



**HAL**  
open science

# Forecasting of Stock Prices Using Machine Learning Models

Albert Wong, Juan Figini, Amatul Raheem, Gaétan Hains, Youry Khmelevsky, Pak Chun Chu

► **To cite this version:**

Albert Wong, Juan Figini, Amatul Raheem, Gaétan Hains, Youry Khmelevsky, et al.. Forecasting of Stock Prices Using Machine Learning Models. 2023 IEEE International Systems Conference (SysCon), Apr 2023, Vancouver, Canada. pp.1-7, 10.1109/SysCon53073.2023.10131091 . hal-04112617

**HAL Id: hal-04112617**

**<https://hal.science/hal-04112617>**

Submitted on 19 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Forecasting of Stock Prices Using Machine Learning Models

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

Juan Figini  
*Mathematics and Statistics*  
*Langara College*  
Vancouver, Canada  
jfigini00@mylangara.ca

Amatul Raheem  
*Mathematics and Statistics*  
*Langara College*  
Vancouver, Canada  
araheem00@mylangara.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

Pak Chun Chu  
*Mathematics and Statistics*  
*Langara College*  
Vancouver, Canada  
jchu37@mylangara.ca

**Abstract**—Stock price prediction with machine learning is an oft-studied area where numerous unsolved problems still abound owing to the high complexity and volatility that technical-factors and sentiment-analysis models are trying to capture. Nearly all areas of machine learning (ML) have been tested as solutions to generate a truly accurate predictive model. The accuracy of most models hovers around 50%, highlighting the need for further increases in precision, data handling, forecasting, and ultimately prediction.

In this paper we present the result of our work on high-frequency (every fifteen minutes) stock-price prediction using technical data with a number of exogenous variables. These variables are carefully chosen to reflect the conventional wisdom in a traditional stock analysis on historical trend, general stock market condition, and interest rate movement. Several simple machine learning (ML) algorithms were developed to test the premise that with the appropriate variables, even a simple ML model could produce reasonable prediction of stock prices. Therefore, the originality of our approach is a rational selection of relevant and useful features and also on-the-fly model re-training taking advantage of the human time scale of inference (price prediction) and moderate size of the models. Moreover we do not mix any trading strategy with our stock-price prediction experiments, to ensure that conclusions are not context-dependent.

Systems that integrate and test sentiment and technical analysis are considered the best candidates for an eventual generalized trading algorithm that can be applied to any stock, future, or traded commodity. However, much work remains to be done in applying natural language processing and the choice of text sources to find the most effective mixture of sentiment and technical analysis. Work on this area will be included in the next phase of our research project and here we have summarized some of the most relevant existing works in this direction.

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

We would like to acknowledge and thank the Co-op Education and Grants-in-Aid (GIA) Fund at Okanagan College as well as the Post Degree Diploma and the Work on Campus programs at Langara College for supporting our research.

## I. INTRODUCTION

A profitable stock trading algorithm will benefit from a forecasting system that can produce accurate short-term forecasts. Based on this premise, we propose this research project to leverage our previous experience in building short-term forecasting models using machine learning (ML) algorithms [1]–[16].

The Algorithmic Trading World is a dynamic area involving forecasting competition that aims to encourage the development of new models to predict the stock market's short-term response following large trades.

We contemplate our central idea, “If I know the past price, could I anticipate the future value?”

To confirm our approach, we use stock price data, in fifteen minutes intervals, for Tesla stocks, extracted from the stock market for two years. We explore, clean, normalize, and build initial forecast models with training and testing datasets, then check our models' accuracy and efficiency.

Data Analysts are expected to use Machine Learning methodology to create forecasting models to predict the stock market behaviour based on specific indicators like SP500, stock behaviour, and seasonality.

The following sections provide a short survey of the most relevant literature, then the technical details of our approach (feature engineering, error measures, models, experimental comparisons), our conclusions about predictive accuracy and ideas for future work.

## II. EXISTING WORK

There exists an incredibly large body of research papers on algorithmic trading including testing and results of various ML models using neural networks (NN), random forest (RF), support vector regression (SVR), XGBoost, and long short-term memory (LSTM) algorithms. Here we

briefly recall the ones that are most relevant and closest to our problem definition: stock-price prediction (independent from, but of course applicable to, trading performance) using structured technical-, structured company- and unstructured natural-language sentiment data.

