CBS COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

Predicting Stock Prices using Machine Learning

Model Analysis for Algorithmic Trading

Final Examination Paper - Report
Data Mining, Machine Learning and Deep Learning

Submitted by	Kitti Kresznai Anastasiya Vitaliyivna Strohonova Natalie Schober dev-almbo
Submitted on	26 May 2020
Number of pages	15
Number of characters	33,050

Abstract

This paper reviews the feasibility of the prediction of stock prices. The Efficient Market Hypothesis suggests stock price prediction is not possible. Therefore, we will analyze a series of models in terms of accuracy and efficiency to determine if we agree with this hypothesis, and if not, which model is best suited for stock price prediction. We determine that there is promise regarding predicting stock prices with the ARIMA and LSTM models, which would benefit from further investigation, further features, and potentially an ensemble approach.
Keywords: Machine Learning, Deep Learning, prediction, SVM, ARIMA, LSTM, investing

Contents

1 Introduction

In the past, stock prices were forecasted using only fundamental and technical analysis methods trying to decipher a company's performance or trends in stock price movement. However, now almost all industries are being transformed by Machine Learning and Deep Learning, which can identify these patterns much more efficiently. This transformation includes the financial industry, especially its data-driven trading and investment sector. With the introduction of machine learning-based methods, it is now possible to develop complex algorithms which consider numerous features and can provide greater accuracy. Due to the growing interest and great value of financial prediction, but the remaining difficulties and skepticism around the feasibility of stock prediction, we have chosen to investigate this topic.

1.1 Motivation

Interest and research in Machine Learning technologies for investment purposes is rapidly increasing despite the Efficient Market Hypothesis (EMH) first postulated by [Fama, 1970]. According to the EMH, the share price reflects all information available on the market which, in this theoretical framework, makes stock price prediction impossible [Fama, 1970]. This paper seeks to contribute to the evidence on the EMH question and to the evidence that predictions using Machine Learning models outperform traditional forecasting methods. To achieve this, five different stock time series and two indices as features from the US were selected. As a foundation to better understand how predictable movement in our data is, we will first review a Logistic Regression on whether prices increase or decrease, and then go on the compare the Support Vector Machine, LSTM, and ARIMA regression models on their ability to predict the close price itself.

1.2 Research Question

We will focus on the following research questions:

- What features are the most important in training models? Are the indices reflective of stock prices?
- Which model is best suited for predicting stock prices? Are long-term trends in stock prices still be predictable even with interruptions such as Covid-19?

With these questions, we hope to develop a foundation for data-driven investment decisions, to help guide decisions and identify limitations of the applications of machine learning algorithms. By first reviewing the existing literature, we identify a route for answering the above questions. Analyzing the structure and features of our data allows us to better apply the models, and later discuss their results and address our research questions, as well as identify any questions that remain or potential improvements to be made.

2 Related Literature

There has been much recent literature in stock price prediction, in which many machine learning methods have been applied to primarily predict short-term, day-to-day prices. However, the fact that stock data is non-linear reduces our choice of algorithms, as non-linear methods such as Deep Learning methods are more successful [Nikou et al., 2019].

Roughly, one can classify the approaches into Machine Learning, Deep Learning, Ensemble-based approaches, and traditional Time Series Forecasting methods. A different axis of classification would be to differentiate between the use of technical-based features or technical features in combination with financial news which requires methodology based on Natural Language Processing. Finally, there is the distinction between classification or regression methods: either the target variable is categorical and only an increase or a decrease is the goal of prediction, or it is numerical which aims for the prediction of exact stock prices.

Under the Machine Learning category can fall approaches such as Regression techniques or Support Vector Machines, which are commonly used within an Ensemble method as in [Zhou et al., 2019]. Ensemble algorithms are aimed at reducing variance and overfitting and at achieving performance improvements over a single classifier/regressor [Zhou et al., 2019]. For their ensemble model of Logistic Regression and Gradient Boosted Decision Trees, [Zhou et al., 2019] find that compared to the individual models, the ensemble performs better.

