Evaluating machine learning algorithms for stock market prediction

Daiki Akiyoshi
Department of Computer Science
Earlham College
Richmond, Indiana, 47374
dakiyo14@earlham.edu

ABSTRACT

Stock market prediction and developing profitable trading strategies have always attracted businesses and academia, and many studies have been conducted in the field to solve the puzzle of stock markets. Some prediction methods use machine learning, which involves processing a vast amount of information that might affect stock markets. Previous studies demonstrated high accuracy in predicting directional movements of stock prices. However, achieving high directional accuracy does not directly translate into great profits, as this ignores the magnitude of stock price fluctuations. There have been studies that address this limitation by predicting stock prices themselves instead of just directional movements. However, studies vary in the evaluation metrics they use, and thus it becomes problematic to compare them effectively. To address this limitation, this study evaluated predictive models based on three metrics: directional accuracy, closeness, and profit generated by trading simulation. These three metrics allow for effective comparison of machine learning algorithms and help determining the potential applicability of machine learning in predicting stock prices in a real setting.

KEYWORDS

Stock prediction, artificial neural network, support vector machine

1 INTRODUCTION

Stock market prediction and developing profitable trading strategies have attracted attentions not only businesses but from academia [1]. Many studies have been conducted in the field to solve the puzzle of stock markets. Some prediction methods use machine learning, which involves processing a vast amount of information that might affect stock markets.

Previous studies demonstrated high accuracy in predicting directional movements of stock prices. For instance, Patel et al. [12] and Leung et al. [9] achieved directional prediction accuracies of 90%. However, achieving high directional accuracy does not directly translate into significant profits, as this neglects the magnitude of stock price fluctuations [17]. There have been studies that address this limitation by predicting stock prices themselves instead of just directional movements. However, each study used a different evaluation method to assess their algorithms, and thus, it is difficult to effectively compare them.

This study addresses this limitation by evaluating predictive models based on three metrics: (1) directional accuracy, (2) closeness, and (3) profit generated by trading simulation as suggested by Xing et al. [17]. For the third metric, I have created a trading simulator, which generates buy and sell signals based on prices predicted by machine learning algorithms. In particular, this study focuses on comparing prediction performances of the two most popular ML algorithms in the field of stock prediction: artificial neural networks (ANN) and support vector machines (SVM) [18].

The remainder of this paper is organized into four sections. Section 2 provides a brief overview of the related work. Section 3 describes the methodology of the project. Section 4 includes the experimental results and discussions on these results. Section 5 concludes the paper with suggestions for the possible future areas of study.

2 RELATED WORK

This section introduces existing research on two widely-used machine learning algorithms for stock prediction. The first method is ANN, and the second is SVM. So far as I have been able to determine, the majority of studies in the field are focused on predicting stock price directional movements rather than predicting prices themselves. Furthermore, previous studies evaluated their algorithms based on one to two metrics, and not the three metrics that will be used in this project. Table 1 summarizes the evaluation measurements used for each work.

Table 1: Measurements used for evaluation

Authors	Measurements	Algorithms
Kimoto et al. [8]	Trading simulation	ANN
Sezer et al. [13]	Directional accuracy, Trading simulation	ANN
Patel et al. [12]	Directional accuracy	ANN, SVM
Shen et al. [14]	Directional accuracy	SVM

2.1 Artificial Neural Networks

ANN are used in some of the most accurate and widely used forecasting models, and they have been applied in various types of forecasting tasks [6]. ANN have become popular forecasting methods due to the four distinguishing features [6] [10]. First, ANN are data-driven methods in that they do not need explicit assumptions on the models between inputs and outputs. The second feature of ANN is their generalization ability, which refers to their strength in recognizing new patterns even if the sample data contain noisy information. Third, ANN are universal function approximators which can approximate any continuous function. The fourth feature of ANN is nonlinearity. Nonlinearity refers to the capability of ANN to identify complex nonlinear relationships between input and output datasets. Soni emphasizes that ANN's ability to capture nonlinear relationships is often preferred over linear regressions because many real-life prediction problems are also nonlinear [15]. However, ANN also have limitations. For instance, it is difficult to determine the optimal configuration of the network structures, such as determining the number of nodes and hidden layers [10].

