K-Means Clustering and RNN for Predicting Stock Prices

Thomas Kahng[#1], Sam Abraham[*2], Hector Lopez[#3], Gehrig Wilcox[#4]

[#]*Engineering and Computer Science (ECS), University of Texas at Dallas*
800 W Campbell Rd, Richardson, TX 75080, United States

[1]ttk190003@utdallas.edu

[2]sca190002@utdallas.edu

[3]hxl190015@utdallas.edu

[4]gaw190001@utdallas.edu

***Abstract*— We propose utilizing RNNs and KNNs to evaluate the rolling averages of the open prices of the APPL stock, along with predicting the next day's open price from open prices of the last documented week. We show that our algorithm produces the number of open prices nearest to each of our k rolling averages, and reaches up to a 79.57% accuracy at predicting the next day's open price for the APPL stock.**

***Keywords*— Machine Learning, RNN, KNN, Stock Prediction, open Price**

I. Introduction and Background Work

The stock market is a way for investors to give money to companies in hopes to make a future profit. Usually investors go through a lengthy and time consuming process of analyzing multiple factors of the stock in order to come up with a personal valuation of that stock. This valuation can quite often be very bad and thus be non profitable.

Since the popular consensus of valuing a stock is that it is some function of parameters, we should be able to design an algorithm to approximate this function.

II. Theoretical and Conceptual Study of Algorithm

A. Recurrent Neural Network (RNN)

The first property about stocks we noted was that they lie within a time domain. We believe that it would be beneficial to provide our algorithm a way to consider the previous prices of the stock. This is why we decided to utilize a Recurrent Neural Network (RNN).

Another property we noted was that stock prices can have various rolling averages. Therefore, we believed that it would benefit us to establish a certain number of rolling averages, and get the count of stock prices nearest to one of however many averages we've decided to keep track of. This led us to utilize the K-Means Clustering or the K-Nearest neighbors (KNN) algorithm.

A Recurrent Neural Network (RNN), is a powerful and robust type of neural network in which the output of the previous step/layer is the input for the next step/layer, for as many layers and hidden layers that exist in the network [5]. This is different from traditional neural networks where inputs and outputs are independent of each other. The idea behind them was created in the 1980's and they have continually found use in problems that deal with sequences [5]. With the invention of Recurrent Neural Networks combined with Long Short Term Memory (LSTM) which was invented in the 1990's, Recurrent Neural Networks have become increasingly useful [5]. In traditional feed-forward neural networks, information moves in one direction, from beginning to end. They can predict the next step based only on what is currently there. There is no consideration about what was there in the computation of what will be there. In other words, feed forward neural networks do not have the ability to compute the derivative of a time series, while a Recurrent Neural Network does.

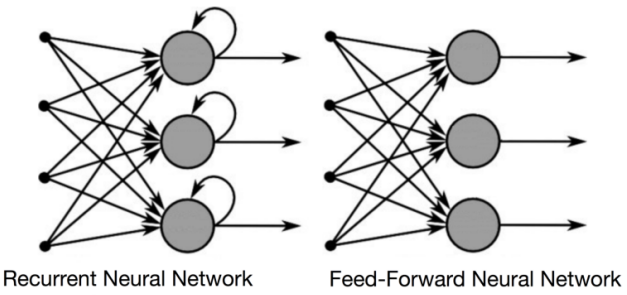

Fig 1. A visual graph representation of the information flow within Recurrent Neural Networks and Feed-Forward Neural Networks

Recurrent Neural Networks can also map inputs to outputs in a one to one, one to many, many to one, or many to many way, making them far more useful compared to Feed Forward Neural Networks [5].

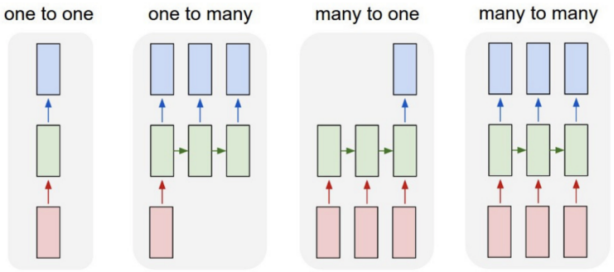

Fig 2. A visual representation of the possible input to output mappings a Recurrent Neural Network can produce.

