# Comparative Analysis of Machine Learning Models for Predicting Airline Stock Price Movements

Mason Hagan mh20gx@fsu.edu
Pietro Candiani pc21d@fsu.edu
Anthony Ingle asi20@fsu.edu
AJ Tello-Rodriguez ajt21i@fsu.edu
Jingwen Zhuo jz19g@fsu.edu

October 2023

**Abstract**: This paper investigates the potential predictive efficacy of Support Vector Machines (SVM), Long Short-Term Memory (LSTM), and Extreme Gradient Boosting (XGBoost) in predicting the stock price movement of an airline company. It employs a range of features encompassing historical stock and crude oil prices, moving averages and volume data. Through a comparative analysis, the study evaluates the models' performance and feature relevance. Results showcase SVM's effectiveness with specific feature sets and kernel types. The LSTM displays nuanced performance based on feature inclusion and timeframes, whereas XGBoost exhibits superior performance pre-COVID-19. The study underscores the predictability of market conditions using technical indicators and proposes considerations for future research in financial market prediction models.

## 1 Introduction

In the rapidly evolving landscape of today's stock market, predicting price movements using machine learning has consistently stood out as a critical component to grow a portfolio. Among available models, Recurrent Neural Networks (RNNs), Support Vector Machines (SVMs), Random Forests (RF), Long Short-Term Memory (LSTM) models, and Gradient Boosting Machines (GBMs) have proven to be the most popular. In this paper, we explore the usage of the Support Vector Machine (SVM), Long Short-Term Memory (LSTM) and Extreme Gradient Boosting (XGBoost) in predicting the price movement of American Airlines (NASDAQ:AAL) stock, using a diverse selection of features that are critical to the airline's performance as a company. Among these features are historical price and volume data, as well as crude oil prices. We compare the performance of these three models and determine which one is the most efficient to apply to the financial markets. This study serves as a comparative analysis to determine which model best works with the selected features, along with how effective and relevant the features are to the movement of the airline's stock price. In our analysis, we hope to create an efficient model that will accurately predict whether a stock price will go up or down.

## 2 Literature Review

There have been several attempts to explore machine learning algorithms in the context of stock price prediction for other asset classes in various exchanges. Choi and Choi [1] used an LSTM (Long Short-Term Memory) model with American Airlines and crude oil price data to predict airline stock price movement. The study shows that this model along with the inclusion of crude oil price movement gives a much more accurate price prediction compared to other studies using other simple machine learning models and only historical price data. Jyothi et al. [2] discusses the benefits of using SVM models for stock price prediction. The findings that are presented find that the model is able to predict price movement to a large degree, however there is space for further development, such as more granular price data for better accuracy during price swings. Qiu et al. [3] use Support Vector Machine (SVM), Decision Tree (DT), Gradient Boosting Decision Tree (GBDT), Random Forest (RF), Naive Bayes (NB), K-Nearest Neighbor (KNN) and Logistic Regression (LR) models in combination with sentiment analysis to predict short-term stock trends. They find that modified sentiment index has huge benefits in reflecting the sentiment of the market, finding that adding a weighted sentiment index to an SVM model can improve its prediction by as much as 68.37%.

# 3 Methodology and implementation

In our experiments, data was collected and processed, and used to train and test models. These models were then compared by their predictive performance using various metrics.

## 3.1 Data Collection

An extensive amount of data was gathered under the general criteria that it had factors that influenced the health of airline companies. For each model, we then tested various factors of data and selected the most fitting parameters for that particular model, because we did not expect every model to work best with the same exact data.

### 3.1.1 Price Data

We collected daily stock price data for open, close and volume markers over the time period starting from 2005 until the most recent data at the point of collection for 2023. We used the *Alpha Vantage API* and *Yahoo Finance API*to obtain this daily historical stock price data for American Airline stock (AAL) and the crude oil commodity stock (WTI). Crude oil stock was chosen because crude oil prices directly contribute to the cost of fuel, which directly impacts the cost of each flight. Therefore, the change in the price of oil can be translated into a change in the operational costs for an airline. It was expected that the price of oil would exhibit some inverse relationship with the price of the stock. This is further justified in the study done by Choi and Choi [1], who demonstrated that including the price of oil in an LSTM model increased its prediction accuracy. The usage of historical price and volume data on the AAL stock itself could help the model find and apply patterns from the past. Simple moving average data was collected explicitly using the *Yahoo Finance API*.

