HIGH SCHOOL EDITION
Journal of Student Research
Volume 10 Issue 1 (2021)

Systematic Optimization of Long Short-Term Memory Model for Predicting NYSE Arca Airline Index (XAL) during COVID-19

Sarah Dong[1] and Amber Wang[1]

[1]Westview High School, San Diego, CA, USA

ABSTRACT

Predicting stock prices has been both challenging and controversial. Since it first spread through the United States, the COVID-19 pandemic has impacted the stock market in a multitude of ways. Thus, stock price prediction has become even more challenging. Recurrent neural networks (RNN) have been widely used in many fields to predict financial time series. In this study, Long Short-Term Memory (LSTM), a special form of RNN, is used to predict the stock market direction for the US airline industry by using NYSE Arca Airline Index (XAL). The LSTM model was optimized through changing different hyperparameters of the model architecture to find the best combination for increased accuracy and performance evaluated by several metrics, including raw RMSE (3.51), MAPE (4.6%), MAPA (95.4%) and R^2 (0.978).

Introduction

The use of artificial intelligence and machine learning by financial institutions to help make trading decisions originates back to the 1970's. Artificial intelligence, also known as AI, involves using computer systems to perform tasks that normally require human intelligence [1]. Predictive analytics can be developed using artificial intelligence. Machine learning (ML) is a specific subset of artificial intelligence based on the idea that systems can learn from data, identify patterns, and make decisions with minimal human intervention [2]. Machine learning is an excellent tool for processing highly structured time series data like stock prices [3].

Deep learning is one of the most popular machine learning algorithms using artificial neural networks (ANN) that allows the computer to recognize and memorize information like neurons do in a human brain [4]. A network of neurons, or nodes, linked to each other, is the foundation of neural networks.

Recurrent neural networks (RNN), a class of artificial neural networks with self-loop in its hidden layers, are designed to have memory and recognize patterns in sequences of time-series data and other sequential data [5]. In a traditional artificial neural network, the data are independent of each other. RNN has memory, and its output depends on previous information. Although RNN has advantages over the traditional neural networks and it is effective with short-term dependencies, it can remember things for only a very short period of time. Thus, RNN has long-term dependency problems, and it does not work well on tasks that require context to make decisions or predictions.

Long short-term memory (LSTM) is a special kind of RNN that has memory that can span over a long period of time, thus avoiding the long-term dependency problems [6]. Like RNN, LSTM also has a chain structure. Additionally, LSTM networks have cell states that function as the long-term memory and run through the entire chain. In the LSTM network, there are also gated cells that are used to selectively add or remove information to the cell states [7]. This design and architecture (Figure 1) make predictions based on historical context possible and efficient. Standard recurrent neural networks have the problem of vanishing and exploding gradients. LSTM solves this problem by introducing the gates, which are logistic functions of weighted sums to manage the contents of the memory. There are

three gates (input, forget, and output) in a LSTM cell. The input and forget gates manage the cell state and allow for better control over the gradient flow while the output gate produces the hidden state and output vector.

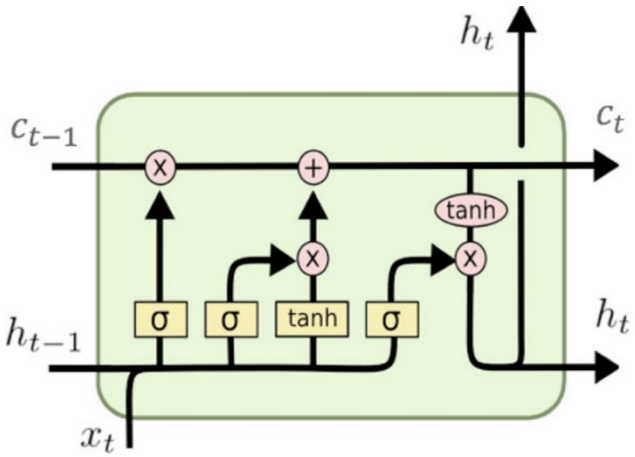

Figure 1. Understanding LSTM (Long Short-Term Memory) cell

Straying away from long-term dependence problems thus allows LSTM to be successfully applied in various fields related to time series, including language translation, speech recognition, image captioning, and stock prediction [8].

