# Predicting Stock Market Movement Direction with Support Vector Machines and Neural Networks

Richard Cloutier

Kate Gleason College of Engineering, Rochester Institute of Technology
Henrietta, Rochester, USA

rxc1028@rit.edu

***Abstract*—Predicting the stock market has been the end goal of every financial analyst, trader, and stockbroker since the opening of the stock market in 1817. Only recently have engineers and computer scientists joined them in their pursuit of predicting the unpredictable. In this project, various classification models and neural networks were used to predict future stock prices, with support vector machines (SVM) being heavily favored. The application of SVM classification with historical stock opening and closing prices on forecasted stock prices resulted in a test accuracy of 51.93% over 10 stocks. The application of forward propagating neural networks with historical stock opening and closing prices on forecasted stock prices resulted in a test accuracy of 50.71% over 10 stocks. Additionally, news articles were analyzed to determine correlations between key words and their impact on a company's stock price. The addition of the news article scanner resulted in a test accuracy of 52.18% and 52.33% for the SVM and neural network models respectively.**

## I. Introduction

PREDICTING the stock market is a complicated and daunting task, due to the seemingly random behavior of the market, the innumerable number of factors that brokers use to determine their buying and selling price of a stock, and the interconnectivity of those factors. While models solely using historical price data might be able to determine historical trends with a reasonable accuracy, the accuracy of those models drops significantly when introduced to new market data. This project attempts to alleviate this accuracy drop by introducing a new feature to the dataset; predictions based off of news articles. By analyzing news articles, specifically the words used in said article, the model will be able to predict and react to a price fluctuation before it happens. This will cause a significant increase in the models accuracy immediately following the release of a news article, which typically correlate to strong fluctuations in stock prices.

The rest of the paper is organized as follows. Section 2 reviews the background information the project was based on. Section 3 overviews Logistic Regression, Support Vector Machines, Neural Networks, and dimensionality reducing algorithms. Section 4 presents the experimental results, and Section 5 contains the concluding remarks.

Support for this paper can be found by contacting Richard Cloutier by the methods below. This report was submitted on December 11th, 2018.

Richard Cloutier is pursuing a B.S. degree in Computer Engineering from the Rochester Institute of Technology. He currently lives in Rochester, NY, while he attends school (email: rcloutier778@gmail.com).

## II. Background

Numerous groups and individuals have attempted to forecast stock market movement via machine learning by utilizing a variety of classification algorithms and neural networks. Huang et al. analyzed various classification models and neural networks to determine the best algorithm for predicting stock prices and from their research, determined that support vector machines (SVMs) produce the highest accuracy score among all classification algorithms [1]. Shen et al. compares SVM, linear, and GLM classification models augmented with temporal correlation between global markets for price forecasting on the NASDAQ, Dow Jones Industrial Average, and the S&P 500[2]. Their findings prove that historical price data alone is not enough to generate an accurate model of the stock market, and that there are several facets of the market that must be considered when determining what features to add to the training and testing datasets. While adding additional, numeric features to the input dataset is relatively straightforward, implementing features based on non-numeric, contextual information is considerably more complex and introduces far more potential error to the model depending on how the feature is analyzed and implemented. In the case of analyzing news articles, Mirvish has explored how a poorly implemented stock price forecaster with news article scanning support mistakes mentions of actress Anne Hathaway with those of holding company Berkshire Hathaway [3]. While humorous, this example demonstrates that the implementation of features based on non-numeric, contextual information is rife with potentially overlooked pitfalls that will *cause undesirable behavior in the resulting model.*

