Enhancing Stock Price Prediction: A Multi-Model Approach with Exponentially Weighted Moving Averages

Hannah E. Hollenbeck*

Westlake High School, Austin, Texas, U.S. 78746.

KEYWORDS. Linear regression, neural network, yfinance, algorithms, trading predictions.

BRIEFS. Multiple algorithmic models for stock price prediction

ABSTRACT. Artificial intelligence (AI) is increasingly being used to predict the stock market. This paper aims to make stock price predictions through using multiple models in conjunction. Two models were created in this paper: a linear regression model and a neural network model both trained on stock prices obtained from yfinance. These models were chosen for their simplicity and prevalence. Despite the neural network having lower error, through Multiplicative Weight Update and Exponentially Weighted Moving Average (EWMA), it was found that no model was explicitly preferred by the algorithm. While the neural network had higher overall performance, in a portion of trials, the linear regression model performed better. Using these two models in conversation with each other with the EWMA algorithm allows for the strengths of both models to be utilized together instead of only relying on one model's strengths. Simple stock market bots utilizing the EWMA algorithm for predictions successfully demonstrated this. Future work applying AI to the stock market should consider using multiple models together.

INTRODUCTION.

The stock market system is a critical facet of the global economy with trillions of dollars traded yearly. In 2023, the total global market capitalization was reported to be over $109 trillion [1]. The stock market provides a platform for companies and investors of any experience to amass capital from this large pool of value; however, the stock market can be difficult to accurately predict prices. A plethora of factors affect a stock's price. From company performance to global political events, a stock price can change drastically with little warning and human managers have been struggling to meet these price variations. From 2012 to 2022, 82% of fund managers traded below the S&P 500 [2]. It is clear that the common current techniques for navigating the stock market are not sufficient. When applied appropriately in industries, artificial intelligence has been shown to provide significant improvements in that field. From security and healthcare to agriculture and beyond, AI has made great advances. Stock price prediction is a subject with large potential with AI. Regardless of the financial situation of those investing in the stock market, success in the market can greatly affect people's lives; however, numerous factors affect a stock's price. Accurate predictions require a thorough understanding of the market, the stock, and contextual detail. The goal of this paper is to reduce the complexity of the stock prediction problem through the use of AI models. With this reduced complexity, the obstacles to financial success will hopefully be mitigated and the stock market can be used successfully by anyone.

This problem is not new and has been approached in many different ways. Kara et al. [3] utilized finely tuned Artificial Neural Networks (ANN) to make predictions; Velay and Daniel [4] used Convolutional Neural Networks (CNN) to find stock price patterns; Guo et al. [5] employed Support Vector Regression (SVR) to dynamically predict prices; and Nelson et al. [6] used Long Short-Term Memory (LSTM) for predictions. While other models have been used, these are some of the most common approaches. Researchers have also attempted to look beyond the quantitative stock data and include situational analysis in their predictions. Thesia et al. [7] consider the current market in predictions, and Lin et al. [8] factor in textual information like daily news titles. These newer studies, which holistically approach the problem, report increased success in stock prediction compared with other purely quantitative methods; however, they also report increased difficulty in sourcing their textual data for analysis. Where the current paper's approach differs from these past papers is the use of multiple models in conversation to make predictions. Models and their predictions are put through a simple Multiplicative Weight Update (MWU) algorithm, which makes predictions based on weights that are updated based on previous performance to determine the final prediction. A technical explanation of the MWU algorithm used is provided in the Materials and Methods section of this paper, and Grigoriadis and Khachiyan [9] provide a more advanced look at the algorithm and its versions. This approach assumes that no one model will always be the best option for stock prediction. The hypothesis tested is that an algorithm making a stock price prediction based on multiple models will be more accurate than just the highest performing model used in the algorithm.

MATERIALS AND METHODS.

Dataset. The training data used for this study was sourced from an open-source library, yfinance [10]. From the large quantity of data included in yfinance, the stock data of Tesla, Inc. (TSLA) was imported for the past five years. From there, all data about the stock other than the open price each day was discarded. Next, the X data was structured. The open prices were organized into a matrix with 1255 rows and 3 columns. The 3 columns consisted of subsequent open prices, with each row increasing the starting day (Table 1).

Table 1. The structure of the dependent data values. Feature refers to the input attributes for training. Iteration refers to each instance of input, output, and weight update. Day (N) refers to the open price on the Nth day.

	Feature 1	**Feature 2**	**Feature 3**
Iteration 1	Day 1	Day 2	Day 3
Iteration 2	Day 2	Day 3	Day 4
Iteration 3	Day 3	Day 4	Day 5
…	…	…	…
Iteration 1256	Day 1256	Day 1257	Day 1258

The Y data was a matrix with 1255 rows. Starting with the fourth open price, the open price for each subsequent row increased by one day (Table 2). These matrices were then split into the models' X and Y training and testing matrices using the train_test_split(X,Y) method. Y training values were used to calculate error and Y testing values were used to evaluate the model. Per the yfinance legal disclaimer, the data should only be used for educational and research purposes. Should these models be commercialized, a different source for the data must be used.

