Stock Market Prediction using Support Vector Regression to Account for Modern Events

Christopher M. Frutos, Andrew B. Garcia
ECE 5424 Advanced Machine Learning, Final Project
Virginia Polytechnic Institute and State University
Department of Electrical and Computer Engineering
Blacksburg, VA 24060
cfrutos@vt.edu
agarcia1296@vt.edu

***Abstract*—As of recently, the stock market has been heavily affected by recent events, such as the war in Ukraine, fluctuation in cryptocurrency, and the coronavirus pandemic. For investment companies, such as Fidelity or Vanguard, knowing the difference between what stocks will perform the best, and the ones that won't, is invaluable to their business plan. The research in this paper will be using publicly available data like stock market prices, commodities, and cryptocurrency. Our approach to achieving stock price prediction is a continuous problem, which is why we are choosing to implement a Support Vector Regression (SVR) model.**

***Index Terms*—Machine Learning, SVM, Stock Market, Investment, Finance**

I. Introduction

A. The Stock Market

The price of stocks in the stock market has always played a huge role in our daily lives. Investment companies, such as Fidelity or Vanguard, make a profit off of buying stocks, bonds, or other assets with investor money like our retirement savings account. The stock market we know today is a place where you can buy and sell shares of publicly traded companies. It is the biggest market in the world that deals with billions of dollars being traded every day. The stock exchange acts as an intermediary between people who want to invest in stocks and those who want to sell them. A company will list its shares on the stock exchange so that investors can buy or sell them at any time, anywhere in the world. Because of these features, stocks are a popular choice for people to invest large portions of their money (retirement or savings account) for the greatest potential of growth over a long period of time. The way people choose to interpret the market is rapidly changing, most notably, with the use of computer-aided decision making such as machine learning [1].

The stock market is constantly being affected by outside factors, such as news articles, social media, earnings reports, government policies, and global events. This is an important factor given the huge impact that the 2020 COVID virus pandemic caused the stock market to drop by nearly 33% in March of 2020 [2]. Currently, there are machine learning models being researched to digest and interpret media as good or bad signs for market prediction [3] [4]. All of these factors have led researchers to investigate the possibility of answering the question "Could we have known sooner?", and we are hoping to find that answer here in this project.

B. Support Vector Machine

SVM is a technique that has been widely used in the field of pattern recognition to find out the relationship between data points. It can be applied to many different fields like medicine, biology, and so on. In medicine, it can be used for finding out which drugs are more effective than others for a particular disease [5]. In biology, it can be used for finding out how certain genes work or what proteins do in cells [6]. In finance, it can be very effective when utilized for predicting stock prices [4] [7].

A Support Vector Machine (SVM) is a type of algorithm that classifies data by finding the hyperplane in an N-dimensional feature space that distinctly classifies the data points [8]. In order to determine the best hyperplane on a feature set, as shown in Figure 1, the algorithm iterates over all possible separations and picks the one with the largest margin represented as epsilon (ϵ).

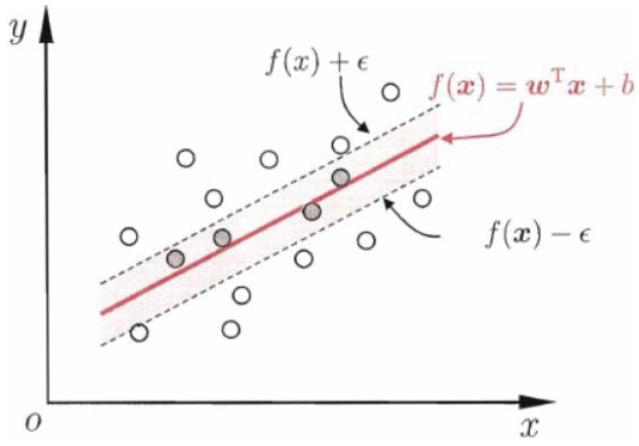

Fig. 1. Implementation principle of Support Vector Machines [8]

SVM's are powerful tools, however, our application is nonlinear which is why we have to use a support vector

regression (SVR). SVR takes the concept of classification and adds Lagrange multipliers to maximize its objective function as seen in Equation 1.

$$Q(\alpha) = \sum_{i=1}^{N} \alpha - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha j d_i d_j K(x_i, x_j) \qquad (1)$$

, where $K(x_i, x_j)$ represents the nonlinear inner-product kernel, and α represents the Lagrange multipliers. There are two inner-product kernels that we plan to test in this paper, (1) polynomial and (2) radial-basis function (rbf), to see if there is any significant difference for our problem. In addition to these SVR models, we are comparing them to a linear SVR model.