The optimization and fine-tuning of the hyperparameters used in SVM models is a central focus for all machine learning projects utilizing SVMs and is the source of a significant amount of time spent during the project. Yeh et al. investigates the application of multiple-kernel learning to remove the need to manually adjust the hyperparameters of commonly used SVM models [4]. This allows for the adjustment and optimization of a far larger number of hyperparameters to a higher degree of precision than what could be accomplished manually, allowing developers to focus on other areas of the project that might have otherwise been ignored. It is widely accepted that the stock market is a highly interconnected network and that no one factor is the leading cause of stock price fluctuations. In order to attempt to forecast stock price movement, multiple different inputs have to be considered and analyzed to prevent garbage data from entering the training dataset of a model. Yu et al. investigates the usage of an evolving least squares multi-kernel SVM algorithm to predict stock market prices [5]. Their research focuses on the utilization of multiple genetic algorithms to select input features for their dataset as well as the tuning of the hyperparameters of their SVM model.

## III. Proposed Methods

A variety of algorithms and classification methods were tested without any news article data weights to establish a baseline accuracy score. Those methods included logistic regression with varying amounts of regularization, SVM with four different kernels, and a handful of forward propagating neural networks with a varying amount of hidden layers and nodes within those hidden layers.

### A. *Logistic Regression*

Logistic Regression is a classification algorithm that iteratively calculates the weights of the given features until the cost has reached an acceptable level, while using the sigmoid function shown in (1) as the activation function.

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

While optional, it is standard practice to regularize the weights while the logistic regression algorithm is still calculating them. Regularization is a technique whose goal is to prevent overfitting when training a machine learning algorithm. For this project, the regularization parameter, λ, was set to 0, 0.01, 0.1, and 1, in order to determine the best possible regularization parameter.

### B. *Support Vector Machines*

Support vector machines are classification algorithms that efficiently classify non-linear data using a kernel, which rotates the data to a higher dimension, reducing the complexity of the matrix arithmetic. SVM's work by using the fewest number of hyperplanes of the largest possible sizes to separate the data into classes. Popular kernels include the radial basis function, N$^{th}$ degree polynomial, and the sigmoid kernel.

### C. *Neural Networks*

A neural network is a machine learning technique that uses groups of nodes called layers to form an interconnected network consisting of one input layer, one or more hidden layers, and one output layer. Each node in the hidden layer represents an activation function while the importance of that node being determined by the weights on the connections to and from that node, with larger weights representing a higher degree of importance. While there are numerous types of neural networks, the multilayer perceptron (MLP) model was the only method implemented in this project. An MLP is feedforward network that trains itself using backpropagation. A feedforward network means that nodes in the i$^{th}$ layer will only feed into nodes in the i$^{th}$+1 layer, while backpropagation indicates that errors calculated at the output of the network are propagated backwards through the layers of the network.

#### 1) *Activation Functions*

Activation functions are functions that define the output of a given node in the hidden layer. Popular activation functions include identity, rectified linear units (ReLU), and the sigmoid function. The identity function does not modify the output of a node, allowing it to be passed into the next layer undisturbed. The function representing ReLU is shown in (2).

$$f(x) = \max(0, x) \tag{2}$$

The sigmoid activation function is shown above in (1).

#### 2) *Weight Optimization Solver*

The purpose of the weight optimization solver function is to adjust the weight of each node in order to minimize the resulting error. Two methods were examined in this project, being stochastic gradient descent (SGD) and a stochastic gradient-based, adaptive moment estimation (ADAM) formula. The function used for SGD is shown below in (3).

$$w = w - \eta \nabla L_i(w) \tag{3}$$

In (3) and (4), w represents the weight array, $L_i(w)$ represents the loss function at the i$^{th}$ sample, and η represents the learning rate. ADAM utilizes the running averages of past gradients $(m_w)$, and squared gradients $(v_w)$ to refine the node weights to a further extent than SGD. The equations used for this method are shown below in (4) through (8).

