Comparison of Differing Machine Learning Algorithms dealing with Stock Market Prediction

Abhishek Parekh and Stephen O'Connor

Abstract
Stock market forecasting using machine learning techniques can aid in gaining insights for future price prediction for a particular stock or financial assets on a stock exchange. This paper will focus on the comparison and contrast between machine learning algorithms used for stock market forecasting, these include linear regression, Convolutional neutral network, and Long Short-term Memory. The experimentation will help illustrate the difference in performance and outline their respective architecture. The models are trained for predicting stock prices, the predicted prices are matched with the actual price of the stock, to see how well the prices were predicted. The stocks used in this experiment include Ford, General Motors, Honda, Navistar, PACCAR, Toyota, and Tesla. Additionally, both the models were trained using the same datasets from Kaggle called *Huge Stock Market Dataset*, it includes well-known companies trading on the New York Stock Exchange. The results from the experiment showcased that the LSTM model outperformed the CNN model, such as for Tesla the improvement in price prediction was by 50%, the CNN model achieved a maximum difference of 73.765 for Tesla, whereas the LSTM model achieved a maximum difference of 37.227. As demonstrated by the results reported in this study, the LSTM model is an excellent choice for stock market forecasting, owing to its ability to extract the predictions from data delivered in a sequential manner that reflects long-term interdependence and temporal patterns. The memory cells in the LSTM model allow it to hold and update previous information; the model can forecast the price based on past information, and it will maintain the knowledge while updating it over a lengthy period. LSTMs, on the other hand, are the best option for handling time series issues since they have the capacity to learn complicated correlations within the data and because stock market data will surely contain a range of fluctuations.

***Index Terms*—Stock Market Prediction, Machine Learning, Deep Learning, Neutral Network, Linear Regression, Recurrent Neural Networks, Long Short-Term Memory, Split Ratio**

I. Introduction

Investing and trading Stocks has been around since the late 18th Century and in the late 20th and early 21st century computer science has been used to predict the value of stocks. A Stock is a security that translates to a fraction of ownership of a public corporation and their assets. These securities can be traded on different exchanges around the world with one of the more popular exchanges being the New York Stock Exchange (NYSE). n more recent years the use of different machine learning and artificial intelligence methods has been used to predict the rise and fall of stock evaluations. Some popular types of machine learning used to predict stocks include Linear Regression, Convolutional Neutral Networks, and Long Short-Term Memory (LSTM) Model. [1]

The initial idea was to use a Linear Regression approach to the Stock Prediction problem. To summarize, a Linear Regression model will train on a series of available input data for a set company. Then the same type of data from different companies will be used to predict their future performance. A Linear Regression Model assigns weights to a set of input variables that are used to calculate a single negative or positive evaluation. Also, the drawback of linear regression is that all evaluations require the same input parameters for training and testing data. After reviewing papers detailing the performance of linear regression model versus more Morden Neural Networks the idea was removed as a candidate. [4]

Algorithm 1 is a Convolutional Neural Network (CNN) approach to the Stock Prediction problem. To summarize a CNN is a Deep Learning algorithm that accepts a series of input data for a company and defines weights and biases to aspects (objects and features) within the data; these items are used to predict the rise or fall of the company's stock. A CNN is broken into multiple parts including the Preprocessing, Convolution Layer, Pooling Layer, and the Classification Layer (or the Fully Connected Layer). [2] CNN will be created using Python in-conjunction with opensource software including but not limited to TensorFlow, and OpenCV.

For algorithm 2, Long Short-Term Memory (LSTM) will be implemented for stock market forecasting. LSTM is a type of

recurrent neural network, with an additional feature that allows it to retain crucial information from the previous output and omit anything else that is irrelevant. A simple LSTM network consists of three gates, these include the input gate, the output gate, and the forget gate [5]. The input gate is used to set up weights and variables that need to be inserted into the algorithm, the output gate allows the end results to be calculated depending on the problem at hand, whereas the forget gate allows the network to decide if a piece of information should retain or not [6].

II. Previous Works

A. Stock Price Prediction Using CNN and LSTM Based Deep Learning Models

The Team of two (Sidra Mehtab and Jaydip Sen) out of Praxis Business School in India designed and implemented a five different deep learning models to predict the NIFTY 50 index in the National Stock Exchange (NSE). The five different models were based on two types of neural network models (CNN and a Long-and-short-term memory (LSTM)). For their experiment they trained and tested using data from December 2008 to July 2020. Each of the models were fed data in weekly increments and each time the model was updated the models were tested on the next week's data for varying stocks. The 5 different models were described as: Univariate CNN Model with one week of data, Univariate CNN Model with two weeks of data, Univariate Encoder-decoder LSTM with two weeks of data, Univariate Encoder-decoder CNN LSTM with two weeks of data, and Univariate Encoder-decoder conversation CNN LSTM Model. The team found that all 5 deep learning models were able to predict the next week's stock price more often than not. Overall, their most accurate model was the LSTM Model using the 2 previous weeks of data to predict the future weeks stock price. However, the CNN Model is trained and executed at a much higher rate. [7]

