Stock Stalkers - CAPP 30254 Project

Alejandro Saenz, Ayako Watanabe, Guo Chen, Jake Nicoll

1 Introduction

As a snapshot of future growth expectations of companies and economy, stock price prediction has long been an important topic in financial and academic studies (Hu et al. 2021). Many variables may affect pricing, including macroeconomic factors and investors' momentum. Technical analysis and fundamental analysis are popular approaches to capture the factors for prediction. Machine learning techniques are further utilized to learn, adapt, and generalize based on large amount of structured and unstructured data in the finance market.

Prevalent market efficiency analysis states that, as long as financial markets are efficient, future stock price will behave as a random walk. Nevertheless, behavioural finance literature offers good reasons to believe investors' momentum is an essential feature that can impact price fluctuation (Dhankar 2016). A typical way to incorporate momentum into the model is through sentiment analysis on textual data from investors' social media interactions. Research also suggests that converting text data into numerical information in a prediction model is challenging but rewarding. If combined with market data and technical indicators, textual data can increase model accuracy (Rouf et al. 2021).

Our research adopted both traditional machine learning models (Logistic Regression, Ensemble Models) and deep learning models (LSTM) to analyze market data as well as textual data. Our goal is to assess the importance of textual data (StockTwits) in stock price prediction. To do this we will compare four versions of a learning setting that includes a long short-term memory (LSTM) neural network at its core, and we will include textual features in half of them. Additionally, given our significant data limitations, two of the settings incorporate a first step of feature selection that uses an ensemble of models, including random forest.

2 Dataset

2.1 Price Data

The historical price data (Open, Close, High, Low, Volume) are obtained from Yahoo Finance. Dates range from 2017-01-01 to 2022-04-28. The target companies are Apple (AAPL) and Microsoft (MSFT) which are the top 2 stocks of companies in the United States Technology Sector, by market capitalization. Figure 1 shows the series for closing price for both stocks.

We create 86 technical features from Technical Analysis library[1] and 12 date features from fastai library[2]. Our goal is to predict the closing stock price direction of the following day. We use y = 1 to refer to the upward movement of closing stock price and y = -1 to refer to the downward movement of closing stock price. We prepare long-term data (from 2017-08-09[3] to 2022-04-28) and short-term data (from 2022-01-01 to 2022-04-28) for both AAPL and MSFT. The distribution of y is shown in Figure 2.

[1] https://github.com/bukosabino/ta

[2] https://docs.fast.ai/tabular.core.html

[3] We start from this point because we need to use the previous period data (from 2017-01-01 to 2017-08-08) to calculate all technical features

Preprint. Under review.

Figure 1: Closing price time series

	AAPL 81d	MSFT 81d	AAPL 5y	MSFT 5y
y = 1 (upward)	35	38	632	658
y = -1 (downward)	46	42	557	530

Figure 2: Distribution of label (y)

2.2 StockTwits

For text data, we collect posts from StockTwits, which is a financial microblog being increasingly used to obtain public sentiment for stock price prediction (Alzazah et al. 2014). We collect posts tagging AAPL and MSFT from 2022-01-01 to 2022-04-28. For each post, we scrape the post's id, the text of the post, the date it was posted, and the user-labeled sentiment.[4] Posts can be labeled "bearish" or "bullish", or the user can omit a sentiment label. We obtain 108,967 posts tagging Apple, for a total of 923 posts per day. For Microsoft, we obtain 55984 posts for 474 per day. For both stocks, approximately 45% were labeled.

In performing data collection, we wish to have sufficiently many posts to evaluate public sentiment, but we also need to obtain data over a sufficiently long period of time. Researchers have scraped social media data for the purposes of stock price prediction over vastly different time scales, ranging from a few weeks to two years (Bujari et al. 2017, Checkley et al. 2017, Li and Pan 2021, Nisar et al. 2017). Researchers who collected data on the time scale of months obtained on the order of 10,000-100,000 posts per stock (Bujari et al. 2017, Xu et al. 2014). We elect to scrape posts over a four-month period and obtain approximately 100,000 posts per stock in order to be consistent with the literature.

We construct two features from the StockTwits data. These represent the counts of bearish posts and the counts of bullish posts by day. We include a subset of unlabeled posts which were assigned labels by a pre-trained sentiment analyzer, following the methodology discussed below. Our decision to separate counts of bearish and counts of bullish posts into distinct features was motivated by Wu et al (2019), who found that negative news articles had more influence on the stock market than positive news articles.

