Short-Term Stock Price Forecasting using exogenous variables and Machine Learning Algorithms

Albert Wong, Steven Whang, Emilio Sagre, Niha Sachin, Gustavo Dutra, Yew-Wei Lim, Gaétan Hains, Youry Khmelevsky, Frank Zhang

Short-Term Stock Price Forecasting using exogenous variables and Machine Learning Algorithms

Albert Wong
Mathematics and Statistics
Langara College
Vancouver, Canada
0000-0002-0669-4352

Steven Whang
Mathematics and Statistics
Langara College
Vancouver, Canada
swhang00@mylangara.ca

Emilio Sagre
Mathematics and Statistics
Langara College
Vancouver, Canada
esagre00@mylangara.ca

Niha Sachin
Mathematics and Statistics
Langara College
Vancouver, Canada
nsachin00@mylangara.ca

Gustavo Dutra
Mathematics and Statistics
Langara College
Vancouver, Canada
gdutra01@mylangara.ca

Yew-Wei Lim
Mathematics and Statistics
Langara College
Vancouver, Canada
ywlim@langara.ca

Gaétan Hains
LACL
Université Paris-Est
Créteil, France
0000-0002-1687-8091

Youry Khmelevsky
Computer Science
Okanagan College
Kelowna, Canada
0000-0002-6837-3490

Frank Zhang
School of Computing
University of the Fraser Valley
Abbotsford, Canada
0000-0001-7570-9805

***Abstract*—Creating accurate predictions in the stock market has always been a great challenge in the finance world. With the rise of machine learning as the next level in the forecasting area, this research paper compares four machine learning models and their accuracy in forecasting three well-known stocks traded in the NYSE in the short term over the period from March 2020 to May 2022. We deploy, develop, and tune XGBoost, Random Forest, Multi-layer Perceptron, and Support Vector Regression models and report the models that produce the highest accuracies from our evaluation metrics: RMSE, MAPE, MTT, and MPE. Using a training data set of 240 trading days, we find that XGBoost gives the highest accuracy despite taking longer (up to 10 seconds) to run. Results from this study may improve with the further tuning of the individual parameters or introducing of more exogenous variables.**

***Index Terms*—Stock Price Predictions, Exogenous variables, Support Vector Regression, Multilayer Perceptron, Random Forest, XGBoost, Machine Learning, Algorithmic Trading.**

I. Introduction

Machine learning models and algorithms have become increasingly popular over the past few years and will continue to be used even more in the future. Researchers, analysts, and other professionals have used machine learning and incorporated it into our daily lives. From corporations to individuals, machine learning can be applied in a wide range of areas. In this research paper, we explore the use of machine learning models to predict stock market price forecasts. This work extended those completed in [1]–[5] and built on the ideas used in [6], [7]. A more detailed survey of these works is presented in Section II.

Machine learning algorithms use past data to create short-term forecasts of the movement of the chosen stock prices. However, predicting stock market prices is known to be difficult because many factors contribute to the movement of these prices. Well-known theories such as the Efficient Market Hypothesis and Random Walk theory suggest that it is impossible to beat the market consistently. By comparing four different machine learning algorithms, this research aims to determine the model that produces the most accurate prediction of the movement of the chosen stocks.

There are many possibilities of exogenous variables that can be used for a machine learning algorithm. For the purpose of this research, we have built on our previous study [8] and include variables that represent short-term interest rate movement (2-year treasury bonds) as well as inflation (the price of gold and the price of crude oil). This is in addition to variables that we believe are central to the prediction of the price of a stock: movements of the overall market (Dow Jones, SI&P, and NASDAQ indexes) and longer-term interest rate (5 and 10-year treasury bonds).

Note that our research has focused on price prediction rather than trading efficiency. Our attempt is to see how

traditional wisdom about factors that impact the price of a stock, such as historical prices, movement of the broad market, interest rate, inflation, and other exogenous variables, could be incorporated into a machine learning algorithm to produce an accurate forecast. We consider that the inclusion of trading strategies, although an ultimate objective, makes it impossible to analyze objectively the quality of price prediction. In other words, strategies could succeed by "chance" even with inaccurate price predictions given a favourable enough market situation.