C. Objectives

The main goal of this paper is to explore and the practical use of a SVM prediction model of the S&P 500 based on key factors like developing technology and pop-culture trends. We use the S&P 500 as our target predictor because it is widely considered to be representative of overall market health. Related data, such as the price of commodities, cryptocurrency, and inflation will all be key factors integrated into our data set for training and testing. For our model, we will be using different types of Support Vector Regressions (SVR), which is a type of non-linear classification algorithm, that will help us predict future stock market prices. We chose to make predictions for 84 days because this is the equivalent to 3 months in business days (ie. starts on a Monday and end on the Monday 3 months away).

II. Methods

In this project, it is important to iterate over the entire data set to eliminate erroneous or missing data. Failure to do so will result in the program crashing or training over false data. All of the data used in this project, such as stocks, inflation rates, and commodities, were found online at publicly available websites. The data source subsection list will go over how and where we found our data online. Once our data files have been gathered from other websites, we then extract the necessary features from each file and combine them all into one data frame we call ”predictors”, as seen in Figure 2. For simplicity, we use the price at the market close of the S&P 500 as our target for the model to predict.

A. Data Source List

In this project, we are attempting to give the model a variety of sources for data to be utilized. We believe that a high level of diversity in the data will allow us to train an accurate model within a reasonable amount of resolution for the scope of this paper. Looking at the first data set on the list, we decided to include the S&P 500 as this will be the acting target variable for the model to ultimately train its prediction on; the S&P 500 tracks the performance of the 500 largest companies involved in the American stock exchange and is commonly used among investors. Next on the list is the index

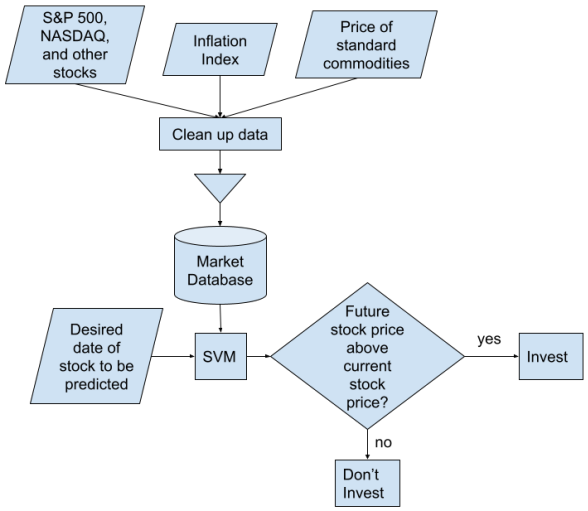

Fig. 2. High-level flowchart depicting the method for data collection and preparation for the SVM.

TABLE I
Data Set List

Index	Data Source Link
S&P500 Index (SPX)	https://www.investing.com/indices/us-spx-500-historical-data
Oil Index (BRN00)	https://markets.businessinsider.com/commodities/oil-price?type=brent
Cryptocurrency Index (BTC)	https://finance.yahoo.com/quote/BTC-USD/
Steel Index (NYSE:STEEL)	https://www.wsj.com/market-data/quotes/index/XX/STEEL/historical-prices
Silicon Index (TSM)	https://www.wsj.com/market-data/quotes/TSM/historical-prices
Defense Index (BA)	https://www.marketwatch.com/investing/stock/ba/
NASDAQ Index (NDAQ)	https://www.nasdaq.com/market-activity/stocks/csv/historical
NYSE Index (NYSE)	https://www.marketwatch.com/investing/stock/csv/download-data

for oil, we believe this has a significant weight to the economy of the United States given the size of the country and its larger consumption/import of oil; as the availability and price of oil changes, we believe this will have a great impact on the stock market. With the United States being the world-leading spender when it comes to the military [9], we also decided to put down the index of the leading company in defense which is Boeing (BA); in addition, we believe an important factor for both technology in and out of the defense economy are the silicon [10] (leading semiconductor manufacturer TSM) and steel [11] (NYSE American Steel index) indices, both of which are taken into account in this project. Last but certainly not least, since we are keeping the time range of all of our data to only the last five years, we believe above all things previously mentioned, Bitcoin [12], a cryptocurrency first started in 2009, has had the greatest rise to prominence and attention as far as investors in the American stock market

are concerned; conveniently, we believe the steel and silicon indices to also hold a close relationship and correlation with the state-of-health of cryptocurrency stocks. To also bolster our prediction of the S&P 500, we will be using the NASDAQ [13] and NYSE [14] indices, which are very similar to what the S&P 500 is indicating.