### 3.1.2 Weather Data

Among those factors that could potentially influence the health of an airline, weather data was a top runner. It was decided to collect weather data from the top 10 cities with the highest air traffic in the US, aiming to find potential correlations between weather conditions and their impact on flight operations. Our decision for this was due to the fact that adverse weather conditions could increase flight cancellations and delays, increasing the amount of tickets needing to be refunded, and thereby potentially lowering profitability at the time of those poor weather conditions. Moreover, such disruptions might erode consumer confidence in the airline, a sentiment that resonates strongly within financial markets. We expected that bad weather conditions would have an inverse relationship to the price of the airline stock. However, what we discovered was that weather data was not a good predictor of immediate stock price in daily movement. Due to the nature of predicting stock price in short intervals of days, we decided to leave out weather data from our models.

## 3.2 Data Processing

In this step, we processed and standardized the data into features and split the data into train and test data sets. We created two different time frames for our data sets, one from 2005 to 2023, and one from 2005 to the end of 2019. We chose this because 2020 marked the beginning of COVID-19 across the world, which influenced the financial markets greatly. The stock market during this time period acted in unpredictable ways and had different volatility compared to prior. We presumed that this would have an effect on the training of our models, so we wanted to compare training the models with the COVID-19 time period, as well as without. For data sets, we created 1 to 5-day lag data sets on the price data for AAL and WTI by shifting each feature the amount of days corresponding to the lag factor. For instance, for a 3 day lag, each data point was the closing price from 3 days prior. Finally, we standardized the features by removing the mean and scaling them to unit variance. This works on the principle of transforming the data such that it has a mean of zero and a standard deviation of one. This standardization helps in achieving better convergence, preventing a single feature from dominating due to its larger scale, and ensuring that all features contribute equally to the analysis. While the pandas datareader and other python tools have proven useful for these tasks specifically, others have used tools such as the WEKA data mining software created by developers from the University of Waikato.[4]

## 3.3 Support Vector Machine

A support vector machine (SVM) is a supervised machine learning algorithm primarily used for classification and regression tasks. It works by creating a hyperplane that best divides a dataset into classes. A support vector machine can be used to classify stock price movement as either upward or downward based on historical data and other relevant features. The application of SVMs in the financial markets is still in its infancy.[5]

Support Vector Regression (SVR) is a variant of Support Vector Machines (SVMs) that is specifically designed for regression tasks. Unlike traditional regression techniques that aim to minimize prediction errors, SVR focuses on fitting a hyperplane in such a way that it captures as many data points as possible within a predefined margin while allowing for a controlled amount of error.

In cases where the data is linearly separable, two hyperplanes are placed in such a way that they perfectly separate the data into two classes. This is called hard-margin classification. When the data is not linearly separable, soft margin classification is used, during which the hinge loss function is useful. This is defined as:

$$max(0, 1 - y_i(w^T x_i - b)) \tag{1}$$

To optimize, we minimize:

$$\lambda\|\|w\|\|^2 + \left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(w^T x_i - b))\right] \tag{2}$$

Where the parameter $\lambda > 0$ determines the correct balance between maximizing the margin and ensuring the classification accuracy of $x_i$.

Deconstructing the hinge loss function further allows us to reach:

$$\begin{aligned} \min_{w,b,\xi} \quad & ||w||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(w^T x_i - b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \\ & \forall i \in \{1, ..., n\} \end{aligned} \tag{3}$$

### 3.3.1 Kernel Trick

The SVM can handle non-linear data by transforming it into a higher-dimensional space using various kernel functions (e.g., linear, polynomial, radial basis function, and sigmoid).