$$m_w^{(t+1)} = \beta_1 m_w^{(t)} + (1-\beta_1)\nabla_w L^{(t)} \tag{4}$$

$$v_w^{(t+1)} = \beta_2 v_w^{(t)} + (1-\beta_2)(\nabla_w L^{(t)})^2 \tag{5}$$

$$\hat{m_w} = \frac{m_w^{(t+1)}}{1-(\beta_1)^{t+1}} \tag{6}$$

$$\hat{v}_w = \frac{v_w^{(t+1)}}{1-(\beta_2)^{t+1}} \tag{7}$$

$$w^{(t+1)} = w^t - \eta \frac{\hat{m_w}}{\sqrt{\hat{v}_w} + \varepsilon} \tag{8}$$

In (4) through (8), $\beta_1$ and $\beta_2$ are the decay rates for gradients and squared gradient, η is the learning rate, and $\varepsilon$ is a small constant which prevents a division by zero scenario. The $w^t$ term represents the weight of node t, L represents the loss function at node t, and $m_w^{(t)}$ and $v_w^{(t)}$ represent the running average of the past gradient and squared gradient at node t.

After establishing a baseline accuracy score for the various classification algorithms, news articles from the New York Times pertaining to a specific stock would be scanned, summarized, and converted to a list of keywords. Those keywords would then be used as inputs to a stock specific SVM model, with the ground-truth outputs being -1, 0, or 1, representing the direction of the stock's price after the release of the article. In order to more effectively gauge the severity of the news article, the movement of the stock price was measured over the next 1, 3, and 5 days following the release of the news article.

Due to the high dimensionality of the input matrices caused by the large number of keywords found per news article, various methods of univariate feature selection were implemented to help boost the accuracy of the model. Univariate feature selection is the process of minimizing the number of dimensions of an input dataset by removing commonly misclassified or unused features, thereby creating a model with a higher accuracy than the original. This is accomplished by isolating each feature, calculating its p-value using its chi-squared statistic ($X^2$), and applying a threshold to the calculated p-value. The chi-squared statistic is the association statistic of each feature under a chi-squared distribution when the null hypothesis is true. The null hypothesis is the probability of a relationship between an input and the test output. A variety of threshold types were applied to the chi-squared statistic of the input training set and were used to train an SVM model for each feature selection method.

### *D. Percentile Selection*

Percentile feature selection works by classifying features of a dataset and preserving the $N^{th}$ percentile of the most accurate features.

### *E. K-Best*

K-best feature selection works by classifying and preserving the k highest scoring features.

### *F. False Positive Rate*

False positive rate feature selection removes features based on their probability of producing a false positive result. Features that produce false positives will be removed.

### *G. False Discovery Rate*

False discovery rate feature selection removes features based on their likelihood of generating a false positive or false negative result. Only features that generate a correct correlation will be preserved.

### *H. Family-wise Error Rate*

Family-wise error rate feature selection applies a threshold based on the probability of generating at least one false positive among all discovered correlations.

While percentile and k-best selection are computationally inexpensive, they tend to be less accurate than false positive, false discovery, and family-wise feature selection. After applying the feature selection methods listed above to the input training set, each model would be scored using the input test set with the most accurate model being selected to represent the SVM model for the given stock. Models that had an accuracy of less than 30% however, were ignored in order to prevent misclassifications from occurring in the eventual stock price direction prediction.

Using the newly generated stock-specific SVM models for interpreting news articles, predictions for stock price movement directions were generated and concatenated to the historical price dataset where applicable. Dates within the historical price dataset that did not coincide with the release of a news article or the days following had their news article prediction feature set to zero.

## IV. Results

The datasets for daily and intraday historical prices were formatted such that each row would represent a time interval, with the columns being the difference in the stock's price at the beginning and end of the interval, and the difference in price between the highest and lowest price within that interval. The difference in price between the beginning and end of the interval was required to represent the movement direction and magnitude of the stock's price. The difference in the stock's highest and lowest price during a given interval was used to obtain a more comprehensive understanding of what occurred during the given interval.

