Original article

Journal of High School Science

The utilization of Artificial Intelligence to predict Apple Inc. stock price

Smith H

Submitted: August 14, 2023, Revised: version 1, June 26, 2024
Accepted: June 28, 2024

Abstract
Forecasting stock prices using statistical analysis alone remains a formidable challenge, owing to the efficient market hypothesis, which asserts that prices embody all accessible information. Nonetheless, a select cohort of investors and funds consistently surpasses the U.S. stock market by leveraging sophisticated strategies. While most financial researchers are skeptical that AI, in isolation, could consistently outperform an S&P 500 index fund, algorithms offer profound insights from vast datasets to assist portfolio managers in making more informed choices. This study endeavors to ascertain the optimal amalgamation of input features and machine learning models for accurately predicting Apple Inc. stock's opening price for the subsequent day. The AI algorithms exclusively draw upon technical investing analysis for predictions, as the research's focal point was to obtain the highest testing accuracy using solely obtainable values from investment websites. Five distinct machine learning models were meticulously evaluated, spanning from the simplest linear regression model to the most intricate neural network regressor. Our hypothesis posits that the neural network and random forest models will be more accurate than all other algorithms due to their intricacy, and that employing a more restricted set of technical indicators will yield superior accuracy by sidestepping complexity and overfitting pitfalls. While predictive precision may not reach the echelons of professional investment standards, diverse models and investment values are dissected to showcase the process of formulating stock prediction programs.

Keywords
Machine Learning, Artificial Intelligence, Stock price prediction, Technical analysis, Investments, Neural network, Apple Inc., Stock market technical indicators, Institutional funds, S&P 500

Harrison Smith, The Lovett School, 4075 Paces Ferry Rd NW, Atlanta, GA 30327, USA.
harrison.smith@student.lovett.org

Introduction

The conventional approach to predicting stock prices involves the expertise of professional analysts who employ fundamental and technical analyses. These methodologies encompass prognostications based on earnings reports, market sentiment, and broader market movements, including cyclic patterns, which exert influence on companies' valuations. Machine learning predominantly aids in the identification of cyclical patterns through technical metrics and other market variables. Nevertheless, existing models are equipped to instantaneously react to objective news, such as adjustments by the Federal Reserve, earnings reports, and economic indicators like unemployment or inflation. Traders and funds predominantly deploy machine learning algorithms for intraday trading and short-term predictions, capitalizing on minor market gains that aggregate over time (8).

Rather than directly forecasting stock prices, numerous companies utilize AI to sift through vast datasets efficiently. These algorithms expedite technical analysis by allowing the algorithm to deduce positive or negative trajectories based on the input data.

For long-term projections, machine learning functions as a supplementary perspective or idea generator for fund managers, indicating that these algorithms involve more than just "button-clicking". Recent startups are striving to pioneer innovative investment strategies by constructing portfolios founded on machine learning recommendations. For instance, GreenKey Technologies (acquired by VoxSmart in 2021), Chicago, IL, leverages AI speech recognition and Natural Language Processing to discern trends within financial conversations and documents (2). Kavout Corporation, Seattle, WA, employs its software and machine learning algorithms to produce stock rankings for the day, helping investors save time on research (6). Auquan, London, UK, a data processing software, empowers data analysts to devise their own algorithms using datasets and personal expertise, thereby democratizing machine learning (10).

AI technology is progressively integrated into finance and stock market forecasts, with the market continuing to expand. Between 60% and 73% of global trading volume is reliant in some way on algorithms, and the current size of the algorithmic trading market is 15.77 billion USD (1). Institutional investors and funds are striving to further harness the capabilities of robust algorithms as machine learning approaches human-like comprehension.

While a substantial portion of the research process hinges on prior investment and machine learning knowledge to validate hypotheses, other papers are consulted to generate ideas and facilitate comparisons. Vijh et al. analyzed the outcomes of a random forest regressor and an artificial neural network regressor in predicting closing prices for companies like Nike, Goldman Sachs, JP Morgan and Co., Johnson and Johnson, and Pfizer Inc. The dataset predominantly considered intraday and weekly movements, such as high minus low, close minus open, and various moving averages. Given that short-term predictions depend significantly on movement patterns, this data was pivotal for next-day predictions. For Johnson and Johnson, the neural network and random forest regressors achieved a MAPE of 0.70% and 0.75%

respectively. The consensus was that the artificial neural network regressor consistently outperformed the random forest regressor (13). Hota et al. predicted future stock prices for American Airlines using an artificial neural network regressor, decision tree regressor, random forest regressor, and support vector regressor. The random forest and artificial neural network yielded MAPE values of 0.37 and 0.36, respectively, whereas the decision tree and SVM exhibited relatively weaker outcomes (9). These papers contribute to the recognition that the random forest and neural network algorithms merit substantial testing and attention due to their consistent superiority in various trials. Additionally, both research papers employed some variation of high, low, open, and close data, which will undergo comprehensive testing in this research.

