The Implementation of GloVe Text Embeddings and Numerical Indicators Analysis for Plastic Resin Price Prediction

Sun Sirisut

A Report Submitted in Partial Fulfillment of the Requirements

for SH701 Independent Study

Master of Science in Computer and Information Technology

School of Science and Technology

University of the Thai Chamber of Commerce

2021

The Implementation of GloVe Text Embeddings and Numerical Indicators Analysis for Plastic Resin Price Prediction

Sun Sirisut

A Report Submitted in Partial Fulfillment of the Requirements

for SH701 Independent Study

Master of Science in Computer and Information Technology

School of Science and Technology

University of the Thai Chamber of Commerce 2021

Research Title	The Implementation of GloVe Text Embeddings and Numerical Indicators Analysis for Plastic Resin Price Prediction
Name	Sun Sirisut
Degree	Master of Science
Major Field	Computer and Information Technology
Thesis Advisor	Asst.Prof.Dr.Watsawee Sansrimahachai
Graduate Year	2021

ABSTRACT

Price forecasting is one of the fundamental techniques used in most businesses to improve the competitiveness and decision-making level. Nonetheless, it is non-trivia task to make a model that provides high accuracy price prediction, especially in modern enterprises with ever longer, and more complex supply chains across the globe. In the classical approach for predictive problem, researchers applied the time series forecasting, but no decent outcome has been developed so far. This work suggests a new way to tackle this problem in the modern complex business world to predict the price of plastic resin by using the integration of textual information and numerical indicators input to deep learning models. Since the traditional methods which based on historical price itself is not sufficient, external data like economic indicators or textual information gathered from news articles, can improve the performance of the models by catching the overall global economic sentiment. Word semantic is retrieved as a vector representation from pre-trained word embeddings called Global Vectors for Word Representation (GloVe). In additional, deep learning models have gained great attention in the past decade after showing promising performance in various applications including Natural Language Processing (NLP), computer vision, and voice recognition. Hence, deep leaning models, Artificial Neural Network (ANN) and Recurrent Neural Network (RNN), are utilized in this research to deal with the complex and fluctuated price of plastic resin. Later, average feature vectors from news headlines constructed with GloVe pre-trained text embedding are then stacked to prices and numerical indicators time series data and fed to the deep learning models for training. The consequences of adjusting the hyper-parameter, including window size, hidden size, number of layers, hidden layers, number of nodes, have been explored. The state of finetuning the embedding layer for extracting features from words are also evaluated. All models are trained for the optimal condition that can generalize well to even new unseen data. The models' performances are validated with root mean square error metric. The outcomes are the robust models that show sufficient and satisfied result for plastic resin price prediction which can improve the decision-making quality. The training, validation, and test loss of ANN is between 100-200, 15-40, 15-40 respectively. While The training, validation, and test loss of RNN is between 150-300, 40-70, 40-70 respectively. The overall validation and test losses are much higher than what have got from ANN. RNN has more training parameters than ANN which could be a reason why RNN shows overfitting problem. This research has also proposed new models designed to handle time series input data with a combination between textual and numerical data and contribute a new alternative strategy in long-complex supply chain petrochemical industry for more accurate price prediction which is the starting point for developing even more sophisticated and more accurate models in the future.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep and sincere gratitude to all my research supervisors, Asst.Prof.Dr.Watsawee Sansrimahachai, Dr.Manachai Toahchoodee, Asst.Prof.Dr.Ittipong Khemapech, and Dr.Chatchai Wangwiwattana who always guide and advise me through all stage of doing this research. This study would not have been possible without the help, support and guidance from the supervisor committee.

Lastly, I would like to acknowledge and give my thanks to my friends and colleagues for the continuous support and help that they had given me during the time I was working on this research.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

INTRODUCTION

1.1 Introduction and Problem Statement

Undeniably, plastics has become parts of people's life. Plastic production has reached 400 million tons annually in 2015 and continue to grow exponentially as you can see from Figure 1. In particularly, over the past two years when people need to keep social distancing and quarantine themselves at home due to covid pandemic outbreak, the plastic consumption is skyrocketing because of substantial growth of online food delivery services which require single use plastics inevitably.

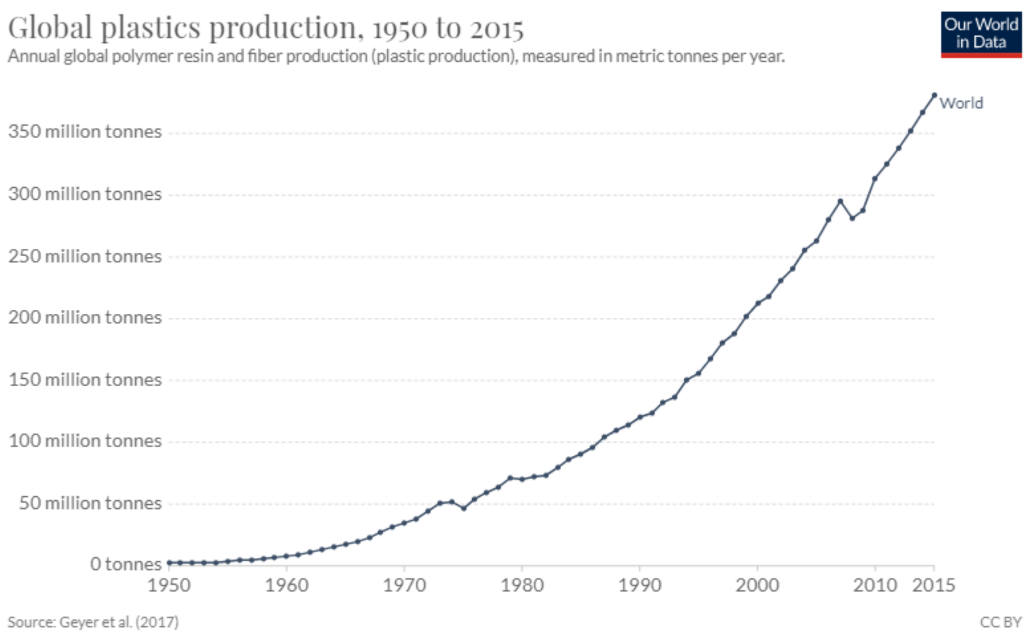

Figure 1 Global plastic production (Geyer et.al. 2017)

Plastic productions are part of petrochemical industry which was gradually developed from oil and gas operation by adding value to some of the products that have no use in fuels business. Afterward, petrochemical producers have sourced this raw material from oil and gas company along with other additives in order to process and turn these materials to finished plastic products and supply to various industries such as packaging, automotive, electronic applicants, adhesive, medical equipment, cable, and water pipe. As a result, petrochemical industry becomes a very long large complex supply-chain organization from sourcing raw materials, managing operations, and delivering products as shown in Figure 2. This leads to price volatility and make it difficult to predict the plastic resin prices. The prices are influenced by multiple numbers of factors in those long-complicated supply-chain. Nevertheless, price forecasting is still the main key challenge to overcome because accurate forecast can be a game-changer and ensures the sizable profit to the petrochemical producers.