Logistic Regression is limited due to its categorical nature, therefore Support Vector Machines can fill this gap. Varying approaches are possible within classification and regression. [Henrique et al., 2018] compare SVMs based on kernel choices as well as data frequencies and find that SVMs are viable options when the data has a daily frequency. [Ballings et al., 2015] and [Gerlein et al., 2016] have implemented SVM along with other classifiers with good performance. The former also evaluated ensemble approaches which provided them with good results compared to single classifiers [Ballings et al., 2015]. Another promising approach is described in [Xiao et al., 2014] who builds various Deep Learning models with integrated SVMs. In general, Deep Learning holds a great advantage over conventional Machine Learning algorithms, as its highly complex nonlinear relationship can fully describe the complex influencing factors such as stock market data [Hu et al., 2021]. Its success in other Machine Learning tasks warrants its feasibility to predict stocks [Hu et al., 2021]. Deep Learning approaches can be divided into four groups following [Hu et al., 2021]: approaches using Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long-Term-Short-Term Memory (LSTM), or Deep Neural Networks (DNNs).

Frequently, Deep Learning approaches are compared to Machine Learning approaches or among each other. For instance, [Maqsood et al., 2020] do the former and compares a CNN model to a Linear Regression and Support Vector Machines, whereas [Hoseinzade and Haratizadeh, 2019] propose a 2D-CNN and a 3D-CNN model using technical indicators. Another common pipeline of Deep Learning models is the combination of several. For example, [Liu et al., 2018] implemented a joint model of a TransE algorithm, a CNN, and an LSTM for the prediction of changes in stock prices. Furthermore, [Wen et al., 2020] use a PCA-LSTM model and compare this to a CNN, MLP, and Moving Average model with the result of the PCA-LSTM model performing the best out of the four.

The meta-study of Deep Learning approaches authored by [Hu et al., 2021] has found that LSTM approaches are among the most common approaches, even in combination with other types of Neural networks. In addition, the majority of papers cited used technical analysis, with the closing price as the most common target variable [Hu et al., 2021]. Some of the most common metrics chosen were the RMSE, MAE, MAPE, MSE, and the accuracy [Hu et al., 2021]. A comparison of the papers according to RMSEs revealed that DNNs achieved the smallest RMSEs among the selected papers [Hu et al., 2021]. In terms of the MAPE, DNN and LSTM models achieved the best performance [Hu et al., 2021]. The papers with the lowest MAE were using CNN and LSTM models, the paper with the lowest MSE used an LSTM model [Hu et al., 2021]. The paper with the highest accuracy also used an LSTM model.

3 Methodology

3.1 The Data Set and Pipeline

The data has been obtained from Yahoo!Finance through the yahoo-fin API. In particular, it comprises historical stock data of five different stocks and the S&P 500 and the Dow Jones indices. In addition, some indicators have been calculated to complete the features for the intended analysis. Originally, the data came from two different datasets downloaded from Yahoo! Finance. Through transformation, they have been joined by their date.

We have chosen to limit the dataset to five stocks to limit the scope and to focus on optimizing the predictive models. Five different stocks from different industries, but all from the United States, have been chosen to ensure comparability. For the banking and financial industry, the Bank of America (BAC) stock is representative. From the health and pharmaceutical industry, Pfizer (PFE) has been selected. The technology and information industry are represented by Google's parent company Alphabet (GOOG). The final two are Boeing Company (BA) for the industrial manufacturing industry and Mondelez International Inc (MDLZ) from consumer goods. These industries have been chosen as they play a major role in the US American economy, and the specific companies are among the top of each industry. The S&P 500 and the Dow Jones have been selected because they are the most important indices for the American economy and are an overall indicator of the American economic situation.