Table 2. The structure of the independent data values. Target variable refers to the independent or Y value that the models are attempting to compute. Iteration refers to each instance of input, output, and weight update. Day (N) refers to the open price on the Nth day.

	Target variable
Iteration 1	Day 4
Iteration 2	Day 5
Iteration 3	Day 6
…	…
Iteration 1256	Day 1259

Algorithmic Modeling. The first model used in this study was a linear regression model which attempts to establish a linear relationship between independent and dependent variables. This model was chosen for its simplicity and how common it is, making it more accessible to a wider variety of people. The use of this model assumes that the relationship between three consecutive open prices and the fourth price is linear. The coefficients of the linear model are determined by finding the optimal linear hyperplane through the data that minimizes the sum of the squared residuals. After training, the model's performance is determined through the mean percent error.

The second model used was a neural network, which utilizes neurons organized into layers to predict the open price. The model was chosen for the same reasons as the linear regression model. This model had four layers with an input layer for the three consecutive open prices, the first hidden layer with one hundred neurons, the second hidden layer with fifty neurons, and the output layer, which is the model's prediction. The model uses the Rectified Linear Unit (ReLU) function as the activation function. The training algorithm used for the model was Adaptive Moment Estimation (Adam), which adapts the parameters' learning rates during the training process. After one pass through the training data, one epoch has been completed. The training for this model had a maximum of five hundred epochs for the model to converge; as the model is a stochastic solver, the "max_iter" value refers to the number of epochs and not the number of gradient steps taken. The neural network's performance was also measured through mean percent error.

After the models were trained and tested, a multiplicative weight update algorithm (MWU) was used. MWU determined the absolute value of the error of the models' predictions and then multiplied the weight of the predictions by ½ to the power of the absolute error. With this method, the weights will only ever decrease or stay the same if the prediction is perfect. To prevent the weights of the model from decreasing to the point where they become negligible and the computer performing the algorithm fails, the new weights of the models were normalized by dividing each by the sum of the new weights. This ensured that the weights always added up to the initial sum of the weights after each iteration. The model with higher error had its weights multiplied by a smaller number than the model with lower error, meaning the less accurate a model is comparatively, the less its prediction was taken into account. This process was repeated for each prediction in the testing dataset to complete the MWU algorithm. The MWU algorithm was then repeated 20 times on the two models after training them again in order to find any trends in the models' relative performance.

In an MWU algorithm, the absolute error for each prediction has an equal impact on the weight throughout the algorithm. As the models were trained and tested on open prices from the past five years, error from potentially five years ago might have the same impact on the weights as error from more recently. To account for the existence of seasonal trends in open prices, the MWU algorithm was modified to only consider more recent absolute error of the models and to place greater emphasis based on how recent it was. Called an Exponentially Weighted Moving Average (EWMA), this algorithm is a common variant of MWU. Arbitrarily, the emphasis was chosen to correlate to (21-n)^2 where n is the number of days since the prediction and n > 0– represented as [1, 4, 9, 16, 25, 36…361, 400]. The weights were then calculated as the day's emphasis multiplied by 1⁄2 to the power of the absolute error.

The next goal of this study was to demonstrate how the use of multiple models with weighted predictions improves the returns when investing in the stock market. Three stock market bots were developed that simulated trading. One traded based off of the linear regression model's predictions, one traded off of the neural network's predictions, and the last used the EWMA algorithm on the two models to determine predictions. All three bots used the same logic and had the same starting budget of $10,000; the only difference was the source of the predictions. The models the bots used were trained over five years of stock data and then set to trade for one year. The bot's final capital–remaining budget plus value held in stocks was then recorded and averaged over 30 trials.

RESULTS.

Predicting against the testing data, the linear regression model had an error of 0.660%. For the testing data, the error was 0.768%. The neural network predicted with an error of 0.314% for the testing data and 0.745% for the training data. The significant decrease between the neural network's testing and training error indicated that the model hadn't overfitted, as testing performance was better than training performance. However, had the neural network overfitted, early stopping and dropout were two methods that could be implemented to mitigate it. After each training epoch, the performance of the model was determined. If the performance of the model after each epoch began a downward trend, the training of the model was stopped before the max number of epochs was reached in a process called early stopping. Early stopping can prevent the model from overfitting to the training data, improving its real performance. Another method of stopping overfitting is dropout. During the training process, random neurons in the model's hidden layers were "dropped out" or set to zero. This causes the neuron not to be considered in the model's calculation, which prevents the model from developing a dependency on training data or certain features of the data, thus mitigating overfitting.

For 19 of the 30 trials of the MWU algorithm, the linear regression model ended up with a higher weight than the neural network. For trials where the linear regression had a higher weight, the mean weight was 0.98855. For the neural network, this number was 0.93538. The linear regression model's larger weights were 5.6843% larger than the neural network's larger weights, suggesting that even when the neural network outperformed the linear regression, the degree of improvement was relatively smaller. After the implementation of the EWMA algorithm, there was significantly less favor of one model over the other, with the greatest weight over the 30 trials being 0.58856876–a 0.08856876 deviation from the center weight of 0.5. The neural network had a higher weight in 19 of those 30 trials.