For an SVM, the primal problem is to find an optimal separating hyperplane that classifies the data correctly. Derived from this primal problem is the Wolfe Dual Problem. This is given by:

$$\begin{aligned} & \max_{\alpha}\left(\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i \cdot x_j\right) \\ & \text{subject to } \alpha_i \geq 0,\ i = 1 \ldots m,\ \sum_{i=1}^{m}\alpha_i y_i = 0 \end{aligned} \tag{4}$$

The dot product $x_i * x_j$ is the main focus of this, however we want to account for higher dimension space. We then rewrite the Wolfe dual problem as:

$$\begin{aligned} & \max_{\alpha}\left(\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j K(x_i \cdot x_j)\right) \\ & \text{subject to} \quad \alpha_i \geq 0, \quad i = 1, \ldots, m, \quad \sum_{i=1}^{m}\alpha_i y_i = 0 \end{aligned} \tag{5}$$

### 3.3.2 Margin Maximization

The SVM aims to maximize the margin between the closest data points (support vectors) of the two classes.

### 3.3.3 Implementation Steps

1. Normalize the sample data using StandardScaler.
2. Divide the normalized data into a training data set and test data set.
3. Input the training data set into the SVM classifier to construct a standard SVR-based price prediction model.
4. Substitute the test data set into the prediction model to obtain the price predictions, iteratively making next-day predictions.
5. Back test the predictive model using a custom back testing algorithm, computing returns over a predetermined time period.

## 3.4 XGBoost

The initial review of the literature revealed that the Random Forest model has been widely applied in the field of stock market prediction. During the training phase, it builds several decision trees that provide a class as a response. Then, the model responds using the tree's most common response. This approach delivers strong prediction power while minimizing overfitting. But after doing additional investigation, we discovered that Random Forest has an advanced version known as XGBoost[6]. It has been demonstrated that Extreme Gradient Boosting performs better in predictive tasks, particularly stock market forecasting. XGBoost uses a gradient boosting framework, which is different from Random Forest in that each tree is generated depending on the ones that came before it since it predicts the residual or error of those trees. By penalizing complex models, XGBoost improves the loss function that results from analyzing the difference between actual and predicted values, more successfully balancing bias and variance. Different kinds of regression functions, such as quantile, poisson, or gamma, can be used to train this model. Since the primary objective in this instance was to determine correlations between the attributes and the stock price, linear regression was employed since it is helpful in predicting continuous data. In the past, regular GBM models have been used to predict stock prices as well using features such as macroeconomic variables.[7]

XGBoost algorithm is base on the tree model which is expressed as follows:

$$\hat{y}^{(t)} = \sum_{k=1}^{n} f_k(x_i) = \hat{y}^{(t-1)} + f_t(x_i) \tag{6}$$

where $\hat{y}^{(t)}$ is the predicted result of sample i after t iteration,$\hat{y}^{(t-1)}$ is the prediction of the tree after t-1 iteration, and $f_t(x_i)$ is the predicted result of i th tree.

The objective function of XGBoost is composed of the loss function and regularization:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{j=1}^{t} \Omega(f_j) \tag{7}$$

$$Obj^{(t)} = \sum_{i=1}^{n} l(y_i, y^{(t-1)}) + f_t(x_i) + \Omega(f_j) + C \tag{8}$$

Where $l(y_i, \hat{y}_i)$ is the loss function of the model, $y_i$is the true label of the i sample in the model, $\hat{y}_i$is the predicted value of the model for the i sample, and $\sum_{j=1}^{t} \Omega(f_j)$ is the regularization term in the function.

Applying Taylor expansion, define $g_i$ and $h_i$ as the first and second derivatives of the loss function, respectively:

$$Obj^{(t)} \approx \sum_{i=1}^{n} [l(y_i, y^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_j) + C \tag{9}$$

### 3.4.1 Implementation

The Root Mean Squared Error (RMSE) was selected as the error function because it emphasizes accurate predictions by penalizing greater errors more than smaller ones. With respect to the data used for training this particular model, the model trained with the price of the stock of the previous five days yielded the best result when tested using various features.