Kimoto et al. proposed a system that recommends timing for when to buy and sell the Tokyo Stock Exchange Price Index (TOPIX) based on modular neural networks [8]. They used six inputs for their model, including interest rates, Dow Jones Industrial Average (DJIA), and foreign exchange rate. They evaluated the prediction performance using a trading simulation. In their simulation, they compared two trading models. The first model is based on the buy-and-hold strategy in which a stock is bought at the beginning of the trading simulation and held until the end. The second model is based on the buy-and-sell strategy in which a stock is bought when the model triggers a buy signal and a stock is sold when the model triggers a sell signal. The results showed that the buy-and-sell strategy made a greater profit than the buy-and-hold strategy. The buy-and-sell strategy was implemented as one-point buying and selling strategy, in which all available money is used to buy stocks and all stocks held are sold at a time. This research was one of the earliest studies that tested ANN-based algorithmic trading in a real environment [15].

Sezer et al. implemented a multilayer perceptron (MLP) ANN to predict the ups and downs of the DJIA index [13]. First, they converted the financial time series data into a series of buy-sell-hold trigger signals based on the three financial technical analysis indicators: Relative Strength Index (RSI), Moving Average Convergence and Divergence (MACD), and Williams %R. These signals were used as the input into the model. The MLP had four layers that consist of four nodes in the input layer, five nodes in the second layer, and four nodes in the third layer and three nodes in the output layer (one for each output class: Buy, Hold, and Sell). The model was trained with the data from 1997 to 2007 with 200 epochs and tested with data from 2007 to 2017. They evaluated their predictive model using directional accuracy and a trading simulation. The results showed that the buy-and-sell strategy based on the predictive model outperformed a buy-and-hold strategy.

2.2 Support Vector Machine

Support Vector Machines (SVM) are supervised machine learning algorithm, which construct the maximum margin hyperplane to classify input into binary groups [3]. In stock price prediction, researchers use SVM to conduct binary classifications of whether the price is going up or down. The SVM's ability to classify input into binary groups with great accuracy is a key reason why many researchers implement SVM for their stock prediction models. Moreover, SVM can be also used as a regression method as implemented in this project. Other characteristic of SVM is that it is resistant to the overfitting problem and can achieve a high generalization performance.

Shen et al. implemented SVM to predict the daily trend of NASDAQ, S&P 500, and DJIA [14]. Unlike research that used local data, such as local interest rate and local stock indexes, they used global stock data, commodity prices, and foreign currency data as input. This approach is based on their assumption that the data of overseas stocks as well as other financial products should correlate with the US stock market. They evaluated their algorithm based on directional accuracy and trading simulations. The results showed that their model outperformed a benchmark buy-and-hold strategy. This work's contribution is that it demonstrated that using global data in the studied time period demonstrated a better result than only using local data.

Patel et al. implemented and compared four different machine learning algorithms to predict the daily movements of CNX Nifty and S&P Bombay Stock Exchange Sensex, either going up or down. These four methods include SVM, ANN, random forest, and naive-Bayes [12]. They used ten technical financial parameters as input into their models, such as open prices, close prices, and simple moving averages. For the dataset, the authors used ten years of historical data from 2003 to 2012. Their main contribution is that they converted the ten continuous financial parameters into trend deterministic data, +1 or - 1. Like many other studies, they evaluated their ML algorithms with the directional accuracy metric. The result showed that the accuracy of 86.69%, 89.33%, 89.98%, and 90.19% was achieved by ANN, SVM, random forest and naive-Bayes.

My capstone project implemented ANN and SVM based on the ten financial parameters suggested by Patel et al [12]. The contribution of this project is that this study focused on predicting the values of stock price instead of predicting the directional movements (up or down) of stock prices. Moreover, this study evaluated stock price predictive models through three measurement metrics: directional accuracy, closeness, and trading simulation.

3 DESIGN

This project focuses on comparing prediction performances of ANN and SVM based on three evaluation metrics: accuracy, closeness, and trading simulation. Moreover, this study focuses on predicting the stock prices of distant future including 1, 3, 5, 10, and 20 days in advance, instead of only predicting the values of close future. This section describes the project design in three sub-sections: research data, predictive models, and evaluation methods. The project framework is shown in Figure 1. The training set was used to train ANN and SVM-based predictive models, and the test set was used to test the models with three metrics: directional accuracy, closeness and trading simulation.