The stock and indices data have a daily frequency. The time series starts in 2017 and ends in 2021. In total, the dataset has 5.520 observations. Our features were either downloaded or calculated. The downloaded features include the "volume" and "open" series which refer to the stock volume in the market and the opening prices for the day. The features we calculated manually were chosen because they are among the most common indicators for technical stock analysis. In particular, these include the 3- day and 5- day moving averages and the weekday. We chose to calculate these because they were indirectly available through the data we already collected from Yahoo!Finance. The moving averages were calculated using the mean of the three/five previous closing stock prices. The weekday was extracted from the date.

3.2 Exploratory Data Analysis

For the Exploratory Data Analysis, we first investigated the closing price time series of all five individual stocks. It was discovered that the stock prices were on different scales (see Figure 1), therefore the closing price needed scaling as part of our pre-processing steps. After our transformations, the time series appeared like this:

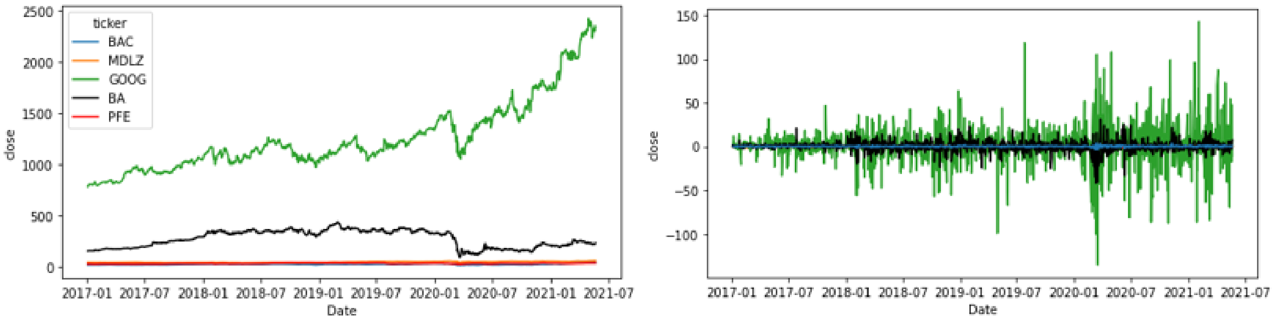

Figure 1: Stock price movement of five stocks before and after scaling

To better understand the behaviour of the stocks within each sample, we investigated the average fluctuations occurring. We found that for 2,799 of our samples the closing price was higher than the opening price, with 2,674 stocks falling in price, and 47 with no change. The average stock price difference between open and close was –0.199, with the maximum decrease being -$72.61 and the largest growth $76.96. These values give us an overview of standard behaviour among the samples.

As part of our Exploratory Data Analysis, we also built a Random Forest to understand which features are the most relevant in training the model. A Random Forest is a potent Machine Learning algorithm that consists of a set of Decision Trees that have been trained on differing random subsets of the data. It selects the most probable result based on all Decision Trees, typically using the bagging ensemble method [Géron, 2019]. Each Decision Tree begins at the so-called root node, which is connected to branches leading to decision nodes [Géron, 2019]. Here, questions are asked leading to either more decision nodes or a leaf node, which represents the output of the model [Géron, 2019].

Among the top features found were “volume” returning a score of about 0.277 followed by “mv_avg_5” at 0.26, “mv_avg_3” at 0.17, “open” at 0.13, and “Date” at 0.09. Five other features scored below 0.03, and the remaining features scored 0.

3.3 Data Preprocessing

After an initial analysis, we confirmed that there were no missing values in the dataset, however noticed that there were several pre-processing steps necessary before we could implement PCA. To avoid biased forecasts, it was necessary to check each stock series for stationarity. This was done using a hypothesis test called Augmented Dickey Fuller Test. We extracted the p-values to test for stationarity. If it was larger than the 0.05 threshold, the data is not stationary and needs to be differenced to become stationary. This was the case for all of our time series.