### 3.4.2 Data Discrepancies

After the data's meaning was examined, it was also discovered that the presence of outliers or "noise" in the data may affect how well this model and possibly other models that are sensitive to the data identify patterns. In this instance, the COVID-19 pandemic was a major global event that caused the stock market to behave significantly differently and volatility in the years 2020–2021. In order to determine whether the model works better, it was trained and evaluated using the entire data set from 2005 to 2023 as well as from 2005 to 2020.

## 3.5 Long Short-Term Memory

For quite some time, many machine learning algorithms have been created with the intention of making an appropriate Time Series Forecasting prediction. Artificial Neural Networks (ANN) have been the main focus of the majority of machine learning forecasting research [2]. However, it has been a lot easier said than done, until Deep Learning was created, in particular, Recurrent Neural Networks(RNN). These models are designed with the intent of capturing long-term dependencies in time series data. Compared to other models, they are able to more effectively model the relationship of historical stock prices, enabling them to capture relevant trends and patterns. Furthermore, what differentiates this model from other Neural Networks, are its memory cells that selectively retain or forget information over time.

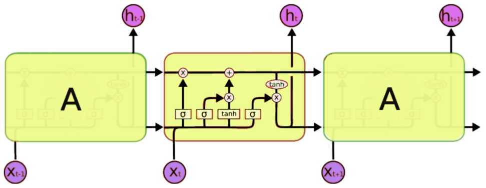


Fig 1. illustrates that in order to help each neuron create a more accurate forecast, they are fed information not only from the related feature but also from the neurons before and after them in the sequence.

An LSTM is a non-linear model, which means that it does not have a linear relationship between predictors and response variables. The model relies on this fact so sometimes when the relationship is linear will output a linear model, so to prevent this from happening before starting to go into the hidden layers, the model uses an Activation function. This Activation Function is always non-linear. The one used for the LSTM is called a rectified linear unit (ReLU). $ReLU(z) = \max(z, 0)$. The model itself uses the form:

$$f(X) = \beta_0 + \sum_{k=0}^{K} \beta_k g(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j + \sum_{s=1}^{K} u_{ks} A_{l-1,s})$$

Where $g()$ is the activation function. It then feeds into the output from the hidden layer. It also has implemented the memory cell as it feeds the information from the previous neurons to the current neuron, resulting in $f(X)$.