This study endeavored to ascertain the most effective machine learning models for predicting Apple Inc. stock prices for the following day, establish the parameters conducive to optimal accuracy, and discern the rationale behind successful model-value combinations. To ensure thorough hypothesis testing, each model operated under uniform conditions, and various permutations of technical indicators were examined (Figure 2).

Materials

Stock Prices and Technical Indicators dataset

The two sources of data we used as parameters were Yahoo Finance and Trading View. Yahoo Finance delivered an array for any stock, ETF, or commodity that included the date, open, close, and stock splits for the past five years. The Trading View website includes almost any technical analysis metric available. The MACD, Signal, ADX, EMA, DI+, DI-, %K, %R, %D, ROC, RSI, CCI, Smoothing Line, Bull Trend History, and Bear Trend History were all imported through downloading a Trading View CSV file. The file contained dates for the last five years, prices, and all of the previously mentioned technical analysis metrics. The Trading View data was preprocessed to ensure that there were exactly 1258 lines of data so as to match the Yahoo Finance data. Slight deviations in line numbers in the two different datasets lead to the algorithms being trained based on metrics that do not correlate to the same day. Obtaining data such as the MACD, CCI, ROC, etc., for the past five years is very difficult to find without subscribing to services such as Trading View. Trading View includes a feature that exports a CSV with all of the values/metrics selected for a certain stock. Then, the data can be read and sorted into data frames in the program that correspond to each CSV column. Yahoo Finance includes all stocks, ETFs, and securities, so data frames with the prices of the S&P 500, Microsoft, Alphabet, Dell, Nvidia, NYFANG, natural gas, copper, bonds, oil, and the dollar were created for testing. In all, 23 features were sorted into arrays to use for testing.

Methods

We used the PyCharm IDE for editing and running code. To determine which technical analysis features performed the best if used to predict Apple prices over the past five years, a professional hedge fund manager assisted by using Bloomberg Professional services for back testing. The only indicators that showed a significant positive correlation to Apple stock price were DMI, ADX, MACD, and Ishomoku. The %K, %R, RSI, Bull Trend History, Bear

Trend History, CCI, and EMA are all displayed on websites such as tipranks.com, thestreet.com, and Investopedia. Therefore, each of these indicators was tested to determine their positive or negative effects on the models. Also, stocks that perform similarly to Apple, such as Microsoft, Nvidia, and Alphabet, were interrogated. Financial experts consider Apple a cyclical stock, so the S&P 500 and Nasdaq were queried as well, since Apple generally follows market patterns. The number of previous days used as parameters also served as a primary factor in the accuracy of the models. To determine the optimal combination of various stock prices and technical indicators, various combinations of values were tested, and then the average and lowest percent error was recorded (Figure 2). When all of the 23 data points used in this research are used as parameters, the percent error became extremely high presumably due to the algorithms' inability to form patterns between so many variables. Accordingly, most tested combinations used less than eight parameters to isolate the effect of certain variables and to allow the algorithms to form patterns.