This work is a continuation of Wong et al. [8].

II. Existing Work

Traditionally, stock price prediction uses technical and/or fundamental analysis. According to Khaidem et al. [1], using machine learning to forecast stock prices specifically is rather new. Despite the difficulty of the inherent instability of the stock market, there is evidence that applying machine learning techniques elevate the accuracy of price forecast (see, for example, [2]).

As with any other model, the list of predictors (variables) a researcher uses as input into a machine learning algorithm is a critical success factor. Nabipour et al. [2] conclude in their research that is utilizing binary data compared to continuous data produces more accurate results when applying machine learning models to the Tehran stock market.

Kim and In [9] suggest a significant relationship between stock prices and bond yield. They conclude that a negative relationship exists between the two variables. Similarly, Engsted and Tanggaard [10] investigate the correlation between the price of stock and bonds in the Danish market. Interestingly, they conclude that the Danish stock and bond prices move similarly to the US market and that the Danish stock and bond prices also have a strong correlation. In his research, Kwan [11] concludes that bond prices and stocks are negatively correlated.

Another variable that may contribute to the movement of stock prices is gold, as it has always been a safe investment in an economic downturn. Smith [12] explores the movement of the price of gold and stock prices in the United States. The researcher concludes that there is a small but positive correlation between stock prices and gold in the short run. Palamalai and Prakasam [13] also conclude in their research that there is a long-term relationship between gold and stock prices in the Indian market.

Similar to gold, another commodity that may have a significant effect on the movement of stocks is crude oil. Narayan [14] finds that the Vietnamese stock market is heavily dependent on the price of oil. In contrast, international stock markets such as Australia, Canada, and France do not respond to shocks in the oil market, according to Apergis and Miller [15]. Although this may be true, Akoum et al. [16] recently found that the relationship between oil and stock prices has strengthened after interest in oil markets spiked in 2007.

Machine learning algorithms are increasingly popular as tools for forecasting stock prices. A comprehensive review of this topic can be found in [17]. In the following, we highlight some of the basic algorithms used in recent years and are considered in this research.

The Random Forest algorithm is mainly implemented for classification and estimation problems through the use of an ensemble of decision trees. With a goal of minimizing noise originating from the stock market, Vijh et al. [4] apply the random forest algorithm on five well-known stocks traded in the NYSE to forecast their closing prices. They find that artificial neural networks (ANN) perform better than the random forest model. However, Kumar et al. [18] find that random forest outperforms Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naive Bayes, and Softmax when predicting stock prices.

The Support Vector Regression (SVR) algorithm has also been used by researchers to forecast stock prices. Henrique et al. [19] conclude in their research on the Brazil, US, and China markets that the SVR model has predictive power and that the SVR model works better during periods of lower volatility. In this study, the Radial Basis Function (RBF) Kernel is apt because it computes the similarity or how close two points X1 and X2 are to each other due to their similarity to the Gaussian distribution.

There have been numerous research studies that have used the Multilayer Perceptron (MLP) algorithm to forecast stock prices. Devadoss and Ligori [5] explore the use of MLP on the Bombay Stock Exchange. They conclude that despite the instability and volatility of the market, their MLP model produces accurate results (MAPE ranges from 1.51% to 5.14%.) In a more recent study, Namdani and Durrani [20] compare the predictive power of MLP to other algorithms such as SVM and Long Short Term Memory (LSTM). They conclude that their MLP forecasts are more accurate than other machine learning models.

The use of XGBoost to forecast stock prices is a relatively new concept as not many researchers have used it for that purpose. Wang et al. [3] conclude in their research that XGBoost can accurately predict the prices of stock indices traded in the NYSE. In another study, Gumelar et al. [21] explore the use of XGBoost and LSTM to create a trading strategy for the Indonesian Stock Exchange. They conclude that XGBoost produces a 99% accuracy rate in terms of predictive power.

III. Methodology

The overall process of the research can be seen in Figure 1.