B. Stock Price Prediction using LSTM, RNN, and CNN-Sliding Window Model

The team out of Amrita University investigated the usage of deep learning architectures in stock prediction. The team used a sliding window approach for feeding data into their models. There were 3 separate models built for their investigation: LSTM, RNN and CNN. The models were training using daily End of Day prices for populate stock Infosys on an exchange within in India. After training the models on the previous day the models would be tested on the next day's end of day market price for different stocks including Infosys, TCS and Cipla. After the prediction, they would calculate the percentage error for each model then compare the results. In the end, the team's conclusion was that the CNN Model performed better in the end due to the CNN Model's ability to change/update based on the sudden changes within the stock market. The results from the study can be seen in figure 1 below. [8]

COMPANY	RNN	LSTM	CNN
Infosys	3.90	4.18	2.36
TCS	7.65	7.82	8.96
Cipla	3.83	3.94	3.63

Fig. 1. Results from the Study out of Amrita University

C. NSE Stock Market Prediction Using Deep-Learning Models

A computer engineering team out of Amrita School of Engineering created 4 different deep learning architectures for predicting the stock price of a company based on the historical prices available. The 4 different architectures include Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN). Like the other experiment out of Amrita University the team trained on the end of day stock market prices for 1 company from NSE of India then tested on 5 stocks off the New York Stock Exchange (NYSE) and the National Stock Exchange of India. The team concluded the CNN Model outperformed the other 3 models as the model was able to successfully predict stocks on both the NYSE and NSE as it is possible to handle sudden changes in a small window. This is a similar conclusion drawn in other studies. Lastly the team compared the results from Deep Learning Models to Linear Model and found the Deep Learning Models all performed at a better rate than the previously built linear models. [9]

D. EA-LSTM: Evolutionary attention-based LSTM for Time Series Prediction

The authors of this paper propose an algorithm called Evolutionary attention-based LSTM (EA-LSTM) for the purpose of tackling the issue of random search within a multivariate time series prediction since LSTMs (Long Short-Term Memory) are slightly insufficient for solving such problems. The authors used a competitive random search for training this model. Through the experimental results, it is highlighted that EA-LSTM achieves a better performance in multivariate time series prediction, while compared to other methods such as LSTM and RNN [10]. These findings prove that EA-LSTM captures the long-term dependencies in a time series prediction problem and successfully and it successfully utilizes the local data in a sample period as per the multiple distribution scales [10].

E. Transductive LSTM for Time-Series prediction: Weather Forecasting

In this paper, the authors defined and proposed an algorithm called Transductive LSTM (T-LSTM), this is done so that it can achieve better performance when predicting using local data points in a time-series prediction. Through the experiments, the team investigated mainly two scenarios based on cosine similarity amongst the training dataset and the training dataset,

in addition, to conduct an unbiased experiment, the team conducted the experiments on two separate time periods of a single year [11]. Moreover, in the end results, it was found that, after thorough experimentation on different lengths of time periods and weights, the results point out that T-LSTM performs much better than LSTM when tested for multiple scenarios [11]. Since applications such as weather forecasting require the highest level of accuracy, T-LSTM proves to be a contender for this particular use case, considering that it achieves exceptional performance on localized data points.

F. NGCU: A New RNN Model for Time-Series Data Prediction

The authors of this paper established and proposed an algorithm called New Gate Control Unit (NGCU), which is based on the RNN model. The purpose of NGCU is to diminish the issues regarding gradient disappearance and explosion when using RNN. Through experimentation, the authors found that, when compared to LSTM and Gate Recurrent Network (GRU), the NGCU model reduces the computational time and resources, in addition, it improves the sensitivity of the model, in other words, it increases the true positive rate (TPR) compared to the rate of false positives [12].

III. Experiment Design

A. Training Data Sets and Test Data Sets

When performed on a sample of data, the algorithms employed in this study will need to be trained and be able to improve the accuracy regarding the prediction of unseen data. The dataset will be split into two portions, the first split will contain data needed for training the models, whereas the second portion will contain data used for testing and evaluating the model [13]. The split percentage for the training dataset will be kept at 80% and for the testing dataset it will be 20% [13]. The information collected from the training set will extract features; the model sees this data for the first time, this aids in retaining attributes from the extracted features and categorization when it encounters fresh data.