The x-array/parameters included the features included for the prediction, and the y-array included stock prices. A CSV file containing the open price, close price, MACD, Signal, ADX, EMA, DI+, DI-, %K, %R, %D, ROC, RSI, CCI, Smoothing Line, Bull Trend History, and Bear Trend History for every date over the past five years was imported into the program. This file included seventeen columns of numerical data for every date the market has been open over the past five years (the U.S. stock market closes on weekends and holidays). This turned out to 1257 lines of data. Then, a 1254-line X-array was created that included the selected features. The X-array varied between 1248 and 1258 rows because some models within the trials considered up to ten previous days, so there needed to be room to do so without trying to access days before the dataset start. The Y-array contained the same number of rows as the X-array and was populated with Apple Inc. Open Prices for the next day. Each row in the X-array contained parameters that corresponded to predicting for the next day and corresponded to a value in the Y-array. Using a randomized loop, the code sorted about 75% of the X-array and Y-array pairs into a dataset for training and 25% into a dataset for testing. Around 945 of these pairs of X-arrays and Y-arrays were sorted into training data, and 314 of the rows of data were sorted into testing data. The pairs of X-array data and Y-array data were randomly selected for placement into the training data set or the testing data set. The X-array contained as many columns as necessary for the number of attributes that needed to be tested. Each X-Array row corresponded to only one Y-Array value. Using the .fit() function through sci-kit learn libraries, the approximately 945 pairs of X-arrays and corresponding Y-array values were fed into the algorithms, which continually adjusted until they created the most accurate results. The algorithms adjusted the weights provided to certain parameters or the pathways between them depending on the specific algorithm to find a pattern that best fit the data. The training error had been calculated to gain an estimate for a potential testing error, as the testing error was almost always anticipated to display higher values due to overfitting. Subsequent to the training stage, the newly trained algorithm was tested on the approximately 314 randomized combinations of X-array and Y-array pairs to determine the percent error of the algorithm.

During the training stage, each algorithm underwent a different process of fitting into the training data set. A linear regression algorithm continually adjusted weights for the features to find the optimal amount that each feature should factor into the prediction. A neural network is much more advanced and adjusts the pathways between various nodes until a more complex pattern is found in the data. Neural networks most accurately model complex processes. KNN regression algorithms assign weights to certain parts of data and determine the closest match between past numerical values and current numerical values. A decision tree works by adjusting the numerical values at each branch of the tree to create an optimal pathway. Each leaf node, which branches lead to, represents a different result that the decision tree can lead to. A random forest regressor uses a combination of decision trees and averages their results with a weighted average to create a numerical output.

The maximum number of iterations for the neural network model was adjusted to determine the ideal maximum of maximum iterations. Too many iterations may lead to overfitting, while a lack of iterations does not let the neural network find ideal weights between nodes. Surprisingly, as shown in Figure 1, there did not exist a linear relationship or pattern between the average percent error and the maximum number of iterations. The training error was calculated in the same way as the testing error and displayed after every trial to discount overfitting. However, the training error alone does not determine the performance of an algorithm. Features are added in and removed based on their positive or negative effects on testing accuracy and training accuracy (Figure 2).

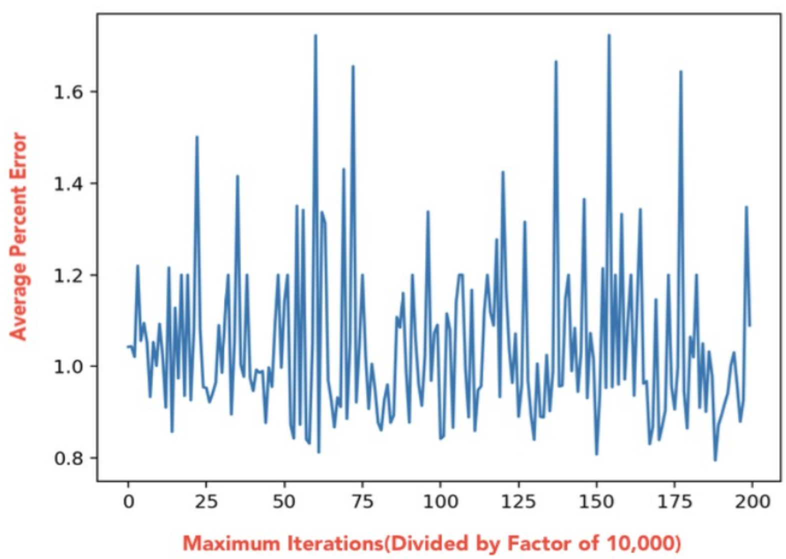

Figure 1. Average percent error for Apple Inc. Stock price prediction versus number of iterations for neural network

Results

The baseline model for predicting stock prices averaged the previous three days' open prices to predict the fourth day's open price for Apple. The percent error, as explained in the "Method" section for the baseline model, was 1.84%. When comparing all five machine learning algorithms, all features and the number of iterations remained constant throughout the trials to eliminate external influences when testing for the superior prediction model. Each model took into account the open for the last two days, previous close price, spread (open minus close), high minus low, ADX, and MACD. The lowest percent error any model achieved was 0.74% for a single trial which used a neural network with parameters/features that included the last three days, close, high, low, MACD, ADX, and open minus close. The lowest average percent error over 100 trials of the same model was 0.90% for a neural network with parameters/features that included the last two days, close, high, low, MACD, ADX, and open minus close (Figure 2). The features were held constant for every model.