First, we had to normalize the data using sklearn's MinMaxScaler with a feature range of (-1,1) to ensure the features had a normal distribution. We used an individual scaler for the X and y sets, to avoid the mixing of their minimum and maximum values. This was a good choice because the data is not linear, and therefore standardization would bias the forecasts. Then, we split the data: 80 % of the initial data set is used as the training set, 10% as validation, and the remaining 10% as test sets. By doing this before the PCA, we ensure that any models we fit to the data are not fit to the validation and test data as well which would be data leakage.

Kernel PCA, an unsupervised learning algorithm and extension of PCA which allows for dimensionality reduction through non-linear projections, was utilized to better fit a model [Géron, 2019]. Its basic idea is the use of a non-linear kernel function in place of the dot product, which means PCA is conducted in a high-dimensional space that is non-linear [Schölkopf et al., 1997]. The process is as follows: At first, the eigenvalues and eigenvectors satisfying the covariance matrix are calculated, and the eigenvalue problem is solved for non-zero Eigenvalues based on the column vectors [Schölkopf et al., 1997]. The solutions are, then, normalized and the projections of the image of a test point are calculated [Schölkopf et al., 1997]. In this step, the kernel functions are used to calculate the dot product [Schölkopf et al., 1997].

After we fit the training data on the model obtained through the Kernel PCA, all three feature sets were transformed using the instantiated PCA. Following this, new data frames were created of the transformed training, validation, and test sets. To these, we added the closing price (the target variable), the corresponding dates, and tickers to be able to interpret the results of our chosen models.

4 Modelling

4.1 Logistic Regression

Our first implementation is a Logistic Regression Model. It classifies the target variable based on the feature vectors according to two categories encoded with a label which makes it a binary classifier [Jung, 2018]. It is based on the conditional probability that the class of an instance is one of the labels given its corresponding feature values [Zhou et al., 2019]. The output of the weighted sum is passed through a sigmoid function to obtain the probability of the instance to be in a certain class. When training data is fit using Maximum Likelihood estimation, the weights "w" are obtained [Zhou et al., 2019]. In addition, the logistic regression can be regularized by adding a penalty in the form of L1 or L2 regularization. The hyperparameter "c" defines the regularization strength in this case [Zhou et al., 2019].

The categorical variable being predicted on is "Price change", which shows an increase as a "1" and decrease as a "0". The price change variable is the difference between the current and the previous value of the stock price replaced by the resulting categorical values as described above. In addition, we shift the values back by one row to represent the price change between the current and the future variable. Another necessary step is the removal of the final rows which contain missing values. Then, we can build the Grid search pipeline where the parameter "C" is optimized. We fit the built grid search pipeline on the training data frame containing the features and the target variable ("Price change") and calculate predictions on the test set. The model is evaluated through the confusion matrix, precision, recall, the f1 – Score, and support.

4.2 Support Vector Machine

For the Support Vector Machine (SVM) we used the Support Vector Regression (SVR) model and the scaled, pre-processed data. The SVR creates a regression that, unlike a simple linear regression, optimizes the number of points within a certain boundary, rather than minimizing the distance to surrounding points [Géron, 2019]. SVM models are ϵ-insensitive, meaning that errors within the ϵ-tube are ignored and additional training values in this range do not lead to improvement in the model [Murphy, 2012].

We applied RandomizedSearchCV to determine the best kernel, C, epsilon, and gamma values for the model. RandomizedSearchCV best suited our need as it allows for a higher number of parameters in the search without severely slowing the fitting of the model. The kernel establishes the form of the model. The C parameter determines the regularization or number of margin violations accepted by the model, epsilon determines the area with no penalty, and gamma determines the range of points that influence the decision boundary [Géron, 2019]. With both C and gamma, higher values can lead to overfitting. Due to C's strong interaction with the kernel parameters, with a larger gamma value such as 5, a small C is needed and vice versa [Murphy, 2012]. Of the options presented, the randomized search selected 'kernel': 'sigmoid', 'gamma': 0.001, 'epsilon': 0.01, 'C': 1.