All SVMs used in this project were generated using the radial basis function (RBF) kernel. The SVMs used to predict stock price movement direction were generated using a cost and gamma of 20.0, while the SVMs used to predict the meaning of key words in news articles used a cost of 1.0 and a gamma of 1/n, where n is the number of features in the input dataset. A cost and gamma of 20.0 was selected by iteratively generating and scoring models with varying values for the cost and gamma parameters. The values used were drawn from a set ranging from 0.0002 to 200000, incrementing in steps of 10. By setting the cost and gamma parameters to 20.0 for the price prediction models, the overall accuracy of said models increased from 50.18% to 51.93%, with the drawback of increasing the computation time from ~3 seconds to over 30 seconds per model.

To establish a baseline accuracy score, the input price data was split 80:20, with 80% of the data being allocated for the training set and the remaining 20% being allocated for the testing set. Using the SVM configuration described above, SVM models for a selection of stocks were trained and tested using the ratio described above. Using the newly generated SVM models, the returns, strategy returns, and Sharpe ratios were calculated for each stock using (2), (3), and (4) respectively.

$$Return_i = \ln\left(\frac{Closing\ Price_{i+1}}{Closing\ Price_i}\right) * 100 \quad (2)$$

$$Strategy\ Return_i = Return_i * Prediction_i \quad (3)$$

$$Sharpe\ Ratio = \frac{Strategy\ Return - Return}{\sigma_{Strategy\ Return}} \quad (4)$$

The Sharpe Ratio is a useful statistic that represents the performance of an investment by adjusting for its risk. The higher the ratio is, the larger the return on investment (ROI) is. Table I shows the accuracies and Sharpe ratios for a selection of stocks found in the Dow Jones Industrial Average using their daily prices.

TABLE I SVM ACCURACIES AND SHARPE RATIO WITHOUT NEWS ARTICLE PREDICTIONS USING DAILY PRICE SET

| Stock | Training Accuracy (%) | Testing Accuracy (%) | Sharpe Ratio |
|---|---|---|---|
| AMD (AMD) | 59.0823 | 49.6198 | -1.72 |
| Verizon (VZ) | 59.4151 | 50.8555 | -0.836 |
| J.P. Morgan (JPM) | 63.457 | 50.8555 | -0.514 |
| PayPal (PYPL) | 69.0828 | 50.8876 | 0.457 |
| Exxon Mobil (XOM) | 62.4108 | 51.1407 | 1.413 |
| Bank of America (BAC) | 61.3172 | 51.9962 | -0.571 |
| Google (GOOGL) | 97.8768 | 52.2949 | -2.218 |
| Macys (M) | 61.1032 | 53.0418 | 2.341 |
| Chevron (CVX) | 64.3842 | 53.327 | 2.178 |
| Visa (V) | 73.513 | 55.2876 | 0.226 |

As shown in Table I, the model for Visa had the highest testing accuracy, while the model for Macys produced the highest Sharpe ratio. By sorting the table by accuracy, it is readily apparent that there exists a correlation between a high testing accuracy and a positive Sharpe ratio.

Due to time constraints, the number of hidden layers and nodes were kept at their default values of one and 100 respectively. The activation function and weight optimization solver for the multi-layer perceptron classifier neural network was ReLU and Adam, as this combination resulted in the highest overall testing accuracy. Table II shows the accuracies and Sharpe ratios when ignoring news article predictions.