3 Methods

3.1 Logistic Regression - Baseline Model

Before the development of deep learning, logistic regression has been widely used for stock price classification and forecasting (Obthong et al. 2020). We use logistic regression as a base-line model to compare with the performance of LSTM (Long Short-term Memory).

Logistic regression can model the probabilities in classification problems. Given the historical stock prices as inputs, we model the probability of the output being 1 (upward price movement) and -1

[4]https://github.com/mtmmy/stock-twits-scraper/blob/master/scraper.py

(downward price movement), as $P(y = 1|x)$ and $P(y = -1|x)$, which are calculated as below:

$$P(y = 1) = \sigma(wx + b) = \frac{1}{1 + exp(-(wx + b))}, \tag{1}$$

$$P(y = -1) = 1 - \sigma(wx + b) = \frac{exp(-(wx + b))}{1 + exp(-(wx + b))}. \tag{2}$$

3.2 Ensemble Models - Feature Selection

Among traditional machine learning models, Random Forest has proven to be one of the best performed models for stock prediction (Kumar et al. 2018, Soni et al. 2022). Other ensemble models such as Adaboost and Gradient Boosting are also widely used (Nabipour et al. 2020). Besides prediction, the three models can also select the most important features with their scikit-learn implementation. This is important for our research purpose since there are many technical indicators available. Selecting the most important features can enhance deep learning model performance, especially when the data size is not big. During the model building process, each feature importance is calculated from the average impurity reduction derived from all decision trees in the forest (Raschka et al. 2022).

According to Raschka et al. 2022, Random Forest is an ensemble of decision trees using bagging. It randomly picks subsets of the sample and features to construct multiple decision trees, and finally aggregates the prediction by each tree to assign the class label by majority vote. Boosting is another ensemble technique in which the weak learners subsequently learn from misclassified training examples. Adaboost uses the whole training dataset to train the weak learners, which learn from the mistakes sequentially by re-weighting training examples each time. Then, it uses weighted majority voting to combine these weak learners. Gradient boosting is similar to Adaboost, but it updates based on a residual (difference between the label and predicted value of the previous tree) using a loss gradient.

3.3 RoBERTa - Sentiment Analysis

When creating posts on StockTwits, users have the option to label tweets as "bearish" or "bullish". This labeling gives us a natural notion of the sentiment contained in the post. However, approximately 55% of posts were unlabeled. We expect that many unlabeled posts contain information about stock movements, which we hope to recover by performing sentiment analysis on these posts. However, we also expect many of the posts to be unlabeled because the creators simply do not believe the posts contain information about stock price movements. Therefore, we expect the pre-labeled posts to contain the highest quality data, and we hope to recover additional information from the other posts using a sentiment analyzer that requires a minimal amount of pre-labled data to train.

Bozanta et al (2021) explored the abilities of different models to predict sentiment in StockTwits posts. In particular, they collected 10,000 posts labeled "bearish" and 10,000 posts labeled "bullish" over a four-month period for each of five stocks (one of which was AAPL). They used these labeled posts to train sentiment analyzers, and found that the pre-trained transformer model RoBERTa outperformed other models in terms of F1 scores. Following this finding, we used a RoBERTa model which was fine-tuned for predicting sentiment in StockTwits posts[5].

To obtain final sentiment labels, we first let posts that were labeled by their creators retain their labels. Then, following Bozanta et al (2021), we restrict our consideration to posts less than or equal to 250 words. We pass every unlabeled post through the sentiment analyzer to obtain a sentiment prediction, as well as a score for that prediction. Taking into account the fact that many unlabeled posts are noise, we compute the median magnitude score amongst the hand-labeled posts. Posts are attributed the label predicted by the sentiment analyzer if the associated score for this label is above the median. If the score is below the median, the post is left unlabeled. In this way, we obtain final sentiment labels for posts, and we thus expand the size of our labeled dataset.

[5]https://huggingface.co/zhayunduo/roberta-base-stocktwits-finetuned

3.4 LSTM - Deep Learning

Recurrent Neural Networks (RNN) are widely used in settings where connections between nodes follow a temporal sequence. Nevertheless, as is not unusual in many neural network designs, they suffer from the vanishing gradient problem, which becomes more prevalent as the data goes further back in time. Long short-term memory (LSTM) are a type of RNN whose architecture addresses the vanishing gradient problem by introducing "gates" that can add or remove information to the state of each cell. For example, the "forget gate", which corresponds to a sigmoid layer, regulates what information is removed from the cell state. The ability of LSTMs to manage long-term dependencies makes them ideal for time series and stock market predictions (Torres et al., 2020; Mahmoud and Mohammed, 2021; Lim and Zohren, 2021; Kalbandi et al., 2021).