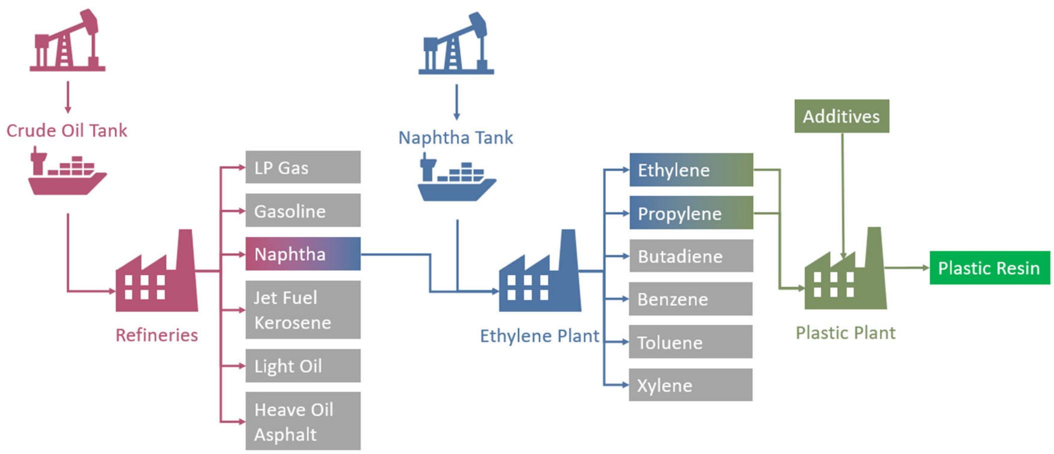

Figure 2 Petrochemical process flowchart

From the scientists' point of view, price prediction is the basic and fundamental problem, yet challenging in this circumstance. In the traditional predictive model as in (Zhao & Shen, 2011) and (Qing, Xiaoli, & Kun, 2012), analysts use a tactic called autoregressive (AR) which is a time series model that predict the future price based on its own historical prices by assuming that the future trend will hold similar to the past trend. However, the drawback of autoregressive time series is, it can only be used to forecast things linked to economics based on prior data. When predicted target is significantly influenced by external factors like social or news, the models would not perform as expected and give poor result. For that reason, there are some limitations since plastic prices are often affected by unknown external factors beyond the price of the products itself.

1.2 Objectives

- Develop deep learning regression models to predict the plastic resin price from both textual and numerical data as inputs.
- Describe the effects of model performance with different text feature extraction techniques.
- Examine relationship between news article and the prediction model.
- Investigate the hyper-parameter of the models such as windows size, number of hidden layers, or number of neural.
- Determine the performance of various deep learning models which are ANN and RNN

1.3 Significance of the Research

As explained earlier, plastic involve in varieties of industries which make it extremely difficult to predict. Not only the historical data of the product itself but also other related numerical indicators such as the price of crude oil or raw material of plastic resin. In additional, news articles have impacted on plastics resin price as well. For instance, throughout the beginning of 2018, there were conflicts between the U.S. and Iran which were the crude oil major producers. There are a lot of negative sentiment news articles associated Iran sanction. The crude oil price had risen from 60 to 75 dollars per barrel or 25% increasing, and it affected the plastic resin as well as finished plastic products. This shows that textual information contains some hidden valuable insights or useful price sentiments which can be added to help improve performance of the models. The data science academic community has extensively explored ways to extract features from wording articles. However, text feature processing is a little trickier than categorical one. The oldest method is to represent the words as one hot encoding vectors which people usually do with categorical data, but these vector representation does not capture the relationship between words since all vectors are orthogonal to each other. Therefore, the key element behind NLP is to find out the best numerical representation of the input text. The more improved model is to represent words as vectors of continuous numbers. The vector representations can be trained and show the relationship between word vectors. In 2013, one of the earliest successful models was proposed by Tomas (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), (Mikolov, Chen, Corrado, & Dean, 2013) called word2vec. Subsampling and negative sampling algorithm was used to increase the computational efficiency, as the result, the model can be trained from larger corpus text and provides more meaningful vector results. One year later, Jeffrey (Pennington, Socher, & Manning, 2014), researchers from Stanford, published Global Vectors for Word Representation (GloVe) model, which claimed to outperform word2vec on many language aspects including word similarity, word analogy, and named entity recognition tasks. Later in 2016, Facebook AI Researchers (Bojanowski, Grave, Joulin, & Mikolov, 2016) displayed word embedding named FastText. The idea is like word2vec, but it can even output a represented vectors for words that are not in the pre-trained text corpus. Afterward, Jacob and a group of researchers at Google AI Language introduced a new designed language representation model Bidirectional Encoder Representations from

Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2018). Unlike previous models that the models deal with the whole words, BERT has it owns way to split a word to smaller chuck of sub-words, hence it can deal with unknown words and outperforms other approaches.

1.4 Scope of the Study

The goal for this research aims to forecast plastic resin prices, as one of the main petrochemical products, through deep learning models by feeding with different kinds of inputs, numerical and wording data. Raw data is cleaned with data preprocessing method in appropriate manners to maximize the full potential of the data. Moreover, as necessity to extract vector representation from the text, word embedding called GloVe technique is investigated. The union features of text and number data is applied with deep learning models, specifically ANN and RNN, to learn the price pattern. Several hyperparameters are examined and finetuned. Up until now, there are no successful research for plastic price has been published yet. It is a common practice for the companies to keep their sale price confidential and shall not share or public the price to third party and make it nearly impossible to get dataset. However, this study is supported by one of the global petrochemical leaders who provides a plastic price dataset and makes this project possible.

CHAPTER 2

LITERATURE REVIEW

Although, price predictive model for plastic resin price has not been intensively studied before, there are tons of research about other products' price forecasting. Much attention has been drawn to determine stock's future price, since the dataset is mass accessible to everyone and would yield massive profit. They have suggested various methodologies and models to deal with price forecasting. Crude oil is another focus that has gotten a lot of attention because the global oil industry is worth trillions of dollars. Since crude oil is related to plastic resin as a raw material, some of those research knowledges can be applied here in quite similar manner. Several methods and techniques have been introduced to forecast the crude oil price and discussed in greater detail on the further sections.

2.1 Technique to Simplify the Problems

To improve proficiency of the model over AR model, many researchers have tried several other methods. According to (Anaghi & Norouzi, 2012) and (Mo & Tao, 2016), the authors applied moving average technique to smooths out price patterns by removing noise from short-term price variations. Nonlinear autoregressive exogenous models are employed in (Chuanjin Jiang & Fugen Song, 2010) and (Thakur, Tiwari, Kumar, Jain, & Singh, 2016) to predict exchange rate and petrol prices respectively. Some works simplify the model into classification model as shows in (Ratto, et al., 2018) when Andrea turned the regression problem to classification by predicting only whether the price going up or going down by using Support Vector Machine (SVM).