TABLE II NEURAL NETWORK RESULTS WITHOUT NEWS ARTICLE PREDICTIONS USING DAILY PRICE SET

| Stock | Training Accuracy (%) | Testing Accuracy (%) | Sharpe Ratio |
|---|---|---|---|
| PayPal (PYPL) | 47.3373 | 42.6036 | -1.507 |
| Macys (M) | 52.1636 | 48.9544 | -0.156 |
| AMD (AMD) | 51.5692 | 50 | -1.203 |
| Chevron (CVX) | 52.9957 | 50.0951 | -1.432 |
| Bank of America (BAC) | 52.0922 | 50.2852 | -1.639 |
| Google (GOOGL) | 52.1754 | 51.0431 | -1.127 |
| Verizon (VZ) | 51.5692 | 51.5209 | 0.678 |
| Exxon Mobil (XOM) | 53.0433 | 52.7567 | 1.543 |
| J.P. Morgan (JPM) | 51.6167 | 52.8517 | -0.018 |
| Visa (V) | 54.7398 | 56.9573 | -0.214 |

As shown in Table II, the model for Visa had the highest testing accuracy, while the model for Exxon Mobil produced the highest Sharpe ratio. The correlation between testing accuracy and a positive Sharpe ratio found for the SVM results was not present in the neural network results. When comparing the two methods it appears that the usage of SVM classification produces more reliable models, as the average testing accuracy for the SVM results are higher than that of the neural network results, 51.93% and 50.71% respectively. The graph of returns, SVM strategy returns, and neural network strategy returns for the stock with the highest testing accuracy, Visa, is shown below in Fig. 1.


Fig. 1: Returns and Strategy Returns for Visa

As shown in Fig. 1, the strategy returns for both the SVM (green) and the neural network (yellow) closely follow the movements of the actual returns (red) throughout the graph, indicating that the accuracy for both models is reasonable. The effects of the positive Sharpe ratio for the SVM model and the negative ratio for the neural network are visible, as the strategy returns line for the SVM model is, on average, above the market return, while the neural network line is below it. Fig. 2 shows the returns, SVM strategy returns, and neural network strategy returns for the stock with the highest Sharpe ratio, out of the SVM models, Macys.


Fig. 2: Returns and Strategy Returns for Macys

As shown in Fig. 2, the strategy returns for both the SVM (green) and the neural network (yellow) closely follow the movements of the actual returns (red) throughout the graph, indicating that the accuracy for both models is reasonable. The effect of the 2.341 Sharpe ratio in the SVM model for Macys is immediately noticeable in the graph, as the SVM strategy plot is far above that of the market returns, indicating that the ROI, if following the predictions made by this model, would be high. Fig. 3 shows the returns, SVM strategy returns, and neural network strategy returns for the stock with the highest Sharpe ratio, out of the neural network models, Exxon Mobil.


Fig. 3: Returns and Strategy Returns for Exxon Mobil

As shown in Fig. 3, the strategy returns for both the SVM (green) and the neural network (yellow) closely follow the movements of the actual returns (red) throughout the graph, indicating that the accuracy for both models is reasonable. The effect of a high Sharpe ratio is again noticeable, as the strategy return plot for the neural network is far above that of the market returns, although by a smaller margin than that which was shown in Fig. 2. As shown in Tables I and II, while the neural network implementation was able to achieve a higher maximum accuracy of 56.96% compared to SVM's maximum of 55.28%, its average accuracy was less than that of the SVM implementation. This indicates that the SVM method is more reliable and that the high maximum accuracy of the neural network could be attributed to sheer coincidence. This could be verified by sampling a larger number of stocks and checking if this pattern remains.

When implementing percentile based feature selection, the cutoff percentage for the highest scoring keywords was set to 10%. When implementing k-best feature selection, the k value used to determine the maximum number of top performing features was set to be the top 200 features. When implementing the false positive, false discovery, and family-wise feature selection methods, the threshold value that the generated probabilities were compared against was set to 0.05. Table III shows a comparison of the average model accuracy when various feature selection methods.