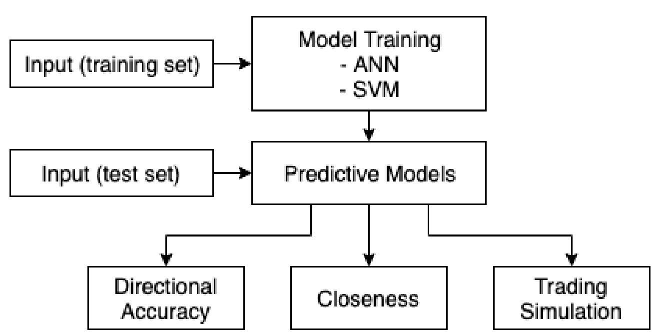

Figure 1: Project framework

3.1 Research Data

This study used 19 years of data on S&P 500 Index from January 2000 to January 2019. All the data was obtained from Alpha Vantage API, which provides stock price data in an organized CSV format free of charge [16]. The collected data was pre-processed to generate ten technical financial indicators used as input into ANN and SVM models. These financial indicators are widely used in the literature [5] [7] [12]. These technical indicators are calculated based on the formulas as described in Table 2. 90% and 10% of the data was used for training and testing respectively.

Table 2: Technical indicators and their formulas [12]

Name of Indicators	Formulas
Simple 10-day moving average (SMA_{10})	$\frac{C_t+C_{t-1}+\cdots+C_{t-10}}{10}$
Weighted 10-day moving average	$\frac{((n)\times C_t+(n-1)\times C_{t-1}+\cdots+C_{10})}{(n+(n-1)+\cdots+1)}$
10-day Momentum	$\frac{C_t-C_{t-10}}{C_{t-10}}$
Stochastic K%	$\frac{C_t-LL_{14}}{HH_{14}-HH_{14}} \times 100$
Stochastic D%	$\frac{K_t\%+K_{t-1}\%+K_{t-2}\%}{3}$
RSI (Relative Strength Index)	$100-\frac{100}{1+RS}$
MACD	$EMA_{12}-EMA_{26}$
Larry William's R%	$\frac{HH_{14}-C_t}{HH_{14}-LL_{14}} \times 100$
A/D Oscillator	$A/D_{t-1}+\frac{((C_t-L_t)-(H_t-C_t))}{H_t-L_t}$
CCI (Commodity Channel Index)	$\frac{M_t-SMA_{20}}{0.015D_t}$

C_t is the close price, H_t and L_t are the high and low prices at time t, HH_{14} and LL_{14} mean highest high and lowest low in the last 14 days, respectively, RS is average gain of last 14 trading days divided by average loss of last 14 trading days, EMA is exponential moving average, M_t: $H_t + L_t + C_t$ / 3, and C_t is the mean deviation.

3.2 Prediction models

This section describes the two machine learning algorithms used in this project.

3.2.1 *ANN.*

This project implemented a three-layered MLP ANN based on the work by Patel et al. [12]. Inputs for the ANN are ten financial technical indicators that are represented by ten neurons in the input layer. The output layer of the ANN consists of one neuron that represents the price of stock prices. The ANN model parameters that must be determined are the number of hidden layer neurons (n), value of learning rate (lr), momentum constant (mc), and number of epochs (ep). To set the parameters efficiently, ten levels of n, nine levels of mc, and ten levels of ep were tested. Initially, the value of lr is fixed to 0.1. The ANN model parameters and their levels are summarized in Table 3. Based on the experiment, the ANN model with 80 neurons, 1000 epochs, and momentum constant of 0.2 performed the best in terms of a mean absolute percentage error measure as calculated by Equation 2. Thus, this project uses hyper-parameters in the baseline ANN model.

3.2.2 *SVM.*

In this study, the polynomial kernel and the radial basis function were used as the kernel functions of SVM, as suggested in the work by Patel et al. [12]. Several levels of the degree of kernel function of polynomial function (d), gamma constant of radial basis function (γ) , and regularization constant c were tested in the parameter settings [12]. To set the parameters efficiently, four levels of d, ten levels of γ, and four to five levels of c were tested. The SVM model parameters and their levels are summarized in Table 4. Based on the experiment, the SVM model with polynomial kernel, degree of 1, and regularization parameter of 1 performed the best in terms of a mean absolute percentage error measure as calculated by Equation 2.