Note that the exogenous (input) variables were chosen for their perceived impact on the stock price. Bond prices are related to interest rates, which can affect the borrowing power of investors. Stock market indexes are composed of

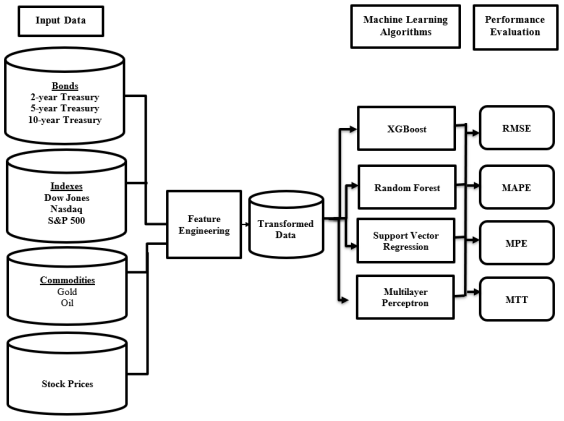

Fig. 1. Model Building Process

the "typical" stocks in the market, so their movement gives the general direction of the market. We use the price of gold and oils as proxies for inflation which we believe have a significant impact on the stock market in general.

In this study, the forecast models were created using data over 60 days or 240 days and assessed using a testing data set of four months.

To produce forecasts every fifteen minutes during a trading day, a forecasting model should use historical data up until the time period just before the forecasting period. The performance of the model developed would then be evaluated interval by interval over the testing period chosen by the researcher. Therefore, the model will need to be trained and re-trained using the "rolling" data set (dropping data from one fifteen-minute interval at the "beginning" interval of the time series and adding data on the most recent fifteen-minute interval). Forecasts from the model for the next fifteen minutes would then be generated and compared to the actual values. As a result, the training and testing cycle will need to be run with the "rolling" data set of roughly two thousand four hundred (4 months times 20 trading days each month times 30 trading periods each day i.e. 7.5 hours times 4 fifteen-minute intervals) times.

We anticipate that the forecasts produced by the models for the next fifteen minutes will be fairly accurate [22]. On the other hand, the training and test time required to build the model could be very high. Therefore, we made modifications to the above process to reduce the number of training and testing cycles by using a longer forecasting period. Accordingly, we built models using a data set that "rolls" every day producing one-day ahead forecasts in fifteen-minute intervals as well as one that "rolls" every five days producing five-day ahead forecasts in fifteen-minute intervals.

The process described above was used to create a forecasting model for the stock price of Tesla (TSLA). For comparison purposes, we also applied the same approach and algorithms to create similar models for the stock prices of Apple (AAPL), and Nvidia (NVDA).

A. Feature Engineering

For this study, data have been collected, on the features described in fifteen-minute intervals, for the period of March 2020 to May 2022. In addition to the input numerical features previously mentioned, one-up categorical features were also created to capture the calendar effects and seasonality patterns on stock prices. A summary of the variables is presented below.

1) Numerical Features:

- Value of Dow Jones Index
- Value of Nasdaq Index
- Value of S&P 500 Index
- Price of two-year treasury bond
- Price of Five-year treasury bond
- Price of Ten-year treasury bond
- Price of gold
- Price of crude oil

The value of these numerical features was shifted by one time period so that the stock prices were predicted based on the value of these features in the previous period.

2) Categorical (One-Up) Features: To capture the seasonal pattern and other calendar effects on stock prices, we created several indicator features for each fifteen-minute interval:

- Months of the year (12 one-hot variables)
- Day of the month (31 one-hot variables)
- Day of the Week (5 one-hot variables for Monday to Friday)
- Hours of the day (6 one-hot variables for hours 9:00 to 16:00)
- Minute Segment of the hour (4 one-hot variables for minute segment between 0,15,30, and 45)
- Whether the time period is on Monday morning (1 indicator variable)
- Whether the time period is on Friday afternoon (1 indicator variable)
- Whether the time period is in a "Pre-holiday" afternoon (1 indicator variable)
- Whether the time period is in a "post-holiday" morning (1 indicator variable)