The simulated trading results were promising. Firstly, it must be said that as the trained linear regression model doesn't have random elements and was trained on the same dataset each trial, the final holdings of the bot using the model's predictions were the same across trials. The linear regression bot recorded a final value of $11,964.77, a $1,964.77 (16%) increase. The neural network bot recorded an average value of $11,613.94, an $1,613.94 (14%) increase. Lastly, the EWMA bot recorded an average value of $12,151.31, a $2,151.31 (18%) increase. The combined predictions of two very simple artificial

intelligence models, outperformed the individual attempts of the models, with a 9.49% profit increase from the linear regression and a 33.30% profit increase from the neural network, respectively.

DISCUSSION.

Stock price prediction with a linear regression model and neural network both models had a low error on the testing data–0.660% and 0.314%, respectively–although the neural network's error was lower. Through the first MWU algorithm, the linear regression model, despite having a lower error, had a larger weight than the neural network in 19 of 30 trials. With the EWMA algorithm, the neural network had a higher weight than the linear regression model in 19 of the 30 trials. As the results of MWU and EWMA algorithms did not completely highlight the dominance of one model, the findings emphasize how not just one model can be used to accurately predict stocks to the best possible degree. Different models perform better in different situations. Simulated application of the EWMA algorithm in stock market prediction showed improved profits of 9.49% over linear regression and 33.30% over the neural network, emphasizing this approach's benefits. This study didn't consider every type of AI model commonly used in stock prediction and, consequently, could have missed certain trends. This study and its findings emphasize the importance of considering multiple models when predicting, as using only one model may limit the performance and quality of stock predictions.

Future approaches to stock price prediction should try to combine multiple approaches to effectively use the strengths of those models together. Analysis of text along with stock data together can produce promising results. Additionally, different types of textual data might provide unique insights into predictions. Text sourced from various social media threads, multiple types of news sources, and news on other stocks and the general market all have potential. With this approach, careful examination of the textual data should ensure no bias creeps into the predictions. Another potential approach to this problem could be establishing a risk index for predictions made with multiple models. As more factors are considered, a risk assessment for predictions would be incredibly useful as the stock market is inherently a risk/reward field. Some last features to consider in future approaches are statistics regarding the national and world economy, such as the price of the dollar, inflation, and employment and unemployment rates, among other factors.

An obvious real-world application of this research is investing. As seen in the simulated trading, making predictions found off of multiple models' predictions leads to improved returns in the stock market. Large or small scale investing and professional or amateur level investing all have potential for success. This research therefore benefits many people. Improvement in stock prediction can lead to greater returns by fund managers and consequently more success for the corporate and individual entities the traders represent.

ACKNOWLEDGMENTS.

I am indebted to Odysseas Drosis for his mentorship throughout this project. I would also like to acknowledge the team at Inspirit AI and thank them for this opportunity.

REFERENCES.

1. K. Kolchin, J. Podziemska, A. Song, SIFMA Quarterly Report: US Equity & Related, 3Q23, October 2023, https://www.sifma.org/wp-content/uploads/2023/07/SIFMA-Research-Quarterly-Equity-and-Related-3Q23.pdf
2. J. Wang, S. Liu, H. Yang, Institutional investor proportions and inactive trading, *Int. Rev. Finan. Anal*. **82,** 102207 (2022).
3. Y. Kara, M. A Boyacioglu, O. K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Syst. Appl.* ***38***, 5311-5319 (2011).
4. M. Velay, F. Daniel, Stock chart pattern recognition with deep learning. arXiv preprint arXiv:1808.00418 (2018).
5. Y. Guo, S. Han, C. Shen, Y. Li, X. Yin, Y. Bai, An adaptive SVR for high-frequency stock price forecasting. *IEEE Access* **6**, 11397-11404 (2018).
6. D. M. Nelson, A.M. Pereira, R.A. Oliveira, Stock market's price movement prediction with LSTM neural networks in *Proceedings of the 2017 International Joint Conference on Neural Networks* (2017), pp. 1419-1426.
7. Y. Thesia, V. Oza, P. Thakkar, A dynamic scenario-driven technique for stock price prediction and trading. *J Forecasting* **41**, 653-674 (2022). 8. Y.-L. Lin, C.-J. Lai, P.-F. Pai, Using deep learning techniques in forecasting stock markets by hybrid data with multilingual sentiment analysis. *Electronics* **11**, 3513-3531 (2022).
9. M. D. Grigoriadis, L. Khachiyan, A sublinear-time randomized approximation algorithm for matrix games. *Oper. Res. Lett.* **18**, 53-58 (1995).
10. R. Aroussi, yfinance, https://github.com/ranaroussi/yfinance (2019).

Hannah Hollenbeck is a student at Westlake High School in Austin, TX; she participated in a research internship through the Inspirit AI program.