Addis Ababa Institute of Technology

School of Information Technology and Engineering

Cement Demand Prediction

Algorithms and Data Structures for Artificial Intelligence

Team Members

Sintayehu Zekarias

Fikir Awoke

Ashenafi Kifleyohans

Instructor:

Dr. Beakal Gizachew

January 2022

Acknowledgement

We would like to express our gratitude to our Course Instructor Dr. Beakal Gizachew, Assistant Professor, School of information Technology and Engineering, Addis Ababa Institute of Technology for his teaching and guidance throughout the course period. WE are very much thankful to the Adinas Construction Material supplier for providing us full access to their cement sales data. We are grateful to the lab assistants of SiTE, for providing us access to laboratory rooms til the completion of this project. We are also thankful to all group members for their undivided attention and cooperation to carry out this work.

Table of Contents

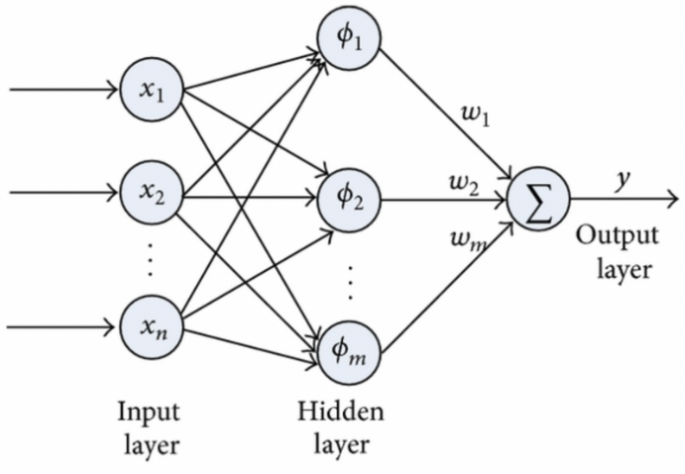
ABSTRACT

The main objective of this project is to predict the stock demand according to previous cement sales data. The company practices a traditional prediction system using spreadsheets which makes it difficult to deal with the big data. And the accuracy of sales forecasting is challenging due to the volatile and non-linear nature of the financial stock demand. Since stock demand forecasting is crucial in any sector, it has recently gained huge popularity to boost market operations and productivity due to new technologies. Therefore, we implemented a time-series prediction to forecast future stock demand. The project attempts to implement four machine learning algorithms including Radial Basis Function (RBF), Multivariate Regression, Random Forest and Long Short Term Memory(LSTM). We compared the performance of these algorithms and tested their accuracy. As a result, Multivariate Regression performed best in relation to the other techniques. Thus, the stock demand prediction will enhance the company to efficiently allocate resources for future growth and manage its cash flow.

1. Introduction

Predicting future demand has the ability to make informed business decisions and develop data-driven strategies. Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting, financial market prediction and so on. It uses the continuous data in a period of time to predict the result in the next time unit. Many time-series prediction algorithms have been implemented throughout this year showing their effectiveness in practice. The machine learning models implemented in this project use two attributes as an input and predict the stock demand. The attributes used are date and quantity of cement sales on a daily basis.[4]

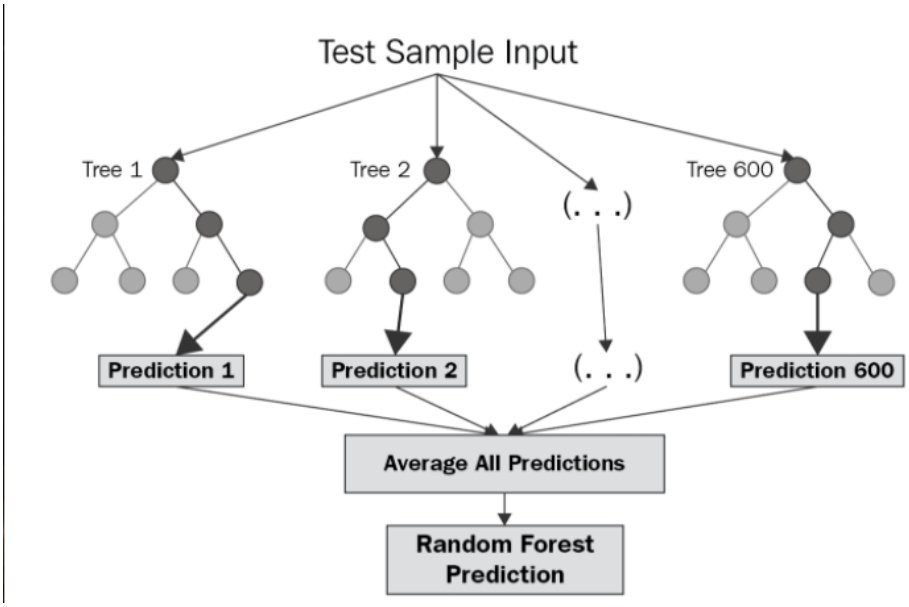
Radial Basis Function (RBF) is an artificial neural network used for function approximation problems. It is composed of three layers, namely the input layer, the hidden layer and the output layer. Each layer is fully connected to the preceding layer. The input layer is responsible for passing the vectors of the input data to the hidden layer. The hidden layer units are activated based on the associated radial basis function(Gaussians). The output layer gives a linear combination of the hidden units. The network outputs are determined on how the hidden layers activation function and weights between the hidden and output layers react to the input vectors.[1]



Multivariate Regression is an extension of multiple regression with one dependent variable and multiple independent variables. Based on the number of independent variables, we try to predict the output. Multivariate regression tries to find out a formula that can explain how factors in variables respond simultaneously to changes in others. The most important advantage of Multivariate regression is it helps

us to understand the relationships among variables present in the dataset. This will further help in understanding the correlation between dependent and independent variables.[8]

Random Forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. It is an ensemble learning method that operates by constructing a multitude of decision trees at training time. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.[6]



The above diagram shows the structure of a Random Forest in which the trees run in parallel with no interaction amongst them. And the output of the mean of the classes are used to predict all the trees.

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) based on artificial recurrent neural network (RNN) architecture used in the field of deep learning. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. A benefit of this type of network is that it can learn and remember over long sequences and does not rely on a pre-specified window lagged observation as input.

LSTMs also help solve exploding and vanishing gradient problems. In simple terms, these problems are a result of repeated weight adjustments as a neural network trains. With repeated epochs, gradients become larger or smaller, and with each adjustment, it becomes easier for the network's gradients to compound in either direction. This compounding either makes the gradients way too large or way too small. While exploding and vanishing gradients are huge downsides of using traditional RNN's, LSTM architecture severely mitigates these issues.[7]

2. Methodology

2.1. Data Collection and Exploration

We collected the Data from Adinas Cement Distribution Company. The collected data was in the form of a csv file. In exploring the dataset, we tried to observe all the attributes of the data, how they are categorized and their relations with each other. The attributes include date, cement type, price before VAT, quantity in quintal, quantity in ton, transport and total price. All the data is labeled on a daily basis and the data shows the amount of cement sold per day in quintal and ton. It also specifies the price before VAT, transportation cost and total price. There are three different cement types as indicated by the data. These cement types are categorized as OPC-Bulk, OPC-Bag and, PPC-Bag.