Due to its complex and dynamic environment, stock market price prediction is a very difficult task. Technical analysis based on historical data of prices is a popular way of modeling and predicting the stock prices. LSTM is especially a powerful approach for time series prediction problems such as stock market prediction because LSTM has memory cells and can store and forget information selectively [9].

Stock prices can be affected by many factors in various ways. Ever since COVID-19 spread through the United States in mid-March 2020, the whole stock market has been affected tremendously. Figure 2 shows the S&P 500 Index in 2020 [10]. The market in general has become much more complicated and unpredictable.

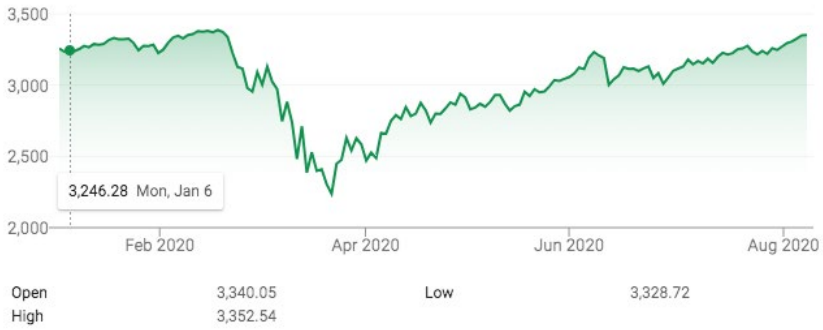

Figure 2. S&P 500 Index in 2020

During this pandemic period, many sectors have been hit hard, especially the hotel and airline industries. For airlines, the TSA passenger volume has dropped to 4.0% of the traffic that was seen a year ago. It has since slowly recovered to about 25% of last year's numbers [11]. As the global airline usages decrease, corresponding stock trends have been affected as well. The stock prices of the airlines have also dropped sharply to only about 20% of their peak value. Even though most airlines have recovered to about 1/3 of their highest values, the situation is still not clear as to when the crisis will end. In this study, the NYSE Arca Airline Index (XAL) was used as an example for the US airline

industry (Figure 3) [10]. The XAL Index tracks the price performance of major U.S. airlines and a few overseas airlines listed on NYSE.

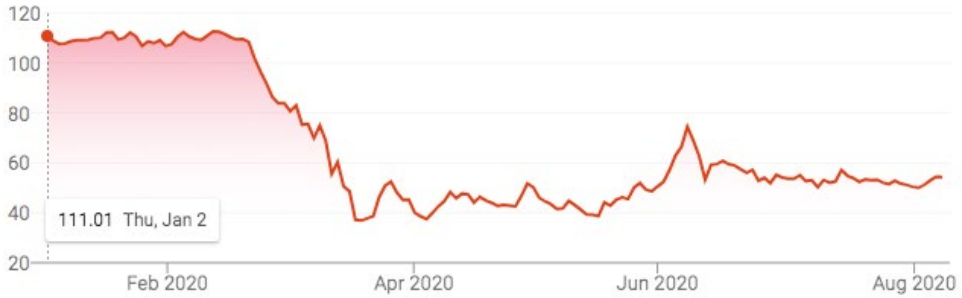

Figure 3. NYSE Arca Airline Index (XAL) in 2020

Based on the current COVID-19 situation, the prediction of airline stock price represented by the XAL index has become more challenging. In this study, LSTM, a special form of RNN that can address long-term dependencies in time series predictions, is used to predict the stock market direction for the US airline industry. The LSTM model is optimized through changing different hyperparameters to find the best combination for increased accuracy and performance. The goal of this paper is to optimize the LSTM model, as well as measure its effectiveness and accuracy through using the XAL index that has been affected dramatically by the COVID-19 pandemic.