4.3 Random Forest Tree

To understand the most relevant features, we implemented a Random Forest Regressor with a maximum depth of thirty. This analysis required splitting the ticker column into five separate columns as dummy variables. After fitting the training data, we evaluated the R^2 score and used the validation set to further optimize the parameters for better performance with the test set. In this case, the maximum depth appeared to be too high and caused the model to overfit to the training data. Decreasing it to three allowed the model to better predict on unseen data, although the performance of the Random Forest was still very poor and only a slight improvement on a simple Decision Tree. Altering the other default parameters only negatively impacted the model.

4.4 Autoregressive Integrated Moving Average

As described earlier, among the models we implement is an ARIMA model to eliminate the possibility that the stock price is a function of its previous values. Autoregressive moving average models are "a combination of past values of the variable" [Hyndman and Athanasopoulos, 2018]. The order of each ARIMA model consists of three parameters: p - the order of the autoregressive part (AR), q - the order of the moving average (MA), and d- the order of differencing. When modeling, parameter estimation is done through Maximum Likelihood estimation. This algorithm maximizes the probability that a given sample of data is obtained from the population. Finally, a model is selected by minimizing an Information criterion such as the Akaike's Information Criterion (AIC), the Bayesian Information Criterion (BIC), or the corrected AIC [Hyndman and Athanasopoulos, 2018].

We compute ARIMA forecasts for each stock series. Our process consists of splitting the data into training and test sets. Then, we fit an ARIMA model using an automatic ARIMA function that searches for the optimal p, q, and d parameters. Our forecasts are calculated with having the long-term and short-term perspectives in mind. For the long-term perspective, we use the length of the full test set. The short-term perspective is done with a period number of five. Finally, for each forecast perspective and each stock price series, the mean absolute error (MAE) and the root mean squared error (RMSE) are computed. We anticipate that the short-term ARIMA forecasts perform better than the long-term forecasts. To find a well-performing model for long-term forecasts, we investigate other Machine Learning techniques such as the LSTM and the SVR.

4.5 Long Short-Term Memory

A model that works well for both the long-term and short-term is the LSTM. LSTM is a type of Recurrent Neural Networks (RNN) used in the field of supervised Deep Learning. Contrary to traditional neural networks, RNNs allow information to persist through loops. The persistence of information is crucial for stock price prediction, as past information can be important for predicting future prices. The LSTM model is trained using backpropagation and prevents the vanishing gradient problem. An LSTM consists of five components, namely cell and hidden state, input, forget and output gate. These gates control what information is stored, persisted and output from a recurrent cell [Charu, 2018] [Jung, 2018].

The LSTM model is instantiated calling keras.Input and layers are added to define the LSTM's forward pass. For the LSTM blocks, our Neural Network uses the tanh activation function whose values range from -1 to 1. The model is then trained with a certain optimizer, batch size, and number of epochs. Whereas the number of epochs determines how many times the weights in the network change, the batch size defines the data samples used in one iteration. As the choice of these hyperparameters is important to avoid over- and underfitting, we have tried various ones (see for example Figure 3). To further improve the accuracy of the model, time lags are introduced. Accounting for time lags is important, as there is often a lag between an event and its subsequent stock price change. With the introduced time lags and tuned hyperparameters, the respective loss values are returned.

As the model is now trained and fitted, its performance can be evaluated. To calculate the error scores, the target variables are scaled back to the original representation so the performance measure and the original data have the same unit.

5 Results

What features are the most important in training models? Are the indices reflective of stock prices?