Many models, such as that seen in [17], attempt to match predictions to real data, but a considerable price and time discrepancy exist between the predicted values and actual values. The discrepancy can be off by several hundred dollars (or any given unit of currency).

Other research groups, like that in [18], tested many neural networks (NN) and ML models to see which is the best for a given data set. They used back propagation NN, radial basis function NN, general regression NN, SVR, and Least Squares-SVR. The testing data was weekly adjusted close price of three individual stocks: Bank of China, Vanke A, and Kweichou Moutai with mean square error (MSE) and mean absolute percentage error (MAPE) as criteria. Though the authors gave largely inconclusive results, it was noted that back propagation had the best results, at least among the tested models, with an MAPE under 5%.

In 2018, Chen and He [19] investigated the reliability of deep learning methods based on a 6-layer convolutional neural network (CNN) at predicting prices on the Chinese stock market. They set the time scale to be a year and input to the opening price, high price, low price, closing price, and the volume for historical stock data sets from the Chinese stock market. The results obtained showed an accuracy of about 73%. Results at this level begin to approach a usable state, but still require fine-tuning to be considered truly reliable and predictive.

Many works have experimented, with success, on stock price predictions with financial news. In [20], the authors examined the relationship between measured sentiment from some media messages about a company and its stock price. The authors used natural language processing (NLP) to perform sentiment analysis on messages from Twitter, called tweets. This research was performed with a shorter duration than what was originally planned and has a noticeably small sample size, but it still highlights the effect of sentiment on market performance as a determining factor to be leveraged for profit.

Weng, Lu, Lang, Megahed, and Martinez [21] sought to predict short-term (1 day or 1 to 10 days) stock value using a rich hybrid data set from many available online sources, including Google search engine results, published news, technical stock indicators, historical stock values and Wikipedia article information. Among the reviewed models, its performance is the best we have observed.

Attempting to tackle the problem of developing a predictive model with integrated hybrid technical and sentiment analysis, Li, Wu, and Wang [22] use NLP for sentiment analysis and an LSTM model for technical analysis in the prediction of Hong Kong stock prices. The hybrid

technique is found to outperform baseline models (SVR and multiple kernel learning) as well as those that use either technical or sentiment data as input. However, the overall accuracy remains low with most results coming at sub 50%.

Qiu and team [23] introduce a wavelet transform, similar to Fourier transforms, to their LSTM model to denoise the volatile data set and separate useful data signal from the noise with the goal of overcoming long-term dependence issues. The performance of the WLSTM+Attention model was excellent in terms of error (MSE, MAE, and RMSE) compared to LSTM along with a gated recurrent unit (GRU). With the addition of sentiment analysis, these techniques are likely to perform very well in stock price prediction.

Patent applications [24] and [25] have the same objectives and methods close to ours. The first one performs stock-price *variation* prediction from structured data (mix of technical and company data) combined with sentiment information from news- and social media. No mention is made of the combination of stock-value changes to produce stock *prices*. The second one presents a technique for stock-price prediction using sentiment analysis, several types of neural nets (CNN, LSTM with multi-channel attention, etc.) and some correlation and dimension-reduction steps blended into a complex process. The sentiment analysis obtains its input from probabilistic web crawling and its results appear to be binary (positive/negative). By comparison with this choice on input data, fixed and less unreliable sources of natural-language information like social media could produce either better predictions or at least context-independent conclusions about public sentiment.

Patent [26] describes stock-values prediction using LSTM neural nets trained with both market data and “mood” data. The mood (“emotion” a.k.a. sentiment here) data comes from a news server on which text mining is applied to produce mood scores in the range  $[1-\ddot{y}, 1+\ddot{y}]$  with  $\ddot{y}$  ranging from 1% to 30%.

We have surveyed dozens of other references but they are not listed here for lack of space. Our work’s originality compared to all the above is to use feature engineering, on-the-fly training and an objective comparison of several ML engines under verifiable conditions.

### III. METHODOLOGY

The overall approach in the model building process is depicted in Figure 1.