Technical Analysis Indicator	Average Percent Error for Model	Change in Percent Error Caused by Given Feature
Baseline(No extra features)	1.14%	N/A
ADX	1.05%	-.09%
MACD	1.05%	0%
CCI	1.61%	+.47%
ROC	1.21%	+.07%
Signal	1.03%	-.11%
EMA	1.14%	0%
DI+, DI-	1.15%	+.01%
K%, D%, R%	1.46%	+.32%
|High - Low|	1.05%	-.09%
|Open - Close|	1.04%	-.10%

After determining which features increase accuracy when individually added, combinations of features are tested. The baseline model is a neural network taking into account two past days. Each subsequent line is the baseline model plus the features mentioned.

Technical Analysis Indicator	Average Percent Error	Change in Percent Error Caused by Given Features
Baseline(No extra features)	1.14%	N/A
ADX, MACD, Signal, |High - Low|, |Open - Close|	1.05%	-.09%
ADX, MACD	1.02%	-.12%
ADX, MACD, Signal	1.18%	+.04%
ADX, MACD, |High-Low|, |Open- Close|	.899%	-.241%

Figure 2. The Average Percent Error over 100 Trials for a Neural Network with hidden layers of size 20, 20, 20 and 800, 000 iterations. The baseline model took three days into account and each subsequent line in the Figure is the baseline model plus the feature stated.

The neural network displayed the lowest average percent error of 0.90% over 100 trials (Figure 3). The linear model performed the second best with a 0.94% error over 100 trials (Figure 4), the random forest performed third with a 1.13% error over 100 trials, the KNN regressor performed fourth best with a 1.36% error over 100 trials, and the decision tree performed the worst with a 1.37% error over 100 trials (Figures 5-7). The best neural network model performed more than 250% better than the baseline model, which shows substantial value in the technical indicators and the model itself, proving the hypothesis.

It was hypothesized that the neural network and random forest would perform substantially better than the Linear Regression, decision tree, and KNN models since stock prices follow very advanced patterns. However, Linear Regression performed better than three models, including the Random Forest, and performed marginally worse than the neural network (Figure 3). The linear model's relative success can be explained by overfitting occurring in the Random Forest and Neural Network models. The Random Forest training error typically fell between 0.3% and 0.6%, while the average percent error was 1.13% (Figure 5). The Neural Network percent error typically fell between 0.70% and 0.90% with an average percent testing error of 0.89%, due to overfitting (Figure 3). Overfitting is evident when the training error differs significantly from the testing error, as the models create complex patterns on the training data that do not necessarily apply to the testing data. The hypothesis regarding the superior models was partly proven as the neural network testing error was consistently the lowest, but the Random Forest experienced severe overfitting. Also, a simple model with just two technical indicators and past stock prices performed the best, which validated the simplicity element of the hypothesis. Some technical indicators have very little correlation to stock movement, so choosing a small number of effective indicators shows the best results.

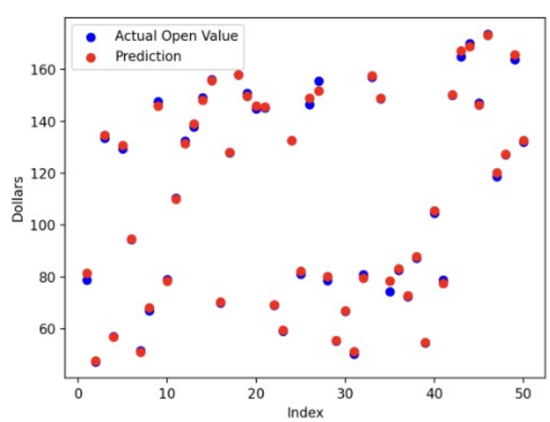

Figure 3. Actual open price of Apple Inc. Stock *versus* Neural network prediction.