Methods

Data Processing

The data of the NYSE Arca Airline Index (XAL) price was collected from Yahoo! Finance [12]. There are 7 columns of data including Date (trading date), High (highest price), Low (lowest price), Open (opening price), Close (closing price), Volume (number of stocks traded), and Adj. Close (adjusted closing price). The High, Low, Open, Close and Adj. Close are highly correlated with each other. For this index, the Close and Adj. Close are basically the same, so closing price is used in the analysis.

The dataset contained historical data of the prices from 2010-07-24 to 2020-07-23 (ten years, 2516 days) (Figure 4). The dynamic range of the graph indicates the different behaviors of stock prices over time, which will make the learning more robust for predictions under a variety of situations and allows for the current COVID-19 pandemic to be taken into consideration.

For data splitting, 95% of the historical data was taken for the training dataset (from 7/2010 to 2/2020), and the remaining 5% of data (from 2/2020 to 7/2020) was used for the testing (validation) dataset. The model was developed using the training dataset, and predictions were made on the testing dataset.

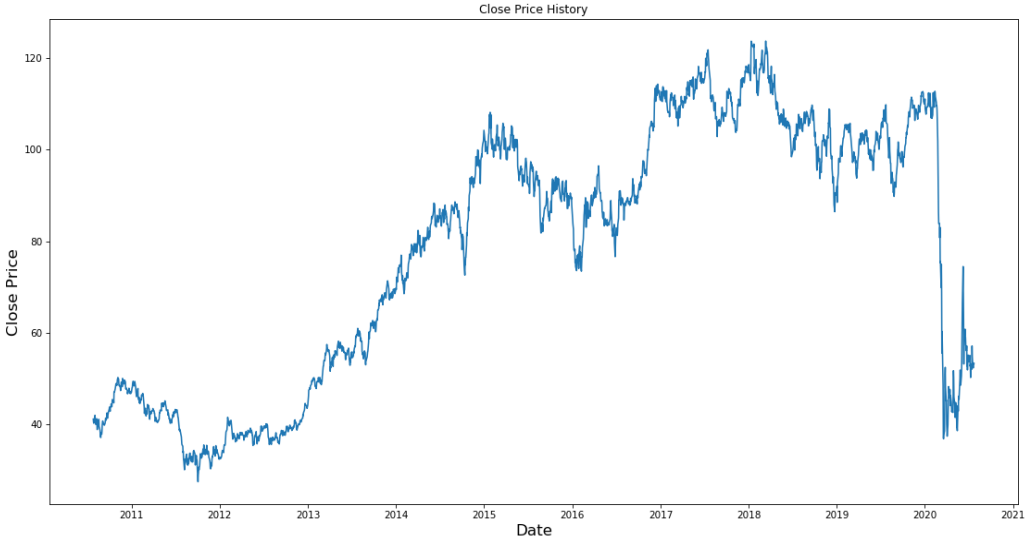

Figure 4. NYSE Arca Airline Index (XAL) Closing Price 7/2010-7/2020

Data Normalization

The stock index increases over time, which makes most values in the testing set out of the scale of the training set. Thus, the model cannot make predictions for values that it has never seen before. For optimal performance, data normalization is required. Data normalization can allow for the LSTM model to be easily trained and can improve the accuracy of the learning model. Data normalization will change the values of numeric columns in the dataset to a common scale. Scikit-Learn's MinMaxScaler was used to scale the dataset to numbers between 0 and 1.

LSTM Model Architecture and Training

In order for the LSTM model to function, its architecture needs to be built first. Like a building, the LSTM architecture contains a series of LSTM layers stacked sequentially on top of one another. The LSTM layers created are then added in the order that they should be connected. After the sequential LSTM layers, a dense layer will be added for the outputting.

Each layer in the LSTM architecture consists of neurons. The neurons can learn by going through the cell state and gates. The three different kinds of gates (input, forget, and output gates) are used to learn what information is important to keep or should be removed along the cell state by applying the respective activation function.

An LSTM layer could be described with an analogy, thought of as consisting of recurrently connected memory blocks, each a different version of a computer's memory chips. Each of these blocks contains recurrently connected memory cells with input, output, and forget gates, which essentially function for writing, reading, and resetting [13].