In comparing the Decision Tree and Random Forest regressors, the Random Forest appeared to better generalize to fit unseen data. However, both models were not able to outperform a horizontal line of the mean, with the R2 of the test data of the Random Forest only reaching -0.069. As described before, the top features were found to be "volume", "mv_avg_5", "mv_avg_3", "open", and "Date". The remaining features played little or no role in the training of the model. Hence, the stock prices seem to be more dependent on their individual features of the stocks than on any of the indices we considered.

Which model is best suited for predicting stock prices? Are long-term trends in stock prices still be predictable, even with interruptions such as Covid-19?

Logistic Regression

Based on the results of the confusion matrix in Figure 2, the logistic regression predicts a price decrease in about half of the cases correctly. In the other half, it faultily predicted an increase, when there was, in fact, a decrease. As far as the increases are concerned, the model was only able to recognize it 8 times out of the 272 instances. The evaluation metrics confirm these results, as the precision is around 50% in both cases, while the value of recall is very low when it comes to correctly predict increasing changes, as there are several false negatives present. The overall accuracy of the model is around 50%.

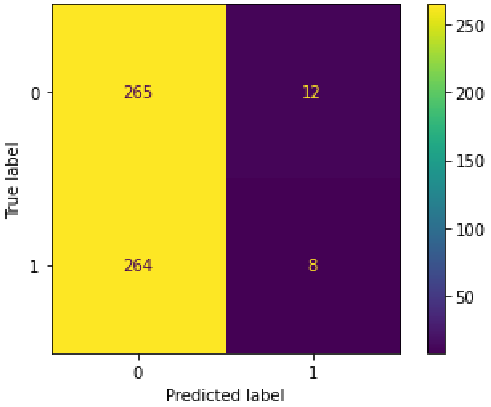

Figure 2: Confusion matrix of the Logistic Regression model

Overall, it can be stated that the logistic regression model is not suitable for predicting price changes in our dataset. There are approximately the same number of instances for increasing and decreasing price changes in the dataset, hence the model's poor performance cannot be explained by class imbalance.

Support Vector Regression

The SVR model performs only slightly better than a simple mean of the close price, with an R^2 of about 0.0193. Upon taking a closer look at the predicted values compared to the target values, 379 out of 526 predictions were higher than the actual close price. This bias may be due to the relatively higher price of the Google stock influencing the model. Although the mean absolute error was only 0.046, this does not necessarily suggest good model performance since the values were scaled down. The larger RMSE of 0.116 shows that there are some large errors within the predictions. Overall, the model is not able to accurately predict the closing stock prices given the independent variables provided.

Autoregressive Integrated Moving Average

The results from the five ARIMA models confirm our early hypothesis. As we have seen from the very low RMSE of the models, ARIMA predicts the stock price for the next day with high accuracy. In other words, ARIMA is a good model for short-term forecasting, but it quickly reaches its limits when predicting stock prices in the long run. A comparison of the short-term and long-term forecast errors shows that long-term errors are larger than short-term errors. For this reason, predictions with longer time forecast periods should be made using other forecast methods. However, the goodness of this model gives evidence that stock prices can be forecasted to some degree, and that Efficient Market Hypothesis (EMH) might not be true after all.

Long Short-Term Memory

From the performance measures and the visualizations, it becomes apparent that the LSTM can be a competent model for predicting stock prices. The results score shows a test loss of about 0.17 indicating a good fit, while the MSE of 3125 is high indicating underfitting.

Figure 3 displays model loss on training and validation set. Starting from the number of epochs of about 30, the plots decrease and the performance of the model on the sets becomes very similar. This can indicate stopping training at this epoch. Figure 4 also visualizes the performance of the model by plotting actual vs. predicted stock prices. We can see that the LSTM model fitted both the training and test set quite well.