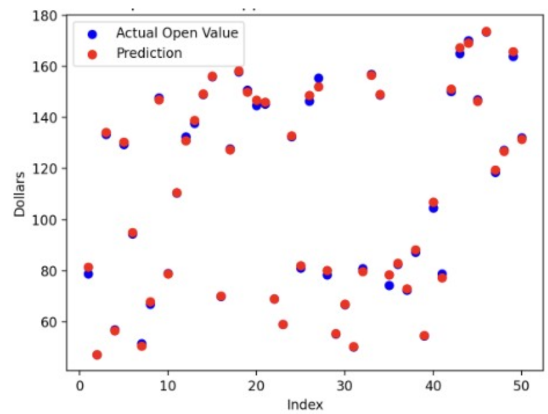

Figure 4. Actual open price of Apple Inc. Stock *versus* linear model prediction.

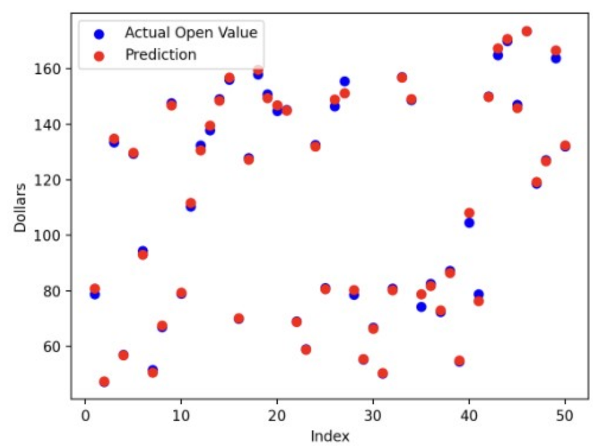

Figure 5. Actual open price of Apple Inc. Stock *versus* Random forest model prediction.

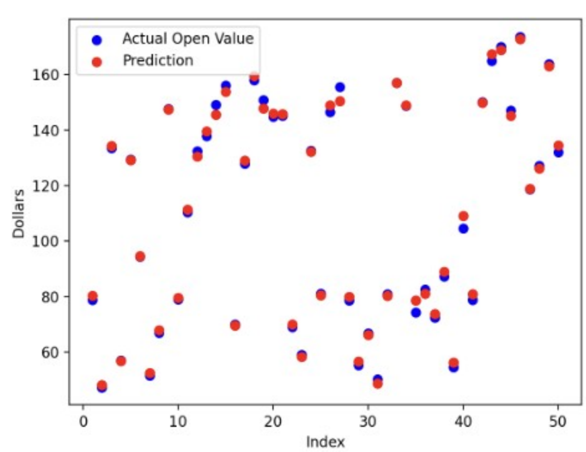

Figure 6. Actual open price of Apple Inc. Stock *versus* KNN model prediction.

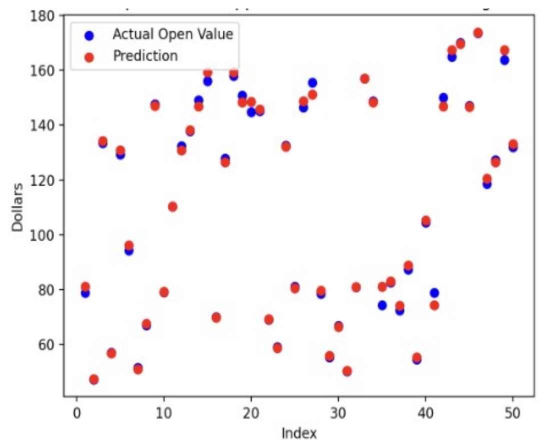

Figure 7. Actual open price of Apple Inc. Stock *versus* Decision tree model prediction.

Discussion

Key insights drawn from the exploration of machine learning in stock price prediction include the supremacy of neural networks, the selective impact of technical indicators, and the adverse effect of an excessive number of features on prediction accuracy. Neural networks exhibit an aptitude for discerning intricate market patterns that may elude human analysts and logical deduction. High-level Investment firms leverage neural networks to enhance the formulation of adaptive investment strategies. Neural networks most directly emulate the elaborate processes analysts and funds follow before making decisions (4). Conversely, linear and KNN models fall short when confronted with intricate datasets involving technical indicators and market prices.