To produce forecasts every fifteen minute during a trading day, a stock price forecast 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. For instance, suppose a forecasting model is created using data over two years and assessed using a testing data set of six months. In this

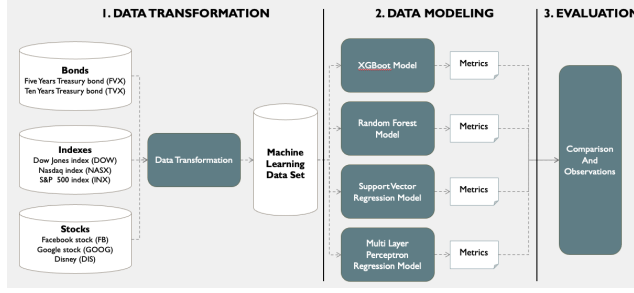


Fig. 1. Model Building Process

case, the model will 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 minute would then be generated and compared to the actuals roughly two thousands four hundred (4 months times 20 trading days each month times 30 trading period each day) times.

From a computational standpoint, this 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 testing time is tracked as part of the evaluation metric. Furthermore, we also develop models and use them to produce forecasts over a longer period (one day or five days). We will discuss Further the training and evaluation process in detail below.

#### A. Dataset Description

To build the model, we collect data, in fifteen minute interval, numerical data on Tesla’s stock price as well as those for other exogenous variables that could have a material impact. These variables (features) are described in the following:

##### 1) Numerical Features:

- Price of Five-year treasury bond
- Price of Ten-year treasury bond
- Value of Dow Jones Index
- Value of Nasdaq Index
- Value of S&P 500 Index
- Price of Facebook Stock
- Price of Alphabet(Google) Stock
- Price of Disney Stock
- Price of Tesla Stock; Target Variable

To capture the seasonal pattern and other calendar effects on stock prices, we created several indicator features for each fifteen minute interval:

##### 2) One-Up Features:

- Year (3 one-up variables for 2020, 2021, and 2022)
- Months of the year (12 one-up variables)
- Day of the month (31 one-up variables)
- Week day (5 one-up variables for Monday to Friday)
- Hours of the day (6 one-up variables for hours 9 to 16)

- Minute Segment of the hour (4 one-up variable for minute segment 0,15,30, and 45)
- Whether the time period is in Monday morning (1 one-up variable)
- Whether the time period is in Friday afternoon (1 one-up variable)
- Whether the time period is in a “Pre-holiday” afternoon (1 one-up variable)
- Whether the time period is in a “post-holiday” morning (1 one-up variable)

For the purpose of this research, the data set for training and testing was created for the period of June 2020 to May 2022.

#### B. Feature Engineering

Once the data is collected, the min-max normalization process (refer to Equation 1) is applied to all numerical variables.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

As well, even though it is trivial, it is worth pointing out that previous period’s value of the numerical variables (except the price of the Tesla stock) are used in the modelling process.

#### C. Performance Evaluation

In this research, we considered several performance metrics for evaluating a model’s performance. Specifically, we used the conventional performance metric, Root mean square error (RMSE), Mean Absolute Percentage Error (MAPE), as well as the Mean Absolute Error (MAE), which are quite common in estimation or forecasting models [14].

1) *Root Mean Square Error*: The root means square error (RMSE) is a popular metric for measuring the predictive model’s performance and comparing different predictive models. It can be calculated as follows:

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

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

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. Equation 3 shows the calculation of the MAPE.

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

Note that results for the MAPE are in percentage

TABLE I  
RESULT OF THE SVR MODELS

Model	RMSE	MAPE	MAE	MTT
SVR				
Train 60 days	121.569	11.9	91.157	0.204
Predict 5 days				
SVR				
Train 60 days	114.050	11.0	84.307	0.163
Predict 1 day				
SVR				
Train 60 days	110.421	10.4	80.034	0.179
Predict 15 minutes				

3) *Mean Absolute Error*: The mean absolute error (MAE) metric measures the average absolute error between the predicted and the actual value. Equation 4 shows the calculation of the MAE.

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad (4)$$

the unit of the RMSE is in dollar.

As stated above, as part of the performance evaluation, we also track the time taken to perform each training and testing circle. The average of these times over the entire testing data set is calculated as the Mean Testing Time (MTT).

#### IV. RESULTS

In this section, results produced by four machine learning models: Support Vector Regression (SVR), multilayer Perceptron (MLP), Random Forest (RF), and Extreme Gradient Boosting (XGBoost) will be presented. Note that these algorithms are rather basic in terms of structure and application. Choosing these simple algorithms allows us to focus on the importance of the features used in the model. They are also used often in the development of forecasting models for price of financial assets. Joiner et al. (reference)

##### A. Support Vector Regression Models

We use a basic SVR algorithm with “rbf” as the kernel. In addition, we use a rolling 60 days testing data and the resulting model is used to predict prices in three different ways: predict prices at the end of all fifteen minute intervals in the next five days, predict prices for all fifteen minute intervals in the next day, and predict prices the price at the end of the next fifteen minute . Intuitively, the prediction for the last scenario should be most accurate. This is indeed the case. The comparison between the actuals and predicted for all three scenarios is depicted in Figures 2–4.