TABLE III
COMPARISON OF FEATURE SELECTION METHODS

| Method | Training Accuracy (%) | Testing Accuracy (%) |
|---|---|---|
| No Feature Selection | 64.23 | 58.30 |
| Percentile | 74.68 | 63.73 |
| K-Best | 72.64 | 63.05 |
| False Positive Rate | 73.49 | 65.76 |
| False Discovery Rate | 74.51 | 64.41 |
| Family-wise Error Rate | 71.70 | 62.03 |

As shown in Table III, the false-positive rate feature selection method produced the highest average testing accuracy of 65.76%, an improvement of 12.8% over no feature selection. The benefit of feature selection is two-fold, as the reduced feature set results in a significant improvement in calculation time, resulting in a more responsive program.

The datasets for the news article keywords were formatted such that each row would represent a time interval of 1, 3, or 5 days after the release of a news article, with each column being a keyword. The elements of the dataset were the number of occurrences of the specified keyword in the given article. The ground truth output used to train and score the SVM models represented the change in price between market opening price the day the article was released and the market closing price of specified date.

Table IV shows the SVM results when combining the predicted impact of a news article on a stock's price the stock's historical daily price data.

TABLE IV SVM RESULTS WITH NEWS ARTICLE PREDICTIONS USING DAILY PRICE SET

| Stock | Training Accuracy (%) | Testing Accuracy (%) | Sharpe Ratio |
|---|---|---|---|
| AMD (AMD) | 59.4151 | 49.3346 | -1.537 |
| Verizon (VZ) | 59.5578 | 50.5703 | -1.617 |
| Google (GOOGL) | 97.8768 | 50.904 | -2.981 |
| J.P. Morgan (JPM) | 63.457 | 50.9506 | -0.497 |
| Exxon Mobil (XOM) | 62.4822 | 51.5209 | 1.806 |
| Bank of America (BAC) | 61.3172 | 51.711 | -0.502 |
| Macys (M) | 61.1032 | 52.4715 | 2.667 |
| Chevron (CVX) | 64.3604 | 52.9468 | 2.239 |
| PayPal (PYPL) | 76.3314 | 54.4379 | 1.303 |
| Visa (V) | 73.5595 | 56.9573 | 0.367 |

As shown in Table IV, the most accurate stock, and the stock with the highest Sharpe ratio remained the same as what was previously found in Table I. However, the testing accuracies and Sharpe ratios of both of these stocks, as well as the overall average testing accuracy and Sharpe ratios increased, with the average testing accuracy now being 52.18%, and the average Sharpe ratio rising from 0.075 to 0.309. Additionally, the correlation between a high testing accuracy and positive Sharpe ratio found in Table I is again present in Table IV, indicating that this correlation remains present regardless of the presence of news article prediction scores. Table V shows the neural network results when combining the predicted impact of a news article on a stock's price the stock's historical daily price data.

TABLE V NEURAL NETWORK RESULTS WITH NEWS ARTICLE PREDICTIONS USING DAILY PRICE SET

| Stock | Training Accuracy (%) | Testing Accuracy (%) | Sharpe Ratio |
|---|---|---|---|
| Verizon (VZ) | 51.8783 | 48.7643 | -1.415 |
| AMD (AMD) | 52.4964 | 49.9049 | -1.439 |
| Google (GOOGL) | 51.9318 | 50.6259 | -2.215 |
| Chevron (CVX) | 52.8293 | 51.0456 | 0.424 |
| Macys (M) | 49.7147 | 51.2357 | 3.191 |
| Bank of America (BAC) | 52.4489 | 51.4259 | -1.682 |
| Exxon Mobil (XOM) | 53.8516 | 52.2814 | 1.688 |
| J.P. Morgan (JPM) | 52.758 | 52.2814 | 1.017 |
| PayPal (PYPL) | 51.6272 | 55.0296 | 0 |
| Visa (V) | 54.461 | 60.6679 | 0.313 |

As shown in Table V, the most accurate stock, while decreasing in accuracy, remained the same. The stock with the highest Sharpe ratio however, had changed to Macys, with a ratio of 3.191. When comparing the results found before and after the addition of news article data, the overall testing accuracy has increased from 50.76% to 52.33%, while the average Sharpe ratio has decreased from -0.5075 to -0.0118. The correlation between testing accuracy and a positive Sharpe ratio found for the SVM results was again, not present in the neural network results. A comparison graph between the SVM and neural network models, with and without the news article data for Visa is shown below in Fig. 4.

