi.org/10.1007/978-981-16-4513-6_59.

[22] Ma, “Comparison of ARIMA, ANN and LSTM for Stock Price Prediction,” in *2020 Int. Symp. Energy, Environ. Sci. Eng. (ISEESE 2020)*, S.O. Oladokun and S. Lu, Eds. Dec. 2020, vol. 218, Art. no. 01026, doi: http://dx.doi.org/10.1051/e3sconf/202021801026.

[23] L. Chen, Z. Qiao, M. Wang, C. Wang, R. Du, and H. E. Stanley, "Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market?," in *IEEE Access*, vol. 6, pp. 48625-48633, Jul. 2018, doi: https://doi.org/10.1109/ACCESS.2018.2859809.

[24] A. Shabri and R. Samsudin, “Daily Crude Oil Price Forecasting Using Hybridizing Wavelet and Artificial Neural Network Model,” *Math. Problems Eng.*, vol. 2014, Jul. 2014, Art. no. 201402, doi: https://doi.org/10.1155/2014/201402.

[25] J. Qiu, B. Wang, and C. Zhou, “Forecasting stock prices with long-short term memory neural network based on attention mechanism,” *PLoS ONE*, vol. 15, no. 1, Jan. 2020, Art. no. e0227222, doi: https://doi.org/10.1371/journal.pone.0227222.

[26] M. Kuhn and K. Johnson, “Data Splitting,” in *Applied Predictive Modeling*, New York, NY, USA: Springer, 2013, ch. 4, sec. 4.3, pp. 67. Accessed: Jun. 24, 2022. [Online]. doi: https://doi.org/10.1007/978-1-4614-6849-3.

[27] K. Kpodar and C. Abdallah, “Dynamic fuel price pass-through: Evidence from a new global retail fuel price database,” *Energy Econ.*, vol. 66, pp. 303-312, Aug. 2017, doi: https://doi.org/10.1016/j.eneco.2017.06.017.

[28] L. H. Ederington, C. S. Fernando, S. A. Hoelscher, T. K. Lee, and S. C. Linn, “A review of the evidence on the relation between crude oil prices and petroleum product prices,” *J. Commodity Markets*, vol. 13, pp. 1-15, Mar. 2019, doi: https://doi.org/10.1016/j.jcomm.2018.09.002.

[29] Y. H. Go, L. K. Chin, Y. C. Kuah, and C. Y. Wei, “Information Spillover Between Crude Oil and Stock Markets: Evidence from Subsidy Cut for RON95 Fuel Price in Malaysia,” *Global Bus. Rev.*, vol. 19, no. 4, pp. 889-901, Nov. 2017, doi: https://doi.org/10.1177/0972150917731409.

[30] X. Zhang, “Dynamic Correlations Between Crude Oil Futures Prices,” *Energy*, vol. 3, no. 1, 2022, doi: https://doi.org/10.46557/001c.30057.

[31] “Petroleum Gas in Malaysia.” oec.world. https://oec.world/en/profile/bilateral-product/petroleum-gas/reporter/mys (accessed Jun. 6, 2022).

[32] J. Chen. “Petrodollars.” Investopedia. https://www.investopedia.com/terms/p/petrodollars.asp#:~:text=No%2C%20petrodollars%20are%20simply%20U.S.,sometimes%20been%20called%20petrodollar%20recycling (accessed Jun. 6. 2022).

[33] J. Beckmann, R. Czudaj, and V. Arora, "The Relationship between Oil Prices and Exchanges Rates: Theory and Evidence," *Independent Statist. Anal.*, pp. 1-62, Jun. 2017. [Online]. Available: https://www.eia.gov/workingpapers/pdf/oil_exchangerates_61317.pdf.

[34] "How to Calculate Petrol Price in Malaysia." carsome.my. https://www.carsome.my/news/item/calculate-petrol-price (accessed Jun 7, 2022).

[35] BBC. "Malaysia to end petrol and diesel subsidies." BBC.com. https://www.bbc.com/news/business-30140728 (accessed Jun 7, 2022).

[36] C. M. Wheeler, J. Baffes, A. Kabundi, G. K. Hanlon, P. S. Nagle, and F. L. Ohnsorge, "Adding Fuel to the Fire: Cheap Oil during the COVID-19 Pandemic," *World Bank Group*, Jul. 2020. [Online]. Available: https://documents1.worldbank.org/curated/en/284371594670190475/pdf/Adding-Fuel-to-the-Fire-Cheap-Oil-during-the-COVID-19-Pandemic.pdf.

[37] N. Antonakakis, R. Gupta, C. Kollias, and S. Papadamou, "Geopolitical risks and the oil-stock nexus over 1899–2016," *Finance Res. Lett.*, vol. 23, pp. 165-173, Nov. 2017, doi: https://doi.org/10.1016/j.frl.2017.07.017.

[38] J. C. Dike, "Does Climate Change Mitigation Activity Affect Crude Oil Prices? Evidence from Dynamic Panel Model," J. Energy, vol. 2014, Dec. 2014, Art. no. 514029, doi: https://doi.org/10.1155/2014/514029.

[39] "Latest Petrol Price for RON95, RON97 & Diesel in Malaysia." comparehero.my. https://www.comparehero.my/transportation/articles/latest-petrol-price-ron95-ron97-diesel (accessed Jun. 24, 2022).

[40] "Weekly Europe Brent Spot Price FOB (Dollars per Barrel)." eia.gov. https://www.eia.gov/dnav/pet/hist/leafhandler.ashx?n=pet&s=rbrte&f=w (accessed Jun. 24, 2022).

[41] "Weekly Cushing, OK WTI Spot Price FOB (Dollar per Barrel)." eia.gov. https://www.eia.gov/dnav/pet/hist/leafhandler.ashx?n=pet&s=rwtc&f=w (accessed Jun. 24, 2022).

[42] "OPEC Basket Price." opec.org. https://www.opec.org/opec_web/en/data_graphs/40.htm (accessed Jun. 24, 2022).

[43] "MYR/USD (MYRUSD=X)." finance.yahoo.com. https://finance.yahoo.com/quote/MYRUSD%3DX/history?p=MYRUSD%3DX (accessed Jun. 24, 2022).