Results of the SVR models are summarized in Table 1.

##### B. Multilayer Perceptron Models

A multilayer perceptron (MLP) is a feedforward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between

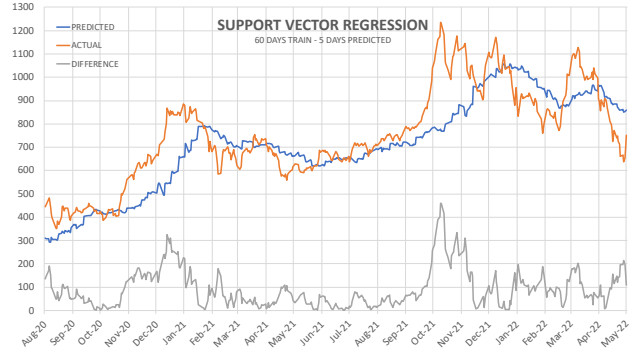


Fig. 2. SUPPORT VECTOR REGRESSION 60 DAYS TRAIN - 5 DAYS PREDICTED.

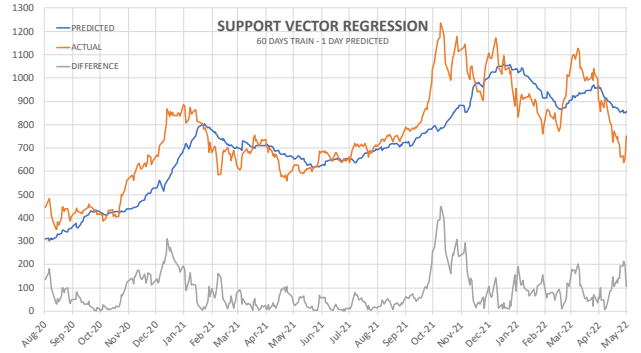


Fig. 3. SUPPORT VECTOR REGRESSION 60 DAYS TRAIN - 1 DAY PREDICTED.

the input and output layers. MLP uses backpropagation for training the network. A MLP model could be quite complicated in structure. For this project, the MLP models built have very simple structure with the following hyper-parameters: neurons = 100, activation function = relu, input dimension = 73, and optimizer = Adam. The comparison between the actuals and predicted for all three scenarios is depicted in Figures 5–7.

Results of the MLP models are summarized in Table 2.

##### C. Random Forest Models

Random forest (RF) is a Supervised Machine Learning algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples

TABLE II  
RESULT OF MLP MODELS

Model	RMSE	MAPE	MAE	MTT
MLP				
Train 60 days	172.100	22.5	141.005	0.275
Predict 5 days				
MLP				
Train 60 days	106.448	11.4	77.306	0.317
Predict 1 day				
MLP				
Train 60 days	54.919	5.5	40.049	0.249
Predict 15 minutes				

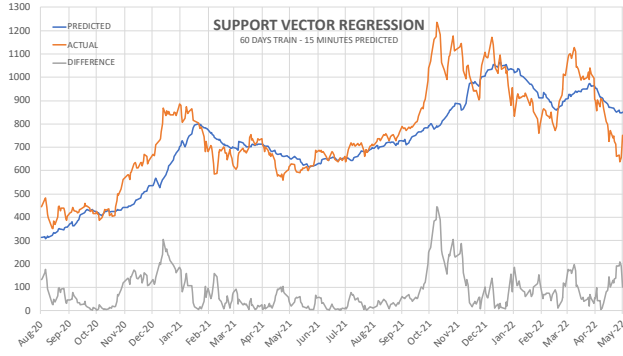


Fig. 4. SUPPORT VECTOR REGRESSION 60 DAYS TRAIN - 15 Mins PREDICTED.