Although this is a clever solution, it does increase the complexity of the algorithm, particularly with backpropagation [5]. Since we are feeding the output back into the network, we have to in some way adjust the previous network to fully train the algorithm.

Like mentioned earlier in the section, there is also an additional tool that can be utilized with the Recurrent Neural Network called the Long Short-Term Memory which allows the network to hold on to information for a longer period of time [5]. In a way, the Long Short-Term Memory can be thought of like a gate, or a latch circuit. Information can be placed into the gate and will remain there until the network decides it is no longer necessary in which case, it will open the gate and lose the information [5].

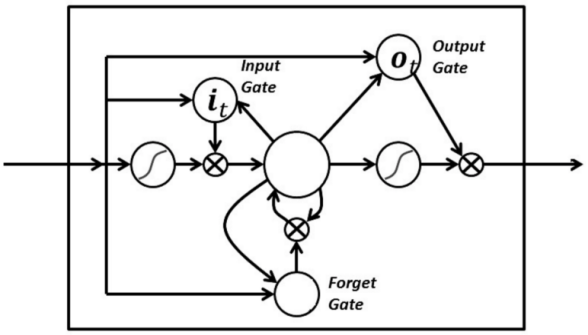

Fig 3. A visual representation of the Long Short-Term Memory network.

.

B. K-Means Clustering or K Nearest Neighbors (KNN)

K-means clustering is a popular unsupervised machine learning algorithm. Unsupervised algorithms make interpretations from data by only using input vectors without referring to known results [1]. K-means is meant to group data points that are similar together and identify any patterns. To achieve this, the algorithm searched for (k) clusters in a dataset [1]. A cluster is a collection of data points that are grouped together because of similarities [1]. K is a pre-defined number, which refers to the number of center points or centroids of the cluster in the given data [1]. Every data point is assigned to a cluster by reducing the sum of squares within each cluster [1]. To process the data, the algorithm starts by randomly assigning data points to centroids [1]. It then performs iterative estimations to optimize the centroids [1]. It stops optimizing clusters when the centroids are stabilized and the values are no longer changing [1]. Or when the defined number of iterations have been achieved.

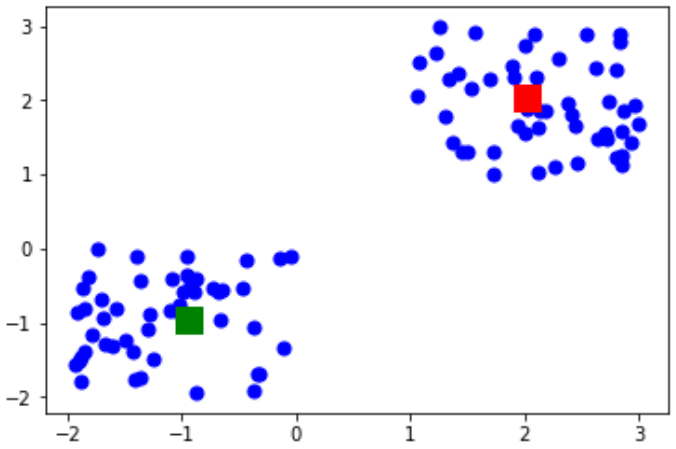

Fig 4. A visual representation of points closest to each centroids (red and green squares are centroids)

III. Results and Analysis

Throughout our experiment, we've kept track of the last recorded week's open prices to predict tomorrow's open price. Also, we've kept track of any k number of centroids for K-Means Clustering on our predicted and actual open prices.

A. Parameters of Our Trials

To evaluate our experiments over the course of five trials, we've kept track of our parameters and results from training a two-layer Recurrent Neural Network (RNN). Throughout all trials, we've consistently used the Tanh activation function and the Mean Squared Error (MSE) error function. However, as seen in the table down below, we've used different values for k for our KNN, the and units, epochs, batch size, and verbose parameters for our RNN