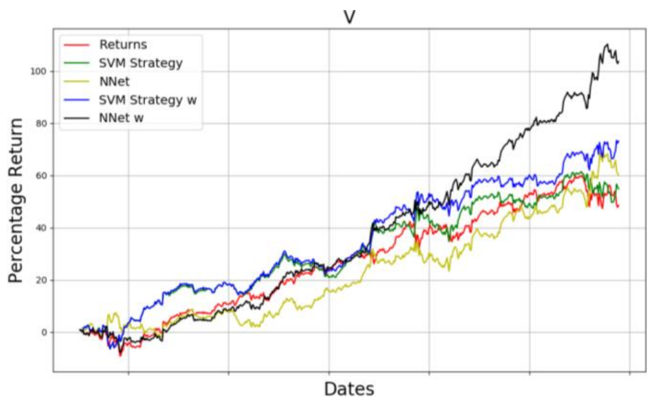



Fig. 4: Strategy Returns Comparison for Visa

As shown in Fig. 4, the introduction of news article data in both the SVM (blue) and neural network (black) models increased their ROI and in the case of the neural network, boosted its ROI above the market returns. The introduction of news article data is noticeable in Fig. 4, as the plots for the models that account for news article data nearly completely overlap their unaware versions, until a news article is released, at which point they diverge, with the aware models rising above their unaware versions. Fig 5 shows a similar effect occurring in Macys' stock.

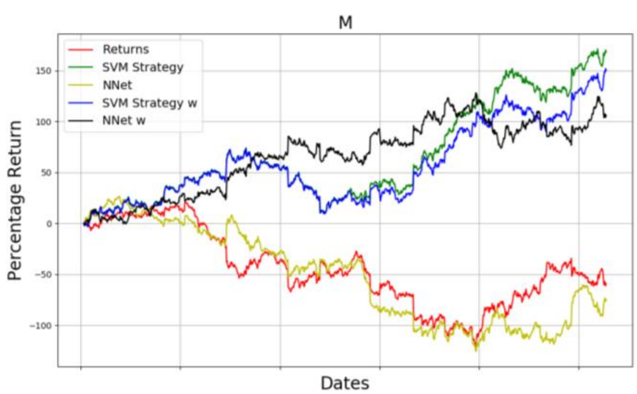



Fig. 5: Strategy Returns Comparison for Macys

As shown in Fig. 5, the introduction of news article data in both the SVM (blue) and neural network (black) models increased their ROI and in the case of the neural network, significantly boosted its ROI above market returns. While the SVM implementation was still favorable on the right most quarter of the graph, the news article aware neural network still held dominance for approximately 33% of the sampled dates and held percentage returns far above that of its unaware version. Fig 6. shows the comparison graph between the SVM and neural network models, with and without news article data for Exxon Mobil.

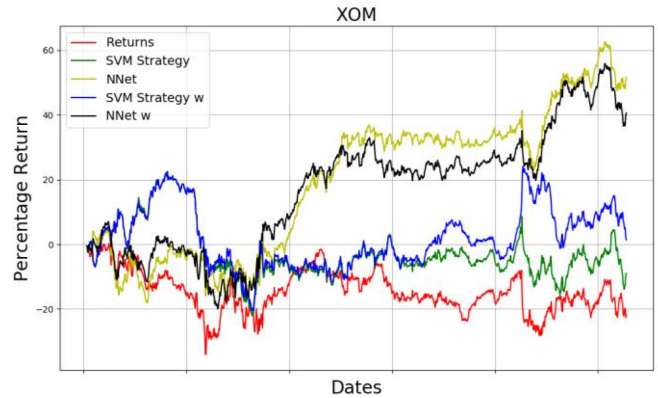



Fig. 6: Strategy Returns Comparison for Exxon Mobil

As shown in Fig. 6, while the introduction of news article data to the SVM (blue) model did increase its percentage returns, its introduction actually decreased that of the neural network model. This shows that the news article parsing, keyword selecting, and keyword weighing algorithms need to be adjusted and refined.