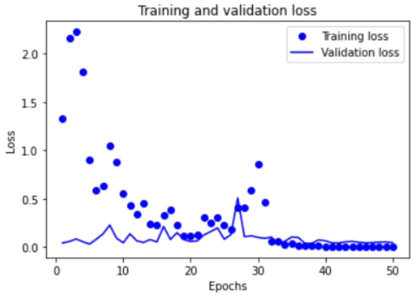

Figure 3: Loss on training and validation set

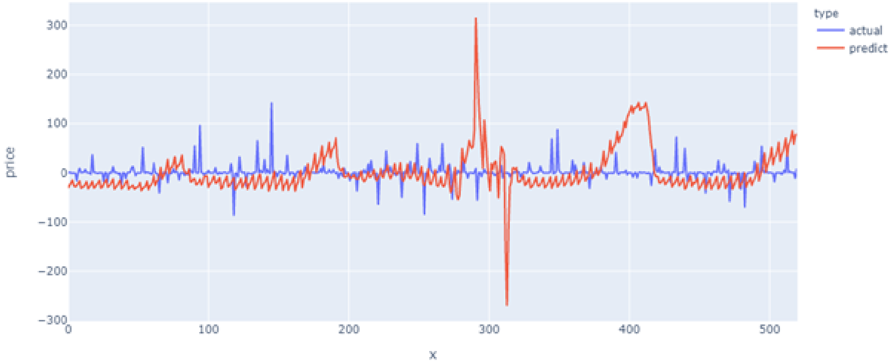

Figure 4: Actual vs. predicted stock prices using LSTM

Running time and model complexity After looking at the performance of the different models from the perspective of the evaluation metrics, this passage will compare them by running time and complexity. The Logistic Regression is one of the fastest out of all models, and this fact is closely related to its relatively simple structure. Compared to this, the SVR model takes more than three times as much to run, although the number of features is the same in both cases. For choosing the hyperparameters, SVR uses Randomized Search, while in the Logistic Regression model, the applied method is Grid Search, which is the more excessive one out of the two. The ARIMA model's running time is around 100 seconds, but in this case, the comparison is not so straightforward, as it calculates forecasts for each stock separately. The LSTM model is by far the most complex one because it is a type of recurrent Neural Network that utilizes backpropagation. Its running time is 1147 seconds.

6 Discussion

We observed that the features volume, the three-day moving average, and the five-day moving average are the most significant features for predicting stock prices, of those we used. Therefore, investors should pay attention to these features when making data-driven investment decisions. We would also recommend including volume and a moving average measure in machine learning models, as they are readily accessible features.

In the modeling process, we predicted both stock prices and stock price changes. For the stock price changes, the Logistic Regression model performed poorly, predicting only 50% of the price changes correctly. We therefore believe that it may be best to pursue another method for classification, such as a boosted classifier. However, we decided to step away from classification to better address our research questions of predicting the prices themselves.

For predicting the stock prices themselves, the best results were obtained by the ARIMA and LSTM models. In the short term, the ARIMA model produced good forecasts with little computational complexity. While we recommend using the ARIMA model for the short-term, the LSTM is suitable for the long-term despite its high computational costs. This implies for investors the usage of ARIMA and LSTM models to improve their ability of stock price prediction. Following these models gives investors a good trading strategy, both in the long-term and short-term. Based on this result, it can be recommended that financial service companies should continue their research into algorithmic trading models in the direction of Deep Learning and Neural Networks.

One limitation we faced was regarding the features we could access: While it is a great additional source of information and features, company financial information is only available on either a quarterly or yearly basis, greatly limiting its potential relationship to forecasting daily stock prices. Text mining of news articles, regular company announcements, or ad-hoc communication can also help in the prediction of stock prices, but due to time constraints, we did not explore this route. Another obstacle was that the underlying data set contains share prices from 2017 to the present, i.e. it includes the Covid-19 period. During the Covid-19 peak periods, stocks were subject to high volatility and economic risk, which makes predicting stock prices particularly difficult.