TABLE I
Parameters chosen for each trial run

Trial	**Parameters**
1	RNN: • Number of Layers = 2 • Units = 50 • Activation Function = tanh • Dropout Parameter = 0.2 • Dense Parameter = 1 • Error Function = MSE • Optimizer = adam • Epochs = 50 • Batch Size = 32 • Verbose = 2 K-Means Clustering: • K = 5
2	RNN: • Number of Layers = 2 • Units = 60 • Activation Function = tanh • Dropout Parameter = 0.2 • Dense Parameter = 1 • Error Function = MSE • Optimizer = adam • Epochs = 60 • Batch Size = 35 • Verbose = 3 K-Means Clustering: • K = 7
3	RNN: • Number of Layers = 2 • Units = 70 • Activation Function = tanh • Dropout Parameter = 0.2 • Dense Parameter = 1 • Error Function = MSE • Optimizer = adam • Epochs = 70 • Batch Size = 40 • Verbose = 4 K-Means Clustering: • K = 9
4	RNN: • Number of Layers = 2 • Units = 80 • Activation Function = tanh • Dropout Parameter = 0.2 • Dense Parameter = 1 • Error Function = MSE • Optimizer = adam • Epochs = 80 • Batch Size = 45 • Verbose = 5 K-Means Clustering: • K = 10
5	RNN: • Number of Layers = 2 • Units = 90 • Activation Function = tanh • Dropout Parameter = 0.2 • Dense Parameter = 1 • Error Function = MSE • Optimizer = adam • Epochs = 90 • Batch Size = 50 • Verbose = 6 K-Means Clustering: • K = 12

As shown in TABLE I, throughout our five trials, we've used values of 5, 7, 9, 10, and 12 for k, 50, 60, 70, 80, and 90 for units, 32, 35, 40, 45, and 50 for batch size, and 2, 3, 4, 5, and 6 for verbose. This helped us partake in feature engineering in an attempt to get results with highest possible accuracy and lowest possible error.

B. Results of Our Trials

To evaluate our parameters on our Recurrent Neural Network (RNN), we've tested accuracy and Root Mean Square Error (RMSE) for both our training and test datasets. We've also calculated the k number of centroids and number of open prices near each centroid.

TABLE II
Results for each trial run

Trial	**Results**
1	Statistics: • Train/Test Split = 80:20 • Size of Dataset = 251 • Training Accuracy = 91.41% • Test Accuracy = 74.59% • Training RMSE = 1.72 • Test RMSE = 2.56 • Tomorrow's open Price = $176.52 Predicted Centroids (Value: Count): • Near $150.32 (predicted): 11 open prices • Near $152.54 (predicted): 12 open prices • Near $158.22 (predicted): 9 open prices • Near $147.59 (predicted): 8 open prices • Near $143.61 (predicted): 5 open prices Actual Centroids (Value: Count): • Near $149.46 (actual): 6 open prices • Near $146.83 (actual): 11 open prices • Near $159.94 (actual): 10 open prices • Near $153.79 (actual): 11 open prices • Near $151.28 (actual): 7 open prices • • Near $151.28 (actual): 7 open prices
2	Statistics: • Train/Test Split = 80:20