2.2 Deep Learning Models

Due to the rapid improvement of computer technologies over the last decade, more modern and sophisticated strategies known as Artificial Intelligence (AI) has adopted as one of the most vital methods used to tackle a broad range of applications. The preferred AI has been relayed on this prediction problem as well. ANN are models combined of multiple neural layers as shown in Figure 3. Activation functions, like Rectified Linear Unit (ReLU), can be applied after each neural layer to improve the model capability to handle with non-linearity. Loss function can be defined at the end of the model to compare with true ground for different kind of problem. For example, sigmoid can be use as loss function for binary classification, while loss mean square error can be used in regression problem. One example for ANN is presented in (Gupta & Nigam, 2020) to estimate the pattern of the crude oil price. The result shows good accuracy even there was an immediate huge change of the price. Researchers in (Mahdiani & Khamehchi, 2016) using ANN combined with Genetic Algorithm (GA) to enhance the efficiency of the prediction model. ANN-GA can perform well even have small amount of data points while ANN alone could not do so.

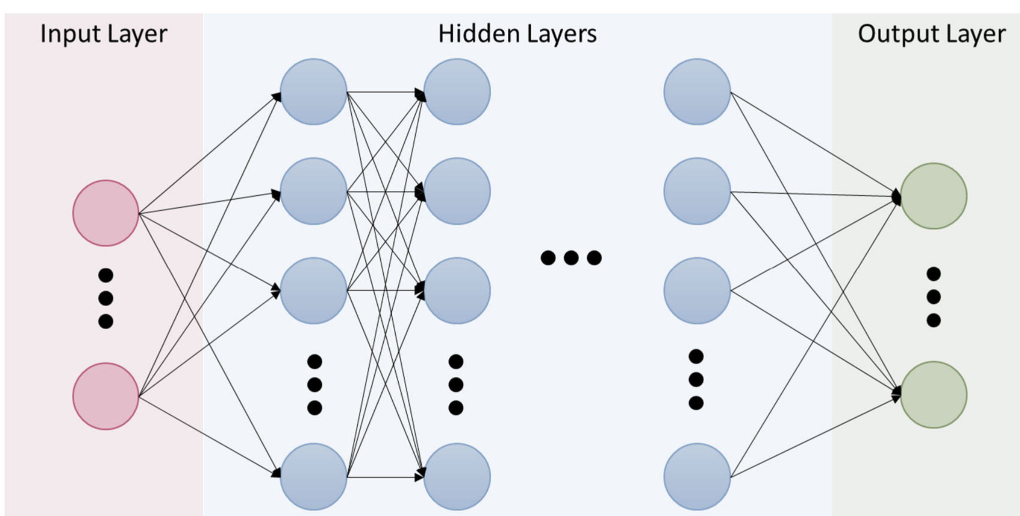

Figure 3 ANN structure (Feedforward Neural Network)

Apart from ANN, there is another type of model, specifically designed to handle sequential data which are RNN, Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). RNN is a kind of ANN that is used specifically in time series or sequential data structure which commonly use in human language or financial stock. Unlike ANN that have different weights parameters of each layer, RNN share the same weight parameter within each layer of the network. RNN can work just fine when it looks back on few time steps of sequential data as demonstrated in Figure 4. However, when a conventional RNN network is faced lengthy sequences data, it tends to lose information as its algorithm mostly relies on the lasted available information at the node. This problem happens when the RNN model backpropagates to a long sequence of time data and the accumulating of gradient becomes smaller and smaller. This problem can be defined as vanishing gradient. To overcome vanish gradient in RNN, derivative versions of RNN are made. Two most popular among them are LSTM and GRU. LSTM is specifically created to solve the vanishing gradient. In regular RNN, the previous input state is passed through tanh activation function which cause the vanishing gradient. On the other hand, in a single LSTM cell has more complex structure which can take a long-term memory from previous state without losing the information. GRU is designed to solve vanishing gradient like LSTM but has less gates comparing to LSTM, so GRU should take less computational power. GRU has only update and reset gate whereas LSTM has input, forget, and output gate. Authors in (Chen, He, & Tso, 2017) predict crude oil price using hybrid approach integrating price forecasting time series model based on the deep learning model. They select deep learning methods which are LSTM and Deep Belief Network (DBN). Random Walk (RW) combined with DBN is proven to be the best model over the others. Researchers in (Bristone, Prasad, & Abubakar, 2020) proposed k-core decomposition and LSTM to forecast the price moment of crude oil. The model was evaluated on 10 different crude oil prices, and the performance show better outcomes compared to traditional predictive model. Not only in petrochemical field that uses predictive model but also financial field.

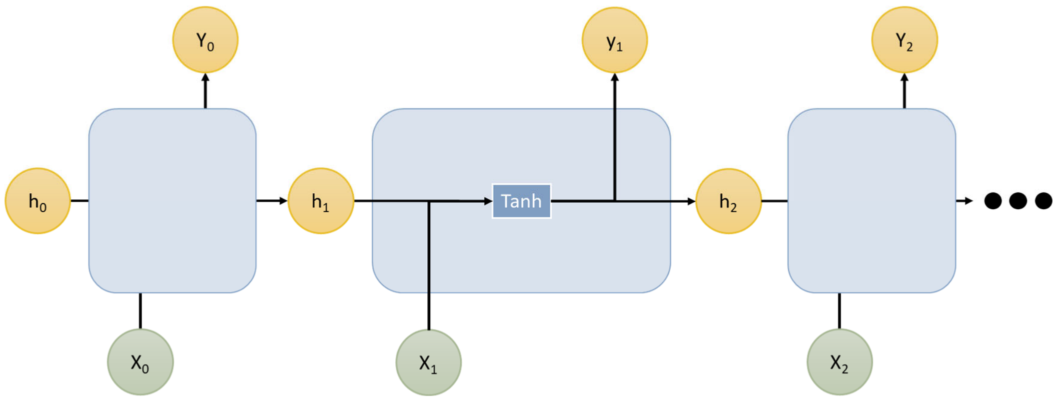

Figure 4 A single layer of sequential standard RNN

2.3 Text Embedding

Many achievements on NLP have inspired new researchers to attempt text sentiment analysis or text embedding to extract feature from the text and use for predictive models. For example, (Hu, Liu, Bian, Liu, & Liu, 2017) (Attanasio, Cagliero, Garza, & Baralis, 2019) (Mohan, Mullapudi, Sammeta, Vijayvergia, & Anastasiu, 2019) combine stock price time series with news to forecast the stock market sentiment. They take different approach to handle text. (Hu, Liu, Bian, Liu, & Liu, 2017) used word2vec pre-trained model, while (Attanasio, Cagliero, Garza, & Baralis, 2019) and (Mohan, Mullapudi, Sammeta, Vijayvergia, & Anastasiu, 2019) used positive-negative count.

Tweet text has been used to predict sentiment of stock market price (Kalyani, Bharathi, & Jyothi, 2016), (Sharma, Khemnar, Kumari, & Mohan, 2019). They both used polarity score which classify each work to whether positive, negative, or neutral and then count the number of each category to extract feature from text. Both works used Naive Bayed and SVM model to classify the sentiment. The accuracy is around 80-90% overall.

One of the researchers in (Picasso, Merello, Ma, Oneto, & Cambria, 2019), forecast the market sentiment of 20 stocks in NASDAQ100 index using both textual data and price data as inputs. News articles were processed to numerical data by two different methods called Loughran and McDonald (L&Mc) and AffectiveSpace. Rain Forest, Support Vector Machine, and feed forward neural network were applied for financial time series data. The integration between textual data and price data performs far better than price data alone. However, the performance of textual and price data together is not significantly shown the outstanding outcomes.