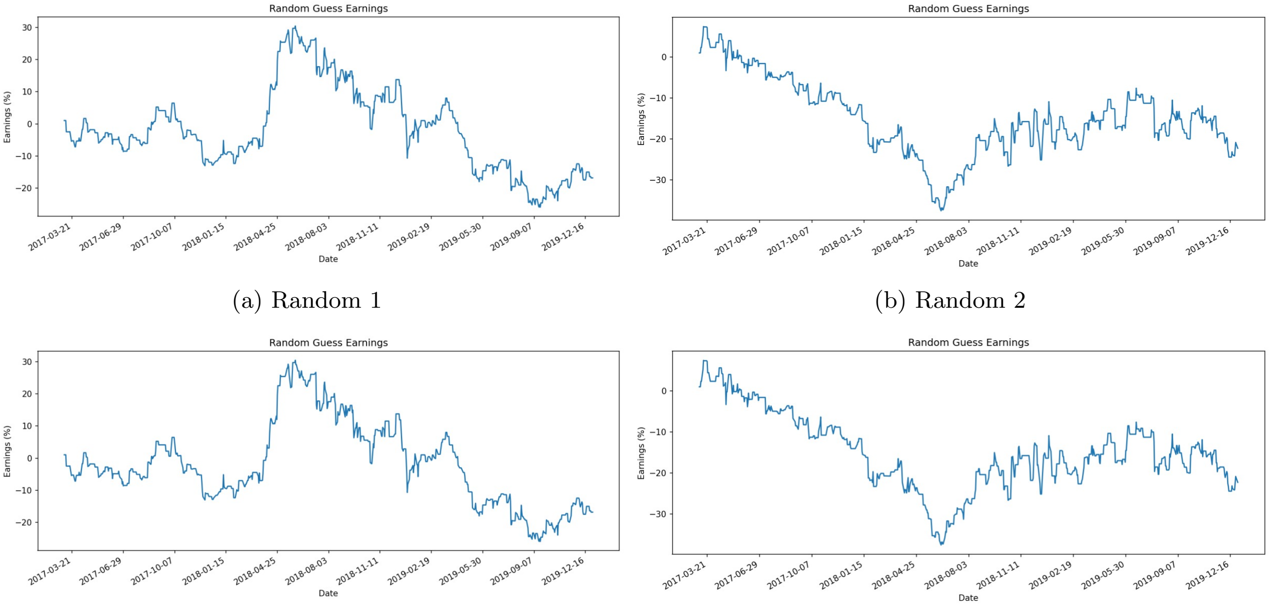
### 3.5.1 Implementation Steps

One well known algorithm in Time Series Forecasting is called window sliding. Since Recurrent Neural Networks (RNN) require large amounts of data, the idea is that window sliding ”synthetically” creates new samples to meet the requirement of data that RNNs need. So, after taking care of the data processing, there were two kinds of models that were created: 1. Using the 5 prior days of lagged closing prices, 2. Using both the 5 day lagged closing prices (Left Side of Figure 1) and the 5 prior days of lagged crude oil futures prices. (Right Side of Figure 1).

# 4 Experimental results

After developing and fine-tuning each model, we compared their predictive performance, and subsequently the quality of the selected features.

While metrics such as Mean Squared Error are commonly utilized in statistics and training to maintain proximity between predictions and actual values, this is not the most important metric in predicting stock price. Rather, what matters most is accurately predicting whether the price will rise or fall relative to the previous price. Thus, we created an algorithm for a simple trading strategy. The algorithm starts with an initial investment, and based on the expected value, it decides whether to purchase long or sell short based on whether the value was higher or lower compared to the price from the previous day. An additional algorithm was created to produce arbitrary buy and sell signals, which were subsequently incorporated into the trading algorithm and plotted. In this manner, we could assess how well the models performed in comparison to possible random behavior.



(a) Random 1

(b) Random 2

(c) Random 3

(d) Random 4

Figure 1: Random Performance

### 4.1 SVM Results

Several SVM models were trained and tested using the aforementioned data, in various forms of feature combinations. The 13 total features used were lagged stock prices ranging from 1 to 5 days prior, lagged crude oil prices ranging from 1 to 5 days prior, a 5 day simple moving average, 20 day simple moving average, and daily volume. For thorough testing, 6 different combinations of features were chosen out of these 13. In table (2) these combinations are shown with a unique key value. This key value can be used to identify models in table 1, where performances and returns of each individual model are also shown. For each feature set, a SVM was trained using linear and RBF kernels. Polynomial and other kernels were left out because their predictive performances were significantly poorer regardless of the features used. With respect to returns, it is evident that the SVM using all features (Lag1-5 (AAL,WTI), 5SMA, 20SMA, Volume) and radial basis function (RBF) kernel has the highest predictive performance.