4 Experiment setup

Our research is aimed at observing the effects of adding sentiment scores to our prediction model. The steps of the experimental setup are defined as follows:

- Step 1. Collect data. Extract technical indicator features. Use RoBERTa to fill the missing sentiment scores. Split the data into training and testing sets.
- Step 2. Run a baseline model with logistic regression.
- Step 3. Use 5-fold cross-validation (CV) in the training set to implement ensemble models and evaluate feature importances.
- Step 4. Define experimental models and similarly use 5-fold CV to find the best hyperparameters for the LSTM.
- Step 5. Retrain the LSTM models using the whole training set to evaluate accuracy scores in the testing set.

Our experimental framework is displayed in the figure below:

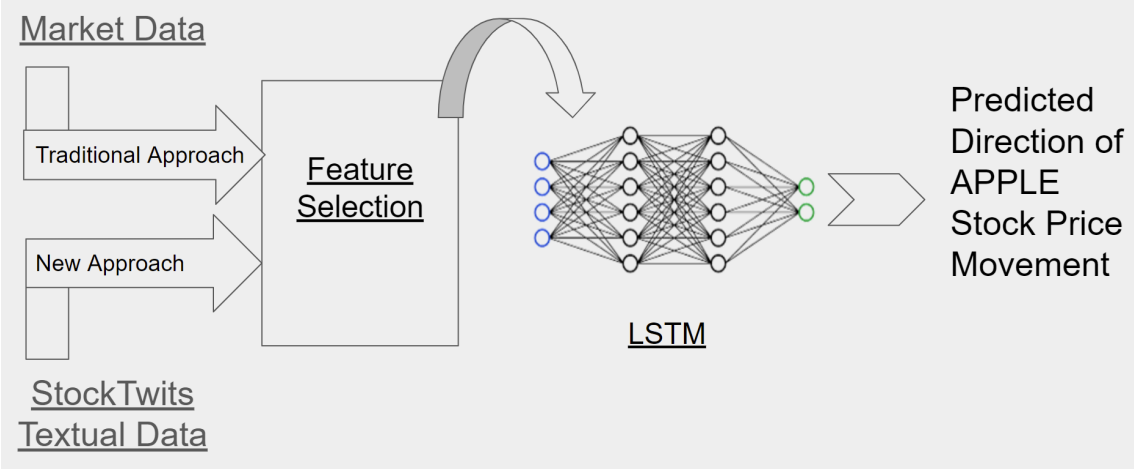

Figure 3: We display the data life cycle for AAPL stock prediction using LSTM.

Our experimental models are briefly described below.

- Model 1. Short-term data and top 20 selected features, including sentiment scores (LSTM)
- Model 2. Short-term data and top 20 selected features, excluding sentiment scores (LSTM)
- Model 3. Short-term data and all features including sentiment scores (LSTM and Logistic Regression)
- Model 4. Long-term data and all features besides sentiment scores (LSTM and Logistic Regression)

4.1 Short-Term and Long-Term Analysis

We conduct research on both short-term (81 days, 2022-01-01 to 2022-04-28) historical price and StockTwits sentiment data and on long-term (2017-04-28 to 2022-04-28) historical price data. Given the caps on rates at which we can scrape StockTwits data, as well as the size of this data, we only collect this data over a period of four months[6]. As discussed in Section 2.2, the scope of our data for short-term analysis is consistent with previous research.

Nevertheless, we are still concerned that the short time frame may not provide sufficient data to train a model that yields robust results. For this reason, we complement this approach with an analysis of technical stock data from the previous five years. This enables us to determine whether adding more data improves the consistency and robustness of the results and to assess the performance of our short-term models (e.g., via comparing the magnitude of training and testing accuracy between models trained on short-term and long-term data).

4.2 Cross Validation, Feature Selection, and Hyperparameter Tuning

Stock price data is time-series data, and working with time-series data requires that we pay special attention to look-ahead bias. This issue results from using future information to predict the past. To address this, we first split the dataset into training (80%) and testing (20%) sets without shuffling. We use the training set to implement CV for feature selection and hyperparameter tuning as discussed below. We then train the final models. The testing set is only used to evaluate the final model performance.