2.4 Influent Factor Analysis

In order to understand the supply demand of physical products like plastic resin, the other physical products have been observed. In (Qin & Xin, 2012), predict the supply demand of polyester filament was studied applied data mining method. The authors extracted information from various sources, such as import, export of the relative products and industries. Finally, they analyze and train prediction model on the data. The consequence of prediction values is closed to the actual values with small error around 1 – 5%.

Majority existing works focus on stock and crude oil time series data to predict the future sentiment of the products. In other words, the problems are simplified to binary classification, whether the products going up or down comparing to the present. Moreover, many researches used either only price or text sentiment as inputs without other numerical indicators. In this study, the integration of time series price, numerical indicators and text sentiment are investigated with deep learning regression models.

CHAPTER 3

METHODOLOGY

This research proposes a new deep learning-based forecasting models that incorporates classical time series autoregressive model, numerical indicators, and news embedding text features as prediction model inputs. Data are gathered from various reliable sources. Next, textual data need to be converted to numerical data. Even though, BERT has been proven to be one of the best algorithms for many tasks in NLP, they are large models. The smallest BERT are consisted of 12 encoder layers and 110 million training parameters. While the dataset on this study has only 371 training data points which is quite small and limited. Therefore, GloVe is selected for word embedding in this study due to its simplicity and efficiency. Wording sentences are applied with GloVe pre-trained text embedding to construct average feature vectors from energy news headlines, which are then stacked to prices and numerical indicators time series data. Finally, this study employs ANN and RNN models to forecast the plastic resin price following by evaluation step to measure the performance of the models. The overall process pipeline is presented in Figure 5.

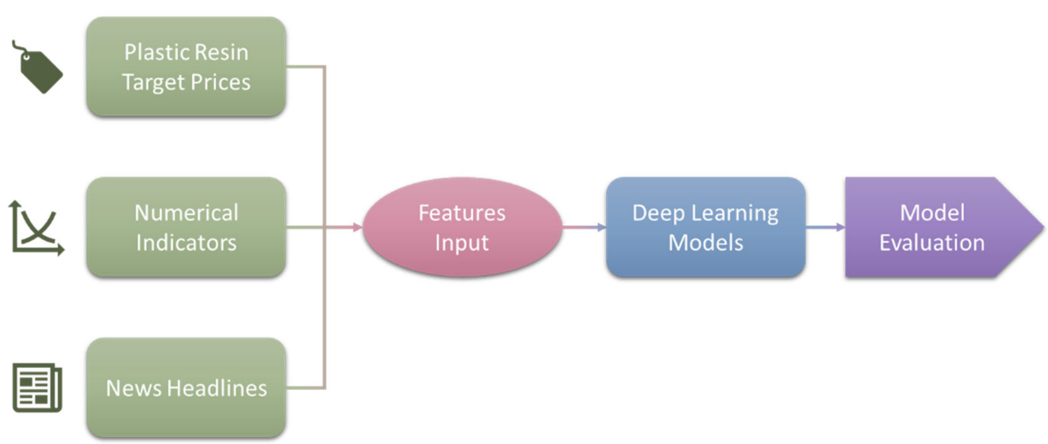

Figure 5 Process pipeline overview

3.1 Data Collecting

The stage of collecting data is important for the successive stages to come, as it is a foundation to build good models. The data must be reliable, accurate, and complete. The first dataset is the plastic resin price dataset which is the target of the regression problem in this study. The data is retrieved from one of the global petrochemical leaders as mentioned earlier. This dataset is a daily time series data from middle of 2014 until end of 2021.

Other numerical indicators, related to the plastic resin price, consists of 20 features obtained from Independent Commodity Intelligence Services (ICIS), a global marketing intelligence organization trusted by most of players in the petrochemical industries. Indicators includes 5 kinds of crude oil prices, 3 kinds of naphtha prices, 4 kinds of ethylene prices, and 8 kinds of polyethylene prices as weekly time interval from middle of 2014 to end of 2021. Crude oil, naphtha, and ethylene are raw material of plastic resin, while polyethylene is the kind of main product in this study.

News headlines are collected from Seeking Alpha (https://seekingalpha.com/) website, which is the world's largest investing community. The news articles are chosen under energy news section since they are the most related topic and contain sentiment features of the petrochemical industries. The dataset is also available from middle of 2014 to the end of 2021. The news releasing and character length distribution are demonstrated in Figure 6. The website usually releases around 20 to 30 articles per day, which are usually 50 to 75 characters long. On the left, there are outliers between 0-1, this could be a problem if the models predict based on daily cycles. However, the news will be concatenated as weekly cyclical patterns as will be discussed on the data preprocessing section. On the right, the distribution is right-skewed which implies that there will be a few longer sentences need to be fed to the models. Next to explore and get some ideas about the text dataset, the highest frequency words after remove stop words are also examined with wordcloud as show in Figure 7. The picture shows many significant words in petrochemical industries such as Exxon, Chevron, and Shell who are global petrochemical suppliers or products in petrochemical industries such as natural gas or crude oil.

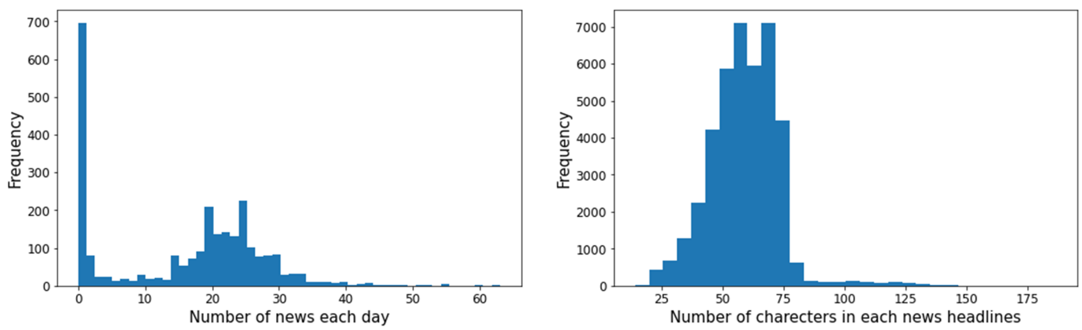

Figure 6 Distribution of variation in news headlines and character length

Figure 7 Most frequent words from wordcloud after removing stop words

3.2 Data Preprocessing

Initially, the frequency of datasets time series observations is not the same, so they need to be adjusted to have the same frequency before processing farther. The frequency of target price and news articles is daily while other indicators frequency is weekly. Hence, there are two options, whether to upsample other indicators frequency to daily or downsample target price and text news to weekly. In this case, to avoid the bias of prediction of upsampling, downsampling is used. Therefore, the target price and news dataset are decreased the frequency of the observation from daily to weekly. Target price dataset is applied quantity-weighted average price (QWAP) formula as shown in (1).

$$QWAP = \frac{\sum(Unit\ Price * Quantity)}{\sum Quantity} \quad (1)$$

Next, the target price for this regression problem is defined as a time series label in (2) where t is the number of time steps of the data, y_0 and y_1 is one week apart.

$$y = [y_0, y_1, \dots, y_t] \quad (2)$$

However, after observing the price moving trend, they are still fluctuating since these plastic resins are premium products, which can be sold on different prices depend on the locations and types of customers. To make the model be able to predict the price and identify the overall trend, this target price dataset needs to be smoothed out. In financial field, there is a method called moving average which usually used by traders and technical analysts to investigate the price movement of stocks. It is simply an average of sequential data points over a given time period. In this study volume weighted moving average is used. The longer the period, the smoother the curve. The effect of 1-week, 4-weeks, and 16-weeks volume weighted moving average is illustrated in Figure 8. The red line is the price average over a week, while the blue line is the price average over 16 weeks which becomes a lot smoother than the red line. After experienced with multiple volume weighted moving average time period, the period of 16 is chosen to be the final target dataset in the regression problem. The selling price values is removed from y axis for confidential reason for the company who provides this dataset.