7 Conclusion

In analyzing the feasibility of algorithmic data-driven investment decision-making, we concluded that even with limited features we could achieve a degree of accuracy with the LSTM and ARIMA models which shows some promise in regards to disproving the Efficient Market Hypothesis. Although we understand the limitations of our model, and the precision required for true stock prediction for investment, we believe that our analysis provides a foundation for future optimization. With additional features providing information external to the stocks themselves, we could potentially account for more of the variation.

Further research is needed in this area to understand if Machine Learning or Deep Learning models can be used for large-scale investment decisions. Our analysis has shown some promising results for Deep Learning models, but even a single model might not have the predictive power and accuracy needed to be trusted enough in the financial sector. For this reason, ensemble approaches in the literature are becoming more and more popular, and an ensemble with a Deep Learning approach could be an improvement. In our future work, we hope to discover further features for prediction, test the ensemble methods which are prevalent in the literature, and apply our model to stocks outside of those in the training set to test if it can be generalized. In selecting stocks from various industries, we hope to have laid the foundation to make the models more widely applicable.

References

[Ballings et al., 2015] Ballings, M., Van Den Poel, D., Hespeels, N., and Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20):7046–7056.

[Charu, 2018] Charu, C. A. (2018). *Neural Networks and Deep Learning.*

[Fama, 1970] Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2).

[Gerlein et al., 2016] Gerlein, E. A., McGinnity, M., Belatreche, A., and Coleman, S. (2016). Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, 54:193–207.

[Géron, 2019] Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O'Reilly Media.

[Henrique et al., 2018] Henrique, B. M., Sobreiro, V. A., and Kimura, H. (2018). Stock price prediction using support vector regression on daily and up to the minute prices. *Journal of Finance and Data Science*, 4(3):183–201.

[Hoseinzade and Haratizadeh, 2019] Hoseinzade, E. and Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129:273–285.

[Hu et al., 2021] Hu, Z., Zhao, Y., and Khushi, M. (2021). A survey of forex and stock price prediction using deep learning.

[Hyndman and Athanasopoulos, 2018] Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice.* OTexts.

[Jung, 2018] Jung, A. (2018). Machine learning: Basic principles. *arXiv preprint arXiv:1805.05052.*

[Liu et al., 2018] Liu, Y., Zeng, Q., Yang, H., and Carrio, A. (2018). Stock price movement prediction from financial news with deep learning and knowledge graph embedding. In

Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 11016 LNAI, pages 102–113. Springer Verlag.

[Maqsood et al., 2020] Maqsood, H., Mehmood, I., Maqsood, M., Yasir, M., Afzal, S., Aadil, F., Selim, M. M., and Muhammad, K. (2020). A local and global event sentiment based efficient stock exchange forecasting using deep learning. *International Journal of Information Management*, 50:432–451.

[Murphy, 2012] Murphy, K. P. (2012). *Machine learning: a probabilistic perspective.* MIT press.

[Nikou et al., 2019] Nikou, M., Mansourfar, G., and Bagherzadeh, J. (2019). Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, 26(4):164–174.

[Schölkopf et al., 1997] Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer.

[Wen et al., 2020] Wen, Y., Lin, P., and Nie, X. (2020). Research of stock price prediction based on PCA-LSTM model. In *IOP Conference Series: Materials Science and Engineering*, volume 790, page 012109. Institute of Physics Publishing.

[Xiao et al., 2014] Xiao, Y., Xiao, J., Lu, F., and Wang, S. (2014). Ensemble ANNs-PSO-GA Approach for Day-ahead Stock E-exchange Prices Forecasting. *International Journal of Computational Intelligence Systems*, 7(2):272–290.

[Zhou et al., 2019] Zhou, F., Zhang, Q., Sornette, D., and Jiang, L. (2019). Cascading logistic regression onto gradient boosted decision trees for forecasting and trading stock indices. *Applied Soft Computing Journal*, 84:105747.