Furthermore, the analysis found that augmenting the number of previous days invariably diminished accuracy for a neural network, holding other parameters constant (Figure 8). The data suggested that the days immediately preceding the first and second previous days hindered accuracy, as recent data combined with appropriate technical indicators proved more accurate in predicting short-term movements. Adding many previous days may help with predicting long term trends and movement, but for short-term prediction, more recent days provided much more relevant data. Including many days as features may add clutter to the algorithm that does not necessarily improve short-term pattern recognition.

Additionally, the amalgamation of 22 technical and stock price features yielded an average error rate of 7% (excluding outliers). This underscores the heuristic that extraneous features divert the model's focus from identifying patterns that directly correlate with predictions. The proliferation of features also increases overfitting, as the algorithm develops non-existent patterns that work on the training set but not on the testing set.

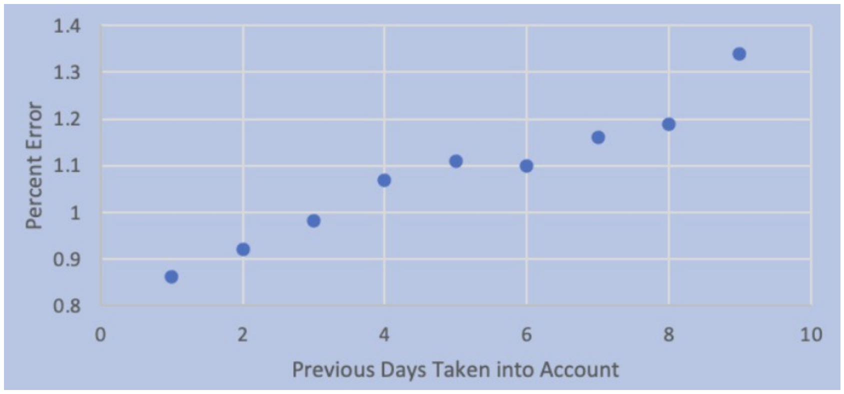

Figure 8. Average percent error *versus* number of previous days the data was drawn from

The optimal parameter combination comprised the opening prices for the last two days, the preceding close price, the spread (open minus close), the range (high minus low), ADX (Average Directional Index), and MACD (Moving Average Convergence Divergence). Notably, Bloomberg Professional Services indicated that ADX and MACD exhibit substantial correlations with predicting Apple prices, thereby substantiating their contributions to the algorithm's enhancement. Volatility indexes, particularly "high minus low" and the spread, are pivotal for intraday traders, aiding in anticipating the magnitude of next-day stock movements (11). Similarly, the opening price for the past two days facilitates modeling inter-day movement, thereby projecting movement for a subsequent day. The most recent close price proved pivotal, serving as the closest value with the following day's opening price. In short-term prediction scenarios, immediate values hold paramount relevance. It remains plausible that alternative parameter combinations might yield heightened accuracy, yet further comprehensive testing is requisite for substantiation.

Stock prediction models based on purely numerical data aid companies in finding ideas and performing intraday trading but are not typically used for long-term prediction. Training long-term models proves extremely difficult due to the constantly changing economy and market, so past financial data does not necessarily relate to current situations. Also, technical analysis, as shown in this project, does not take into account extenuating circumstances such as company news, public sentiment, and the feel of the market that many professionals possess (7). A significant obstacle to algorithmic trading is the irrationality with which stocks move. Insider trading, uneducated investors, and meme stock traders are all examples of reasons that stocks do not trade solely based on the performance of a company (3). Models that can decipher public sentiment and predict the irrationality with which the market acts hence have the potential to generate considerable amounts of revenue. Another limitation of this research is the number of combinations of parameters tested. A program can be written that tests thousands of combinations of parameters and returns the best performances for further analysis.

To include fundamental analysis in machine learning stock prediction, separate machine learning algorithms can assign scores to company reports or news headlines. Also, separate algorithms can consider professional opinions and help determine which websites, analysts, and news sources best predict the direction and future price of certain stocks. A dataset with five well-respected analysts and their 12-month price forecasts for Apple over time offers the potential to create algorithms that discover patterns in the predictions of successful analysts that rely on sophisticated fundamental analysis. The most significant limiting factor to machine learning algorithm prediction is the knowledge of the user. Analysts and money managers must understand under what economic and political conditions algorithms can perform accurately. For example, algorithms will not adequately perform under irrational circumstances, such as market manipulation by large companies and firms or investors overreacting or underreacting to news.