Table 3: ANN parameter tested in parameter setting [12]

Parameters	Level(s)
Number of neurons (n)	10, 20, ... , 100
Epochs (ep)	1.000, 2.000, ... , 10.000
Momentum constant (mc)	0.1, 0.2, ... , 0.9
Learning rate (lr)	0.1

Table 4: SVM parameter tested in parameter setting [12]

Parameters	Level (polynomial)	Level (radial basis)
Degree of kernel function (d)	1,2,3,4	-
Gamma in kernel function (γ)	-	0.5, 1, 1.5, ... ,5, 10
Regularization parameter (c)	0.5, 1, 5, 10, 100	0.5, 1, 5, 10

3.3 Evaluation methods

This project evaluates prediction models based on three different metrics: accuracy, closeness, and trading simulation. The idea of using three different evaluation metrics are based on the work by Xing et. al [17].

3.3.1 *Directional accuracy.*

The first measurement is a directional accuracy which evaluates prediction models based on whether a selected stock price will go up or down. The directional accuracy was calculated based on four parameters: true positive (TP), false positive (FP), true negative (TN), and false negative (FN) [12]. TP indicates the cases where a prediction model predicts a positive movement of stock price correctly, whereas TN indicates the cases where a prediction model predicts a negative movement correctly. On the other hand, FP indicates the cases where a prediction model predicts that a stock price will go up, but the price, in fact, goes down. FN indicates the cases where a prediction model forecasts that a stock price will go down, but the price actually goes up. The directional accuracy is calculated using Equation 1.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

3.3.2 *Closeness.*

The second measurement is closeness between the predicted prices and the corresponding real-world prices. This project calculated the closeness based on the average Mean Absolute Percentage Error (MAPE) used by Bollen et. al [2]. The closeness is calculated using Equation 2.

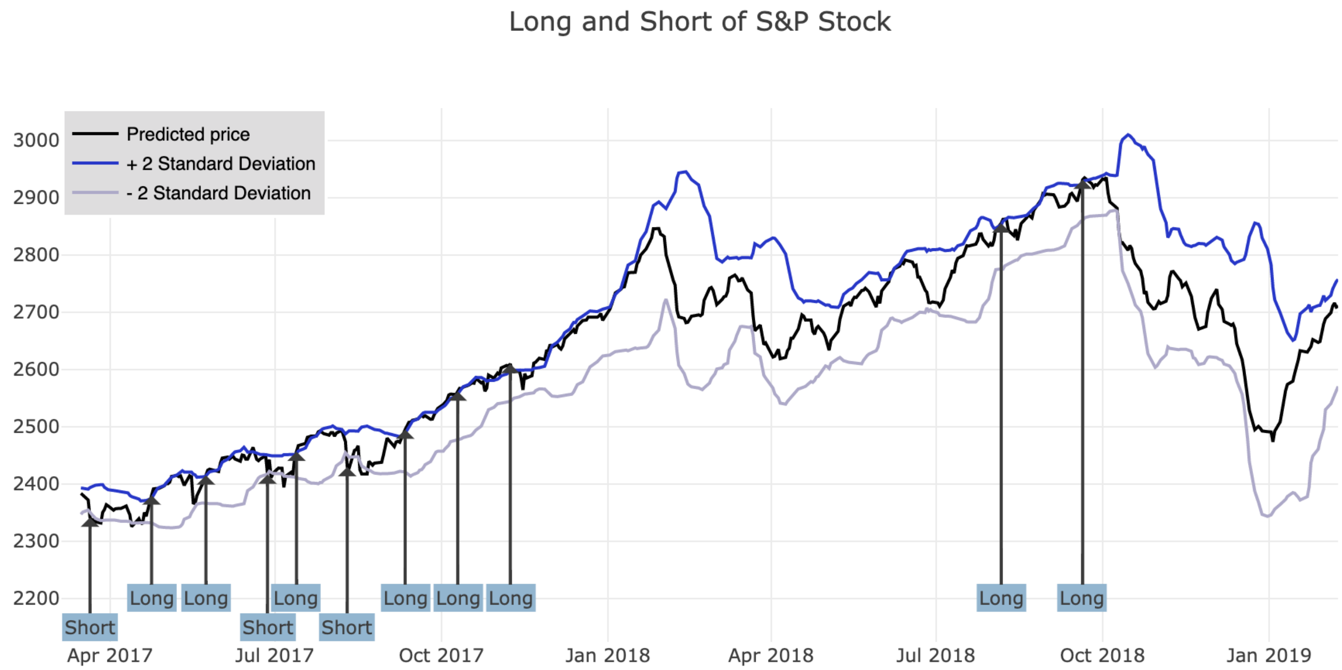

Figure 2: Trading simulation ANN 20-day ahead predictions

$$MAPE = \frac{1}{n} \sum \left| \frac{Actual - Forecast}{Actual} \right| \quad (2)$$