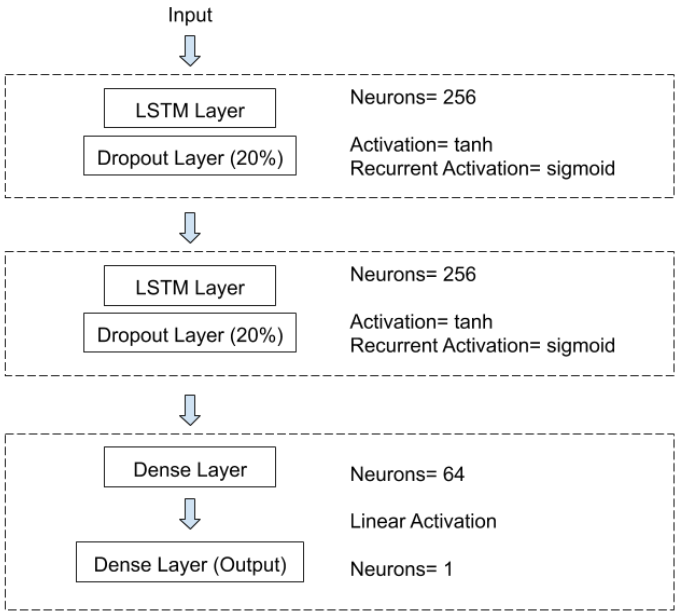

Figure 5. Model Architecture and Layers

In order to train the proposed model, deep learning usually needs a lot of computational resources and time. Therefore, an optimization algorithm is needed for faster speed and less resources. In this study, the model uses the Adam optimization algorithm, which is good for large data sets and parameters and has computational efficiency [15]. Mean Square Error (MSE) is used as the loss function in the model.

Model Evaluation Metrics

In order to obtain optimal results, the accuracy and performance of the model are evaluated and compared through regression and classification metrics, which can help to make decisions such as finding the right hyperparameters to use in the model. The two different metrics can then be used to evaluate the model from different perspectives. Regression metrics are used for the continuous value, such as stock price, while classification metrics are used for the categorical value, such as the stock price movement (up or down).

(1) Regression metrics

The regression metrics are used to measure the prediction fit. The following metrics are calculated:

(a) Root-mean-square error (RMSE)

Root-mean-square error (RMSE) is an approach to calculate the accuracy or error of a prediction model, such as LSTM. It is used to evaluate how closely the predicted values match the observed values. The smaller the RMSE, the more reliable the results produced by the LSTM model.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)\text{^}2}{N}}$$

(b) Mean absolute percentage error (MAPE)

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N} \left| \frac{Actual_i - Predicted_i}{Actual_i} \right| * 100$$

MAPE is another commonly used metric for evaluating the prediction accuracy. The smaller MAPE the better.

(c) Mean absolute percent accuracy (MAPA)

$$\text{MAPA} = 100 - \text{MAPE}$$

The higher MAPA, the more accurate the model is.

(d) Coefficient of determination (R^2)

The coefficient of determination can explain how much variability of one factor can be caused by its relationship to another factor. It is also called "goodness of fit." The coefficient is between 0 and 1. The higher the coefficient, the better and more reliable the model is.

(2) Classification metrics

There are four classes used for comparing the prediction price movement against the actual price movement.

- *TP* (true positive)
- *TN* (true negative)
- *FP* (false positive)
- *FN* (false negative)

		Prediction Price Movement	
		Increase	*Decrease*
Actual Price Movement	*Increase*	True Positive	False Negative
	Decrease	False Positive	True Negative

The classification metrics are calculated based on the above four classes.

(a) Precision

$$Precision = \frac{TP}{TP+FP}$$

(b) Recall

$$Recall = \frac{TP}{TP+FN}$$

(c) F1-score

F1-score is the combination of Precision and Recall. It is between 0 and 1, and a higher score is better.

$$F1\text{-}score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

(d) Movement Direction Accuracy (MDA)

MDA is the proportion of all the correct predictions of price movement direction corresponding to the whole price movement.

$$MDA = \frac{TP+TN}{TP+FP+TN+FN}$$

Software

The software used for analysis was Python 3 (https://www.python.org/) in Anaconda Jupyter Notebook (https://www.anaconda.com). Keras neural network API (https://keras.io), which uses TensorFlow (https://www.tensorflow.org/), was used as the backend for LSTM deep learning analysis.