B. Test Execution and Parameters

The dataset utilized for the purpose of training and testing the algorithms in this study contains historical daily stock prices and trading volume for US stocks and ETFs trading on NASDAQ and the New York Stock Exchange [14]. The different models will take in the daily stock data on the stocks then attempt to predict the future day's Closing Price. The stocks in question will be a subset of the Automotive industry stocks, specifically the manufacturing of automobiles. The automotive manufacturers that will be used are Ford, General Motors, Honda Motor Company, Navistar, PACCAR, Toyota Motors, and Tesla Motors. These stocks start at varying dates from 1977 (Ford) to 2010 (Tesla) but all end in the year 2017. Given the end date is 2017, the models will not be affected by the COVID19 pandemic from the start of 2020. Figure 2 Below provides key data points for the stocks for reference. The crucial features within this dataset include the date, Open, High, Low, Close, Volume, and OpenInt [14]. The average error along with other metrics including accuracy will be calculated then compared to determine which model produces the best overall results.

Name	Start Date	Total	Train	Min Value	Max Value
f	1977-01-03	10305	8244	0.21277	24.973
gm	2010-11-17	1759	1407	16.099	46.48
hmc	2005-02-25	3201	2560	17.385	42.521
nav	1970-01-02	12075	9660	6.23	452.5
pcar	1986-07-09	7901	6320	0.9379	74.651
tm	2005-02-25	3201	2560	53.594	137.72
tsla	2010-06-28	1858	1486	15.8	385.0

Fig. 2. Input Data Information

IV. Results

A. Algorithm I: Convolutional Neutral Network Implementation and Results

The Convolutional Neutral Network Algorithm will be based off a combination of examples accessible through Kaggle including but not limited to the "Deep Reinforcement Learning on Stock Data", "Recurrent Neural N. (RNN) Tutorial for Beginners" and "RNN & LSTM Comparison for Beginners." The resulting algorithm is broken into four separate parts, first is the Preprocessing Stage, second is the Model Building/Training Stage, Prediction of Test Data Stage, and lastly is an analysis Stage. [15]

In the Preprocessor Stage there are 4 distinct parts: categorizing training/test, normalization of the input data, identifying input vs output variables, and reshaping the data. After splitting the data between training and testing the data is normalized. Normalization is important because normalized data is consistent and has a uniform scale, this enables the model to have more accurate predictions by reducing the impact of outliers and skewed data. Lastly, normalization helps against overfitting based on the input data. The next part of the preprocessing stage is identifying the input and output variables, allowing the model to learn the underlying patterns and relationships between them, which enables the model to perform accurate prediction. Note that the output data point we are using is the End of Day 'Close' and we are inputting the previous 32 days' worth of data into the model as an input. The last part of preprocessing is the reshaping of the data to become more compatible with the architecture of the CNN model. From there the algorithm creates and trains the CNN model. [15].

The CNN Machine Learning Model is implemented as a Sequential Model using TensorFlow meaning the model is made up of a plain stack of layers, these layers have one input and one output. The sequential model allows for adding processing layers throughout the model for tweaking performance. The First Layer is a 1-Dimensional Convolution Layer using the Rectified Linear Unit activation function; the

layers are made up of 32 nodes, respectively. Note that the Activation function of RELU was chosen due to its speed of computation enabling the team to run the models without expensive hardware. The Convolution Layers apply a sliding convolutional filter to 1D input attempting to extract features from the Array of inputs. From there the model flattens the output into a single-dimensional array. The single-dimension array supports our next set of layers. Next, we have a combination of off and dense layers. The Dense layers, known as the fully connected layers, aid in mapping the representation between the inputs and the outputs. The dropout layer protects the model from overfitting to the training data. After the model has been built and trained the next step is testing against testing data. Note that the model will be trained against each stock's individual history and then tested against the known future, meaning the model will not be trained against all stock's training data prior to testing. A Summary of the Model can be found in Figure 3. [16]

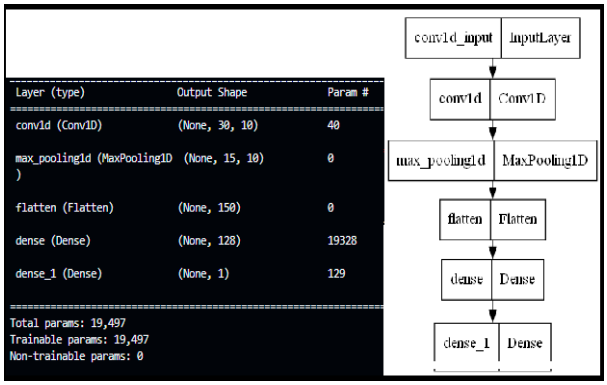

Fig. 3. CNN Model Summary

Figures 4 and 5 show the results of predicting the test data against the model and figure 71 highlights the key performance metrics against each of the data sets. The bullets below describe the performance of the model for each stock price:

- For Ford, the CNN Model overvalued the stock price and routinely did not account for drastic negative peaks resulting in significant differences between the predicted stock price and the actual stock price. Overall, the difference in stock price was not large, but constantly off.
- For General Motors, the CNN Model undervalued the stock price and routinely overcompensated for declines by dropping larger than the negative peaks.
- For Honda, the CNN Model overvalued the stock price and routinely did not anticipate the negative peaks well. The performance of this stock was like the performance of Ford stock.
- For Toyota, the CNN Model undervalued the stock price and constantly lagged changes in stock price.
- For Navistar, the CNN Model slightly under evaluated the stock, but most of the time it is accurate. The one outstanding flaw is that the model sometimes does not accurately predict the dips within the stock price.
- For PACCAR, the CNN Model overvalued the stock price and routinely did not account for drastic negative peaks resulting in significant differences between the predicted stock price and the actual stock price. Overall, the difference in stock price was not large, but constantly off.
- For Tesla, the CNN Model regularly underestimated the model by a significant margin and significantly lagged the actual stock prices. This is likely due to the major changes in stock price with a limited number of data points.

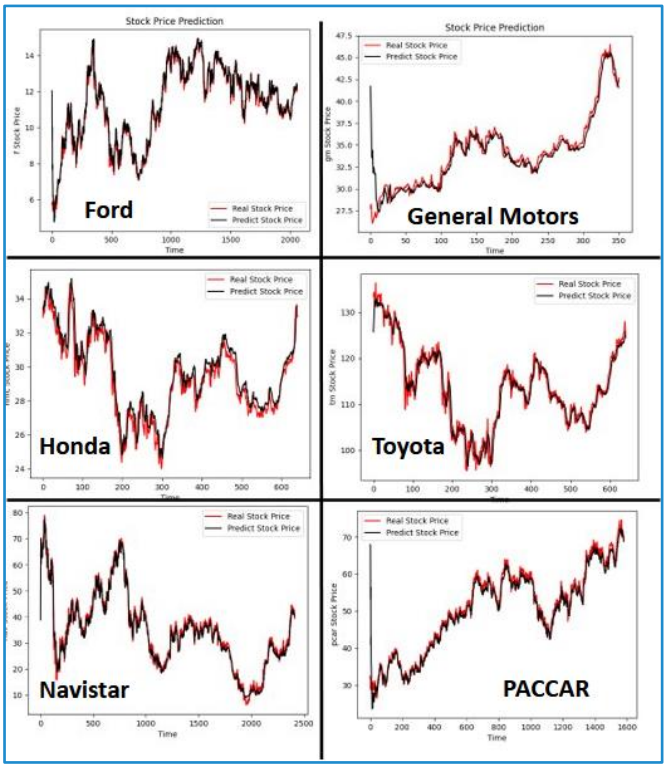

Fig. 4. Similar Performance against traditional automotive companies

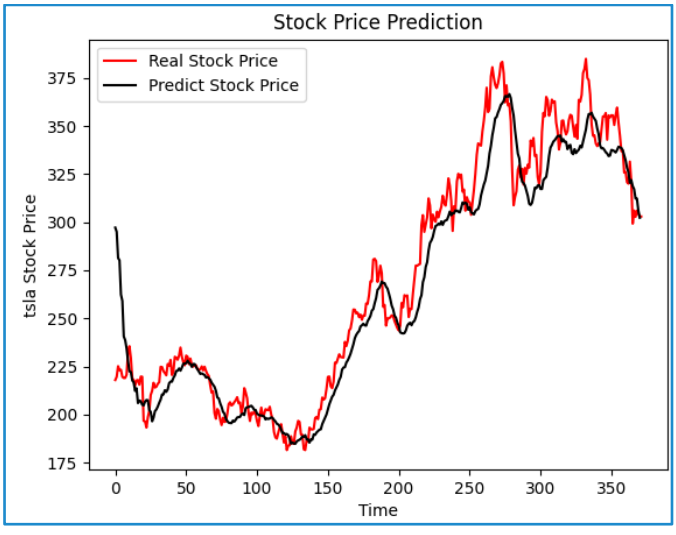

Fig. 5. Poor Performance against electric automotive Company Tesla

In order to easily judge the performance of the Models metrics were calculated against each data set, The following metrics were chosen Maximum Difference between the actual price and the predicted price, the Minimum Difference between the actual price and the predicted price, the Mean Squared Error (MSE), Root Mean Squared Error (RMS), and the Absolute Mean Error. The MSE measures the average squared difference between the predicted and actual values therefore the lower the value the better the performance, the RMS root of the average squared difference between the predicted and actual values, and the AME measures the average absolute difference between the predicted and actual values which is less sensitive to outliers compared to the MSE or RMS. Using these metrics, it is possible to analyze the model's performance a bit more. If the RMS and MSE are large and have similar values, then the model's performance is bad across the board, but if they have major differences then there are outlying pieces of data the model did not handle well. Based on that knowledge it is easy to see that the model did not handle outliers in the following datasets: General Motors, and PACCAR.