Given the limited amount of data used for short-term analysis, we conduct feature selection to observe its effects on the performance of LSTM . We use scikit-learn's TimeSeriesSplit, which employs expanding window CV, to split the training and validation data. We implement a 5-window CV and within each window, we run Random Forest, Adaboost, and Gradient Boosting to obtain feature importance scores. We average scores across models and windows to find the top 20 most important features. Note that we do not conduct feature selection when using the long-term dataset.

Since the performance of LSTM depends on the choice of hyperparameters, we conducted a similar 5-window CV to find the best hyperparameters for each model. We split the complete training data set using TimeSeriesSplit, define a grid of hyperparameter values, and use optuna to find the combination of values which minimize validation loss. Following Bozanta et al. (2021), we choose to tune the 1) learning rate, 2) number of hidden layers, 3) number of nodes per layer, and 4) dropout rate. Once we obtain the best hyperparameters, we retrain the LSTM model using the complete training set to evaluate the training set accuracy. We repeat this process for Models 1-3. Due to the computational complexity of tuning hyperparameters for Model 4 (which uses the long-term data), we select the same hyperparameter combinations used for Model 3, which has the most overlap with Model 4 in terms of the features used.

4.3 Normalization

Each feature is normalized so that the training set is in range between 0 and 1. We apply the scalar to validation and testing set to evaluate the models.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3}$$

[6]Getting StockTwits data is expensive and it took us 5 continuous dates to get the current text data.

4.4 Evaluation Metrics

We use accuracy, precision, and recall as model performance metrics, which are commonly used in previous literature (Rouf et al. 2021).

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total}} \tag{4}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False}} \tag{5}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False negative}} \tag{6}$$

5 Results

5.1 Feature Selection

With TimeSeriesSplit for cross validation and ensemble models for feature importance, we obtain the top 20 features for APPLE and for MSFT (both with and without sentiment features). The features are displayed below (Figure 4). The top 20 features are boxed, and there is a clear 'elbow' trend of feature importance scores. Besides top features such as momentum pvo, volume mvi, the sentiment-related features - "bearish" and "bullish" - are all included as important features for both Apple and Microsoft. This supports the hypothesis that these sentiment analysis features are useful to incorporate into the stock prediction model.