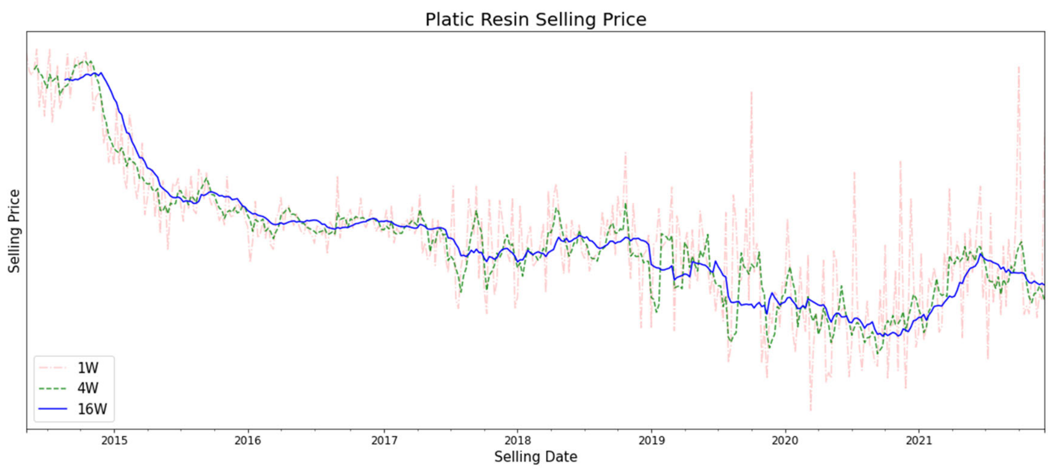

Figure 8 Target price on different volume weighted moving average

For the other indicators dataset, there are some missing values in number indicators, so they are filled with the average between the previous time step and the next time step. Next, as these features need to be fed to the deep learning models and used to update the weight parameters through gradient decent during backpropagation, they should be scaled to the same range with standardization to ensure that the gradient descent moves smoothly towards the minima and that gradient descent steps for all features are updated at the same rate. Moreover, standardization can help to speed up the convergence and make each individual indicator contributes to the models equally. In this particular study, StandardScaler() from scikit-learn is used as shown in eq (3)

$$z = \frac{x - \mu}{\sigma} \tag{3}$$

Lastly, the text information is unstructured data which cannot be input directly to the models. As the news dataset need to downdample from daily to weekly, all the headlines news is concatenated to multiple sentences over a week. The long lines of text must be tokenized to small pieces of token or word. Since the tokens in each row (each week) are of varying length, thus padding must be utilized to make all the text input have the same length. When examining the token lengths in the dataset as demonstrated in Figure 9, most of the text have 1,000 to 1,300 tokens long. Even though the longest token text is almost 2,000 tokens long, 1,300 is selected for the padding length If 2,000 padding is selected, this only adds padding token (0) to most of the text that would not help model to learn anything worthwhile and on top of that it increases the computational cost which will slow down the training duration. Then each token word is converted to token id corresponding GloVe pre-trained embedding weight matrix. Finally, text dataset is converted to a sequent of 1,300 token ids on each time interval (weekly)

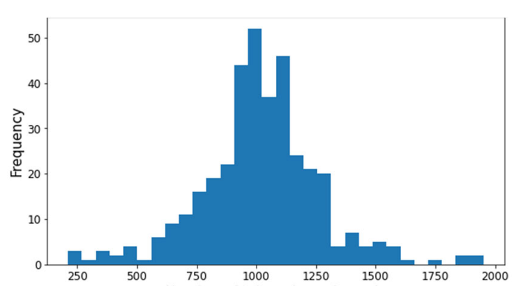

Figure 9 Distribution of variation in token lengths

3.3 Predictive Models

The goal of this research is to build a predictive model that can deal with complex data structure. Thus, two kinds of models, namely ANN and RNN is chosen, ANN is flexible and powerful while RNN is designed to handle time series and complex data structure. Even thought, there are more sophisticated RNN models like LSTM or GRU that built to solve problems that require long-term learning long-term temporal dependencies, this problem only study on 2-4 previous time steps (window size). Moreover, as the tanh activation function inside the RNN cell is main root cause for vanishing gradient as backpropagating back over multiple time step, ReLU can be used stead of tanh. Therefore, the vanishing gradient will not be the problem.

Since both ANN and RNN need to convert token ids to vectors, pre-trained GloVe weight matrix is loaded to embedding layer. However, since this weight matrix is already pre-trained and would already capture most of the semantic properties of the data, this embedding layer will be frozen at the beginning of the training process. After the models converge to the minimal training loss, this embedding layer will be unfrozen and finetuned the weight to improve the overall performance of the model. The embedding layer is simply a look up matrix. Each row is a vector corresponding to a token id of the row number. In this study, glove.6B.50d is used to extract the text features, and hence each token id is converted to 50-dimensional vector. To get the weekly news embedding vector, an average of all vectors over the week is calculated as (4) where n is number of words in all sentences. This will be done in embedding bag layer.

$$Vec_{sentence} = \frac{\sum Vec_{word}}{n} \quad (4)$$

Target price, numerical features and text features are concatenated to train models. This input features can be defined as a matrix (5) consisting of F rows and t columns, where F is number of features and t is number of window size. The result is a 71-dimensional vector constructed with 20 features from numerical indicators, 50 features from text embedding vectors, and 1 feature from target price. All features are described in Table 1.

Table 1 The features used to feed predictive machine learning models

Feature No.	Feature Name	Description
1-50	Text embedding	The vector representation from pre-trained GloVe model
51-55	Crude oil price	The prices of crude oil which are raw material of plastic
56-58	Naphtha price	The prices of naphtha which are raw material of plastic
59-62	Ethylene price	The prices of ethylene which are raw material of plastic
63-70	Polyethylene price	The prices of related plastic products
71	Target price	The prices of the target product itself

The window size is a hyper parameter and vary on this study to see the effects of different window size. For instance, the time-series windows size of 4 means, the current time step is input along with 4 previous time steps to train the model. The intuition is that the noise held in one time step may reduce by learning from multiple time steps. However, too large windows size can lead to over fitting the model to the training data and diminish model performance. Time-series windows size of 2, 3, 4 are chosen in this experiment.

$$X = \begin{bmatrix} x_0(0) & \cdots & x_t(0) \\ \vdots & \ddots & \vdots \\ x_0(F) & \cdots & x_t(F) \end{bmatrix} \quad (5)$$

Now all three datasets are preprocessed and ready to be fed to the models. There are total of 371 records (rows) on the dataset as weekly frequency from 2014 to 2021. Hence, if the window size is specified as 3, the total input will be 371 – 3 = 368. Then the data set will be split to 3 small sub datasets consisting of train, validation, and test set. Since the dataset is small and overlapping sets of sequential data, it is not practical to shuffle the data and randomly select train and validation chuck. On the other words, cross validation will not be utilized here. Thus, the dataset will be split to 3 part and each part will be input to the models at once as shown in Figure 10.