2.2. Data Preparation and Preprocessing

As the dataset has a lot more than what we need, we prepared it so that it can be suitable for our specific task. In the data preparation step we began by removing the unnecessary attributes from the dataset. These unnecessary attributes for our task are price before VAT, quantity in ton, transport and total price. After removing these elements from the given data, we found those elements important for the prediction process, these elements are the date, cement type and quantity in quintals.

The data preprocessing step focused on normalizing the cement types. While doing that we generated a unique number for each cement type and replaced the cement types with the generated numeric value. Then we filtered the cement types from the entire dataset and stored them separately as ppc-bag-dataset, opc-bag-dataset and opc-bulk-dataset. And lastly we filtered and got a unique date for each cement type, we did this because the data has a lot of duplicate dates for the same cement type.

In the process of normalizing the date, we followed almost the same step we used in normalizing the cement type. We generated a unique number for the unique dates in each of the cement type dataset. Then replaced the date with the uniquely generated numeric values. We also added up the quantity of quintals that are purchased at the same date. This will pull together the duplicated dates.

Last but not least we implemented smoothing. Smoothing techniques are kinds of data preprocessing techniques to remove noise from a data set. This allows important patterns to stand out. In market analysis, smoothed data is preferred because it generally identifies changes in the economy compared to unsmoothed data. [5]

In the dataset the quantity in quintal value doesn't have an evenly increasing or decreasing value that goes smoothly, instead it shoots to a higher pick once and rolls down immediately to a lower value, it doesn't have a smooth flow. For this reason we applied the moving average smoothing technique as it is a simple and common type of smoothing used in time series analysis and forecasting.

$$S_t = \frac{(X_{t-k} + X_{t-k+1} + X_{t-k+2} + \ldots\ldots + X_t)}{k}$$

In the following diagram we tried to show the difference between the smoothed and unsmoothed data on the OPC-Bulk Dataset.

3. Algorithm Implementation

3.1. Radial Basis Function (RBF)

RBF is implemented using the Exact Interpolation approach; we calculated gaussian RBF with standard deviation.[2]

$$\varphi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

To calculate the distance between the data points and center we used
get_distance(x,xk)

A Gaussian RBF to scale, and to approximate given functions

gauss_rbf(r)

To get the RBF of unknown data point x with respect to all centroids and to calculate the RBF and yk to get W, we used
fit (xk, yk)

To get the RBF of unknown data point xn with respect to all centroids and predict the output by calculating the dot product of RBF and W, we used
predict(xn)

3.2. Multivariate Regression

The mathematical equation for Multivariate Regression is as follows.

$$y = m1x1 + m2x2 + c$$

Where y is the dependent variable, x1 is the independent variable, m1 is the slope of x1, x2 is the second independent variable, m2 is the slope of x2, and c is the constant.

And for n number of inputs the equation must be represented as

$$y = \beta0 + \beta1.x1 + \beta2.x2 + \ldots.. + \beta n.xn$$

The cost of Multiple linear regression is calculated as

$$MSE=\frac{1}{2m}\Sigma\left(h_\theta(x)^{(i)} - y^i\right)^2$$

We performed the following tasks in implementing this algorithm to predict the future demand.

1. First we need to convert the data to supervised learning in order to get an input and an output as we only have input data which is the time series. This helps our model to learn the relationship between the input and the output in order to make a prediction.

   To do this we used the past seven days(a one week) data to make a prediction for today. So we considered seven values by shifting the quantity in quintal column by one unit, two units,.....seven units respectively. After doing this the dataset the first row will have a NAN value

for the quantity in quintal on the last day as there is no information prior to the first data. Similarly for 2 to 7 days back we will have NAN value. Before getting to the next step we need to remove the NAN values and then we will have seven input columns and one output column which is the quantity in quintal. The following diagram shows the idea in detail.

	Date	Quantity In Quintal
0	2.0	700.000000
1	3.0	733.333333
2	4.0	750.000000
3	5.0	744.000000
4	6.0	742.222222
5	7.0	750.476191
6	8.0	756.666667
7	9.0	756.543210
8	10.0	752.888889
9	11.0	747.254362

	Date	Quantity In Quintal	Quintal_LastDay	Quintal_2Dayssback	Quintal_3Dayssback	Quintal_4Dayssback	Quintal_5Dayssback	Quintal_6Dayssback	Quintal_7Dayssback
0	2.0	700.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	3.0	733.333333	700.000000	NaN	NaN	NaN	NaN	NaN	NaN
2	4.0	750.000000	733.333333	700.000000	NaN	NaN	NaN	NaN	NaN
3	5.0	744.000000	750.000000	733.333333	700.000000	NaN	NaN	NaN	NaN
4	6.0	742.222222	744.000000	750.000000	733.333333	700.000000	NaN	NaN	NaN
5	7.0	750.476191	742.222222	744.000000	750.000000	733.333333	700.000000	NaN	NaN
6	8.0	756.666667	750.476191	742.222222	744.000000	750.000000	733.333333	700.000000	NaN
7	9.0	756.543210	756.666667	750.476191	742.222222	744.000000	750.000000	733.333333	700.000000
8	10.0	752.888889	756.543210	756.666667	750.476191	742.222222	744.000000	750.000000	733.333333
9	11.0	747.254362	752.888889	756.543210	756.666667	750.476191	742.222222	744.000000	750.000000

	Date	Quantity In Quintal	Quintal_LastDay	Quintal_2Dayssback	Quintal_3Dayssback	Quintal_4Dayssback	Quintal_5Dayssback	Quintal_6Dayssback	Quintal_7Dayssback
7	9.0	756.543210	756.666667	750.476191	742.222222	744.000000	750.000000	733.333333	700.000000
8	10.0	752.888889	756.543210	756.666667	750.476191	742.222222	744.000000	750.000000	733.333333
9	11.0	747.254362	752.888889	756.543210	756.666667	750.476191	742.222222	744.000000	750.000000
10	12.0	757.205387	747.254362	752.888889	756.543210	756.666667	750.476191	742.222222	744.000000
11	13.0	765.231008	757.205387	747.254362	752.888889	756.543210	756.666667	750.476191	742.222222
12	14.0	775.877773	765.231008	757.205387	747.254362	752.888889	756.543210	756.666667	750.476191
13	15.0	784.597033	775.877773	765.231008	757.205387	747.254362	752.888889	756.543210	756.666667
14	16.0	793.372218	784.597033	775.877773	765.231008	757.205387	747.254362	752.888889	756.543210
15	17.0	808.987001	793.372218	784.597033	775.877773	765.231008	757.205387	747.254362	752.888889
16	18.0	822.067970	808.987001	793.372218	784.597033	775.877773	765.231008	757.205387	747.254362

2. Then we Splitted the data as Train and Test. Training data = 80% and Test data = 20%

3. Gave the train and test data for the fit and predict function in the linear regression class, which calculates the slope, y-intercept, cost and other values that need to be estimated for the prediction to be made.
4. Made a prediction, and tested the prediction in two ways. First we tested the prediction with already trained data, on the second one we gave the test data for the model so that it can make a prediction.
5. We calculated the Mean Squared Error

3.3. Random Forest Regression

We performed the following tasks in implementing this algorithm to predict the future demand.