Name	Max Dif	Min Dif	MSE	RMS	AME
f	6.627	0.0	0.239	0.489	0.35
gm	14.894	0.004	1.895	1.377	0.754
hmc	2.137	0.0	0.248	0.498	0.375
nav	22.91	0.001	3.883	1.971	1.394
pcar	40.51	0.0	5.402	2.324	1.221
tm	10.541	0.001	4.614	2.148	1.64
tsla	73.765	0.052	227.568	15.085	11.433

Fig. 6. Metric Summation on each stock

Overall, the CNN Model performed differently for each of the data sets (Stock) therefore over all the ability to rate its performance or to identify improvements is difficult. The use of the basic CNN with minimum preprocessing would not be recommended as stocks were routinely incorrect and the algorithms were unable to predict changes in stock accurately or quickly.

B. Algorithm II: Long Short-Term memory Implementation and Results

As noted earlier, LSTM is a type of RNN, it is recognized for time series problems, and plays a crucial role when dealing with long-term dependencies [6]. For this paper, the dataset will go through a preprocessor that will allow it to be easy to work with, this includes, the dates will be sorted in ascending order, and the 'Date' column will be set as the index of the data frame containing the data, by doing this it will be easily accessible and to manipulate the data based on dates. Furthermore, the 'Close' column from the data will be extracted and reshaped into a 2-dimensional array with one column and an appropriate number of rows, this column is reshaped due to further processing, which includes scaling or feeding it into the machine learning model Moreover, the data needs to be normalized, this is required so that all the stock prices will be within a common range, the range can be of personal choice. Once the dataset has been normalized and stored within a variable, the training and testing sets are created by selecting a split ratio, the model will 80% for training and 20% for testing purposes. Since the problem deals with sequential data, sequences of data need to be generated for training and testing the model, this is because the data has a long-term dependency, such that the data must be carefully inputted into the model, to prevent misrepresentation or loss of information. Once the LSTM architecture has been created, it consists of 64 memory units to learn the patterns from the input sequence, the model can be trained using the training set for 10 epochs and a batch size is set to 10. Once it has completed training, the testing set is inputted to evaluate the model's performance by predicting the price of the stock.

Figures 7 – 13 display the plots of the predicted and actual stock price values, while figure 14 holds the crucial values depicting the performance of the LSTM model on chosen stocks for this paper. The LSTM Model anticipated a raised stock price for Ford, but there was no significant difference between the predicted and actual stock price when compared to the CNN model. Overall, the variation in stock price was not significant; the largest difference was 2.09.

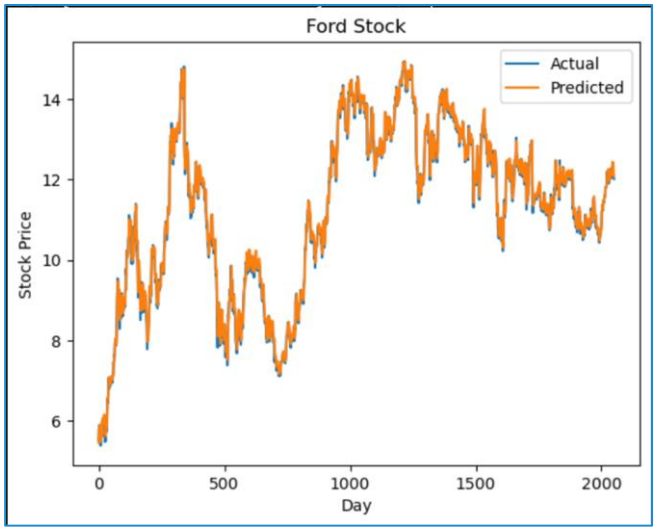

Fig. 7. Forecasting the stock price for Ford

The LSTM Model underestimated General Motors' stock price and frequently inadequately compensated for falls by dropping more than the negative peaks. The LSTM model had the smallest maximum difference of 3.426 when compared to the CNN model, which had a maximum difference of 14.894.