	• Size of Dataset = 251 • Training Accuracy = 90.40% • Test Accuracy = 78.99% • Training RMSE = 1.81 • Test RMSE = 2.33 • Tomorrow's open Price = $183.10 Predicted Centroids (Value: Count): • Near $150.95 (predicted): 6 open prices • Near $153.78 (predicted): 11 open prices • Near $148.11 (predicted): 7 open prices • Near $143.90 (predicted): 5 open prices • Near $159.50 (predicted): 9 open prices • Near $149.13 (predicted): 1 open prices • Near $151.84 (predicted): 6 open prices Actual Centroids (Value: Count): • Near $152.57 (actual): 6 open prices • Near $153.70 (actual): 6 open prices • Near $156.08 (actual): 4 open prices • Near $159.94 (actual): 8 open prices • Near $147.81 (actual): 9 open prices • Near $143.97 (actual): 4 open prices • Near $150.64 (actual): 8 open prices
3	Statistics: • Train/Test Split = 80:20 • Size of Dataset = 251 • Training Accuracy = 91.03% • Test Accuracy = 79.57% • Training RMSE = 1.75 • Test RMSE = 2.29 • Tomorrow's open Price = $177.31 Predicted Centroids (Value: Count): • Near $148.18 (predicted): 2 open prices • Near $147.63 (predicted): 1 open prices • Near $151.88 (predicted): 5 open prices • Near $151.07 (predicted): 3 open prices • Near $147.88 (predicted): 3 open prices • Near $144.02 (predicted): 5 open prices • Near $158.04 (predicted): 7 open prices • Near $160.43 (predicted): 2 open prices • Near $150.36 (predicted): 3 open prices • Near $149.84 (predicted): 3 open prices • Near $147.22 (predicted): 2 open prices • Near $153.23 (predicted): 9 open prices Actual Centroids (Value: Count): • Near $147.71 (actual): 2 open prices • Near $148.03 (actual): 2 open prices • Near $147.05 (actual): 3 open prices • Near $153.70 (actual): 6 open prices • Near $152.81 (actual): 3 open prices • Near $152.16 (actual): 3 open prices • Near $156.08 (actual): 4 open prices • Near $159.94 (actual): 8 open prices • Near $148.90 (actual): 3 open prices • Near $143.97 (actual): 4 open prices • Near $150.21 (actual): 4 open prices • Near $151.19 (actual): 3 open prices
4	Statistics: • Train/Test Split = 80:20 • Size of Dataset = 251 • Training Accuracy = 89.82% • Test Accuracy = 69.27% • Training RMSE = 1.86 • Test RMSE = 2.81 • Tomorrow's open Price = $178.58 Predicted Centroids (Value: Count): • Near $148.36 (predicted): 3 open prices • Near $159.52 (predicted): 2 open prices • Near $150.65 (predicted): 7 open prices • Near $147.02 (predicted): 4 open prices • Near $146.72 (predicted): 3 open prices • Near $152.10 (predicted): 8 open prices • Near $154.37 (predicted): 3 open prices • Near $149.46 (predicted): 4 open prices • Near $143.37 (predicted): 5 open prices • Near $156.90 (predicted): 6 open prices Actual Centroids (Value: Count): • Near $148.04 (actual): 6 open prices • Near $158.86 (actual): 5 open prices • Near $150.09 (actual): 5 open prices • Near $147.05 (actual): 3 open prices • Near $143.97 (actual): 4 open prices • Near $159.94 (actual): 1 open prices • Near $156.08 (actual): 3 open prices • Near $162.44 (actual): 3 open prices • Near $153.11 (actual): 10 open prices • Near $151.28 (actual): 5 open prices
5	Statistics: • Train/Test Split = 80:20 • Size of Dataset = 251 • Training Accuracy = 91.98% • Test Accuracy = 77.22% • Training RMSE = 1.66 • Test RMSE = 2.42 • Tomorrow's open Price = $181.64 Predicted Centroids (Value: Count): • Near $150.26 (predicted): 4 open prices • Near $152.55 (predicted): 3 open prices • Near $149.36 (predicted): 2 open prices • Near $152.15 (predicted): 2 open prices • Near $151.38 (predicted): 5 open prices • Near $147.80 (predicted): 2 open prices • Near $151.85 (predicted): 1 open prices • Near $153.31 (predicted): 6 open prices • Near $143.94 (predicted): 5 open prices • Near $148.46 (predicted): 2 open prices • Near $158.70 (predicted): 9 open prices • Near $147.36 (predicted): 4 open prices Actual Centroids (Value: Count): • Near $152.35 (actual): 4 open prices • Near $153.88 (actual): 7 open prices • Near $143.97 (actual): 4 open prices • Near $148.04 (actual): 6 open prices • Near $151.28 (actual): 2 open prices • Near $150.64 (actual): 2 open prices • Near $153.11 (actual): 2 open prices • Near $162.44 (actual): 3 open prices • Near $157.32 (actual): 3 open prices • Near $147.05 (actual): 3 open prices • Near $159.30 (actual): 5 open prices • Near $150.09 (actual): 4 open prices

Throughout our trials, we've evaluated our training accuracies to be 91.41%, 90.40%, 91.03%, 89.82%, and 91.98% and our test accuracies to be 74.59%, 78.99%, 79.57%, 69.27%, and 77.22%. Our RMSE values were evaluated to be 1.72, 1.81, 1.75, 1.86, and 1.66 for training data, and 2.56, 2.33, 2.29, 2.81, and 2.42 for our testing data.

Through five trials from feature engineering, we've evaluated that trial 5 has the greatest training accuracy of 91.98% and the lowest training RMSE of 1.66. This concludes that trial 5's data has best fit and trained our RNN mode.

However, we've evaluated that trial 3 had the greatest test accuracy and lowest RMSE for test data. Therefore, we decided to focus on trial 3's prediction, as it had the most accurate test data. Although training accuracy and RMSE is important, our objective is to find the most accurate stock prediction, which is derived from our result, so we've focused on more accurate test data.