1. Here in the first step we performed the same thing as we did for Multivariate regression, we used the past seven days(a one week) data to make a prediction for today. So we considered seven values by shifting the quantity in quintal column by one unit, two units,.....seven units respectively. After doing this the dataset the first row will have a NAN value for the quantity in quintal on the last day as there is no information prior to the first data. Similarly for 2 to 7 days back we will have NAN value. Before getting to the next step we need to remove the NAN values and then we will have seven input columns and one output column which is the quantity in quintal and we got the following result.

	Date	Quantity In Quintal	Quintal_LastDay	Quintal_2Dayssback	Quintal_3Dayssback	Quintal_4Dayssback	Quintal_5Dayssback	Quintal_6Dayssback	Quintal_7Dayssback
7	9.0	756.543210	756.666667	750.476191	742.222222	744.000000	750.000000	733.333333	700.000000
8	10.0	752.888889	756.543210	756.666667	750.476191	742.222222	744.000000	750.000000	733.333333
9	11.0	747.254362	752.888889	756.543210	756.666667	750.476191	742.222222	744.000000	750.000000
10	12.0	757.205387	747.254362	752.888889	756.543210	756.666667	750.476191	742.222222	744.000000
11	13.0	765.231008	757.205387	747.254362	752.888889	756.543210	756.666667	750.476191	742.222222
12	14.0	775.877773	765.231008	757.205387	747.254362	752.888889	756.543210	756.666667	750.476191
13	15.0	784.597033	775.877773	765.231008	757.205387	747.254362	752.888889	756.543210	756.666667
14	16.0	793.372218	784.597033	775.877773	765.231008	757.205387	747.254362	752.888889	756.543210
15	17.0	808.987001	793.372218	784.597033	775.877773	765.231008	757.205387	747.254362	752.888889
16	18.0	822.067970	808.987001	793.372218	784.597033	775.877773	765.231008	757.205387	747.254362

2. Then we Splitted the data as Train and Test. Training data = 80% and Test data = 20%
3. Gave the train and test data for the fit and predict function in the random forest class, and

    3.1. Pick at random k data points from the training set.

3.2. Build a decision tree associated with these k data points.

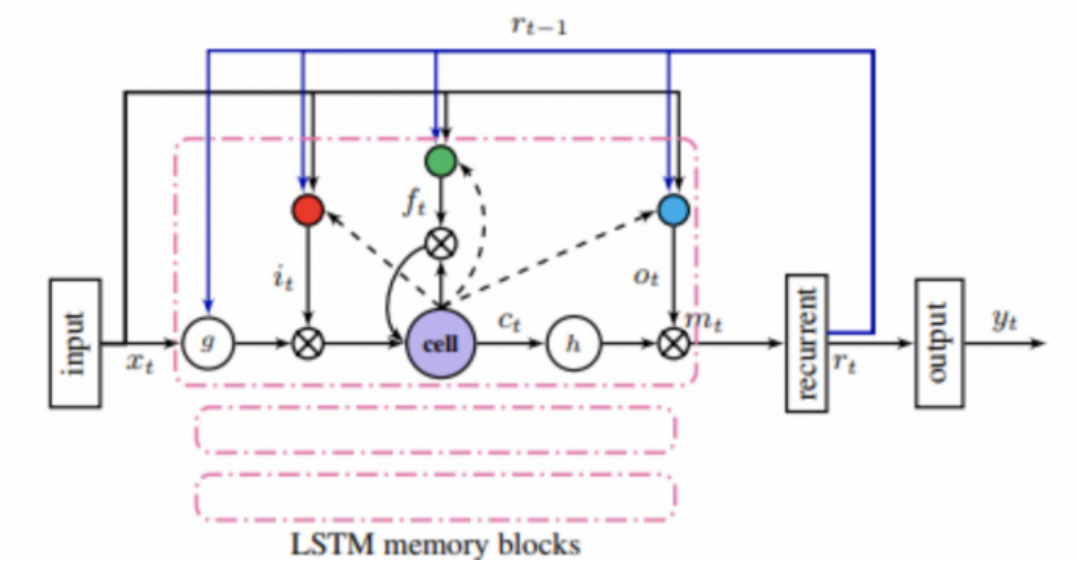
3.3. Choose the number N of trees you want to build and repeat steps 1 and 2.

3.4. For a new data point, make each one of your N-tree trees predict the value of y for the data point in question and assign the new data point to the average across all of the predicted y values.

4. Made a prediction, and tested the prediction in two ways. First we tested the prediction with already trained data, on the second one we gave the test data for the model so that it can make a prediction.
5. We calculated the Mean Squared Error

3.4. Long Short-Term Memory LSTM

LSTM uses a memory block containing memory cells that store the state of the network and additional units called gates. There are three different types of gates, namely the input gate, the output gate and the forget gate. The input gate controls the activations into the cell. The output gate controls the activation to the rest of the network. The forget gate can rest the memory of the cell. LSTM are most often fully connected to an output layer to make a prediction.