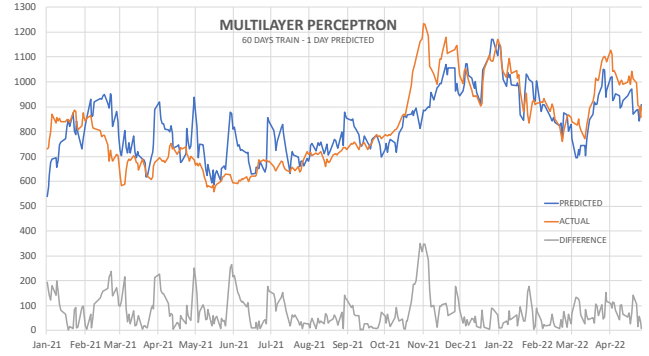


Fig. 6. MULTILAYER PERCEPTRON 60 DAYS TRAIN - 1 DAY PREDICTED.

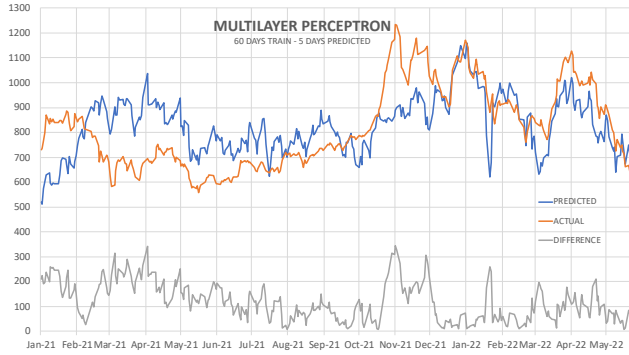


Fig. 5. MULTILAYER PERCEPTRON 60 DAYS TRAIN - 5 DAYS PREDICTED.

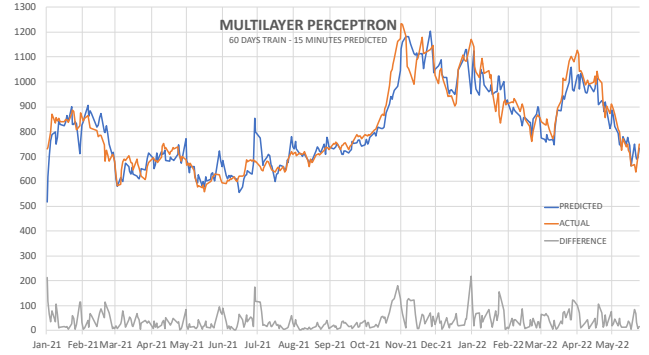


Fig. 7. MULTILAYER PERCEPTRON 60 DAYS TRAIN - 15 Mins PREDICTED.

and takes their majority vote for classification and average in case of regression. For this project, the RF models built have very simple structure with the following hyper-parameters: number of estimators = 100 and maximum depth = 100. The comparison between the actuals and predicted for all three scenarios is depicted in Figures 8–10.

#### D. Extreme Gradient Boosting Models

Extreme Gradient Boosting (XGBoost) is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of

simpler, weaker models. For this project, the RF models built have very simple structure with the following hyper-parameters: n of estimators = 100, maximum depth = 100, learning rate = 0.1, objective = reg:squarederror, and alpha = 10s. The comparison between the actuals and predicted for all three scenarios is depicted in Figures 11–13.

#### E. Models Comparison

Observations on the above would lead us to the following conclusions:

- 1) Models XGBoost and Random Forest have an excellent performance to predict a future stock value.
- 2) Consistent with our intuition, accuracy increases if the duration for prediction is shorter. For example,

TABLE III  
RESULT OF RF MODELS

Model	RMSE	MAPE	MAE	MTT
RF				
Train 60 days	65.598	5.9	45.649	1.278
Predict 5 days				
RF				
Train 60 days	39.958	3.7	27.417	1.150
Predict 1 day				
RF				
Train 60 days	13.788	1.0	7.132	1.138
Predict 15 minutes				

TABLE IV  
RESULT OF XGBOOST MODELS

Model	RMSE	MAPE	MAE	MTT
XGB				
Train 60 days	64.677	5.9	45.796	1.589
Predict 5 days				
XGB				
Train 60 days	39.649	3.7	27.676	1.550
Predict 1 day				
XGB				
Train 60 days	12.389	0.9	6.727	1.536
Predict 15 minutes				

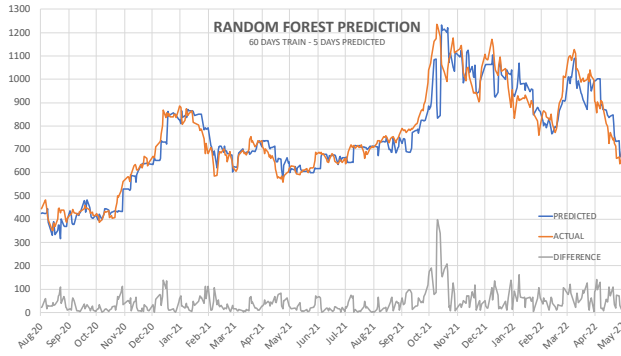


Fig. 8. RANDOM FOREST 60 DAYS TRAIN - 5 DAYS PREDICTED.