Results and Discussion

Optimization of Hyperparameters

The LSTM performance and results are highly dependent on and extremely sensitive to the model's hyperparameters. Thus, in order to obtain good results and performance, the hyperparameters need to be chosen very carefully. While some hyperparameters can be determined by rules of thumb, in most cases, testing various configurations is necessary to obtain optimal results. For each hyperparameter, different scenarios were tested to find the optimal result with the smallest error based on the lowest RMSE (on the normalized value).

Optimization of Dropout

Dropout is a technique for neural network models where randomly selected neurons based on a probability are ignored during training. The goal of dropout is to improve performance and prevent neural networks from overfitting [14]. Overfitting means that the model cannot be generalized because the training data are modeled too well, analogous to memorizing the answers to a problem instead of learning the formula to solve it. By adding the dropout layer after each LSTM layer, overfitting can be effectively prevented. Tests are performed for different dropout values from 0.1 to 0.8 (increasing by 0.1) for getting the optimal dropout value. In each testing scenario, the timestep window was set to 60, batch size was set to 150, and epoch number was set to 50. Neuron numbers for the two LSTM layers and two dense layers are 256, 256, 64, and 1.

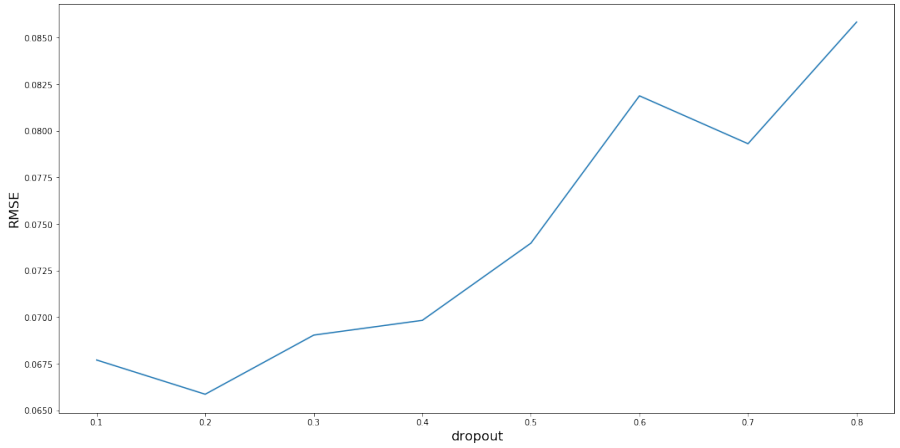

Figure 6. Testing plot for finding optimal dropout values

From the above resulting plot, we can see that the optimal dropout value is 0.2 (meaning that the dropout rate is 20%) because this value results in the smallest RMSE (Figure 6).

Optimization of Number of Training Epochs

An epoch is one complete pass through the training data along the network. The number of epochs is a hyperparameter for the number of times of the learning algorithm will go through the entire training data. In order to sufficiently minimize the error from the model, the number of epochs is usually large for the learning algorithm to run. Tests are performed for different epoch numbers with from 1 to 400 for getting the optimal epoch number. In each scenario of testing, the timestep window was set to 60, batch size was set to 150, and dropout was set to 0.2. Neuron numbers for the two LSTM layers and two dense layers are 256, 256, 64, and 1.