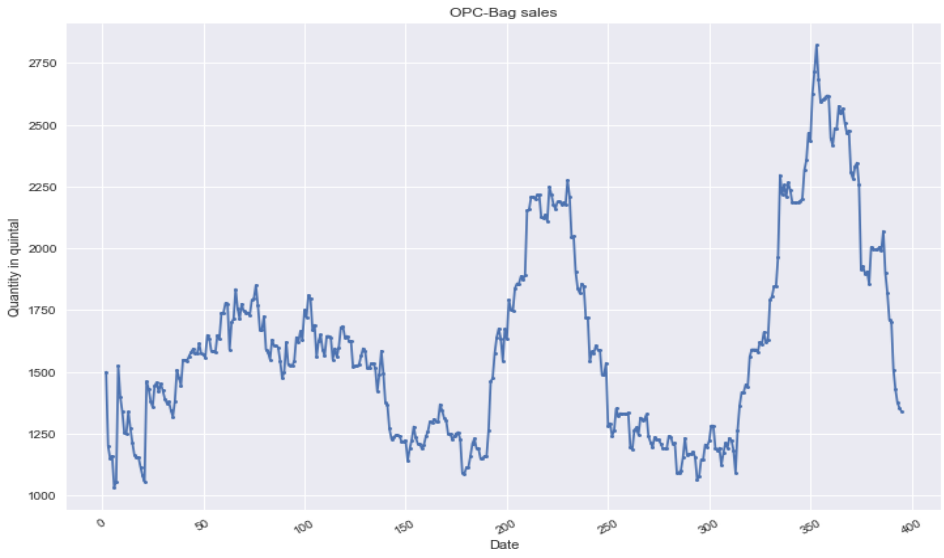


In our prediction the LSTM works by a lookback, it looks one day back. We performed the following tasks in implementing this algorithm to predict the future demand.

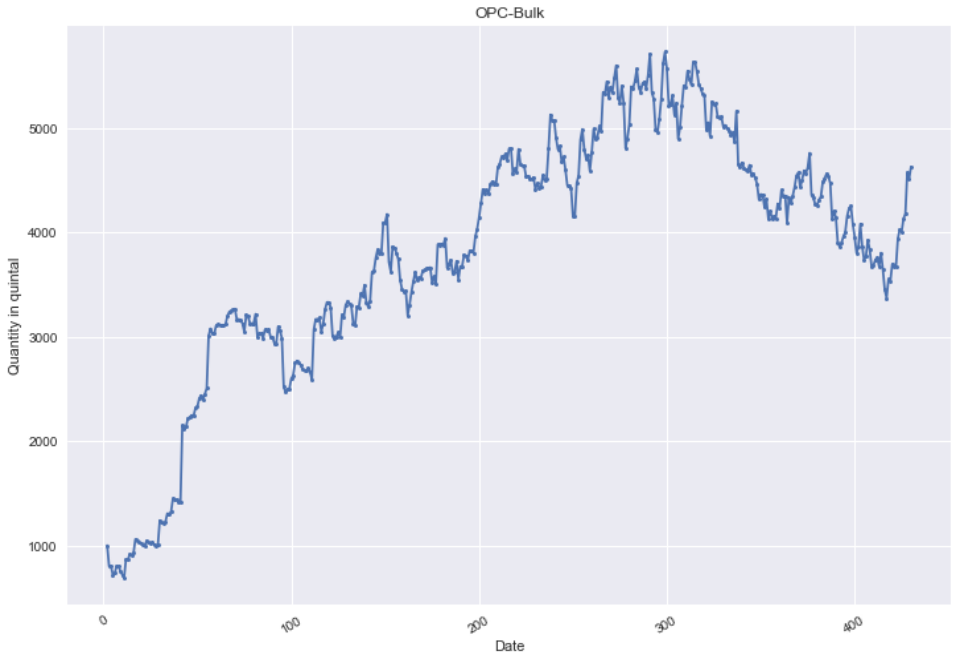
1. Before building our models, we did some basic feature engineering. We created a matrix of lagged values out of the time series using a window of a specific length. Say the window length or look back value is 1. So in our prediction models, the last column would serve as the labels and the rest of the columns as the predictors.
2. Then we splitted the data as Train and Test. Training data = 80% and Test data = 20%
3. Gave the train and test data for the fit and predict function in the LSTM class.
4. Made a prediction, and tested the prediction in two ways. First we tested the prediction with already trained data, on the second one we gave the test data for the model so that it can make a prediction.
5. We calculated the Mean Squared Error

4. Result

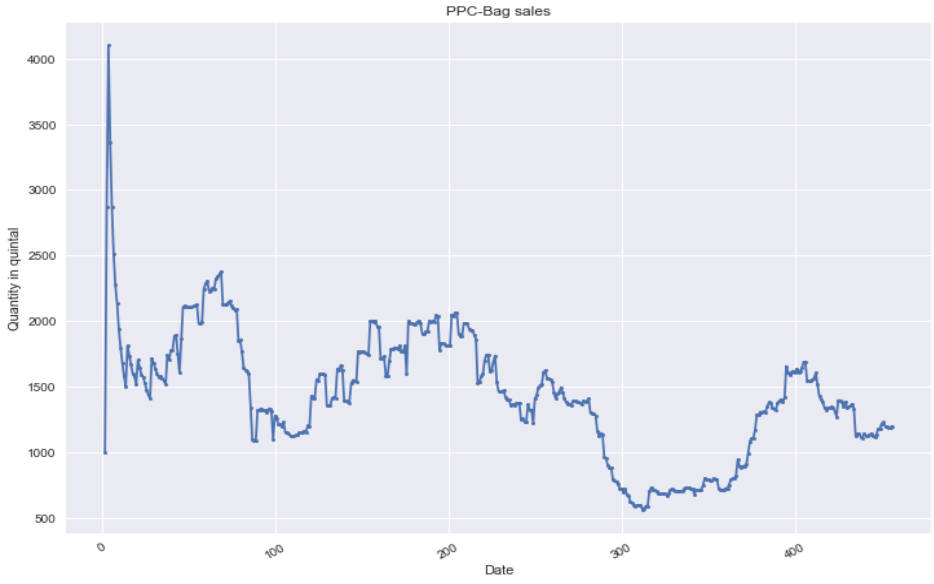
OPC-Bag original data time series visualization



OPC-Bulk original data time series visualization



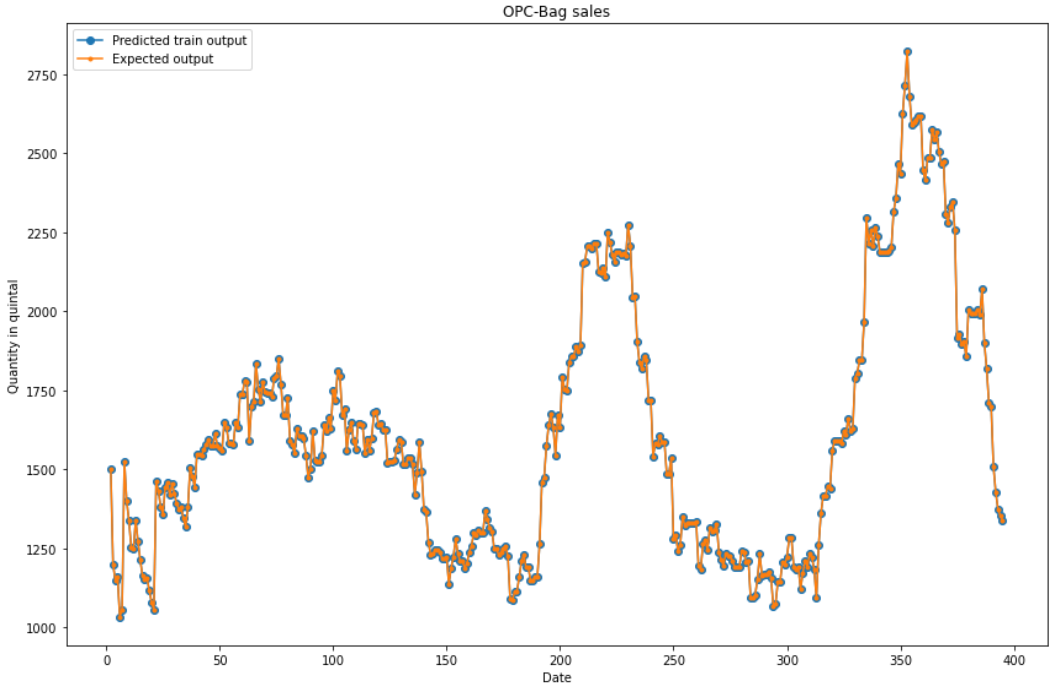
PPC-Bag original data time series visualization