B. Data Preparation

In order to prepare the data for making stock price predictions, features must be added to the data frame that represent this future stock price. The features that were added, for every day that we will generate a prediction, will have the next three days in the future as an immediate trend for the model to train on, as well as the 84-days-in-the-future feature which will ultimately be the day that our model will decide its prediction for. For every dataset, (silicon, steel, Bitcoin etc...) we used around 20 different features: open, close, adjusted close, low, and high stock price. This set of features is then also provided for one, two, and three days in advance. In addition, the inflation rate is also included as a single feature; the target variable is listed as the close price for the S&P 500 for 84 days later.

C. Scoring Each Model

For both Polynomial and RBF, we will be adjusting the **regularization factor** (λ) which is the value for determining how general the model should be, represented in Equation 2,

$$min \sum_{i=1}^{n} V(f(x_i), y_i) + \lambda R(f) \tag{2}$$

where V is the underlying loss function that describes the cost of predicting $f(x)$. $R(f)$ is chosen to impose a penalty on the complexity of the function f [15]. The larger the regularization factor is the more general the model will be. It is easy to go with a low regularization factor and call the model done, however there comes a problem with over-fitting that must be addressed to account for future data. A popular method to prevent over-fitting is by implementing a cross-validation method. In this project, we use a 70/30 split of training to testing. The model is then trained on 70% of the data then tested on 30%.

There are 3 metrics we are using for scoring our models. First, Equation 3 shows the **coefficient of determination**, defined as,

$$\mathbb{R}^2 = (1 - \frac{u}{v}) \tag{3}$$

where u is the residual sum of squares `((y_true - y_pred)**2).sum()` and v is the total sum of squares `((y_true - y_true.mean())**2.sum()`. The best possible score for the coefficient of determination is 1, meaning it has a perfect prediction rate. [16]

Second, is the **mean squared error (MSE)** given in Equation 4 as,

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2 \tag{4}$$

where $\hat{y}_i$ is the predicted value of the i-th sample, and y_i is the corresponding true value, then the mean squared error (MSE) estimated over $n_{samples}$. [17]

Finally, we set an arbitrary metric for calculating **profits**. For example, we assume the user will want to invest $1,000 when the model says the value of the stock will go up. Equation 5 shows us how that is calculated:

$$Profit = \sum_{i=0}^{N} (\frac{\$1000}{current\_price_i}) * future\_price_i - \$1000 \tag{5}$$

where the summation of i consists of dates where the future stock price is predicted to be higher than the current stock price.

III. Results

To visualize the results better, we have to understand that whenever the prediction of the stock price (orange) as well as the actual future stock price (blue) are both above the current stock price (green), it would indicate that the user would have earned profits, assuming they trusted the prediction. Conversely, if the prediction (orange) sits below the current stock price (green), along with the actual future stock price (blue) then the model would have prevented the user from having a loss on an investment. The rule previously described applies to all resulting charts discussed, and is a core part of how we analyze our results throughout this paper. Ideally, the prediction model should be as close as possible to the actual future stock price to give the user the best outcomes. Telling the user that a stock is not as high as it really is can make the user lose out on investing opportunities or perhaps lose money on a shorting trading option. In short, having the predicted stock price as close as possible to the future stock price is what determines a good model.

A. SVR Polynomial

TABLE II
Polynomial SVR Model Scores

Reg. Factor	$\mathbb{R}^2$	MSE	Profit
1	0.5986	182987.3033	$25,310.64
0.1	0.7963	92849.1007	$30,768.64
0.01	0.8923	49085.8438	$44,787.08
0.001	0.9250	34181.7226	$48,704.88
0.0001	0.9434	25784.0102	$52,221.32
0.00005	0.9452	24997.6102	$52,450.90