B. Normalization of Numerical Features

The min-max normalization process (Equation 1) is applied across all numerical features.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

C. Performance Evaluation

After building the machine learning models, we must evaluate how accurate the forecasts are. In this research, we use the following four performance metrics: Root mean squared error, Mean absolute percentage error, Mean

positive error, and Mean training time. The first two performance metrics are common in estimation or forecasting models and would allow for the comparison of results across models or studies. [7], [8]

1) Root Mean Square Error: The root mean square error (RMSE) is a popular metric for measuring the predictive model's performance and comparing different predictive models. The calculation for finding the value of RMSE can be seen in Equation 2 below.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{2}$$

In this project, the unit of the RMSE is in dollars.

2) Mean Absolute Percentage Error: The mean absolute percentage error (MAPE) metric measures the average absolute error percentage between the predicted and the actual value. It is independent of the scale of the data and therefore can be used for comparison of performance between different models. The calculation to find MAPE can be seen below in Equation 3.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{|Y_i|} \tag{3}$$

3) Mean Positive Error: The mean positive error (MPE) is developed specifically for the evaluation of stock price prediction. It focuses solely on errors made when the predicted value is larger than the actual value. The calculation to find MPE can be seen below in equation 4.4.

$$MPE = \frac{1}{n}\sum_{i=1}^{n} max(\hat{Y}_i - Y_i, 0) \tag{4}$$

The unit of the MPE is in dollars.

4) Mean Training Time: The Mean Training Time (MTT) measures the average amount of time an algorithm takes to train with the dataset and create the prediction model. From a computational standpoint, the development process described above could be demanding if we were to produce forecasts every fifteen minutes for the entire stock universe in a sizable stock market. Therefore, the mean training time is tracked as part of the evaluation metric.

As you can see in the following section, this metric is not critical for the current study as the training time for the development of a forecasting model for one stock is insignificant. This measure will become critical when training a model that allows for the forecasting of multiple stocks.

The unit of the MTT is in seconds.

IV. Results

In this section, we compare the performance of the models developed in this study based on the evaluation metrics mentioned above. Naturally, a lower value for all performance metrics is favourable for a model as this implies that its predictions are closer to the actual values in general. In the following, we will focus our comparisons on the use of a 60 days training data set versus a 240 days training data set for producing one-day ahead and five-day ahead forecasts.

The four basic machine learning algorithms were used to develop the forecasting models with different hyper-parameters to optimize their performance. The following tables summarize the top-performing models developed for each algorithm with details on the hyper-parameters used.

Model Number	Hyperparameter Details
XGB 1.0	n_estimators = 100, max_depth = 100
XGB 2.0	n_estimators = 300, max_depth = 100
RF 1.0	n_estimators = 100, max_depth = 100
RF 2.0	n_estimators = 300, max_depth = 100
MLP 1.0	neurons = 100, activation = relu, dropout = 0.25, opt = Adam (amsgrad=True, lr =0.001,beta_1=0.79, beta_2 = 0.999), loss = mse
MLP 2.0	neurons = 100, activation = relu, dropout = 0.25, opt = Adam (amsgrad=True, lr =0.001,beta_1=0.79, beta_2 = 0.999), loss = mse, epochs=8, batch_size=256
MLP 3.0	neurons = 100, activation = relu, dropout = 0.25, opt = Adam (amsgrad=True, lr =0.001,beta_1=0.79, beta_2 = 0.999), loss = mse, epochs=20, batch_size=256
SVR 1.0	kernel = 'rbf', C=1.0, gamma = "scale"
SVR 2.0	kernel = 'rbf', C=5.0, gamma = "scale"
SVR 3.0	kernel = 'rbf', C=10.0, gamma = "scale"

Fig. 2. Top-Performing Models for Each ML Algorithm

The first set of results, presented in Figure 3, is on the comparison of these models for forecasting the price of Tesla stock for the next day (1 day ahead forecast). Two sets of results were produced: one for models developed and evaluated using "rolling" 60-day training data sets, the other using "rolling" 240-day training data sets.