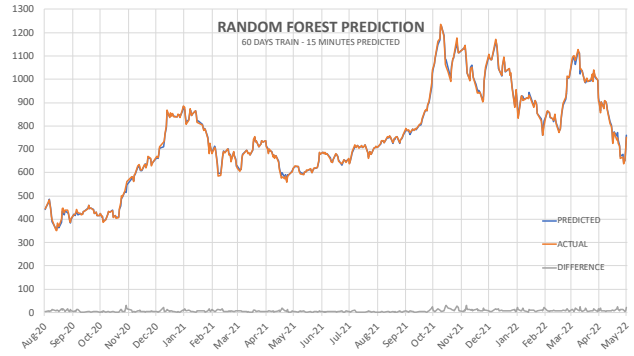


Fig. 10. RANDOM FOREST 60 DAYS TRAIN - 15 Mins PREDICTED.

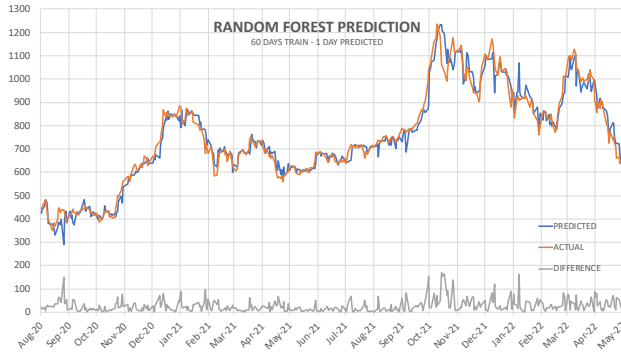


Fig. 9. RANDOM FOREST 60 DAYS TRAIN - 1 DAY PREDICTED.

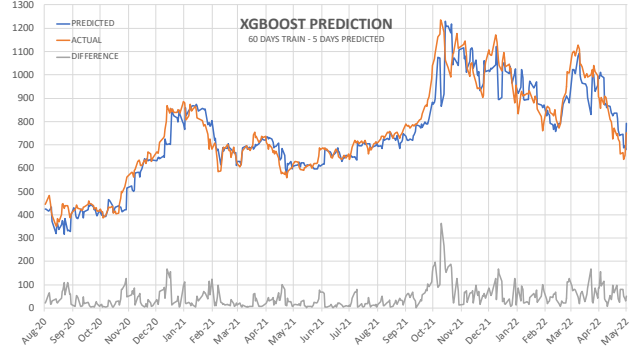


Fig. 11. XGBOOST 60 DAYS TRAIN - 5 DAYS PREDICTED.

predictions are more accurate in general if the trained model is used to generate predictions for the next days comparing to those for the next five days

- 3) Creating a prediction with any of these models takes less than 2 seconds by projection.
- 4) It is possible to predict the whole week of stock in 15 minutes in less than 3 minutes using Google Colab.

In summary, we have learned how a careful choice of input variables and a rational comparison of basic ML engines can produce higher and somehow *explainable* prediction quality than has been previously published. On-the-fly training is possible at our time scale, even with non-optimized implementations so this counter-intuitive aspect of our approach could also be novel.

## V. FUTURE WORK

The research will follow two defined paths. Building on the work presented here, fine-tuning some of the models and building more sophisticated machine learning models such as Long Short Term Memory (LSTM) models would be a natural extension. Adding other exogenous variables (features) such as short-term interest rate indicators (for example 2 years treasury bond prices), inflation related indicators (for example the price of gold) and sentiment data from social media and/or news reports would be another. Other possible and minor refinement would be the consideration of impact of dividend and other shareholder

related events. We will also investigate the possible use of parallel or distributed computing on training time relative to the frequency of our data collection and stock-price predictions, and also the size of our models.