C. Graphical Representations of Our Best Trial

After evaluating all trials, we've found that trial 3's predicted open price would be the most accurate. This was mainly due to finding the highest test accuracy and the lowest test RMSE, both of which were critical in determining the accuracy of test data. Here we prioritized test data as it was critical in determining a final factor, tomorrow's open price predicted from the dataset's last recorded week's open prices.

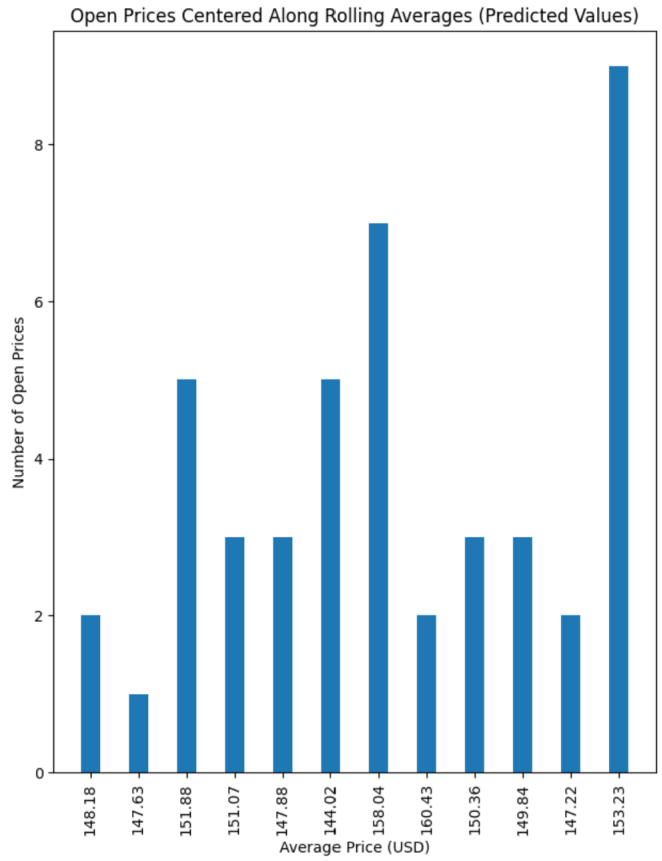

Fig 1. 9 open prices (centroids), for predicted open prices, along with the number of open prices closest to any given centroids

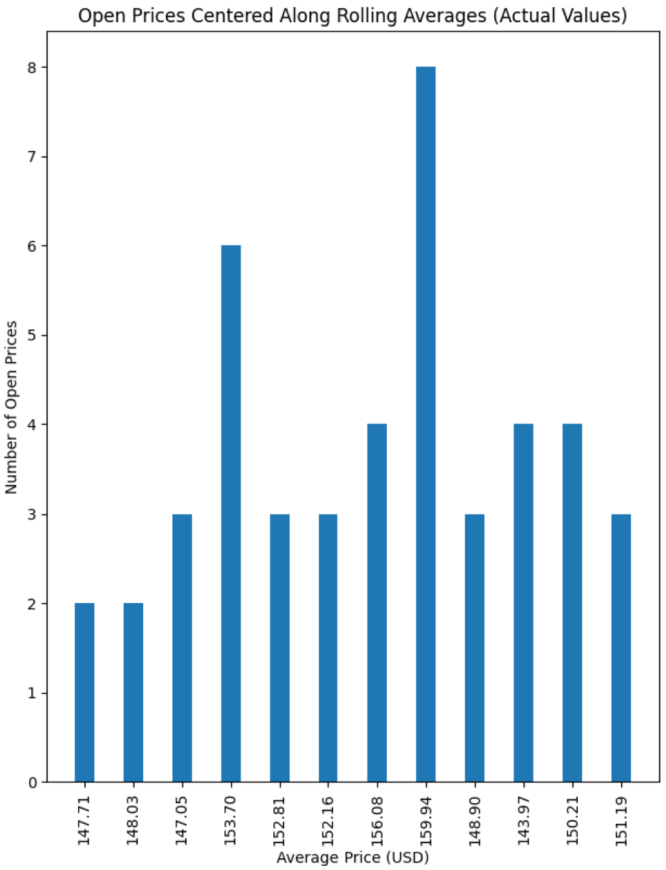

Fig 2. 9 open prices (centroids), for actual open prices, along with the number of open prices closest to any given centroids

Figure 1 showed the rolling average open prices (centroids), for predicted open prices. Here, a k value of 9 was used for K-Means Clustering, and 9 centroids were recorded since our algorithm last converged. Since 9 open prices have been randomly selected, in the beginning, the end results will always vary due to reassignment. Since each non centroid is assigned to a cluster associated with its closest centroid, and each cluster's centroid is updated to the point closest to the centroid, reassignment is based on previous centroids and associated data. This happens until we have an iteration without a change in cluster values, so results will vary based on our data and initially selected clusters.