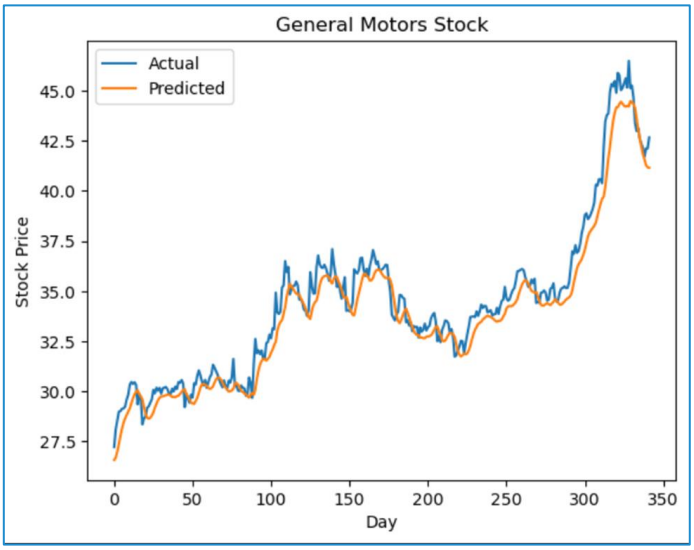

Fig. 8. Forecasting the stock price for General Motors

The LSTM model performed similarly to the CNN model in this prediction, with the greatest differences being 2.137 for CNN and 2.698 for LSTM. Both models raised the stock price and did not account for any price decrease that may get induced by a sudden dissatisfaction against the firm.

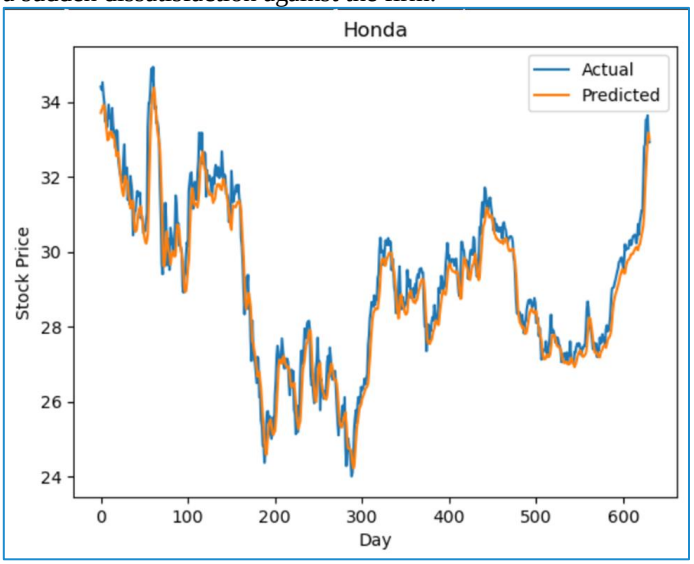

Fig. 9. Forecasting the stock price for Honda

The LSTM model has a significant flaw, it may not reliably forecast stock price declines. This is because the algorithm may be unaware of spontaneous human errors or accidents that may have caused the price to decrease on a certain day. Furthermore, the LSTM model outperforms the CNN model by around 45%.

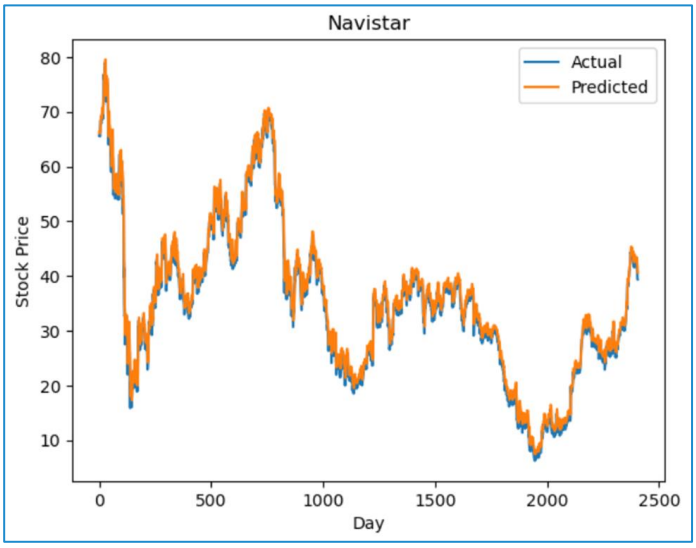

Fig. 10. Forecasting the stock price for Navistar

For PACCAR, the LSTM model demonstrates that it is an excellent choice for forecasting stock prices because it outperforms the CNN model by a significant margin; the maximum difference for the LSTM model is 4.037, whereas the CNN model has a maximum difference of 40.51, representing a 95% improvement.