## VI. CONCLUSIONS

In this paper, we presented results on forecasting the price of Tesla stocks using a historical price time series as well as several exogenous variables (features) that are considered relevant from a stock analysis standpoint. Four simple machine learning algorithms: support vector regression, multilevel perceptron, random forest, and XGBoost, were implemented to validate the appropriateness and accuracy of using these features for forecasting of stock prices. The outcome of this experiment confirms that this approach has merit even with machine learning models with simple structure. Future research would therefore focus on the inclusion of other relevant economic variables, such as inflation and short term interest rate, as well as sentiment data from social media and financial news sources. As well, the implementation of more sophisticated machine learning algorithms such as Long Short Term Memory would also be explored.

## REFERENCES

- [1] Y. Khmelevsky, "SW Development Projects in Academia," *WCCCE 2009 - Proceedings of the 14th Western Canadian Conference on Computing Education*, vol. 1, no. 250, pp. 60–64, 2009. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1536292>

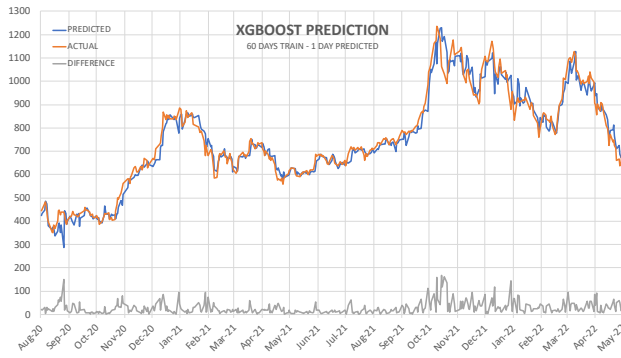


Fig. 12. XGBOOST 60 DAYS TRAIN - 1 DAY PREDICTED.

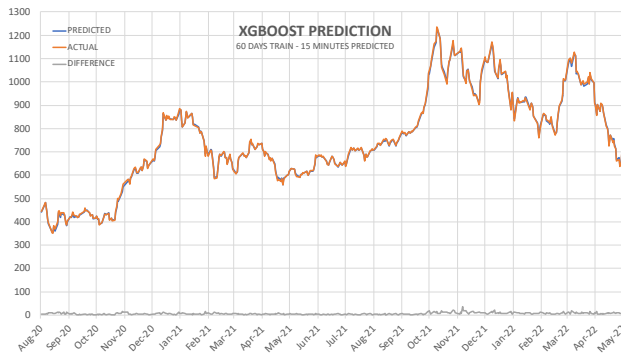


Fig. 13. XGBOOST 60 DAYS TRAIN - 15 Mins PREDICTED.

[2] Y. Khmelevsky, V. Ustimenko, G. Hains, C. Kluka, E. Ozan, and D. Syrotovsky, "International collaboration in SW engineering research projects," in *Proceedings of the 16th Western Canadian Conference on Computing Education - WCCCE '11*, 2011.

[3] Y. Khmelevsky and V. Voytenko, "Hybrid Cloud Computing Infrastructure in Academia," in *WCCCE 2015 - the 20th Western Canadian Conference on Computing Education, At May 8-9, 2015*. Vancouver Island University (VIU), Nanaimo, British Columbia, Canada., 2015.

[4] G. Hains, C. Li, N. Wilkinson, J. Redly, and Y. Khmelevsky, "Performance analysis of the parallel code execution for an algorithmic trading system, generated from UML models by end users," in *Parallel Computing Technologies (PARCOMPTECH), 2015 National Conference on*. IEEE, 2015, pp. 1–10.

[5] G. Hains, Y. Khmelevsky, R. Bartlett, and A. Needham, "Game private networks performance: Analytical models for very-large scale simulation," in *2016 IEEE International Conference on Cybercrime and Computer Forensic, ICCCF 2016*, 2016.

[6] Y. Khmelevsky, "Ten Years of Capstone Projects at Okanagan College: A Retrospective Analysis," in *Proceedings of the 21st Western Canadian Conference on Computing Education*. New York, NY, USA: ACM, 2016, pp. 7:1–7:6. [Online]. Available: <http://doi.acm.org/10.1145/2910925.2910949>

[7] B. Ward, Y. Khmelevsky, G. Hains, R. Bartlett, A. Needham, and T. Sutherland, "Gaming network delays investigation and collection of very large-scale data sets," in *11th Annual IEEE International Systems Conference, SysCon 2017 - Proceedings*, 2017.