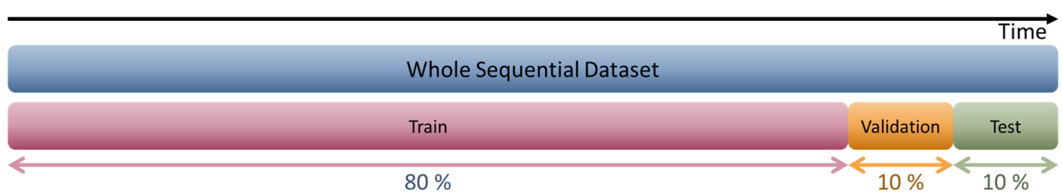

Figure 10 Train, validation, test dataset split

In the case of ANN, the input matrix X of dimension F by t will be flatten down to 1 dimension of F x t elements before feeding to Linear layer with ReLU activation function following by Batchnorm1D and Dropout (p=0.5) layer. This combination of layers is repeated multiple time (4 and 8) to make it comparable with number of layers in RNN. Number of nodes in feedforward also vary as 64, 128, and 256 to make it comparable with hidden size in RNN. The full model's architectures are illustrated in Figure 11.

For RNN, the matrix input X can be directly fed to the model since RNN can deal with time series data. The number of layers is varied as 4 and 8 as well as hidden size of 64, 128, and 256. On the last layer, only the last time step will be passed to the single Dropout(p=0.5) and Linear layer.

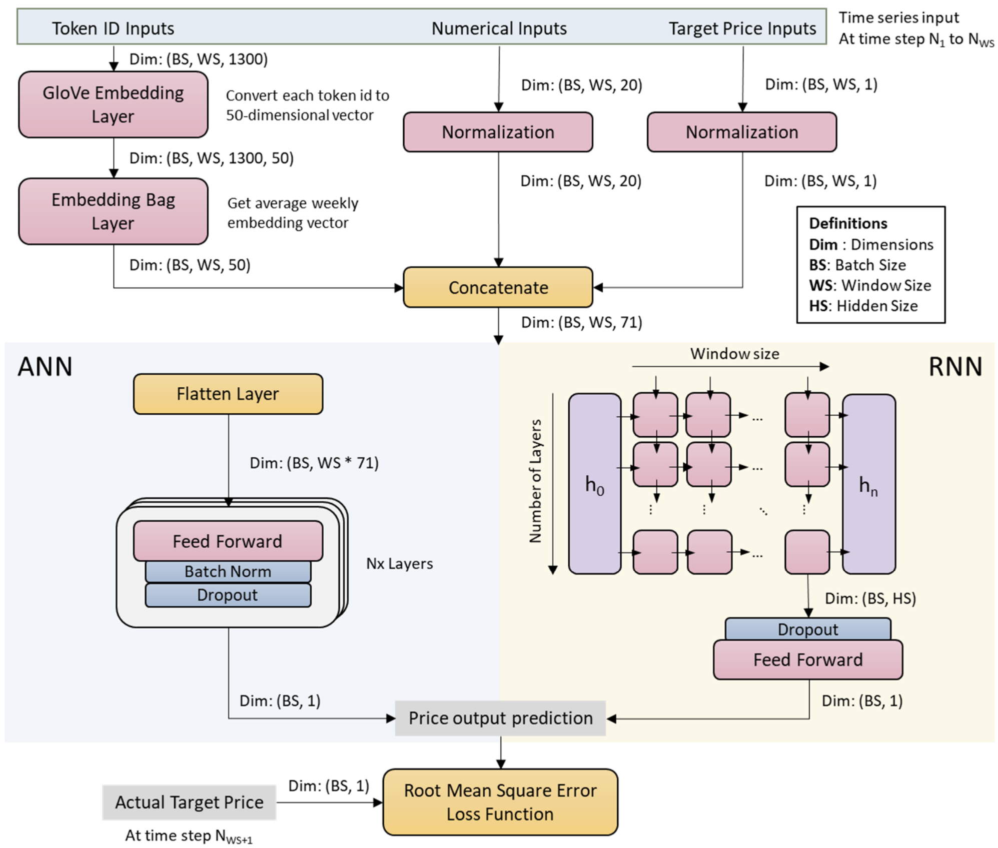

Figure 11 ANN and RNN model architectures.

The output of both ANN and RNN is a single value which is the predicted price of plastic resin which later is used to compute the root mean square error loss function by compare with actual price. The gradient of the loss will be backpropagated to adjust the weight of the models. Adam is used as optimizer with learning rate of 0.001. Validation set is utilized to set a rule for early stop to prevent the overfitting. If there is no improvement of validation loss in 100 epochs, the training process will be terminated. After the models have learned to fit the training data, now ANN or RNN layers will be frozen and embedding layer is unfrozen so it can finetune with this specific text dataset. The finetuned training continues with 500 epochs on Adam optimizer 0.003 learning rate.

CHAPTER 4

RESULTS

The models are implemented with PyTorch, open-source Python libraries for deep learning, on Visual Studio Code (VS Code). Some preprocessing methods are retrieved with scikit-learn as discussed in methodology section.

4.1 ANN Predictive Models

First, ANN is examined by initializing with various parameters including number of nodes (64, 128, 256) in each Linear layer and number of layers (4, 8). Each individual models are tested with different window sizes input from 2 to 4. Then the number of epochs before overfitting is recorded as well as root mean square loss values from training, validation, and test set. After finetuning of another 500 epochs, train, validation, and test loss values are tracked again. The complete records are listed in Table 2. The window size of the input doesn't make much different for the training loss but may have little effect on validation and test loss. As the window size increasing, the validation and test loss tends to increase slightly. This causes by the models have remembered the long-time pattern in training set which differ from validation and test set. As presented in Figure 12, the plastic resin price prediction from ANN 128 nodes and 4 hidden layers, after finetuning the embedding layer, the model can predict a little closer to the actual prices.

Overall, number of nodes in the hidden layers may help to reduce number of training epochs and the loss on the training set but this is the overfitting issue since it heightens the loss on validation and test set. Number of hidden layers help improve the models' performance by lowering the validation and train loss although it doesn't show much effect the on the training set. Training on word embedding layer will modify the predicted price more identical to the actual price.