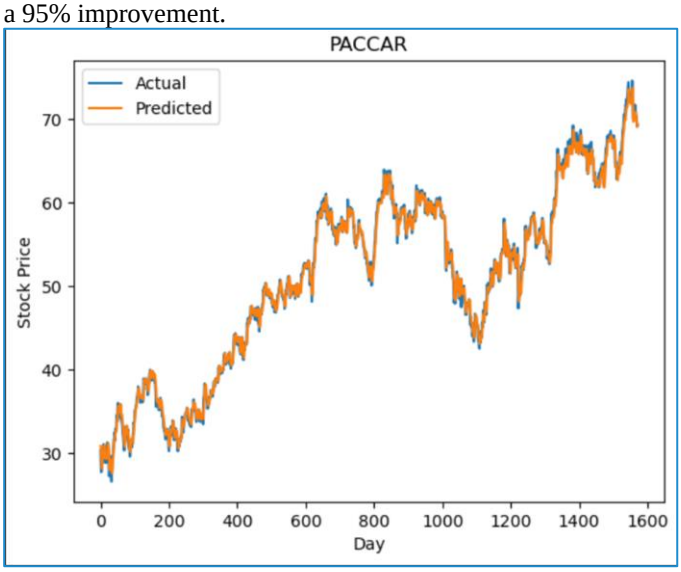

Fig. 11. Forecasting the stock price for PACCAR

When evaluating their relative maximum discrepancies from actual prices, the two models performed similarly on the Toyota stock. As a result, when there is a lot of variation in the data, the CNN model suffers, resulting in more inaccuracies as compared to the LSTM model.

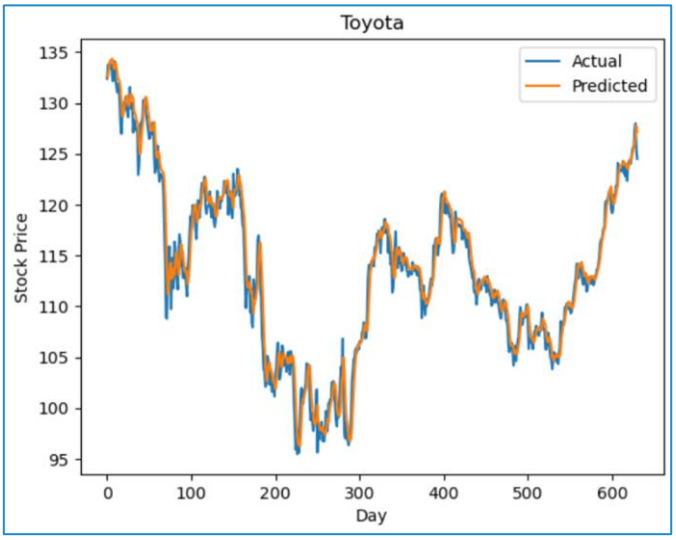

Fig. 12. Forecasting the stock price for Toyota

The LSTM Model regularly underestimated the anticipated value for Tesla by a substantial margin; yet, when comparing the two models, the LSTM model outperforms the CNN model. The largest difference achieved by the LSTM was 37.22, whereas the maximum difference achieved by the CNN model was 73.765. The huge deviation is most likely due to the large fluctuations in stock price with a small number of data points.

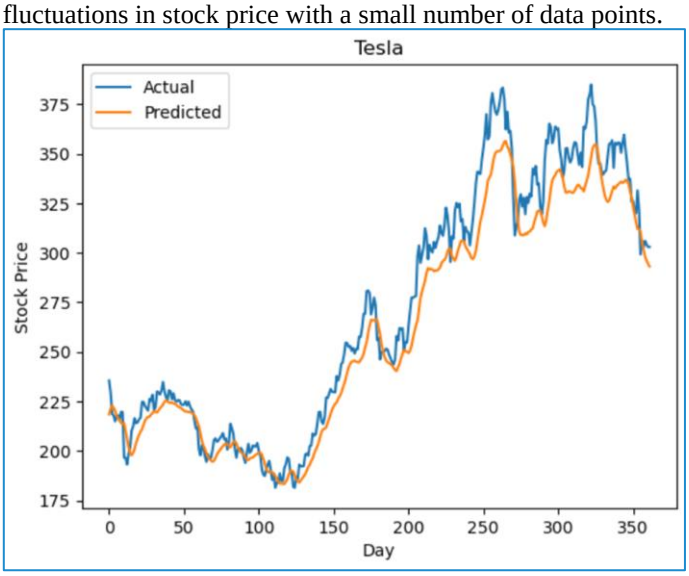

Fig. 13. Forecasting the stock price for Toyota

Metrics were calculated against each data set in order to quickly assess how well the models performed. The following metrics were selected: Maximum Difference between the Actual Price and the Predicted Price, Minimum Difference between the Actual Price and the Predicted Price, the MSE, the RMS, and the AME. By comparing figures 6 and 14, it can easily be determined that the LSTM model outperforms the CNN model, however, there are instances where the where the CNN model does have similar performance almost eqaulivalent to the LSTM model, but the LSTM model consistently did better and provided most possible accurate predictions.