Similarly, figure 2 shows the rolling average open prices (centroids), for actual open prices. Since we've implemented the same K-Means Clustering algorithm, our final centroids and number of open prices closest to that centroids will vary from that of our predicted values. This is because we've used different data and probably started with different random centroids.

We implemented K-Means Clustering in our stock prediction algorithm to show different averages for our open price of our predicted and actual values. From this, we've evaluated the open price with most other open prices associated with and/or closest to it to be 9 open prices closest to

$153.23 for our training data and 8 open prices closest to $159.94 for our test data.

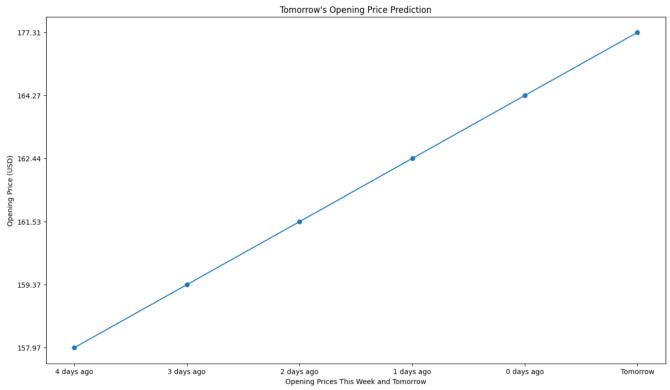

Fig 3. open prices of the last 5 days found on our dataset, followed by tomorrow's predicted open price

Fig 3 showed our predicted open price (tomorrow's open price), and the open prices of the last week shown on our original dataset. From Table 2 of previous subsection B (Results of Our Trials), we've evaluated our predicted open price to be $177.31, and this is shown in the final point of Fig 3 as well.

Fig 3 shows a positive linear correlation among open prices from the last week up to the next day following the last week. Therefore, we can conclude that the AAPL stock may be increasing in value, based on the most recently recorded data.

IV. Conclusion and Future work

After all, we've trained our RNN to predict the next day's opening price, based on the last recorded week's prices. Upon training our RNN, we utilized K-Means Clustering to get multiple average open prices and number of open prices associated with each one, and finally predicted our next day's opening price shortly after.

We focused on our third trial mainly due to accurate test data, which was crucial in determining our prediction. Aside from this fact, we've had relatively well performing training accuracy and training RMSE, which helped in training our RNN.

In the future, we plan on predicting high, low, close, and adjusted close prices in a similar fashion, as a short term goal. Furthermore, we plan on applying this prediction on different stocks, and predict prices over a greater amount of days.

V. References

[1] E. E. (LEDU), "Understanding K-means clustering in machine learning," *Medium*, 12-Sep-2018. [Online]. Available: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1. [Accessed: 29-Apr-2023].

[2] S. Mulani, "RMSE - root mean square error in Python," *AskPython*, 16-Feb-2023. [Online]. Available: https://www.askpython.com/python/examples/rmse-root-mean-square-error. [Accessed: 29-Apr-2023].

[3] O. Ozturk, "Stock price prediction by simple RNN and LSTM | Kaggle," Stock Price prediction by simple RNN and LSTM, 2019. [Online]. Available: https://www.kaggle.com/code/ozkanozturk/stock-price-prediction-by-simple-rnn-and-lstm. [Accessed: 30-Apr-2023].

[4] "Sklearn.metrics.r2_score," *scikit*, 2007. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html. [Accessed: 29-Apr-2023].

[5] B. Whitfield, "A guide to recurrent neural networks: Understanding RNN and LSTM Networks," *Built In*, 28-Feb-2823. [Online]. Available: https://builtin.com/data-science/recurrent-neural-networks-and-lstm. [Accessed: 29-Apr-2023].