Table 2 Results of training ANN and finetuning embedding layer

Window Size	Nodes	Hidden Layers	ANN Parameters	Train ANN Layers				After Finetune Embedding		
				Epochs	Train Loss	Val Loss	Test Loss	Train Loss	Val Loss	Test Loss
2	64	4	22,209	3,832	208.79	17.22	28.96	207.66	16.94	29.34
2	64	8	39,361	3,924	206.83	15.77	18.21	207.86	17.65	16.99
2	128	4	68,993	2,611	152.22	17.35	22.29	149.02	22.29	19.10
2	128	8	136,065	2,596	150.17	21.10	26.03	150.68	26.58	26.26
2	256	4	236,289	1,766	110.66	31.29	16.00	109.25	37.40	15.58
2	256	8	501,505	1,764	110.54	25.88	38.81	111.27	37.48	42.27
3	64	4	26,753	3,827	208.96	18.29	24.93	206.78	18.51	24.08
3	64	8	43,905	3,861	208.13	17.36	21.21	207.38	21.42	21.27
3	128	4	78,081	2,593	151.86	22.60	14.46	149.30	22.77	15.86
3	128	8	145,153	2,603	149.14	23.62	27.45	149.62	35.70	32.15
3	256	4	254,465	1,758	110.29	32.04	23.88	109.54	45.86	32.03
3	256	8	519,681	1,784	109.84	32.72	44.40	110.87	30.83	36.78
4	64	4	31,297	3,941	207.06	16.91	18.21	208.34	17.87	20.55
4	64	8	48,449	3,841	207.80	20.07	18.12	208.88	23.15	20.02
4	128	4	87,169	2,578	152.76	24.38	15.20	149.80	23.88	15.91
4	128	8	154,241	2,600	149.60	18.79	20.96	149.72	30.32	23.31
4	256	4	272,641	1,798	109.79	43.94	36.69	110.79	46.03	36.65
4	256	8	537,857	1,767	110.19	28.38	46.27	111.34	36.72	51.60

Figure 12 Plastic resin price forecast from ANN 128 nodes, 4 hidden layers

4.2 RNN Predictive Models

In the case of RNN, since the RNN structure is more complicated than ANN, the results are not quite obvious. However, as tabulated in Table 2, it can be summarized as following. When the window size increasing, the number of epochs also increases as well as validation and test loss. The train loss reduces when the window size and hidden size increase. The RNN models show more problem of an overfitting since it is built to look back thru the time, so it can remember the pattern of the training set. The overall validation and test losses are much higher than what have got from ANN. Nevertheless, as appeared on Figure 13, which is the plastic resin price prediction from RNN 256 hidden sizes and 4 layers, the data on the training and validation set are on the different range. Even on the training set data itself, the prices are varied from the upper most part to the very bottom, so it makes RNN a hard time to learn this dataset. Then when the validation dataset is provided to the models, the validation dataset comes in even lower range than training set and that why the RNN models perform a little poorer than ANN. The training, validation, and test loss of ANN is between 100-200, 15-40, 15-40 respectively. While The training, validation, and test loss of RNN is between 150-300, 40-70, 40-70 respectively. RNN also has more training parameters than ANN which could be another reason why RNN shows overfitting problem. When the models have lots of training parameters, they tend to remember data instead of looking for relationship in the data. Predicted prices after finetuning the embedding layer show improvement. The predicted line is refined and get the new shape that is even closer to the line of actual prices.

Table 3 Results of training RNN and finetuning embedding layer

Window Size	Hidden Size	Layers	RNN Parameters	Train RNN Layers				After Finetune Embedding		
				Epochs	Train Loss	Val Loss	Test Loss	Train Loss	Val Loss	Test Loss
2	64	4	33,793	1396	254.55	39.10	39.64	255.49	32.85	38.47
2	64	8	67,073	1113	275.31	54.50	41.69	277.35	60.65	41.94
2	128	4	124,929	882	200.91	54.98	42.71	200.08	62.49	43.61
2	128	8	257,025	992	195.58	76.54	71.65	192.88	57.60	71.60
2	256	4	479,233	1686	156.12	55.61	60.89	152.23	57.32	66.96
2	256	8	1,005,569	1528	156.80	47.93	56.44	153.19	46.39	56.01
3	64	4	33,793	1960	206.29	49.28	46.12	206.91	50.20	45.49
3	64	8	67,073	954	228.98	55.67	37.47	228.84	61.90	36.80
3	128	4	124,929	1823	155.76	46.87	65.74	153.99	47.34	64.40
3	128	8	257,025	1322	179.40	46.79	43.47	183.15	46.62	41.43
3	256	4	479,233	1460	143.29	43.10	54.27	142.21	31.93	50.61
3	256	8	1,005,569	1344	146.20	33.70	51.32	144.65	25.22	43.06
4	64	4	33,793	1759	208.63	45.94	50.02	203.98	56.02	52.58
4	64	8	67,073	1099	215.42	56.91	42.17	212.09	34.41	37.95
4	128	4	124,929	1416	155.38	37.15	44.56	149.31	33.81	43.64
4	128	8	257,025	1243	167.90	38.49	39.11	167.85	54.22	48.35
4	256	4	479,233	1997	135.23	37.33	51.88	134.06	29.64	50.74
4	256	8	1,005,569	1497	145.39	46.33	43.15	143.73	35.99	34.91

Figure 13 Plastic resin price forecast from RNN 256 hidden size, 4 layers

CHAPTER 5

CONCLUSION

The aim of this research is to develop a new approach for plastic resin price forecasting in long-complex supply chain petrochemical industry in order to improve the decision-making quality. This requires multiple sources of information to capture the overall market sentiment so both numerical data from economic indicators and textual data from news headlines are combined and used as inputs for the ANN and RNN models. Text inputs need to be converted to numerical representation before feeding to the models, and hence pre-trained GloVe has been employed. The consequences of adjusting the hyper-parameter, including window size, hidden size, number of layers, hidden layers, number of nodes, have been explored. The state of finetuning the embedding layer for extracting features from words are also evaluated. All models are trained for the optimal condition that can generalize well to even new unseen data. Even thought, we can force the models to train on more epochs and reduce the training loss, but it will lead to overfitting issue which reduce the price prediction accuracy on the validation and test set. RNN tends to be strongly influenced by the training set than ANN. As more data can be collected in the future, then, this larger dataset can help to solve the overfitting issue. Moreover, the algorithms like GloVe are dependent on a large-scale dataset to become effective and lessen the overfitting issue. The results of this experiment have proposed a new business strategy for plastic resin price prediction and shown great potential models for practical use in real businesses. Not only the petrochemical business, but also this can be applied to other business to forecast the prices of their end products by utilizing the datasets with combination between textual and numerical data. However, it is still too early to say that this is the best design. This is only the beginning to make regression models based on a merger of text and number. We hope that this contribution can offer some beneficial and enlighten more researchers to come up with even more sophisticated and more accurate models in the future. One idea to improve the models is changing the text feature extraction part from GloVe to a newer algorithm like BERT. BERT can understand the context of language better than GloVe which lead to better vector representations of the text which in turn can lead to improvement of the models. Moreover, the textual dataset in this study is scraped from an online website. Although, it was chosen from specific section that is most related to petrochemical industry and cleaned with various methods, it still contains a lot of noise in the dataset. With a large amount of additional meaningless noisy information, it confuses the models to get the accurate prediction. A company may put an effort to collect their own news dataset instead of scraping from online website. Then, this dataset is a clean, reliable, accurate, and complete dataset for specific price prediction of their own products. Obviously, it can help the models to fit the data better.

REFERENCES

Anaghi, M., & Norouzi, Y. (2012, Dec.). A model for stock price forecasting based on ARMA systems. 2012 2nd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA).