Figure 3 shows the current stock price of the S&P 500 on any given day (green), what that stock's predicted value will be in 84 days (orange), and what the actual value of that stock will be 84 days in the future (blue). At last, the main determinant of how accurate our model will be lies in the distance between the real value of the stock in the 84 days' future (blue), and the prediction made by our model (orange). This analysis is the main focus for the rest of the graphs. In the case of our SVR polynomial, it appears to have a general understanding of the trend of the graph, but appears to lack the quickness to change

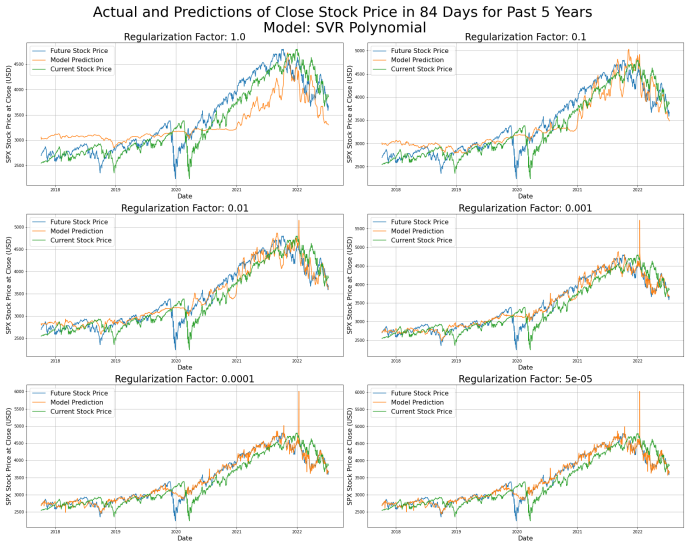

Fig. 3. Polynomial SVR prediction shown in orange.

its predictions, as well as the magnitude of change, based on the trends it is following. The best way to describe it would be that it lacks confidence when presented change, and is delayed in deciding to stray its predictions even when the current stock price has risen well above or below where it is at. Looking at all the provided figures, the regularization factor is minimized by approximately powers of 10. As the regularization becomes increasingly smaller, the prediction becomes more accurate. By the time the regularization factor has been reduced to 5e-05, the prediction model has mostly fully overlayed the actual future stock price except when there are sharp dips and rises in the current stock price. Right around when the regularization factor is turned to .01, can one *visually* see a significant amount of profitability in the charts.

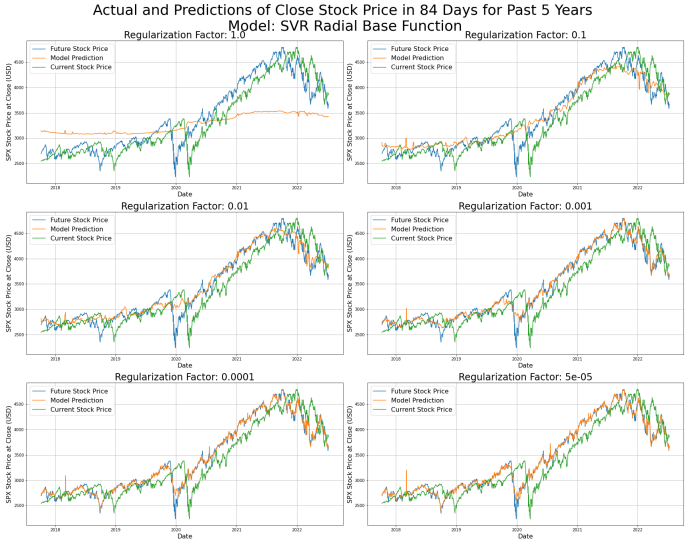

Fig. 4. Radial Base Function SVR prediction shown in orange.

Figure 4 Shows the Radial Base Function. This one has the least amount of accuracy when it comes to predicting the stock's price 84 days in the future. it can be seen that the model sits correctly within the direction of change of the stock price but not its magnitude. Right around the great dip shown in the beginning of 2020, the model is able to flatten out the increasing trajectory of the stock, but fails to capitalize on the growth shown after the drop. It can be seen that the SVR Radial Base Function model would have prevented the user from making any real profits in the entirety of the year 2021. that being said, it is not the worst outcome as it could have made the user lose a lot of money at the start of 2022 had it overshot it's influence with the growth found throughout 2021. However, when the regularization factor is reduced to 5e-05, the prediction model has fully overlaid the actual future stock price even when there are sharp dips and rises in the current stock price.