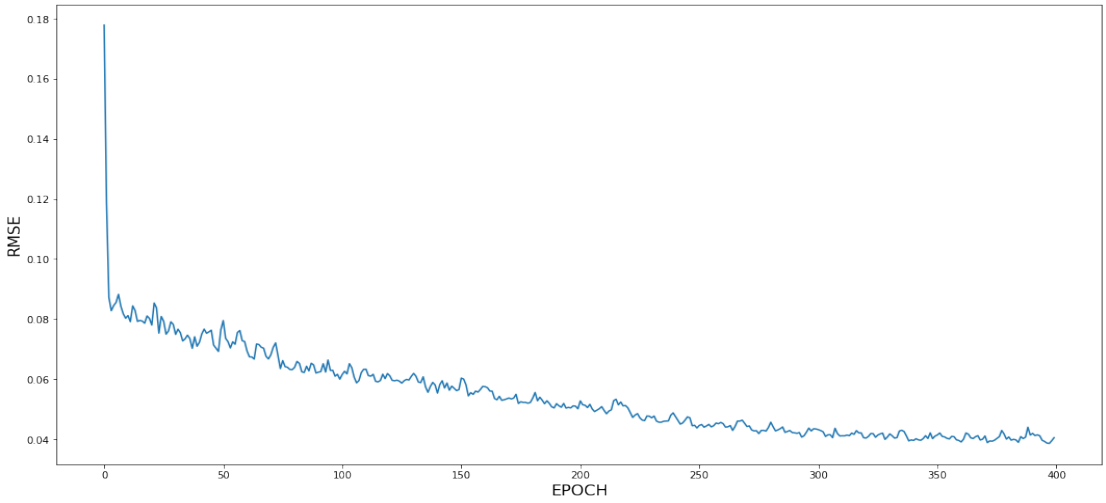

Figure 7. Testing plot for finding optimal Epoch number

The results clearly show a downward trend in RMSE over the training epochs, suggesting more epochs will provide more accuracy to the model (Figure 7). The optimal number of epochs is 300 because that value results in almost the smallest RMSE, and higher numbers of epochs above 300 result in similar RMSE.

Optimization of the Number of Neurons in each Layer

Generally speaking, more neurons would enable the system with more learning capacity at the price of longer training time and the possibility of overfitting the training data.

A combination of different numbers of neurons (nine scenarios) in the two LSTM hidden layers and two dense layers were tested to find the optimal value of neurons (Table 1). For finding the optimal neuron numbers, the timestep window was set to 60, batch size was set to 150, dropout was set to 0.2, and epoch number was set to 50. The following table contains the summary results for the nine testing scenarios and corresponding RMSE.

Table 1: Testing table for finding optimal neuron numbers.

Neuron Number Testing Scenario	RMSE
[256, 256, 16, 1]	0.07224
[256, 256, 32, 1]	0.06531
[256, 256, 64, 1]	0.06502
[128, 128, 16, 1]	0.07359
[128, 128, 32, 1]	0.07366
[128, 128, 64, 1]	0.06945
[64, 64, 32, 1]	0.08727
[64, 64, 16, 1]	0.08538
[32, 32, 16, 1]	0.10280

From the testing result table, we can see that the optimal neuron number combination in each layer was 256, 256, 64, 1, which produces the smallest RMSE.

Optimization of the Size of Timesteps

Timesteps define how many units back in time you want your network to see. For example, for time series data such as stock prices, a timestep with a value of 60 means that we will look into 2 months (60 days) of data to predict the next day's price. Tests are performed for different timestep values from 10 to 70 (increasing by 10) for getting the optimal timestep value (Table 2). In each testing scenario, the batch size was set to 150, epoch number was set to 50, and dropout was set to 0.2. The neuron numbers for the LSTM layers and dense layers are 256, 256, 64, and 1.

Table 2: Testing table for finding optimal timestep.

Timestep	RMSE
10	0.06862
20	0.07475
30	0.06616
40	0.07053
50	0.07005
60	0.06996
70	0.06831

From the testing result table, we can see that the optimal timestep size is 30 because that value results in the smallest RMSE.

Optimization of Batch Size

An epoch has one or more batches. In one epoch, when the learning algorithm passes through the training data, the data can be split into batches of the same size (batch size). Batch size controls how often to update the weights of the model network. A small batch size will reduce the speed of training but too big of a batch size (such as the whole dataset) will reduce the model's ability to generalize to different data. Tests are performed for different batch sizes for getting the optimal batch size value (Table 3). In each testing scenario, the timestep is set to 60, dropout is set 0.2,

epoch number is set to 50; and neuron numbers for the LSTM layers and dense layers are 256, 256, 64 and 1, respectively.

Table 3: Testing table for finding optimal batch size.