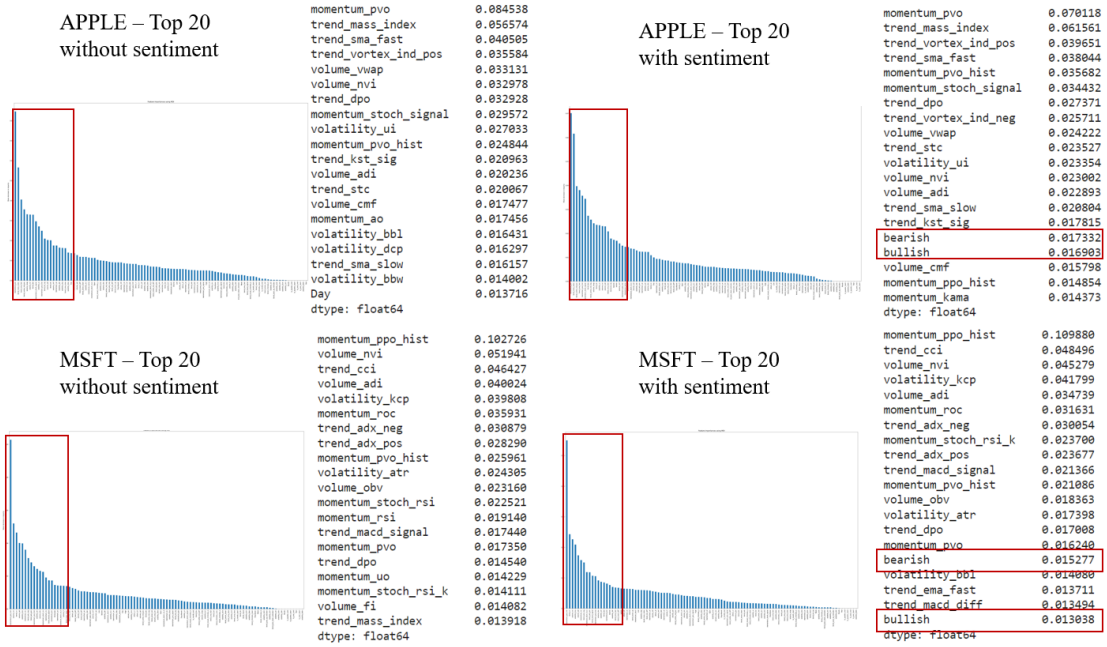

Figure 4: Feature Selection Result

5.2 Model Comparison

For APPLE (Figure 5), overall we observe better accuracy and precision scores with LSTM models than baseline logistic regression models. In the short-term LSTM models, we observe the same testing accuracy of 0.65 for including and excluding sentiment features. However, the precision and recall for upward prediction is 0 if sentiment features are excluded, suggesting the importance of including textual data and sentiment analysis into the stock prediction model.

For the LSTM models, while including all technical features to produces the highest testing accuracy as 0.71 for the short-term period, the training accuracy of 1.0 suggests that there could be over-fitting, which we may observe with a larger test sample.

	Training	Testing		
Model - Features	Acc.	Acc.	Precision / Recall: down	Precision / Recall: up
LR - All technical + sentiment (81d)	0.92	0.65	0.67 / 0.91	0.50 / 0.17
LR - All technical (5y)	0.61	0.56	0.69 / 0.17	0.55 / 0.93
LSTM - Selected tech. + sentiment (81d)	0.75	0.65	0.69 / 0.82	0.5 / 0.33
LSTM - Selected technical (81d)	0.69	0.65	0.65 / 1.00	0 / 0.00
LSTM - All technical + sentiment (81d)	1	0.71	0.71 / 0.91	0.67 / 0.33
LSTM - All technical (5y)	0.97	0.53	0.51 / 0.75	0.59 / 0.33

Figure 5: Model Performances for APPLE

We observe similar patterns for the models for MSFT (Figure 6), indicating that adding more data could increase the consistency and robustness of our model. The overall performance of the LSTM models is better than the logistic regression models. For LSTM running on short-term datasets, the model with sentiment analysis did a better job in terms of both accuracy score and precision score than the one without sentiment. Passing all the features into the model without selection still results in some over-fitting issues.

	Training	Testing		
Model - Features	Acc.	Acc.	Precision / Recall: down	Precision / Recall: up
LR - All technical + sentiment (81d)	0.80	0.62	0.71 / 0.56	0.56 / 0.71
LR - All technical (5y)	0.61	0.53	0.51 / 0.74	0.56 / 0.31
LSTM - Selected tech. + sentiment (81d)	0.81	0.75	0.86 / 0.67	0.67 / 0.86
LSTM - Selected technical (81d)	0.83	0.62	0.80 / 0.44	0.55 / 0.86
LSTM - All technical + sentiment (81d)	0.91	0.69	0.75 / 0.67	0.62 / 0.71
LSTM - All technical (5y)	0.96	0.54	0.53 / 0.67	0.57 / 0.42

Figure 6: Model Performances for MSFT

6 Conclusions

The main purpose of this project is to assess the contribution of textual data to stock price prediction. A first glance at the accuracy of models suggests that StockTwits data did not have a significant impact on our model's prediction power on our data set. Nevertheless, results on other indicators lead us to speculate that including textual data can be helpful in other circumstances. For example, results for precision and recall indicate that predictions tend to be less variable when we do not include

sentiment data (i.e., the model is more likely to predict mostly in one direction). This behavior of always predicting downward movement produces reasonable accuracy rates for our testing data, but with other or more extended data, a more sensible prediction could yield higher accuracies, especially in highly volatile markets.

Additionally, our feature selection analysis consistently finds sentiment features to be amongst the 20 most relevant from a collection of more than 100 examined features. This also supports the idea of the importance of textual data for stock price prediction, beyond our LSTM implementation.

An important caveat is that we have serious data restrictions due to the computational limitations in scraping StockTwits. This means that our results lack the robustness needed to be more definitive and have to be considered as suggestive. In practice, we experience this problem in that different runs of the learning algorithm yielded somewhat different results (due to randomness introduced by dropout rates).