[8] Y. Khmelevsky, K. Chidlow, K. Sugihara, and K. Zhang, "Engaging and Motivating Students Through Programming Competitions and GIS Applied Research Projects," *Proceedings of the 22nd Western Canadian Conference on Computing Education*, 5 2017. [Online]. Available: <http://dx.doi.org/10.1145/3085585.3088491>

[9] Y. Khmelevsky, X. Li, and S. Madnick, "Software development using agile and scrum in distributed teams," in *11th Annual IEEE*

*International Systems Conference, SysCon 2017 - Proceedings*, 2017.

[10] G. Hains, A. Jakobsson, and Y. Khmelevsky, "Towards formal methods and software engineering for deep learning: Security, safety and productivity for dl systems development," in *12th Annual IEEE International Systems Conference, SysCon 2018 - Proceedings*, 2018.

[11] G. Hains, C. Mazur, J. Ayers, J. Humphrey, Y. Khmelevsky, and T. Sutherland, "The WTFast's Gamers Private Network (GPN®) Performance Evaluation Results," in *2020 IEEE International Systems Conference (SysCon)*. IEEE, 2020, pp. 1–6.

[12] C. Mazur, J. Ayers, J. Humphrey, G. Hains, and Y. Khmelevsky, "Machine Learning Prediction of Gamer's Private Networks (GPN®S)," in *Proceedings of the Future Technologies Conference*. Springer, 2020, pp. 107–123.

[13] A. Wong, C. Chiu, G. Hains, J. Behnke, Y. Khmelevsky, and C. Mazur, "Network Latency Classification for Computer Games," in *The IEEE International Conference on Recent Advances in Systems Science and Engineering (submitted)*, 2021.

[14] A. Wong, C. Chiu, G. Hains, J. Humphrey, Y. Khmelevsky, C. Mazur, and H. Fuhrmann, "Gamers Private Network Performance Forecasting - From Raw Data to the Data Warehouse with Machine Learning and Neural Nets," 2021. [Online]. Available: <https://arxiv.org/abs/2107.00998>

[15] 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.

[16] A. Wong, C. Chiu, A. Abdulgapul, M. N. Beg, Y. Khmelevsky, and J. Mahony, "Estimation of Hourly Utility Usage Using Machine Learning," in *SysCon 2022 (accepted for the publication)*, 2022.

[17] I. Parmar, N. Agarwal, S. Saxena, R. Arora, S. Gupta, H. Dhiman, and L. Chouhan, "Stock Market Prediction Using Machine Learning," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 12 2018, pp. 574–576.

[18] Y.-G. Song, Y.-L. Zhou, and R.-J. Han, "Neural networks for stock price prediction," 2018. [Online]. Available: <https://arxiv.org/abs/1805.11317>

[19] S. Chen and H. He, "Stock Prediction Using Convolutional Neural Network," *IOP Conference series: materials science and engineering*, vol. 435, p. 012026, 11 2018. [Online]. Available: <https://doi.org/10.1088/1757-899x/435/1/012026>

[20] R. Gómez-Martínez, C. Prado-Román, M. de la Cruz, and others, "ALGORITHMIC TRADING SYSTEM ON NASDAQ TWEETS," *Review of General Management*, vol. 29, no. 1, 2019.

[21] B. Weng, L. Lu, X. Wang, F. M. Megahed, and W. Martinez, "Predicting short-term stock prices using ensemble methods and online data sources," *Expert Systems with Applications*, vol. 112, pp. 258–273, 12 2018.

[22] X. Li, P. Wu, and W. Wang, "Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong," *Information Processing & Management*, vol. 57, no. 5, p. 102212, 9 2020.

[23] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long-short term memory neural network based on attention mechanism," *PLoS ONE*, vol. 15, no. 1, 2020. [Online]. Available: <https://doi.org/10.1371/journal.pone.0227222.g001>

[24] Li Qing Wang, Jun Tan, Jinghua Jiang, and Li Ling, "Stock price fluctuation prediction model and device based on media information tensor supervised learning," patents.google.com, Chengdu, China, 2021.

[25] Liu Peng, Zhang Zhen, Qiang Zhang, and Kun Yan Lele, "Stock price prediction method combining network public sentiment and dependency," patents.google.com, China, 2021.

[26] 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)*, 4 2018, pp. 1003–1007.