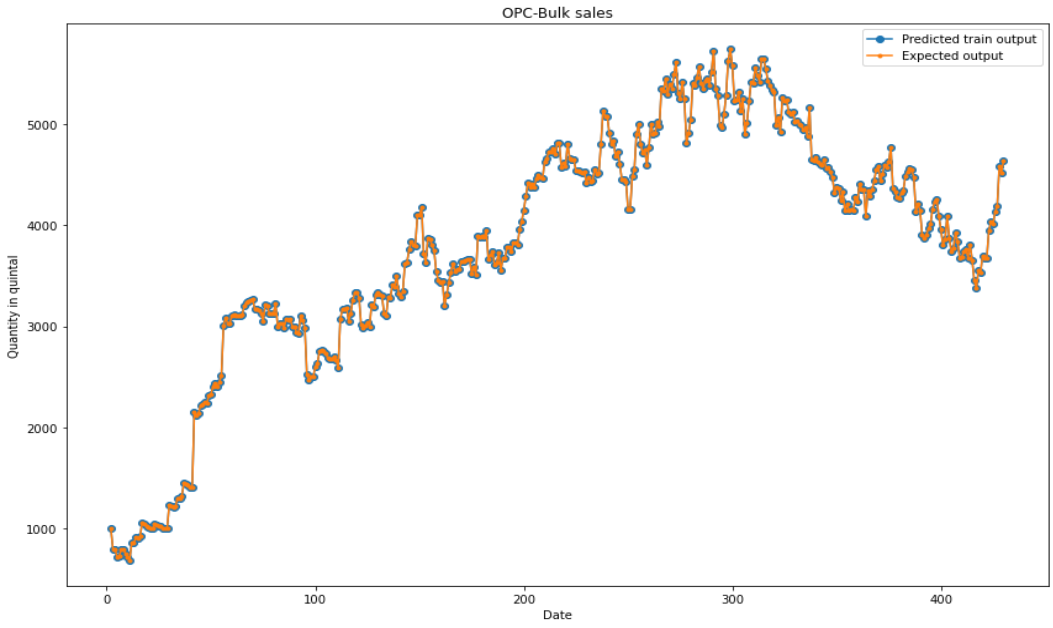
4.1. RBF

The following diagrams illustrate the prediction result for the respective cement types as performed by the Radial basis function.