To partially address this concern, we implemented the mentioned feature selection and compared it with a model that included 5 years of technical (non-textual) data. On one hand, adding feature selection helps us control potential overfitting situations. On the other hand, the lower accuracies (consistently) obtained with a 5-year horizon suggest that the relatively high accuracies shown in the results for the short-term models are probably a particular product of our data and not a general characteristic of the model. Nevertheless, as the most important conclusions are made comparing short-term models which all suffer from the same data limitations, we have grounds to believe they constitute suggestive evidence.

As an concluding remark, it is well documented that developed financial markets such as the ones studied behave relatively efficiently. In this sense, any improvement in stock price prediction is expected to be small and, thus, it would require large amounts of data to be robust. Moreover, any better-than-market performance of a model at a certain moment in time will disappear once the market incorporates it. Thus, as the benefits of using textual data to enhance stock price predictions become more salient, the smaller those benefits will become.

References

Alzazah, Faten, Cheng, Xiaochun. "Recent Advances in Stock Market Prediction Using Text Mining: A Survey". E-Business - Higher Education and Intelligence Applications, edited by Robert Wu, Marinela Mircea, IntechOpen, 2020. 10.5772/intechopen.92253.

Bozanta A., S. Angco, M. Cevik and A. Basar, "Sentiment Analysis of StockTwits Using Transformer Models," 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 2021, pp. 1253-1258, doi: 10.1109/ICMLA52953.2021.00204.

Bujari A., Furini M., Laina N. On Using Cashtags to Predict Companies Stock Trends. 2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)

Checkley, M. S., D. Añón Higón, and H. Alles. "The hasty wisdom of the mob: How market sentiment predicts stock market behavior." Expert Systems with applications 77 (2017): 256-263.

Dhankar, R., Maheshwari, S. Behavioural Finance: A New Paradigm to Explain Momentum Effect (May 27, 2016). Available at SSRN: https://ssrn.com/abstract=2785520 or http://dx.doi.org/10.2139/ssrn.2785520

Hu , Z.; Zhao, Y.; Khushi, M. A Survey of Forex and Stock Price Preduction Using Deep Learning, Appl. Syst. Innov. 2021, 4, 9.

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) (pp. 1003- 1007), IEEE, 2018

Kalbandi, I., Jare, A., Kale, O.V., Borole, H., Navsare, S. (2021). Stock Market Prediction Using LSTM.

Li, Y., Pan, Y. A novel ensemble deep learning model for stock prediction based on stock prices and news. Int J Data Sci Anal 13, 139–149 (2022). https://doi.org/10.1007/s41060-021-00279-9

Lim, B., Zohren, S. (2021). Time-series forecasting with deep learning: a survey. Philosophical Transactions of the Royal Society A, 379(2194), 20200209.

Mahmoud, A., Mohammed, A. (2021). A Survey on Deep Learning for Time-Series Forecasting. In: Hassanien, A.E., Darwish, A. (eds) Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges. Studies in Big Data, vol 77. Springer, Cham. https://doi.org/10.1007/978-3-030-59338-4$_1$9

M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, "Deep Learning for Stock Market Prediction", Entropy, 22, 840, 2020. M. Obthong, N. Tantisantiwong, W. Jeamwatthanachai, G. Wills, “A Survey on Machine Learning for Stock Price Prediction: Algorithms and Techniques”, Conference: 2nd International Conference on Finance, Economics, Management and IT Business, 2020.

S. Raschka, Y. Liu, V. Mirjalili, "Machine Learning with Pytorch and Scikit-Learn", Packt, 2022.

Torres, Hadjout, Sebaa, Martínez-Álvarez, and Troncoso. Big Data. Feb 2021. 3-21. http://doi.org/10.1089/big.2020.0159

N. Rouf, M. B. Malik, T. Arif, S. Sharma, S. Singh, S. Aich, H.-C. Kim, "Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions.", Electronics, 10, 2717, 2021.

P. Soni, Y. Tewar, D. Krishnan, "Machine Learning Approaches in Stock Price Prediction: A Systematic Review", Journal of Physics: Conference Series, 2161 012065, 2021.

Rouf N, Malik MB, Arif T, Sharma S, Singh S, Aich S, Kim H-C. Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions. Electronics. 2021; 10(21):2717.

Wu GG, Hou TC, Lin JL. Can economic news predict Taiwan stock market returns? Asia Pacific Management Review. 2019;24(1):54-59

Xu, Feifei, and Vlado Keelj 2014. "Collective sentiment mining of microblogs in 24-hour stock price movement prediction." 2014 IEEE 16th conference on business informatics. Vol. 2. IEEE, 2014.