Attanasio, G., Cagliero, L., Garza, P., & Baralis, E. (2019). Combining News Sentiment and Technical Analysis to Predict Stock Trend Reversal. 2019 International Conference on Data Mining Workshops (ICDMW), 514-521.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016, Jul.). Enriching Word Vectors with Subword Information.

Bristone, M., Prasad, R., & Abubakar, A. (2020). CPPCNDL: Crude oil price prediction using complex network and deep learning algorithms. Petroleum, 6(4), 353-361. Retrieved from https://www.sciencedirect.com/science/article/pii/S2405656119301117

Chen, Y., He, K., & Tso, G. (2017). Forecasting Crude Oil Prices: a Deep Learning based Model. Procedia Computer Science, 122, 300-307. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050917326169

Chuanjin Jiang, & Fugen Song. (2010). Forecasting chaotic time series of exchange rate based on nonlinear autoregressive model. 2010 2nd International Conference on Advanced Computer Control.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018, Oct.). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Gupta, N., & Nigam, S. (2020). Crude Oil Price Prediction using Artificial Neural Network. Procedia Computer Science, 170, 642-647. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050920305913

Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T.-Y. (2017, Dec.). Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction.

Kalyani, J., Bharathi, P., & Jyothi, P. (2016, Jul.). Stock trend prediction using news sentiment analysis.

Mahdiani, M., & Khamehchi, E. (2016). A modified neural network model for predicting the crude oil price. Intellectual Economics, 10(2), 71-77. Retrieved from https://www.sciencedirect.com/science/article/pii/S1822801116300121

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, Jan.). Efficient Estimation of Word Representations in Vector Space.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013, Oct.). Distributed Representations of Words and Phrases and their Compositionality.

Mo, Z., & Tao, H. (2016, Aug.). A Model of Oil Price Forecasting Based on Autoregressive and Moving Average. 2016 International Conference on Robots & Intelligent System (ICRIS).

Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P., & Anastasiu, D. (2019, Apr.). Stock Price Prediction Using News Sentiment Analysis. 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService).

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Picasso, A., Merello, S., Ma, Y., Oneto, L., & Cambria, E. (2019). Technical analysis and sentiment embeddings for market trend prediction. Expert Systems with Applications, 135, 60-70. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417419304142

Qin, Y., & Xin, X. (2012). Research on the price prediction in supply chain based on data mining technology. 2012 International Symposium on Instrumentation & Measurement, Sensor Network and Automation (IMSNA), 2, 460-463.

Qing, C., Xiaoli, Z., & Kun, Z. (2012, Mar.). Research on Precipitation Prediction Based on Time Series Model. 2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring.

Ratto, A., Merello, S., Oneto, L., Ma, Y., Malandri, L., & Cambria, E. (2018). Ensemble of Technical Analysis and Machine Learning for Market Trend Prediction. 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 2090-2096.

Sharma, V., Khemnar, R., Kumari, R., & Mohan, B. (2019, Sep.). Time Series with Sentiment Analysis for Stock Price Prediction. 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT).

Thakur, A., Tiwari, A., Kumar, S., Jain, A., & Singh, J. (2016, Sep.). NARX based forecasting of petrol prices. 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO).

Zhao, Y., & Shen, L. (2011, May.). Application of time series auto regressive model in price forecast. 2011 International Conference on Business Management and Electronic Information.

APPENDICES

APPENDIX 1

BUILDING GLOVE EMBEDDING WEIGHT MATRIX

```
from torchtext.legacy.data import Field
# initiate Field instance
text_field = Field(
    tokenize='basic_english',
    lower=True
)
# apply preprocess manually
preprocessed_text = text_df['title'].apply(lambda x: text_field.preprocess(x))
# load glove6B simple embedding with 50d
text_field.build_vocab(
    preprocessed_text,
    vectors='glove.6B.50d'
)
# get the vocab instance
vocab = text_field.vocab
```

APPENDIX 2

CUSTOM DATASET FOR LOADING TO THE MODELS

```
class CustomerDatasetForWordEmbedding(Dataset):
    def __init__(self, num_df, text_df, target_df, window_size):
        self.window_size = window_size
        self.target = torch.tensor(target_df.values)
        # self.num = torch.cat((torch.tensor(target_df.values),
torch.tensor(num_df.values)), dim=1)
        # self.num = (self.num-mean)/std
        self.num = torch.tensor(num_df)
        self.text = text_df.apply(lambda x: text_field.numericalize([x]).reshape(-
1)).values
        self.text = pad_sequence([*self.text[:]], batch_first=True)
    def __len__(self):
        return len(self.num) - self.window_size
    def __getitem__(self, index):
        num_features = self.num[index:index+self.window_size]
        text_features = self.text[index:index+self.window_size]
        target_features = self.target[index+self.window_size]
        return num_features, text_features, target_features
```

APPENDIX 3

FEEDFORWARD NEURAL NETWORK MODEL DECLARATION

```
class ANN(nn.Module):
    def __init__(self, embedding_vector, window_size, in_size, node_size, out_size,
num_layers):
        super().__init__()
        self.embedding = nn.Embedding.from_pretrained(embedding_vector,
freeze=True, padding_idx=0)

        layerlist = []
        n_in = in_size*window_size

        for i in range(num_layers):
            layerlist.append(nn.Linear(n_in, node_size))
            layerlist.append(nn.ReLU(inplace=True))
            layerlist.append(nn.BatchNorm1d(node_size))
            layerlist.append(nn.Dropout(0.5))
            n_in = node_size

        layerlist.append(nn.Linear(node_size, out_size))

        self.ann = nn.Sequential(*layerlist)
```

```
def forward(self, N, T):
    T = self.embedding(T)
    mask = T != 0
    mask = mask.to(device)
    T = ((mask*T).sum(dim=2)/mask.sum(dim=2))
    X = torch.cat((N, T), dim=2)
    out = self.ann(X.view(X.size(0), -1))

    return out
```

APPENDIX 4

RECURRENT NEURAL NETWORK MODEL DECLARATION

```
class RNN(nn.Module):
    def __init__(self, embedding_vector, in_size, hidden_size, out_size, num_layers):
        super().__init__()
        self.in_size = in_size
        self.embedding = nn.Embedding.from_pretrained(embedding_vector,
freeze=True, padding_idx=0)
        self.hidden_size = hidden_size
        self.out_size = out_size
        self.num_layers = num_layers

        self.rnn = nn.RNN(
            input_size = self.in_size,
            hidden_size = self.hidden_size,
            num_layers = self.num_layers,
            nonlinearity = 'relu',
            dropout = 0.5,
            batch_first = True
        )
        self.dropout = nn.Dropout(0.5)
        self.dense = nn.Linear(self.hidden_size, self.out_size)

    def forward(self, N, T):
```

```
T = self.embedding(T)

mask = T != 0

mask = mask.to(device)

T = ((mask*T).sum(dim=2)/mask.sum(dim=2))

X = torch.cat((N, T), dim=2)

h_0 = torch.zeros(self.num_layers, X.size(0), self.hidden_size).to(device)

out, _ = self.rnn(X, h_0)

out = out[:,-1,:]

out = self.dropout(out)

pred = self.dense(out)


return pred
```