3.3.3 Trading simulation.

The third measurement is trading simulation results generated by stock predictive models. An algorithm may achieve a high directional accuracy, but that does not guarantee that the strategy can make a profit. Therefore, testing predictive models on trading simulations provides a better understanding of whether these models work in a real-world situation.

This project implemented Bollinger Bands trading strategy to determine when to buy and sell S&P 500 Index [4]. A Bollinger band is a financial technical analysis tool developed by John Bollinger in the 1980s. Bollinger Bands consists of two bands, the upper band and lower band. The upper and lower bounds are constructed by adding or subtracting standard deviations from a simple moving average, respectively. In this project, the upper and lower bands are defined as two standard deviations above or below from a 20-day simple moving average.

These upper and lower bands were used to generate buy and sell signals based on the predicted S&P 500 Index prices. A point where predicted price is above both the upper band and the actual close price indicates a long (buy) signal as the predicted price is significantly higher (two standard deviations) than the normal price. A point where predicted price is below both the lower band and the actual close price indicates a short (sell) signal as the predicted price is significantly lower (two standard deviations) than the normal price. Theoretically, the trading signal generated based on these conditions never incur losses if prices are perfectly predicted. Therefore, using these conditions allows this study to purely assess the profit performance of the predictive models.The predictive models are assessed based on the absolute value of the difference between the predicted profit and the actual profit. This shows how accurately the predictive models can forecast profits (the predicted profit accuracy). Thus, a smaller difference indicates that the predictive model accurately predicts profit.

Figure 2 shows the examples of trading signals generated by the Bollinger Bands when the ANN model predicts the S&P 500 Index prices 20 days in advance. The black line indicates predicted closing price 20 days in advance: that is, the price shown on April 1st, 2017 on the graph is the forecasted price for 20 trading days later. The blue and grey lines show the upper and lower bands of Bollinger Bands, respectively. As seen in Figure 2, the points where 20-day predicted prices are above the upper band indicate long signals, whereas the points where 20-day predicted prices are below the lower band indicate short signals. The long and short positions are assumed to be closed after 20 days of the trades: that is, if the trading system buys a stock today, the stock will be automatically sold 20 days later. Repeated long and short signals within a 20-day period were discarded in order to ignore repeated signals generated due to small price fluctuations around the Bollinger Bands.

$$\text{Return} = (R_l(\text{return from long}) + R_s(\text{return from short})) \quad (3)$$

where:

R_l = ln((Total long prices at T+n)/(Total long prices at T))
R_s = ln((Total short prices at T)/(Total short prices at T+n))
n = number of days a model predicts ahead of time

Based on trading signals, two types of percentage returns are calculated with Equation 3. For instance, Table 5 shows 11 trades that were generated based on the ANN 20-day ahead predictions,