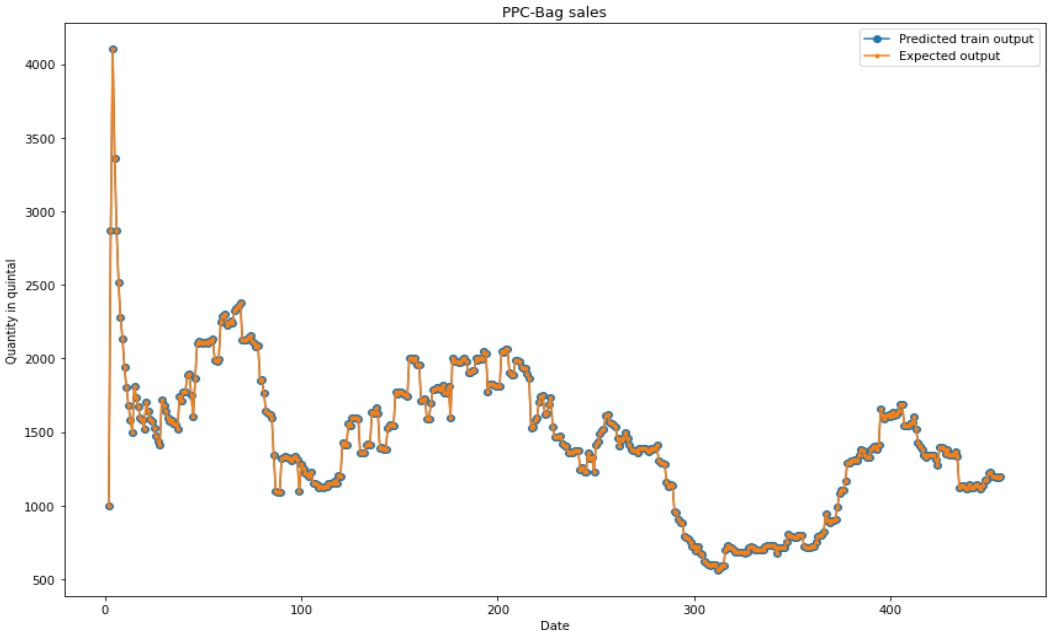
OPC-Bag



OPC-Bulk



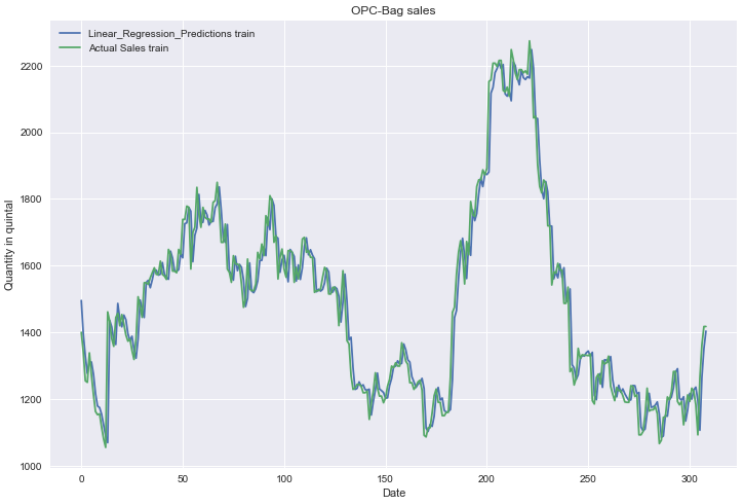
PPC-Bag

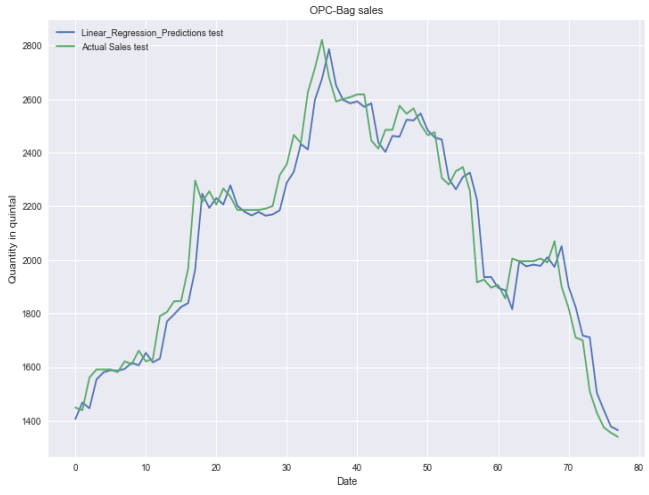


4.2. Multivariate Regression

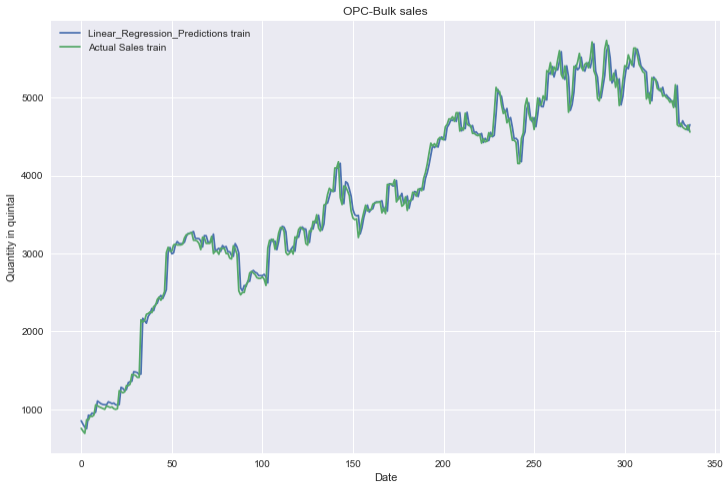
The following diagrams illustrate the prediction result for the respective cement types as performed by Multivariate Regression.

OPC- Bag

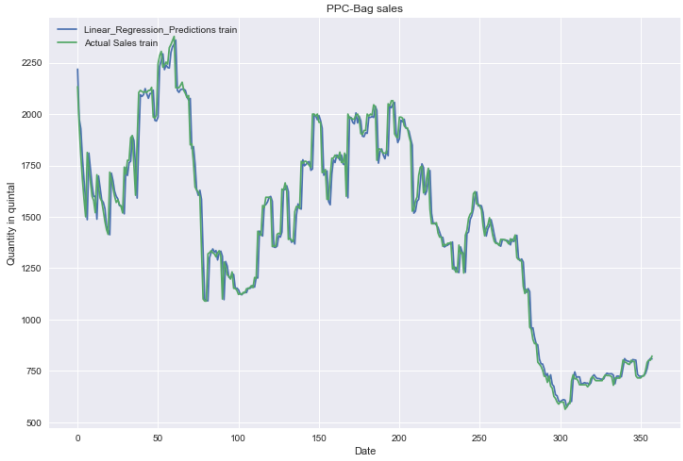


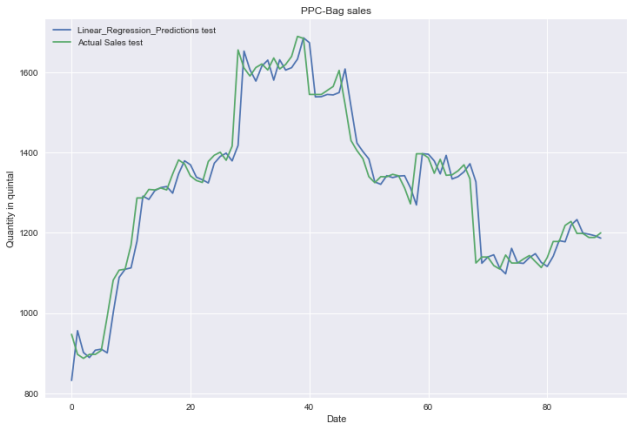


OPC-Bulk



PPC - Bag

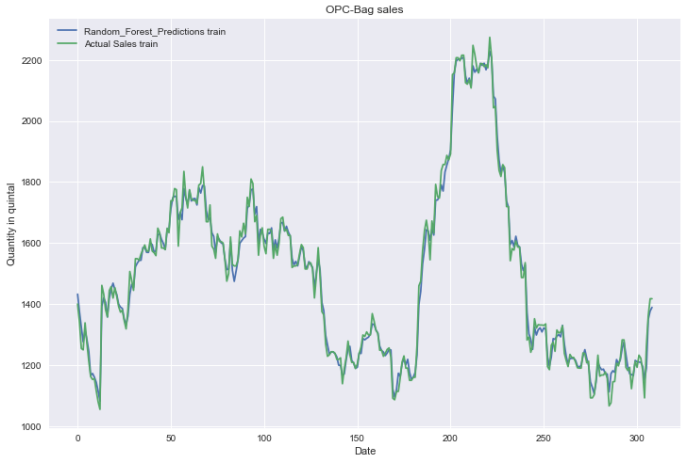


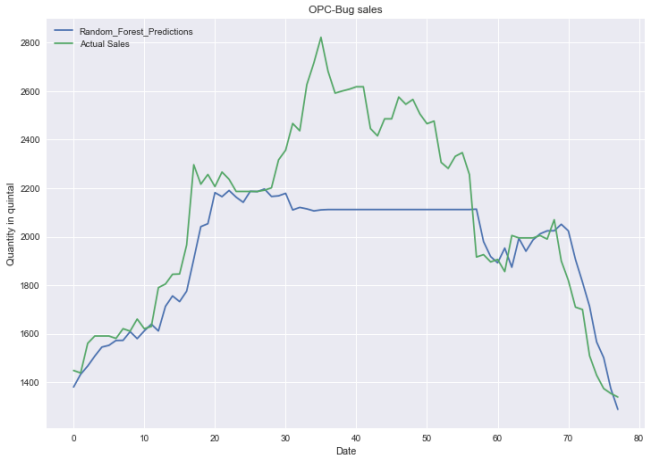


4.3. Random Forest

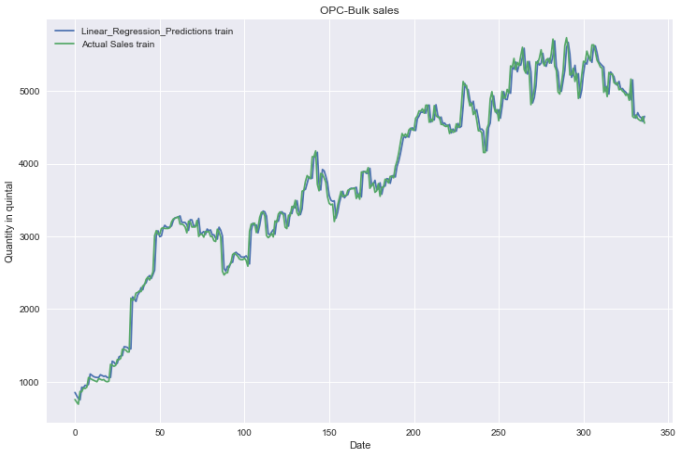
The following diagrams illustrate the prediction result for the respective cement types as performed by Random Forest Regressor.