It is apparent that, on average, the XGBoost models produce the lowest errors compared to the other machine learning models considered. The Random Forest models are not far behind.

Model	Model	RMSE		MAPE (%)		MPE (%)		MTT (in seconds)	
		60 Days	240 Days	60 Days	240 Days	60 Days	240 Days	60 Days	240 Days
XGBoost	XGB 1.0	28.70	33.41	2.30	2.39	7.00	8.54	0.48	3.65
	XGB 2.0	28.61	33.31	2.29	2.38	7.07	8.62	0.97	9.86
Random Forest	RF 1.0	31.21	39.05	2.40	2.43	9.77	11.16	1.00	4.50
	RF 2.0	31.31	38.58	2.40	2.40	9.74	11.05	3.11	12.00
Multilayer Perceptron	MLP 1.0	71.04	68.90	7.16	5.91	20.92	15.99	0.16	0.32
	MLP 2.0	63.85	68.24	6.35	5.88	18.76	17.35	0.25	0.46
	MLP 3.0	56.22	62.47	5.69	5.36	16.66	16.73	0.44	0.84
Support Vector Regression	SVR 1.0	115.34	159.83	11.08	11.82	23.64	9.34	0.13	1.71
	SVR 2.0	73.82	101.56	6.43	6.78	15.70	12.24	0.17	1.58
	SVR 3.0	60.42	84.78	5.12	5.34	13.54	11.60	0.12	2.17

Fig. 3. Performance of Forecasting Models on Price of Tesla Stock - One Day Ahead

The same can be said about the performance of forecasting models developed for the Apple and Nvidia stocks.

To give the reader a sense of the performance of the models, below are the graphs that show the best models, in terms of MAPE, in producing 1 day-ahead forecast for the three stocks.

It can be observed from the graphs that prediction accuracy is higher during periods with low volatility. A higher level of prediction errors occurs when the actual price of the stocks fluctuates more. This is completely understandable.

Model	Model	RMSE		MAPE (%)		MPE (%)		MTT (in seconds)	
		60 Days	240 Days	60 Days	240 Days	60 Days	240 Days	60 Days	240 Days
APPLE									
XGBoost	XGB 1.0	2.49	3.15	1.12	1.21	0.45	0.38	0.56	1.86
	XGB 2.0	2.46	3.11	1.09	1.17	0.46	0.38	0.67	3.18
Random Forest	RF 1.0	2.53	3.27	1.04	1.13	0.67	0.64	1.10	5.41
	RF 2.0	2.55	3.30	1.05	1.13	0.67	0.64	3.52	12.40
Multilayer Perceptron	MLP 1.0	9.07	7.65	3.74	3.56	1.93	1.09	0.59	1.02
	MLP 2.0	7.31	5.26	3.10	2.45	1.58	1.06	0.67	0.97
	MLP 3.0	5.12	5.52	2.46	2.51	1.15	0.84	0.98	1.95
Support Vector Regression	SVR 1.0	4.22	6.39	1.92	2.93	0.85	0.72	0.25	2.98
	SVR 2.0	3.03	4.45	1.26	1.82	0.63	0.64	0.45	5.06
	SVR 3.0	2.81	4.19	1.16	1.63	0.61	0.62	0.56	7.47
NVIDIA									
XGBoost	XGB 1.0	6.24	8.22	1.67	1.97	1.20	1.46	0.34	1.94
	XGB 2.0	6.29	8.29	1.70	2.00	1.18	1.44	0.19	0.85
Random Forest	RF 1.0	6.93	9.21	1.70	1.97	1.65	2.01	2.04	8.77
	RF 2.0	6.91	9.33	1.70	2.00	1.66	2.05	0.65	2.94
Multilayer Perceptron	MLP 1.0	13.71	14.74	4.51	4.60	3.10	3.71	0.11	0.19
	MLP 2.0	11.94	12.79	3.99	4.05	2.51	3.05	0.16	0.30
	MLP 3.0	10.68	12.42	3.58	3.96	2.34	3.10	0.27	0.63
Support Vector Regression	SVR 1.0	16.80	29.88	4.92	8.71	3.66	2.43	0.10	1.41
	SVR 2.0	9.80	15.92	2.72	4.28	2.29	2.30	0.15	2.36
	SVR 3.0	8.59	13.57	2.29	3.55	2.12	2.30	0.22	4.46