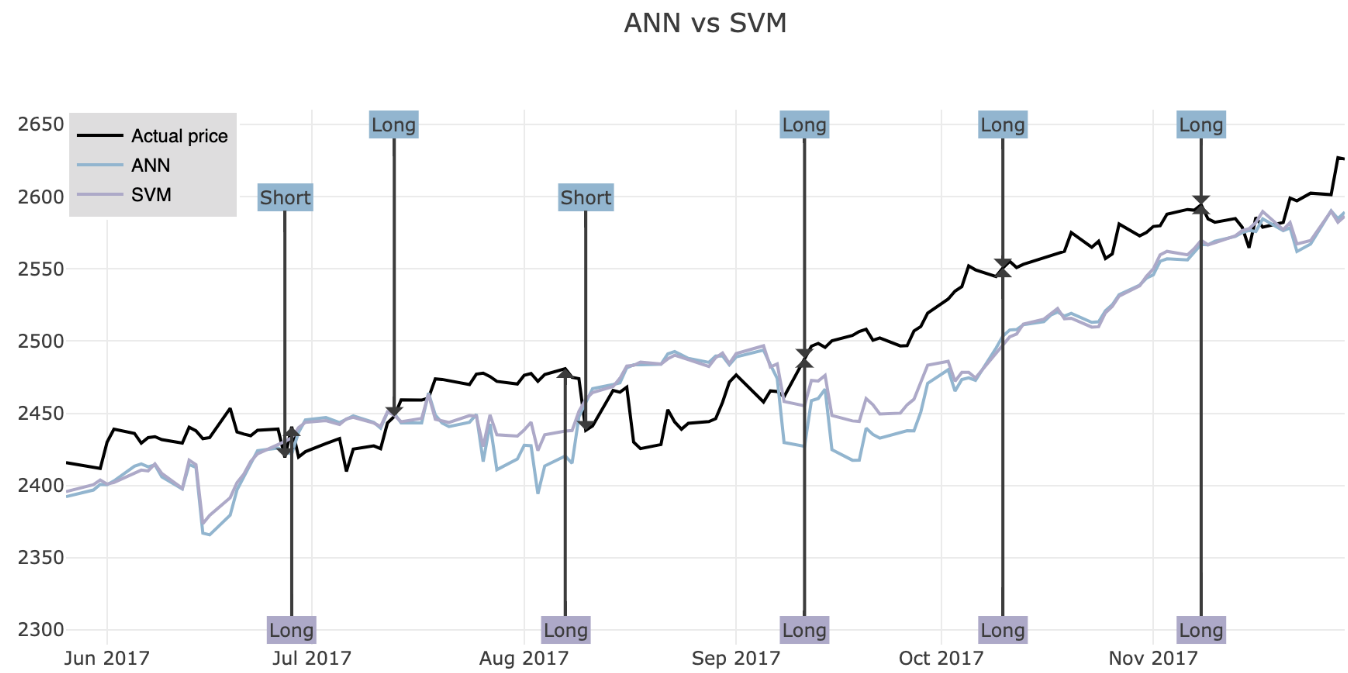

Figure 3: ANN vs SVM for 20-day ahead predictions

Table 5: Trade examples for ANN 20-day ahead predictions

Date	Long/Short	P at T	Predicted (T+20)	Actual (T+20)
2017-03-21	**Short**	2344.02	2341.54	2338.17
2017-04-24	**Long**	2374.15	2380.25	2394.02
2017-05-24	**Long**	2404.39	2415.16	2434.50
2017-06-27	**Short**	2419.38	2416.58	2477.83
2017-07-13	**Long**	2447.83	2456.56	2438.21
2017-08-10	**Short**	2438.21	2429.78	2461.43
2017-09-11	**Long**	2488.11	2495.16	2544.73
2017-10-10	**Long**	2550.64	2561.63	2590.64
2017-11-08	**Long**	2594.38	2609.73	2636.98
2018-08-06	**Long**	2850.40	2855.06	2896.72
2018-09-20	**Long**	2930.75	2931.40	2768.78
Total(Long)	-	20640.65	20704.96	20704.58
Total(Short)	-	7201.61	7187.90	7277.43

with the dates of trade, whether long or short, prices on the initial trades day (P at T), predicted prices 20 trading days later (Predicted P at T + 20), and actual prices 20 trading days later (Actual P at T + 20). Based on Equation 3 and trades in Table 5, the return for ANN 20-day ahead predictions is calculated as follow.

$$
\begin{aligned}
R_l &= \ln(20704.96/20640.65) &= 1.0031\% \\
R_s &= \ln(7201.61/7187.90) &= 1.0019\% \\
Return &= (1.0031 + 1.0019) &= 2.0050\%
\end{aligned}
$$

4 EXPERIMENT RESULTS

Table 6 and Table 7 show the experiment results for ANN and SVM, respectively. Figure 2 shows the predicted prices and trading signals generated by both ANN and SVM from the June 2017 to December 2017 period. As seen in Figure 3, the predictions by these model similarly predict prices. However, it also shows that the small deviations in the predictions between these two models have led to different trading signals. For instance, ANN generated a short signal, while SVM generated a long signal at around early August 2017.