OPC-Bag

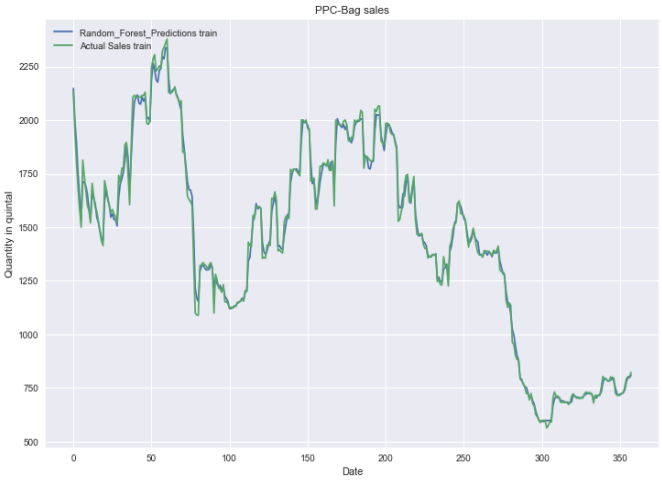


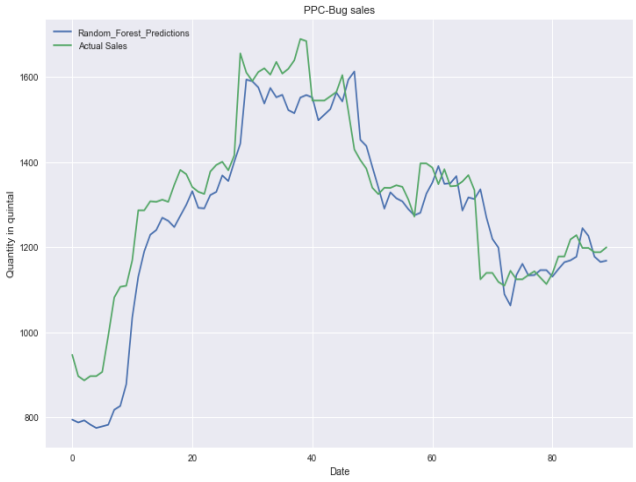


OPC-Bulk



PPC-Bag

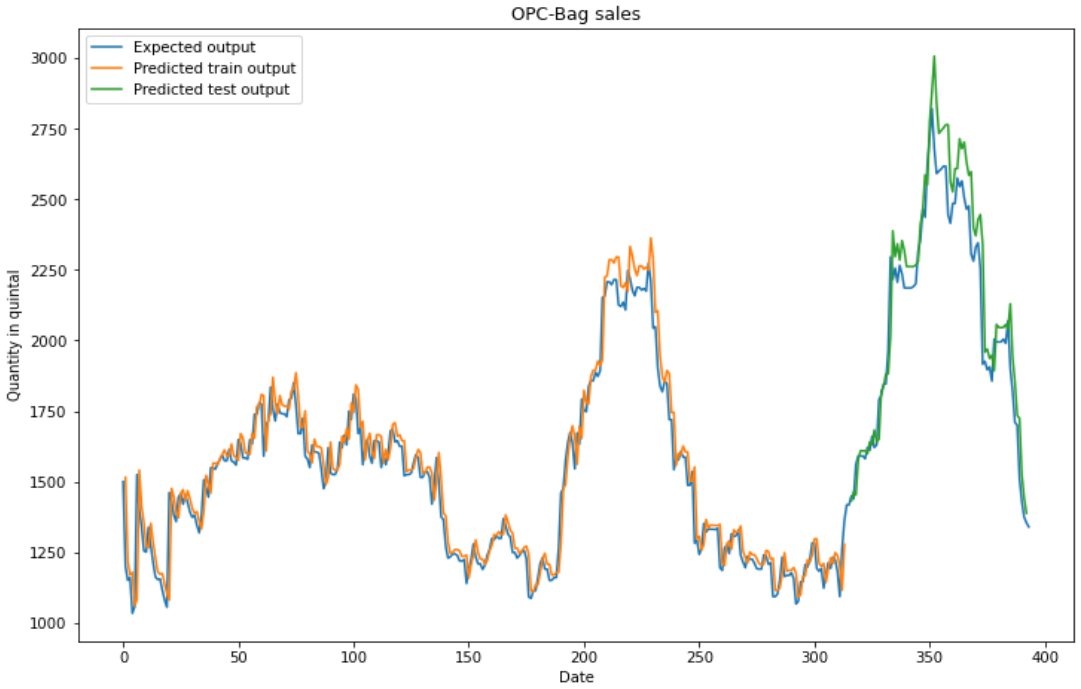




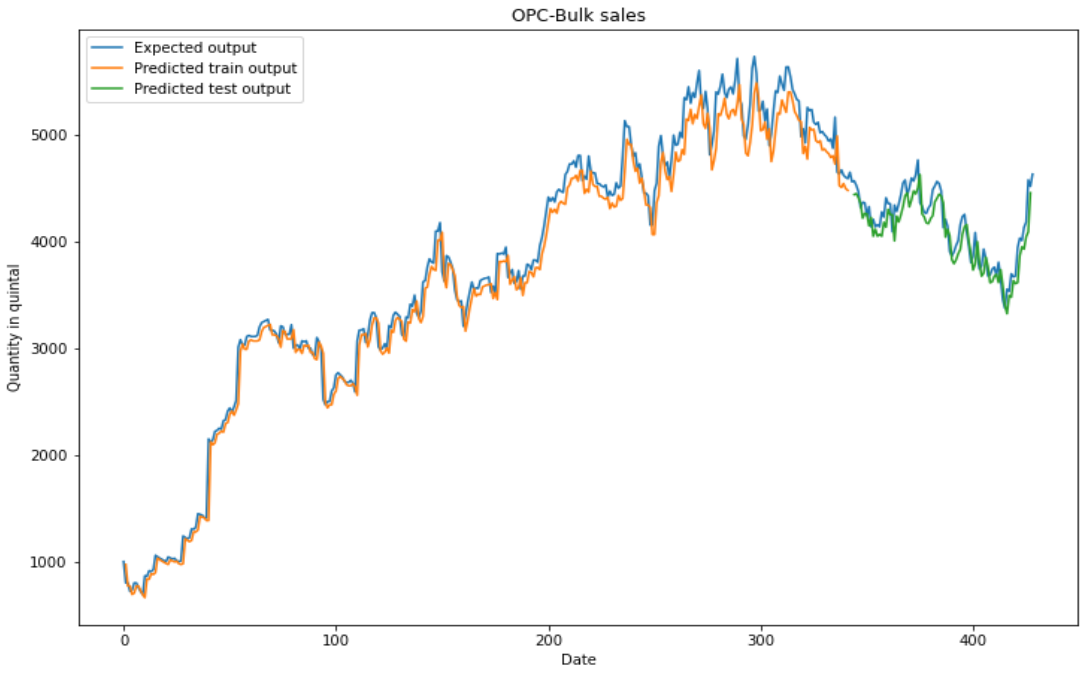
4.4. LSTM

The following diagrams illustrate the prediction result for the respective cement types as performed by Long short term memory.

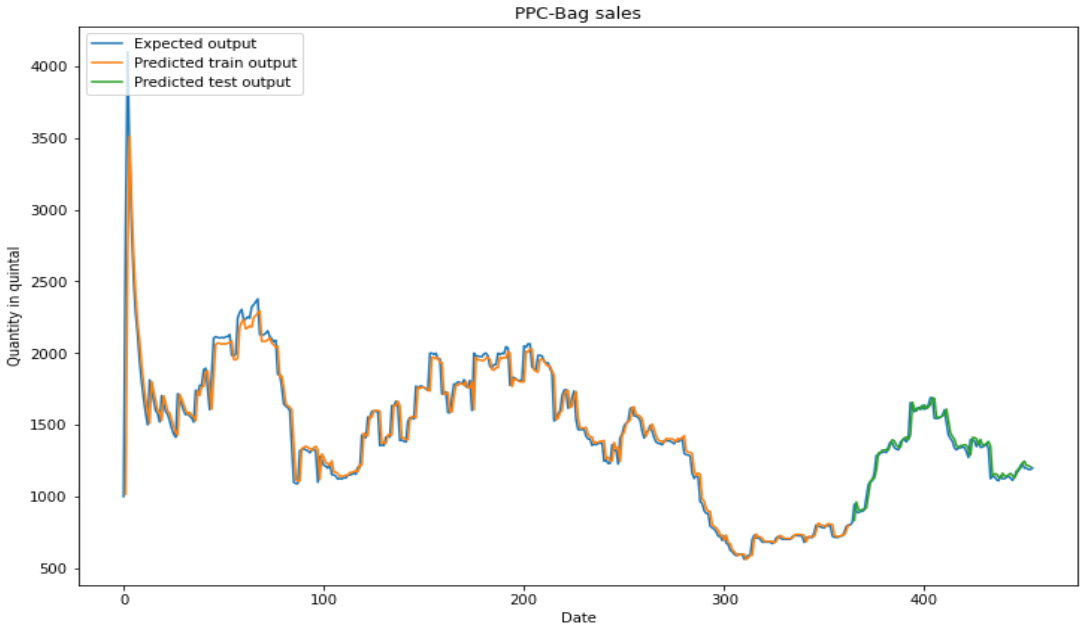
OPC-Bag



OPC-Bulk



PPC-Bag



4.5. Performance Comparison

To evaluate the effectiveness of the models, a comparison is made between the three techniques on cement types namely, OPC-Bag, OPC-Bulk, and PPC-Bag using both Multivariate regression , Random forest regressor and Long short term memory models. Predicted closing stock level quantity values are subjected to Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Squared Error (MSE) for finding the final minimum errors in the predicted price.

RMSE is computed using eq. 1.

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

Where ‘Oi’ refers to the original closing stock level quantity value, ‘Fi’ refers to the predicted closing stock level quantity value and ‘n’ refers to the total window size.

MSE, take the observed value, subtract the predicted value, and square that difference. It is computed using eq. 2.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Where 'Yi' refers to the original closing price, 'Y hat i' refers to the predicted closing price and 'n' refers to the total window size.

MAE, will find all of your absolute errors, $x_i - x$. Add them up, then divide by the number of errors. It is computed using eq. 3.

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n}$$

The following table shows the results clearly:

Cement Type	**Random forest regressor**			**Multivariate regression**			**Long short term memory**		
	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE
OPC Bag	487.47	237626.65	402.70	91.08	8295.38	62.66	130.09	16922.67	99.41