Using continuous testing and machine learning/investing knowledge, the best percent error obtained for a model was 0.74%, and the best average percent error over 100 trials was 0.899% (Figure 2). Therefore, machine learning and technical analysis values have a significant impact on prediction as opposed to the 1.84% error achieved by averaging the three previous days. Neural network regression almost always achieved increased accuracy in line with the hypothesis, as seen in their lowest average and overall percent error. One place for improvement to this research would be testing more advanced neural network models through Keras and other advanced machine learning models. The only technical indicators that improved price prediction were MACD and ADX, which was expected due to prior research and consulting with a hedge fund manager (Figure 2). Sophisticated AI investing algorithms are worth billions of dollars, and this research shows the first steps involved in building and testing such algorithms. Obviously, thousands of financial values exist that can be incorporated and tested as parameters for the neural network.

The linear regression model finished a close second to the neural network model. The linear regression model created far fewer complex patterns, took less time, and used less power. However, when dealing with stock price prediction, miniscule differences in percent error matter. Trading algorithms make consistent small profits that eventually matriculate into larger gains. The fact that the simple linear regression algorithm outperformed three other more complex algorithms shows the difficulty of modeling the stock market. A simpler approach obtained better results than trying to find complex patterns that may not exist. The neural network formulates the most complex prediction processes, which explained why its performance was consistently the best. The neural network found patterns in the Apple stock price movement that no other algorithm could find, which explains why large firms see immense potential in constantly evolving neural networks.

This project could easily be continued by incorporating GPT-4 to analyze Apple and other companies' financial reports, including professional opinions as parameters, and creating or finding an algorithm to label news as having positive or negative connotations

(12). The vast majority of day traders relying on technical metrics struggle to outperform the market through its ups and downs. However, a machine learning model that continuously learns based on technical analysis and incorporates the thought process of experienced investors offers considerable potential. Algorithms are exponentially faster than humans, do not deal with personal bias, rarely make mistakes, and take emotion out of investing, so machine learning in investing will only grow as models start to emulate the most brilliant human minds. Similar processes, as displayed in this paper, can be applied to any stock, commodity, or ETF. Machine learning certainly performs much better to predict prices of certain stocks than others, so a project that loops through thousands of major securities and returns the best ones helps with further exploration of AI in investing. Even the most experienced and nuanced investors in the world can use machine learning to detect interesting trends in markets to aid their decision-making.

If the algorithms are retrained with the most recent data for every next-day prediction, then the parameters will continually adjust over time. If researchers modeled the manner in which the weights within algorithms continuously change, information could arise pertaining to shifts in the relevance of certain factors in the stock market. To the author's knowledge, correlating real-time hyperparameter weight changes in algorithms to a combination of input feature importance has not yet been attempted.

Finally, more testing and evaluation should be conducted on the random forest model since its MAPE was relatively low. Research and trials can be done to determine the best way to limit overfitting, as the model performed extremely well on training data. The neural network performed significantly better than the random forest regressor in this research, but with further optimization, it is very likely that the random forest could perform better.

Limitations

Stock price prediction with only technical analysis and numerical values does not emulate the process taken by human investors and leaves out many key components. The model of prediction discussed in this paper does not include research on company fundamentals, news, or an analysis of economic conditions. The predictions generated using the machine learning algorithms apply only to Apple Inc. Stock during the given time frame. These results should not – and probably cannot - be generalized for other stocks. Also, this research tested next-day price predictions which may not necessarily correlate with long-term price predictions. However, further research can be conducted on stocks similar to Apple, such as tech and cyclical stocks.

Conclusion

The optimal amalgamation of input features and machine learning models for accurately predicting Apple Inc. stock's opening price for the subsequent day was found. The optimal input feature combination comprised the opening prices for the last two days, the preceding close price, the spread, the range, ADX, and MACD. A neural network yielded the most accurate price prediction.

Acknowledgments

Inspirit AI provided valuable resources and insights to help with the successful formation of this research. The Inspirit AI program connected Harrison Smith and Odysseus Drosis to work together on machine learning and financial research. Gary D. Smith provided valuable analysis help by using Bloomberg Professional Services to back test which technical indicators performed best for Apple over the last five years. He also provided a list of technical indicators and different stocks/commodities for incorporating in the predictions.