Batch Size	RMSE
25	0.04501
50	0.05161
100	0.06852
150	0.07012
200	0.07120
250	0.07838

From the testing result table, the optimal batch size value is 25 because that value results in the smallest RMSE.

Final Model Accuracy Evaluation

Based on the above hyperparameter testing results, the optimal size of timesteps is 30, dropout is 0.2, epoch number is 300, batch size is 25, and the neuron numbers are 256, 256, 64, and 1 for each layer. By using these optimal hyperparameters, the final model was built and executed. The accuracy metrics are calculated for this final model with the testing (validation) data.

Performance Result of Predicting the Testing Data for Regression Metrics

Table 4: Evaluate final model accuracy using regression metrics.

Metrics	Values
RMSE (Scaled)	0.037
RMSE (Raw)	3.510
MAPE	4.6%
MAPA	95.4%
R^2	0.978

The experimental results showed the optimal combination of hyperparameters with improved accuracy for prediction with very small root-mean-square error (RMSE) and mean absolute percentage error (MAPE), as well as a very high mean absolute percent accuracy (MAPA) and coefficient of determination (R^2) (Table 4).

Performance Result of Predicting the Testing Data for Classification Metrics

Table 5: Evaluate final model accuracy using classification metrics.

Metrics	Values
Precision	0.412
Recall	0.389
F1-Score	0.400
MDA	0.492

From Table 5, we can see that the optimal combination of hyperparameters provides decent classification metrics results for predicting the NYSE Arca Airline Index (XAL) movement direction, but the values are not extremely high. This may be due to the COVID-19 pandemic, as it is a black swan event that would affect the LSTM model's predicting capability. For the testing dataset, actual and predicted prices are very close to a regressed diagonal line, indicating a good fit (Figure 8).

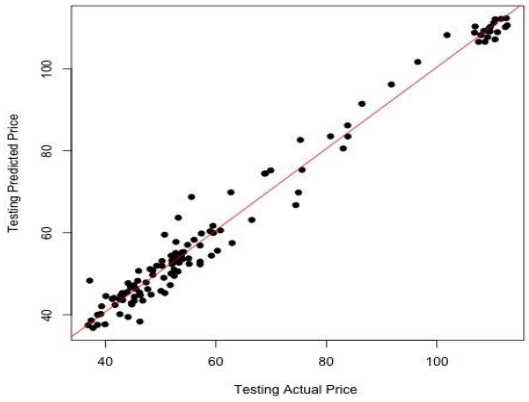

Figure 8. Final model result: actual price vs predicted price for testing dataset

In the final model, the validation and prediction results are very close to each other using the optimal hyperparameter combination (Figure 9). From these results, we conclude that our LSTM model has been optimized with the best combination of different hyperparameters to achieve the increased accuracy and performance. Even with the dynamic change of the stock market during the pandemics, our model still provides decent predictions.