Fig. 4. Performance of Forecasting Models on Price of Apple and Nvidia - One Day Ahead

Fig. 5. Forecasting 1-Day ahead using 240 training days for Tesla using XGBoost

Across all experiments, the XGBoost models produced the lowest errors on average compared to the other machine learning models. However, it is noticeable from the MTT metric that the Random Forest algorithm takes the longest to train, followed by XGBoost. More interesting is the fact that increasing the N-estimators parameter from 100 to 300 for both the XGBoost and the Random Forest algorithm showed little to no signs of improvement with respect to the MAPE performance measure. This is true regardless of whether the training data set is for 60 days or 240 days.

The second set of results, presented in Figure 8, is on the comparison of these models for forecasting the price of all three stocks for the next five days (5-day ahead forecast). As before, two sets of results were produced: one for models developed and evaluated using "rolling" 60-day training data sets, the other using "rolling" 240-day training data sets.

Overall, all values of performance metrics considered for the 5-day ahead forecasts are higher than those for the 1-day ahead forecasts. This outcome is expected as it is more difficult to accurately forecast further into the future, thus errors on average are higher. However, observations on the 5-day ahead forecasts are similar to those presented for the 1-day ahead forecasts.

With MAPE at 5% or lower, these forecasting models demonstrated the usefulness of the approach taken in this research. Further work will be required to make these models implementable inside a profitable algorithmic trading system.

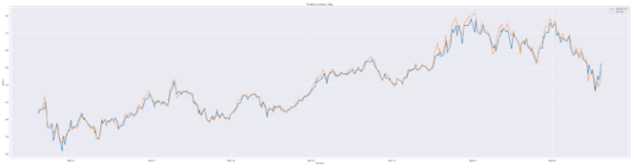

Fig. 6. Forecasting 1-Day ahead using 240 training days for Apple using XGBoost

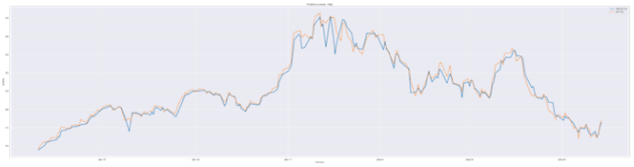

Fig. 7. Forecasting 1-Day ahead using 240 training days for Nvidia using XGBoost

V. Future Work

Although the results have been insightful, a number of improvements could be made to these models and worked on in the next phase of the research:

- the development of a data depository or data house to acquire, store, and manage the volume of stock price and related data over time. Data in this facility will be the data source for the research proposed below.
- expansion of the stock universe being considered in the research.
- additional features on the economic environment. Examples would be the inflation rate, unemployment rate, etc.
- features on market sentiment through data from social media and financial-related forums, etc.
- demographic features related to the companies: industrial classification, size, profitability, etc.
- further tuning of the above models on the hyper-parameters.
- consideration of other training data periods
- monitoring of the mean training time as the models become more complicated as a result of the above.

VI. Conclusions

This study explores the predictive power of four different machine learning algorithms (XGBoost, Random Forest, Support Vector Regression, and Multilayer Perceptron) and how accurately they can forecast the prices of several technology stocks (Apple, Nvidia, and Tesla). As well, this research follows the "fundamental" approach in finance for evaluating the stock market and picked features as input to the models accordingly. The results of these models showed promise as the values on several performance metrics are quite reasonable. We believe that the extension of these models to include other economic- or financial-related measures such as inflation rate and company demographics will lead to a stock forecasting