Abbreviations

ADX: Average Directional Index, MACD: Moving Average Convergence Divergence, CCI: Commodity Channel Index, ROC: price Rate Of Change, Signal: Oscillators for Moving Average Convergence Divergence or Stochastic, EMA: Exponential moving average, DI (+ and -): Components of the Average Directional Index, K% and D%: Derived forms of the Stochastic Oscillator Indicator, R%: Momentum indicator that is the inverse of the Fast Stochastic Oscillator, High: Highest stock price of the day, Low: Lowest stock price of the day, Open: Stock price at market open, Close: Stock price at market close, RSI: Relative Strength Index, Smoothing line: Smoothed Rate of Change, Bull Trend History: Sustained period of rising prices, Bear Trend History: Sustained period of falling prices, ETF: Exchange Traded Fund, MAPE: Mean Absolute Percentage Error, GPT-4: Generative Pre-trained Transformer 4, Ishomoku: a collection of technical indicators that show support and resistance levels, as well as momentum and trend direction. DMI: Directional Movement Index, NYFANG: NYSE FANG index comprising of stock tickers META, AAPL, AMZN, NFLX and MSFT.

References

1. *Algorithmic trading market size & share analysis – growth trends & forecasts (2023 - 2028)*. (n.d.). Mordor Intelligence. https://www.mordorintelligence.com/industry-reports/algorithmic-trading-market

2. *Artificial Intelligence Provider GreenKey Technologies Receives Strategic Investment From Motorola Solutions*. (n.d.). Business Wire. https://www.businesswire.com/news/home/20191003005068/en/Artificial-Intelligence-Provider-GreenKey-Technologies-Receives-Strategic-Investment-From-Motorola-Solutions

3. Carosa, C. (n.d.). *How Irrational Markets Trick You (And What You Can Do About It*. Forbes. https://www.forbes.com/sites/chriscarosa/2020/06/05/how-irrational-markets-trick-you-and-what-you-can-do-about-it/

4. Chanda, S. (n.d.). *How neural networks are used in the investment world https://economictimes.indiatimes.com/markets/stocks/news/how-neural-networks-are-used-in-the-investment-world/articleshow/94043039.cms?utm_source=contentofinterest&utm_medium=text&utm_campaign=cppst*. The Economic Times.

https://economictimes.indiatimes.com/markets/stocks/news/how-neural-networks-are-used-in-the-investment-world/articleshow/94043039.cms?from=mdr

5. Downey, L. (n.d.). *Efficient Market Hypothesis (EMH): Definition and Critique*. Investopedia. https://www.investopedia.com/terms/e/efficientmarkethypothesis.asp

6. Foy, P. (n.d.). *8 Companies Working on AI for Trading & Investing*. MLQ.ai. https://www.mlq.ai/ai-companies-trading-investing/

7. *Fundamental Analysis: A Complete Guide*. (n.d.). Winvesta, https://www.winvesta.in/blog/fundamental-analysis-a-complete-guide/

8. *Hedge funds rely on algos to trade majority of portfolios in market stress*. (n.d.). The Trade. https://www.thetradenews.com/hedge-funds-rely-on-algos-to-trade-majority-of-portfolios-in-market-stress/

9. Hota, J., Chakravarty, S., Paikaray, B. K., Bhoyar, H. (n.d.). *Stock Market Prediction Using Machine Learning Techniques*. ceur-ws. https://ceur-ws.org/Vol-3283/Paper86.pdf

10. Jain, C. (n.d.). *Adoption of AI in Investment Management: How can firms setup for success*. medium. https://medium.com/auquan/how-investment-management-can-setup-data-science-teams-for-successul-ai-implementation-40ced831660f

11. Konchar, P. (n.d.). *What is Volatility? And Strategies to Trade It*. My Trading Skills. https://mytradingskills.com/volatility-trading-strategies

12. Schmitz, C. (n.d.). *The GPT-4 Revolution: Towards an Augmented CFO*. LinkedIn. https://www.linkedin.com/pulse/gpt-4-revolution-towards-augmented-cfo-c%C3%A9dric-schmitz/

13. Vijh, M., Chandola, D., Tikkiwal, V. A., Kumar, A. (2020). Stock Closing Price Prediction using Machine Learning Techniques. *ScienceDirect*, 167, 599-606, https://www.sciencedirect.com/science/article/pii/S1877050920307924