With the addition of the news article data, the accuracy of the neural network implementation have pulled ahead of that of the SVM implementation, indicating that with the addition of the news article data, the neural network implementation is the most consistent method. However, the average Sharpe ratio of the SVM implementation when using news article data is far greater than any other implementation, indicating that the ROI with this implementation has the potential to be much greater than that of the more reliable neural network implementation.

As shown in Tables IV and V, the introduction of news article data results in an increase in accuracy when forecasting stock price movement, indicating that the objective of the project was accomplished and that the project was a success.

## V. Conclusion

As the addition of news article data increased the overall accuracy of the SVM and neural network implementations, the project is considered a success. The magnitude of success should not be understated, as machine intelligence focused hedge funds such as Blackrock and Two Sigma are known to offer high paying (>$120,000) positions to developers of stock price forecasting models with accuracies of 55%-60%. If one were to invest using the models generated by this project, it is recommended to use both the SVM and neural network implementations, as the neural network would provide a slow, but reliable ROI while the SVM models would act as a high risk, high reward strategy.

While project was a success, several things could be done to further improve the accuracy of both implementations. In order to ensure that the models have not overfitted to the 10 stocks examined, a greater number of stocks should be added to the input dataset. Additionally, the methods for selecting keywords in news articles could be improved by the implementation of various other parsing and weighing techniques, which would further increase the resulting price forecasting accuracy. The selection of which news articles to parse could also be improved to allow non-company-specific articles to be selected if they pertain to the focus of said company. The neural network implementation in particular could be improved by optimizing the number of hidden layers and nodes, as they were left at their default values of one and 100.

## References

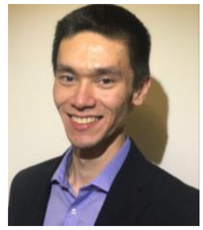
[1] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," ScienceDirect, 25-May-2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054804000681. [Accessed: 09-Nov-2018].

[2] Shen and H. Jiang, "Stock Market Forecasting Using Machine Learning Algorithms," Semantics Scholar. [Online]. Available: https://pdfs.semanticscholar.org/b68e/8d2f4d2c709bb5919b82effcb6a7bbd3db37.pdf. [Accessed: 09-Nov-2018].

[3] D. Mirvish, "The Hathaway Effect: How Anne Gives Warren Buffett a Rise," The Huffington Post, 07-Dec-2017. [Online]. Available: https://www.huffingtonpost.com/dan-mirvish/the-hathaway-effect-how-a_b_830041.html. [Accessed: 09-Nov-2018].

[4] C.-Y. Yeh, C.-W. Huang, and S.-J. Lee, "A multiple-kernel support vector regression approach for stock market price forecasting," ScienceDirect, 15-Aug-2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417410007876. [Accessed: 09-Nov-2018].

[5] L. Yu, H. Chen, S. Wang and K. K. Lai, "Evolving Least Squares Support Vector Machines for Stock Market Trend Mining," in IEEE Transactions on Evolutionary Computation, vol. 13, no. 1, pp. 87-102, Feb. 2009.
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4632148&isnumber=4769002



**R. Cloutier** was born in Whippany, NJ, USA in 1996. He is currently pursuing a B.S. degree in Computer Engineering from the Rochester Institute of Technology, Rochester, in 2018.

For the summers of 2017 and 2018, he was a Technology Analyst Intern at J.P. Morgan Chase. From January through May of 2018, he was an intern at UTC Aerospace Systems.