Model	Model	RMSE		MAPE (%)		MPE (%)		MTT (in seconds)	
		60 days	240 days	60 days	240 days	60 days	240 days	60 days	240 days
TESLA									
XGBoost	XGB 1.0	53.34	59.76	4.83	4.63	13.40	15.18	1.53	3.90
	XGB 2.0	53.28	59.72	4.82	4.62	13.45	15.24	2.93	13.75
Random Forest	RF 1.0	57.97	58.86	5.19	4.77	17.62	19.58	0.79	4.03
	RF 2.0	58.21	58.36	5.19	4.73	17.60	19.35	2.46	11.87
Multilayer Perceptron	MLP 1.0	102.93	91.09	11.09	7.61	30.71	13.88	0.14	0.40
	MLP 2.0	96.35	89.38	10.17	7.60	26.89	13.51	0.24	0.46
	MLP 3.0	79.53	78.93	8.17	6.61	23.80	15.71	0.43	0.92
Support Vector Regression	SVR 1.0	123.08	164.57	12.15	12.45	26.08	10.07	0.15	1.84
	SVR 2.0	85.93	112.99	7.99	7.96	19.32	14.24	0.14	2.18
	SVR 3.0	75.32	98.01	7.01	6.69	17.99	14.07	0.16	1.92
APPLE									
XGBoost	XGB 1.0	3.95	5.04	1.94	2.19	0.80	0.61	0.31	1.62
	XGB 2.0	3.93	5.01	1.93	2.16	0.81	0.61	0.79	3.59
Random Forest	RF 1.0	4.05	5.26	1.99	2.26	1.16	1.01	1.15	4.76
	RF 2.0	4.07	5.17	1.99	2.23	1.16	1.02	3.13	12.03
Multilayer Perceptron	MLP 1.0	16.69	9.76	6.91	4.82	4.14	1.87	0.42	1.04
	MLP 2.0	14.03	9.27	5.70	4.54	2.97	1.39	0.66	1.25
	MLP 3.0	8.75	8.70	4.09	8.70	2.28	1.37	0.95	1.88
Support Vector Regression	SVR 1.0	5.43	8.08	2.66	8.08	1.12	0.85	0.28	3.42
	SVR 2.0	4.76	7.05	2.28	3.13	1.02	0.88	0.49	6.60
	SVR 3.0	4.57	6.91	2.19	3.04	1.01	0.87	0.31	9.57
NVIDIA									
XGBoost	XGB 1.0	11.50	14.68	0.03	0.04	2.24	2.65	0.33	1.97
	XGB 2.0	11.53	14.73	0.03	0.04	2.22	2.63	0.18	0.84
Random Forest	RF 1.0	11.96	15.53	3.54	4.12	3.11	3.76	0.64	8.86
	RF 2.0	12.06	15.56	3.57	4.15	3.16	3.84	0.66	3.06
Multilayer Perceptron	MLP 1.0	22.20	20.14	8.00	6.26	5.56	3.66	0.11	0.21
	MLP 2.0	20.22	20.50	7.10	6.00	4.52	3.98	0.16	0.30
	MLP 3.0	16.13	17.11	5.20	5.05	3.57	3.62	0.25	0.65
Support Vector Regression	SVR 1.0	19.51	32.93	6.12	10.04	4.41	2.73	0.11	1.48
	SVR 2.0	14.09	20.93	4.52	5.96	3.44	2.85	0.16	2.42
	SVR 3.0	13.48	20.03	4.28	5.71	3.33	3.02	0.23	4.50

Fig. 8. Performance of Forecasting Models - Five Days Ahead

model as the foundation of a profitable algorithmic trading system.

REFERENCES

[1] L. Khaidem, S. Saha, and S. R. Dey, "Predicting the direction of stock market prices using random forest," *arXiv preprint arXiv:1605.00003*, 2016.

[2] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150 199–150 212, 2020.

[3] J. Wang, Q. Cheng, and Y. Dong, "An xgboost-based multivariate deep learning framework for stock index futures price forecasting," *Kybernetes*, no. ahead-of-print, 2022.

[4] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock closing price prediction using machine learning techniques," *Procedia computer science*, vol. 167, pp. 599–606, 2020.

[5] A. V. Devadoss and T. A. A. Ligori, "Forecasting of stock prices using multi layer perceptron," *International journal of computing algorithm*, vol. 2, no. 1, pp. 440–449, 2013.