TABLE III
RADIAL BASED FUNCTION SVR MODEL SCORES

Reg. Factor	$\mathbb{R}^2$	MSE	Profit
1	0.3838	280893.9250	$28,645.25
0.1	0.9133	39524.0962	$45,207.40
0.01	0.9459	24643.6002	$51,163.43
0.001	0.9731	12281.5533	$52,850.36
0.0001	0.9844	7110.0923	$54,028.05
0.00005	0.9863	6243.5809	$54,168.51

IV. CONCLUSIONS

In conclusion, the Radial Base Function model had the best coefficient of determination score for a lower regularization factor and yielding the most profits at each iteration. RBF is very good at fitting to the markets quick ups and downs, whereas the polynomial model struggles to do the same. This project stands as an example of the power of modern technology and its influence on how financial decisions can be managed. SVR models are shown here to help make smart financial decisions by showing when is a good time to buy and when it's not. Future work could include the impact of positive and negative media on the target stock by using the latest in Python technology. This project was a great experiment for giving a deeper understanding to non-linear Support Vector Regressors.

REFERENCES

[1] M. Usmani, S. H. Adil, K. Raza and S. S. A. Ali, "Stock market prediction using machine learning techniques," 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), 2016, pp. 322-327, doi: 10.1109/ICCOINS.2016.7783235.

[2] R. Jindal, N. Bansal, N. Chawla and S. Singhal, "Improving Traditional Stock Market Prediction Algorithms using Covid-19 Analysis," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021, pp. 374-379, doi: 10.1109/ESCI50559.2021.9396887.

[3] Chen, J. (2021). Application of support vector machines and Holt-Winters Model in local finance forecast. 2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS). https://doi.org/10.1109/acctcs52002.2021.00076

[4] Shah, V. H. (2007). Machine learning techniques for stock prediction. Foundations of Machine Learning— Spring, 1(1), 6-12.

[5] P. Dhanush and N. Nalini, "Drug Review System Using Machine Learning by Comparing Linear Support Vector Machine with Naïve Bayes Classifier to Measure Accuracy," 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2022, pp. 1-5, doi: 10.1109/ICSES55317.2022.9914150.

[6] I. M. Kamal, N. A. Wahid and H. Bae, "Gene Expression Prediction using Stacked Temporal Convolutional Network," 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), 2020, pp. 402-405, doi: 10.1109/BigComp48618.2020.00-41.

[7] Zhu, Y. (2021). Research on financial risk control algorithm based on machine learning. 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI). https://doi.org/10.1109/mlbdbi54094.2021.00011

[8] Gandhi, R. (2018, July 5). Support Vector Machine - introduction to machine learning algorithms. Medium. Retrieved October 14, 2022, from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[9] Conte, N. (2022, August 18). Ranked: Top 10 countries by military spending. Visual Capitalist. Retrieved December 10, 2022, from https://www.visualcapitalist.com/ranked-top-10-countries-by-military-spending/

[10] Producer price index by industry: Semiconductors and related device manufacturing: Other semiconductor devices, including parts such as chips, wafers, and heat sinks. FRED. (2022, October 12). Retrieved October 14, 2022, from https://fred.stlouisfed.org/series/PCU334413334413A

[11] Journal, W. S. (n.d.). Steel — NYSE American Steel Index Historical Prices - WSJ. The Wall Street Journal. Retrieved October 14, 2022, from https://www.wsj.com/market-data/quotes/index/XX/STEEL/historical-prices

[12] Blockchain developer apis. (n.d.). Retrieved October 15, 2022, from https://www.blockchain.com/api

[13] NASDAQ - Market activity. Nasdaq. (n.d.). Retrieved October 20, 2022, from https://www.nasdaq.com/market-activity/stocks/csv/historical

[14] Download CSV data: Carriage Services Inc.. Price Data. MarketWatch. (n.d.). Retrieved October 20, 2022, from https://www.marketwatch.com/investing/stock/csv/download-data

[15] Wikimedia Foundation. (2022, November 27). Regularization (mathematics). Wikipedia. Retrieved December 9, 2022, from https://en.wikipedia.org/wiki/Regularization_(mathematics)

[16] Sklearn.svm.SVR. scikit. (n.d.). Retrieved December 9, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

[17] 3.3. metrics and scoring: Quantifying the quality of predictions. scikit. (n.d.). Retrieved December 9, 2022, from https://scikit-learn.org/stable/modules/model_evaluation.htmlmean-squared-error