In terms of directional accuracy, ANN performed better for 1-day, 3-day, and 10-day ahead predictions, while SVM performed better for 5-day and 20-day ahead predictions. The highest directional accuracy was 71.46 percent for 20-day ahead predictions by SVM. However, these results are lower than the ones reported in other studies such as 90 percent directional accuracy achieved by Patel et al. [12]. This difference is not surprising as the study by Patel et al. focused on predicting binary classifications of whether stock prices are going up or down; that is, their ML models were optimized so that they were able to predict directional movements of stock prices accurately. On the other hand, this current study focused on predicting stock prices rather than predicting directional movements. Therefore, the ML models were optimized so that they could predict stock prices more accurately. This difference in the focuses of these studies might explain the varying directional accuracy performances.

In terms of MAPE, SVM outperformed ANN in all of the five different predictions. For both ANN and SVM, MAPE was increasing as the models predicted for the distant future values. In other words, the closeness of the predicted values to the actual values were becoming wider as the models attempted to predict further distant future values. This result is as expected given how the ML models predict future values. For instance, a model predicts a stock price at T + 1 given the input at time-point T. Similarly, when a model predicts a stock price at T + 20, it also uses the input at time-point

T. In other words, there are more missing days between the date of input and the date the model attempts to predict. Therefore, MAPE becomes larger as there are more missing days between T and T + n. The MAPE results that the ANN and SVM yielded are comparable with other studies. For instance, an ANN implemented by Naeini et al. [11] achieved MAPE of around 1 percent in predicting the next day's price.

In terms of profit prediction, ANN performed better for 1, 3, 10-day ahead predictions, while SVM performed better for 5 and 20-day ahead predictions. This pattern is consistent with the directional accuracy metric, which may indicate that higher directional accuracy leads to a higher accuracy in predicting profits. Besides this, there seems no pattern in the performance difference between ANN and SVM across distant future predictions, and the predicted profit accuracy gained through this experiment seem arbitrary and did not yield great results. This result suggests that the ANN and SVM-based predictive models may be able to predict stock prices and directional movements in reasonable accuracy, however, they are not suitable for predicting profits in the real world.

Table 6: Experiment Results for ANN

	1-day	3-day	5-day	10-day	20-day
Directional accuracy	51.88%	59.66%	57.17%	66.52%	57.52%
MAPE	0.83%	1.26%	1.36%	1.81%	2.54%
Predicted Return (P)	0.59%	0.81%	0.52%	0.83%	0.50%
Actual Return (A)	-0.14%	0.22%	-1.32%	-0.14%	-0.74%
|P - A|	0.73%	0.47%	1.84%	0.97%	1.24%

Table 7: Experiment Results for SVM

	1-day	3-day	5-day	10-day	20-day
Directional accuracy	50.0%	56.72%	59.07%	65.03%	71.46%
MAPE	0.66%	1.03%	1.31%	1.81%	2.45%
Predicted Return (P)	0.29%	0.53%	0.4%	0.33%	0.34%
Actual Return (A)	-0.78%	-0.48%	-1.22%	-2.01%	0.2%
|P - A|	1.07%	1.01%	1.62%	2.34%	0.14%

5 CONCLUSION

This study implemented ANN and SVM to predict S&P 500 Index prices and examined the predictive models based on three metrics: directional accuracy, MAPE, and trading simulation. The directional accuracy yielded from this experiment was lower than the previous studies which reported the accuracy of 90 percent. This difference is as expected because the previous studies focused on predicting directional movements of stock prices, and their ML models were optimized to do so. On the other hand, the models in this study were optimized so that they could predict stock prices more accurately. This difference in the focuses might explain the varying results on directional accuracy between the previous studies and this research. In terms of MAPE, the results the predictive models yielded are comparable with previous studies. The result showed that SVM outperformed ANN in all of the five different predictions. Finally, the trading simulation showed that ANN and SVM did not predict profits accurately, which suggests that they may not be suitable for predicting profits in the real world. As a future area of study, it will be an interesting to explore why ANN outperformed in some predictions and SVM outperformed for other predictions. As seen in Figure 3, ANN and SVM similarly predicted prices overall but with some deviations. Analyzing and understanding the differences require extensive knowledge in these machine learning algorithms and statistics used behind them. Therefore, this type of analysis is not reported in the current study since it is beyond the scope of this capstone project. However, understanding the differences may help to develop better predictive models that could be used in real life applications.