OPC Bulk	466.64	217755.78	378.03	132.35	7516.37	99.84	157.34	24756.31	131.53
PPC Bag	299.37	89628.27	236.19	51.77	2679.83	32.09	52.40	2745.56	34.10

5. Conclusion

Predicting stock demand is a challenging task due to the inconsistency of stock levels which are dependent on multiple parameters. This work consists of three parts which are data extraction and pre-processing of the dataset, carrying out feature engineering, and stock level trend prediction. Four algorithms were implemented for the prediction. These are the Radial Basis Function Network, Long Short Term Memory, Random Forest Regressor and Multivariate Regression. Different concepts were implemented throughout the process in order to obtain a better performance and higher accuracy. Based on our implementation the comparative analysis in RMSE, MSE and MAE values indicates that Multivariate regression has a better stock level prediction as compared to the other models implemented. Results show that the best values obtained from Multivariate regression model has RMSE (91.08), MSE (8295.38) and MAE (62.66) values for OPC-Bag ,RMSE (132.35), MSE (7516.37) and MAE(99.84) values for OPC-Bulk and RMSE (51.77), MSE (2679.83) and MAE(32.09) values for PPC-Bag .

Reference

[1]https://www.researchgate.net/figure/Architecture-of-RBF-network_fig6_273610577

[2] https://www.csie.ntu.edu.tw/~yien/papers/tnn0485.pdf

[3] https://ieeexplore.ieee.org/document/8126078

[4]https://www.researchgate.net/publication/311755721_Stock_market_prediction_using_machine_learning_techniques

[5] https://medium.com/@srv96/smoothing-techniques-for-time-series-data-91cccfd008a2 and https://www.solver.com/smoothing-techniques

[6]https://towardsdatascience.com/master-machine-learning-random-forest-from-scratch-with-python-3efdd51b6d7a

[7]https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/

[8]“Multivariate Linear Regression and Machine Learning” (by rupka nimbalkar)