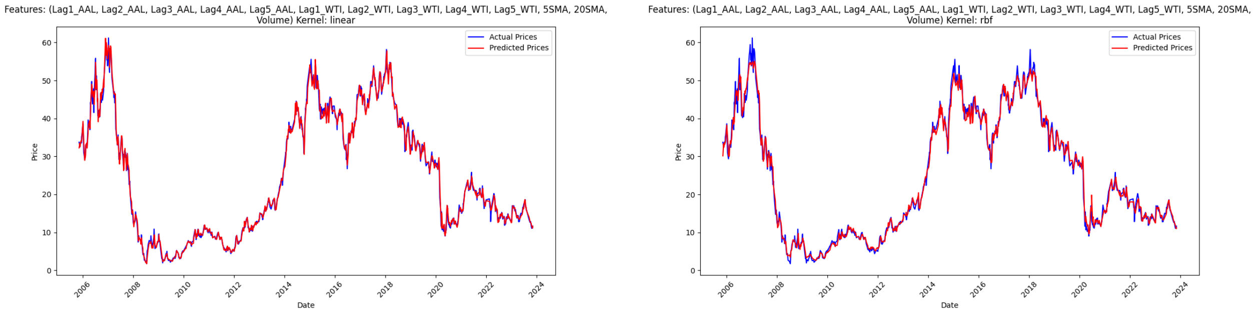
| Key | Kernel | MSE | R2 | Returns |
|---|---|---|---|---|
| 1 | linear | 1.2765 | 0.9950 | -85.71% |
| 1 | rbf | 2.2606 | 0.9911 | 64.18% |
| 2 | linear | 2.3646 | 0.9907 | -89.54% |
| 2 | rbf | 2.5009 | 0.9902 | -78.22% |
| 3 | linear | 293.9556 | -0.1524 | -44.83% |
| 3 | rbf | 229.5552 | 0.1000 | 60.91% |
| 4 | linear | 1.5849 | 0.9938 | -96.65% |
| 4 | rbf | 1.9067 | 0.9925 | -70.36% |
| 5 | linear | 1.3159 | 0.9948 | -75.92% |
| 5 | rbf | 1.7395 | 0.9932 | -53.23% |
| 6 | linear | 1.5816 | 0.9938 | -96.37% |
| 6 | rbf | 2.1230 | 0.9917 | -87.25% |

Table 1: SVM Performance

| Key | Feature Set |
|---|---|
| 1 | Lag1-5 (AAL,WTI), 5SMA, 20SMA, Volume |
| 2 | Lag1-5 (AAL) |
| 3 | Lag1-5 (WTI) |
| 4 | 5SMA, 20SMA, Volume |
| 5 | 5SMA, 20SMA, Lag1-5 (AAL) |
| 6 | 5SMA, 20SMA, Lag1-5 (WTI) |

Table 2: Feature Sets

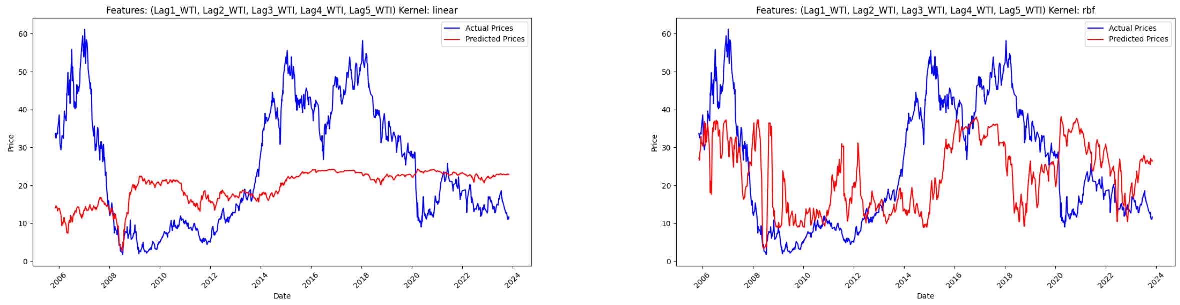
The predictive performances of 4 of the 12 SVM models are shown below, SVM (3) being the least successful, likely due to a poor feature selection. SVM (1) with the linear kernel was the most successful with respect to returns, generating around 64% returns when back tested.



(a) SVM 1 Linear

(b) SVM 1 RBF

Figure 2: SVM Performance: All Features



(a) SVM 3 Linear

(b) SVM 3 RBF

Figure 3: SVM Performance: Poor Feature Selection

## 4.2 LSTM Results

The predictive performances of the various LSTM models are pictured below. Training LSTMs are a lot more time consuming than other models, so we only displayed the 4 types that displayed the most promising results. The results that came out were really interesting, since as it can be see training the two different set of features shows its copying the movement quite well. However, it should be remarked how it had an inverse relationship the two data sets, when it was only used the price of stock it performed well with the whole data set, whereas when the price of crude oil was included it performed better before COVID-19.