ACKNOWLEDGMENTS

I would like to thank Ajit Chavan and David Barballa for providing me detailed feedback to improve this research. I also would like to thank Xunfei Jiang for providing me feedback on topic proposal in previous semester.

REFERENCES

[1] George S. Atsalakis and Kimon P. Valavanis. 2013. *Surveying stock market forecasting techniques - Part I: Conventional methods.* 49–104.

[2] Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science* 2, 1 (2011), 1 – 8. https://doi.org/10.1016/j.jocs.2010.12.007

[3] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. 2005. Forecasting Stock Market Movement Direction with Support Vector Machine. *Comput. Oper. Res.* 32, 10 (Oct. 2005), 2513–2522. https://doi.org/10.1016/j.cor.2004.03.016

[4] Investopedia. 2019. Bollinger Band Definition. https://www.investopedia.com/terms/b/bollingerbands.asp Accessed: 2019-03-13.

[5] Yakup Kara, Melek Acar Boyacioglu, and Ömer Kaan Baykan. 2011. Predicting Direction of Stock Price Index Movement Using Artificial Neural Networks and Support Vector Machines. *Expert Syst. Appl.* 38, 5 (May 2011), 5311–5319. https://doi.org/10.1016/j.eswa.2010.10.027

[6] Mehdi Khashei and Mehdi Bijari. 2010. An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with applications* 37, 1 (2010), 479–489.

[7] Kyoung-jae Kim. 2003. Financial time series forecasting using support vector machines. *Neurocomputing* 55, 1-2 (2003), 307–319.

[8] Takashi Kimoto, Kazuo Asakawa, Morio Yoda, and Masakazu Takeoka. 1993. Stock market prediction system with modular neural networks. In *Neural Networks in Finance and Investing.* 343–357.

[9] Carson Kai-Sang Leung, Richard Kyle MacKinnon, and Yang Wang. 2014. A Machine Learning Approach for Stock Price Prediction. In *Proceedings of the 18th International Database Engineering & Applications Symposium (IDEAS '14).* ACM, New York, NY, USA, 274–277. https://doi.org/10.1145/2628194.2628211

[10] Seyed Niaki, Akhavan Taghi, and Saeid Hoseinzade. 2013. Forecasting S&P 500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International* 9, 1 (Feb. 2013), 1. https://doi.org/10.1186/2251-712X-9-1

[11] Mahdi Pakdaman Naeini, Hamidreza Taremian, and Homa B. Hashemi. 2010. Stock market value prediction using neural networks. 132 – 136. https://doi.org/10.1109/CISIM.2010.5643675

[12] Jigar Patel, Sahil Shah, Priyank Thakkar, and K Kotecha. 2015. Predicting Stock and Stock Price Index Movement Using Trend Deterministic Data Preparation and Machine Learning Techniques. *Expert Syst. Appl.* 42, 1 (Jan. 2015), 259–268. https://doi.org/10.1016/j.eswa.2014.07.040

[13] Omer Berat Sezer, A. Murat Ozbayoglu, and Erdogan Dogdu. 2017. An Artificial Neural Network-based Stock Trading System Using Technical Analysis and Big Data Framework. In *Proceedings of the SouthEast Conference (ACM SE '17).* ACM, New York, NY, USA, 223–226. https://doi.org/10.1145/3077286.3077294

[14] Shunrong Shen, Haomiao Jiang, and Tongda Zhang. 2012. Stock Market Forecasting Using Machine Learning Algorithms. http://cs229.stanford.edu/proj2012/ShenJiangZhang-StockMarketForecastingusingMachineLearningAlgorithms.pdf

[15] Sneha J Soni. 2011. Applications of ANNs in Stock Market Prediction : A Survey. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.404.1909

[16] Alpha Vantage. 2018. Alpha Vantage. https://www.alphavantage.co/ Accessed: 2018-10-30.

[17] Frank Z. Xing, Erik Cambria, and Roy E. Welsch. 2018. Natural Language Based Financial Forecasting: A Survey. *Artif. Intell. Rev.* 50, 1 (June 2018), 49–73. https:

//doi.org/10.1007/s10462-017-9588-9

[18] P. D. Yoo, M. H. Kim, and T. Jan. 2005. Machine Learning Techniques and Use of Event Information for Stock Market Prediction: A Survey and Evaluation. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Vol. 2. 835–841. https://doi.org/10.1109/CIMCA.2005.1631572