Stock	Max Dif	Min Dif	MSE	RMSE	AME
F	2.091	0.00	0.0687	0.262	0.206
GM	3.426	0.0011	0.825	0.908	0.695
HMC	2.699	0.0008	0.250	0.500	0.377
NAV	12.120	0.001	1.373	1.172	0.807
PCAR	4.038	0.001	1.393	1.180	0.976
TM	9.005	0.0007	3.037	1.742	1.276
TSLA	37.227	0.045	114.010	10.677	8.164

Fig. 14. Crucial Performance Metrics for all automotive stocks

V. Conclusion

As can be seen from the results presented in this paper, the LSTM model is the ideal choice for stock market forecasting, this is mainly due to its ability to capture the predictions from data supplied in a successive order that captures long-term interdependence and temporal trends. The memory cells in the LSTM model allow it to store and update past information, the model can predict the price according to past information, and it will retain the information while updating it over a long duration. Furthermore, CNN models require a fixed length, whereas the LSTM model can change its input sequence length according to its needs for solving the problem. Moreover, from the results the CNN model struggles with noisy data leading to inaccurate predictions, on the other hand, LSTMs are capable of learning complex relationships within the data, and since stock market data will inevitably contain a variety of fluctuations, thus proving that LSTMs are the ideal choice for solving time series problems.

For future projects I would like to re-tune the CNN Model to create a more consistent model. Additionally, I would like to change the approach to training the CNN Model, instead of training and testing on a single model I would want to train on all automobile stock data. Where all stock data was fed into the model to train then test each of the stocks to perform individually. Lastly, I would add another metric of how the measurement: percentage of time the model correctly predicted a change in direction of the stock price and similar metrics.

Acknowledgment

We thank Dr. Wagnar for his continuous support and guidance throughout our research. We would like to thank Syracuse University for providing us with access to the necessary cloud-based technologies and research articles to implement machine learning algorithms.

References

[1] "Public companies," *Public Companies | Investor.gov*. [Online]. Available: https://www.investor.gov/introduction-investing/investing-basics/how-stock-markets-work/public-companies. [Accessed: 06-May-2023].

[2] S. Saha, "A comprehensive guide to Convolutional Neural Networks - the eli5 way," *Medium*, 16-Nov-2022. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed: 19-Feb-2023].

[3] T. M. H. Hope, "Linear regression," *Machine Learning*, pp. 67–81, 2020.

[4] D. Maulud and A. M. Abdulazeez, “A review on linear regression comprehensive in machine learning,” *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 140–147, 2020.
[5] [5] N. Lang, “An introduction to long short-term memory networks (LSTM),” Medium, https://towardsdatascience.com/an-introduction-to-long-short-term-memory-networks-lstm-27af36dde85d (accessed May 10, 2023).
[6] [4] “Deep learning: Introduction to long short term memory,” GeeksforGeeks, https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/ (accessed May 10, 2023).
[7] S. Mehtab and J. Sen, “Stock price prediction using CNN and LSTM-based deep learning models,” *2020 International Conference on Decision Aid Sciences and Application (DASA)*, 2020. doi:10.1109/dasa51403.2020.9317207
[8] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” *2017 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)*, 2017. doi:10.1109/icacci.2017.8126078
[9] [1] H. M, G. E.A., V. K. Menon, and S. K.P., “NSE stock market prediction using deep-learning models,” *Procedia Computer Science*, vol. 132, pp. 1351–1362, 2018. doi:10.1016/j.procs.2018.05.050
[10] [2] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, “EA-LSTM: Evolutionary attention-based LSTM for time series prediction,” *Knowledge-Based Systems*, vol. 181, p. 104785, 2019. doi:10.1016/j.knosys.2019.05.028
[11] [3] Z. Karevan and J. A. K. Suykens, “Transductive LSTM for Time-series prediction: An application to weather forecasting,” *Neural Networks*, vol. 125, pp. 1–9, 2020. doi:10.1016/j.neunet.2019.12.030
[12] [1] J. Wang, X. Li, J. Li, Q. Sun, and H. Wang, “NGCU: A new RNN model for time-series data prediction,” *Big Data Research*, vol. 27, p. 100296, 2022. doi:10.1016/j.bdr.2021.100296
[13] T. Shah, “About train, validation and test sets in machine learning,” *Medium*, 10-Jul-2020. [Online]. Available: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7. [Accessed: 24-April-2023].
[14] B. Marjanovic, “Huge stock market dataset,” Kaggle, https://www.kaggle.com/datasets/borismarjanovic/price-volume-data-for-all-us-stocks-etfs [accessed May 24, 2023].
[15] [1] B. Marjanovic, “Huge stock market dataset,” Kaggle, https://www.kaggle.com/datasets/borismarjanovic/price-volume-data-for-all-us-stocks-etfs (accessed Jun. 4, 2023).
[16] “TensorFlow dense: How to use function tensorflow dense?,” *EDUCBA*, 12-Jul-2022. [Online]. Available: https://www.educba.com/tensorflow-dense/. [Accessed: 27-Feb-2023].