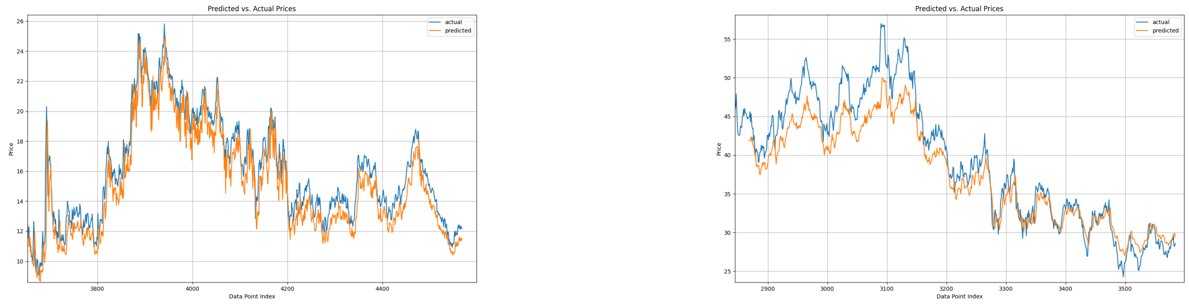


Figure 4: LSTM

| Key | RMSE | Returns |
|---|---|---|
| 1 | 0.88 | -41.92% |
| 2 | 1.71 | 3.00% |
| 3 | 2.89 | 18.23% |
| 4 | 1.60 | -18.20% |

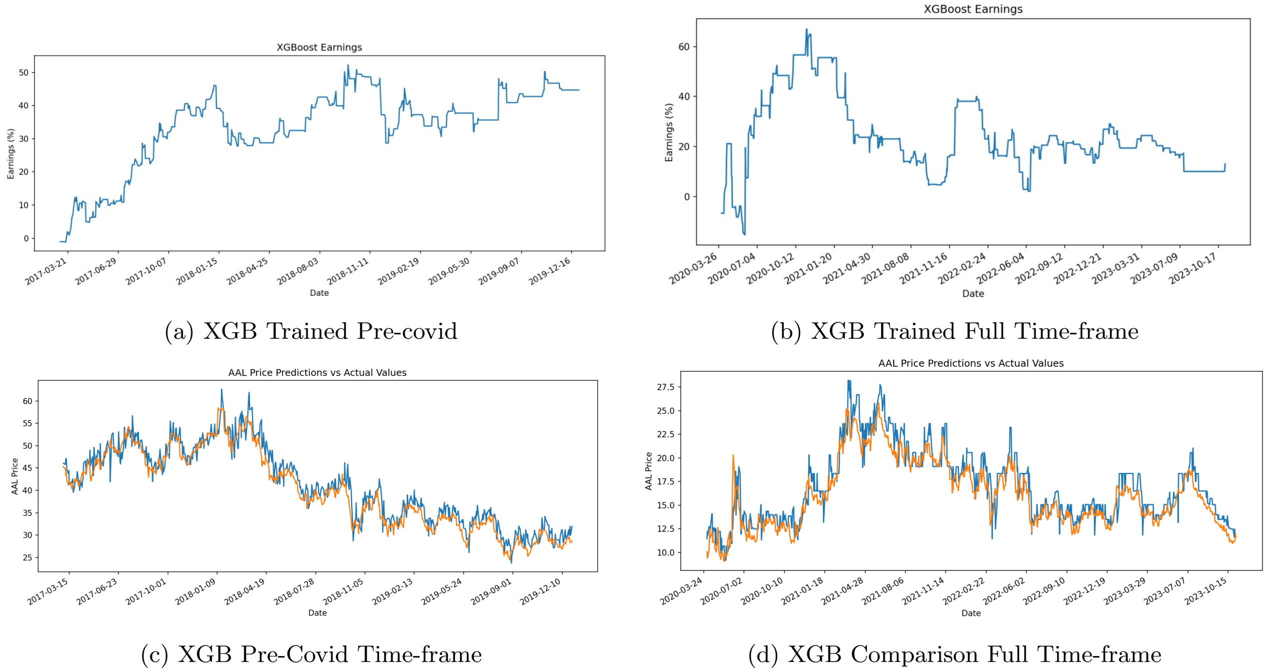
Table 3: LSTM Performance

| Key | Feature Set |
|---|---|
| 1 | Lag1-5 (AAL), Date=2019-12-31 |
| 2 | Lag1-5 (AAL), Date=2023-11-28 |
| 3 | Lag1-5 (AAL,WTI), Date=2019-12-31 |
| 4 | Lag1-5 (AAL,WTI) Date=2023-11-28 |