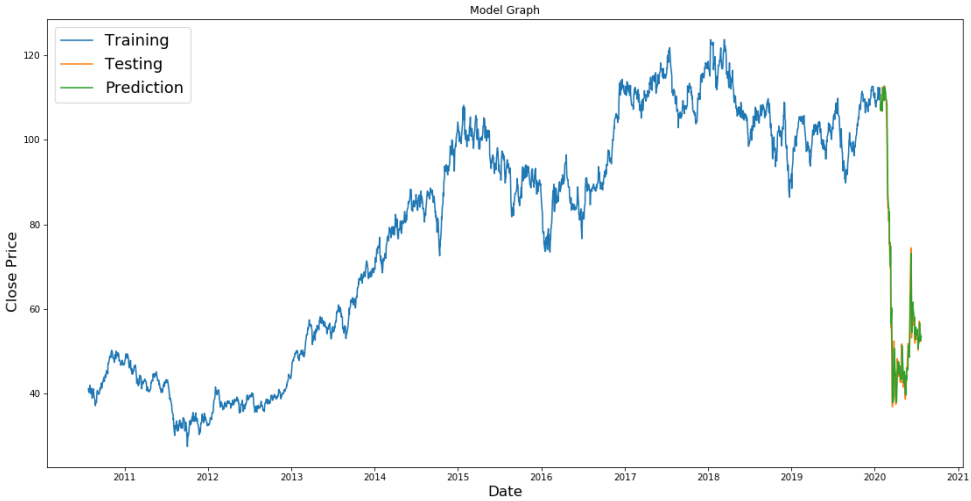

Figure 9. Prediction of XAL closing price with optimal hyperparameters

Future directions

An unexpected situation such as the COVID-19 pandemic can have a dramatic impact on the fundamentals of the stock market in the short term, which will impose a great challenge to the LSTM model. As the situation improves,

the stock index will continue move into the positive territory. More training data from this stabilization and recovery period will allow better prediction of the future direction of the index.

The goal of this study was to predict the future stock price movement, not the prices themselves. The future work has several directions:

Backtesting has been widely used for stock price prediction based on the past data with almost perfect results. However, there is a huge difference between "backtesting" and "real prediction". For "real prediction", the last prediction should be fed into the LSTM model and will become the last entry data of the training set for prediction of the next closing price. This process will continue until we have predicted the price over a certain amount of trading days. Then we can compare the prediction to backtesting. If they are not near to each other, that means the model has overfitted the data. Optimizing hyperparameters would allow us to overcome the overfitting problem to achieve better accuracy.

Stock prices are considered to be very dynamic and susceptible to quick changes due to unknown factors. Historical data on price trends are not sufficient for accurately predicting the price on a given day. Both technical and fundamental data should be used. While the technical data is in the form of historical stock prices, and the fundamental data is in the form of news stories and analyst opinions. Especially during the COVID-19 pandemic, there is a lot of news and daily events that can affect the stock market. We can combine time series analysis with information from Google Trends and Yahoo! Finance websites to forecast stock prices. By doing so, improvements are likely to be seen with the price movement direction accuracy. The hypothesis is that news and "external" factors have a very large impact on how stock prices evolve. In one case, investor sentiment tendency was combined with LSTM for more accurate stock forecasts [16]. More factors need to be considered for more complex models in future works.

Acknowledgments

I would like the thank my advisor Amber Wang for helping me with the project.

References

[1] Russell SJ and Norvig P. Artificial intelligence: a modern approach. Pearson Education Limited, 4th Edition, 2020.

[2] Michalski R, Carbonell J and Mitchell T. Machine Learning: An Artificial Intelligence Approach (Volume I). Morgan Kaufmann, 2014.

[3] Nasrabadi NM. Pattern recognition and machine learning. Journal of electronic imaging, 16(4):049901, 2007.

[4] LeCun Y, Bengio Y and Hinton G. Deep learning. Nature 521:436–444, 2015.

[5] Dupond S. A thorough review on the current advance of neural network structures. Annual Reviews in Control. 14: 200–230, 2019.

[6] Hochreiter S and Schmidhuber J. Long short-term memory. Neural computation, 9 (8):1735–1780, 1997.

[7] Olah C. Understanding lstm networks. 2015. URL http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[8] Van Houdt G, Mosquera C and Nápoles G. A review on the long short-term memory model. Artif Intell Rev, 2020.

[9] Nelson DMQ, Pereira, ACM and de Oliveira RA. Stock market's price movement prediction with LSTM neural networks. Proceedings of the International Joint Conference on Neural Networks (IJCNN), 1419–1426, 2017.

[10] Google Finance: URL https://google.com/finance

[11] Martin G. Summer Holiday Traffic Pushes Air Travel Volumes To New Pandemic Highs. Forbes, Jul 12, 2020.

[12] Yahoo! Finance. NYSE ARCA AIRLINE INDEX (^XAL), 2020. URL

[13] Graves A and Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18(5-6): 602-610, 2005.

[14] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1): 1929–1958, 2014.

[15] Kingma DP and Ba JL. Adam: A method for stochastic optimization. A conference paper at ICLR 2015.

[16] Jin Z, Yang Y and Liu Y. Stock Closing Price Prediction Based on Sentiment Analysis and LSTM. Neural Computing and Applications (32): 9713–9729, 2020.