[6] A. Wong, D. Joiner, C. Chiu, M. Elsayed, K. Pereira, Y. Khmelevsky, and J. Mahony, "A Survey of Natural Language Processing Implementation for Data Query Systems," in *2021 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE)*, 2021, pp. 1–8.

[7] A. Wong, C. Chiu, A. Abdulgapul, M. N. Beg, Y. Khmelevsky, and J. Mahony, "Estimation of Hourly Utility Usage Using Machine Learning," in *IEEE International Systems Conference (SysCon) 2022*, 2022.

[8] A. Wong, J. Figini, A. Raheem, G. Hains, Y. Khmelevsky, and P. C. Chu, "Forecasting of stock prices using machine learning models," *Submitted*, 2022.

[9] S. Kim and F. In, "On the relationship between changes in stock prices and bond yields in the g7 countries: Wavelet analysis," *Journal of International Financial Markets, Institutions and Money*, vol. 17, no. 2, pp. 167–179, 2007.

[10] T. Engsted and C. Tanggaard, "The danish stock and bond markets: Comovement, return predictability and variance decomposition," *Journal of Empirical Finance*, vol. 8, no. 3, pp. 243–271, 2001.

[11] S. H. Kwan, "Firm-specific information and the correlation between individual stocks and bonds," *Journal of financial economics*, vol. 40, no. 1, pp. 63–80, 1996.

[12] G. Smith, "The price of gold and stock price indices for the united states," *The World Gold Council*, vol. 8, no. 1, pp. 1–16, 2001.

[13] S. Palamalai and K. Prakasam, "Gold price, stock price and exchange rate nexus: The case of india," *Srinivasan P. and Karthigai, P.(2014), Gold Price, Stock Price and Exchange Rate Nexus: The Case of India, The IUP Journal of Financial Risk Management*, vol. 11, no. 3, pp. 1–12, 2015.

[14] P. K. Narayan and S. Narayan, "Modelling the impact of oil prices on vietnam's stock prices," *Applied energy*, vol. 87, no. 1, pp. 356–361, 2010.

[15] N. Apergis and S. M. Miller, "Do structural oil-market shocks affect stock prices?" *Energy economics*, vol. 31, no. 4, pp. 569–575, 2009.

[16] I. Akoum, M. Graham, J. Kivihaho, J. Nikkinen, and M. Omran, "Co-movement of oil and stock prices in the gcc region: A wavelet analysis," *The Quarterly Review of Economics and Finance*, vol. 52, no. 4, pp. 385–394, 2012.

[17] D. Joiner, A. Vezeau, G. Hains, Y. Khmelevsky, and A. Wong, "Algorithmic trading and short-term forecast for financial time series with machine learning models; state of the art and perspectives," *2022 IEEE International Conferences on Recent Advances in Systems Science and Engineering*, 2022.

[18] I. Kumar, K. Dogra, C. Utreja, and P. Yadav, "A comparative study of supervised machine learning algorithms for stock market trend prediction," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. IEEE, 2018, pp. 1003–1007.

[19] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Stock price prediction using support vector regression on daily and up to the minute prices," *The Journal of finance and data science*, vol. 4, no. 3, pp. 183–201, 2018.

[20] A. Namdari and T. S. Durrani, "A multilayer feedforward perceptron model in neural networks for predicting stock market short-term trends," in *Operations Research Forum*, vol. 2, no. 3. Springer, 2021, pp. 1–30.

[21] A. B. Gumelar, H. Setyorini, D. P. Adi, S. Nilowardono, A. Widodo, A. T. Wibowo, M. T. Sulistyono, E. Christine *et al.*, "Boosting the accuracy of stock market prediction using xgboost and long short-term memory," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*. IEEE, 2020, pp. 609–613.

[22] A. Wong, J. Figini, A. Raheem, G. Hains, Y. Khmelevsky, and P. Chu, "Forecasting of Stock Prices Using Machine Learning Models," in *IEEE International Systems Conference (SysCon) 2023*, 2023.