Table 4: Feature Sets

## 4.3 XGBoost Results

XGBoost was trained euristically using distinct feature subsets, just like the previous models. The subset made of the stock price for the previous five days was the best subset. Additionally, as previously indicated, pre-COVID-19 data produced higher outcomes for training and testing when compared to all of the data from 2005 to 2023:



(a) XGB Trained Pre-covid

(b) XGB Trained Full Time-frame

(c) XGB Pre-Covid Time-frame

(d) XGB Comparison Full Time-frame

Figure 5: XGB Performance

## 5 Conclusion and future research directions

Ultimately, given the metrics of performance for these three models (including returns), there is a clear indication that the selected variables have at least some predictive abilities for the general movement of their respective stock price. Whether this is because some of the predictors are (in nonvolatile market conditions) innately close to the response variable, we do not know. If a predictive model is a ”bad” model and generates profit, how do we define bad with respect to the underlying goal of generating a substantial profit?

In terms of future research endeavors, it is evident that there is some predictability of financial market conditions, and that technical indicators can be used to some extent to make reasonable estimates of future price movement. We believe that more complex technical indicators could be used in place of some of our features, potentially resulting in better returns.

Looking forward, the advancement of deep learning poses an intriguing frontier for enhancing stock price prediction models. While our implementations of XGBoost, LSTM, and SVM have provided valuable insights, there exists potential for refinement. Models like Convolutional Neural Networks (CNNs) could be adapted to interpret temporal sequences in stock data, capturing patterns over varying time windows. Similarly, Transformer-based architectures, which have revolutionized natural language processing, could be tailored to decode complex dependencies and long-range interactions in time-series data. These sophisticated models could assimilate a broader set of features, including high-frequency trading data, order book information, and intra-day price fluctuations, to possibly reveal subtle signals that precede significant market movements. Experimentation with hybrid models, combining elements of LSTM for sequence recognition with the attention mechanisms of Transformers, may yield a new class of algorithms that better understand the intricacies of market trends. The quest for enhanced precision in predictions will likely benefit from these deep learning approaches, furthering our capacity to anticipate market dynamics and improve investment strategies.

# References

[1] Jae Won Choi and Youngkeun Choi. A study of prediction of airline stock price through oil price with long short-term memory model. *International Journal of Advanced Computer Science and Applications*, 14(5), 2023. doi: 10.14569/ijacsa.2023.0140509.

[2] Aruna T Jyothi, Venkateshwara T Chowdary, A Abhishek, and Chetan U Anand. Exploring the benefits of support vector machines for stock prediction. *International Journal of Creative Research Thoughts (IJCRT)*, 11(5), 2023. URL `https://ijcrt.org/papers/IJCRT2305597.pdf`.

[3] Yue Qiu, Zhewei Song, and Zhensong Chen. Short-term stock trends prediction based on sentiment analysis and machine learning. *Soft Computing*, 26(5):2209–2224, 2022. doi: 10.1007/s00500-021-06602-7.

[4] C.C. Kao, C.Y. ChiangLin, and K. C. Yang. Applying three deep learning techniques to predicting stock price. *2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2022. doi: 10.1109/ieem55944.2022.9989878.

[5] Alec N. Kercheval and Yuan Zhang. Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance*, 15(8):1315–1329, 2015. doi: 10.1080/14697688.2015.1032546.

[6] Tianqi Chen and Carlos Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. doi: 10.1145/2939672.2939785.

[7] Jarrett Yeo Shan Wei and Yeo Chai Kiat. Calixboost: A stock market index predictor using gradient boosting machines ensemble. *Artificial Intelligence Trends*, 2022. doi: 10.5121/csit.2022.121009.