Abstract—Stock forecasting is challenging because of stock volatility and dependability on external factors, such as economic, social, and political factors. This motivates investors to seek tools to identify stock trends to reap profits.

In this research, we compared several heterogeneous ensembles for financial forecasting, including averaging, weighted, stacking, and blending ensembles. In addition, we used a random forest regressor as the baseline.

Regression was used to predict the next day’s closing stock price. We used classification to label closing stock value as HIGH or LOW by comparing with the opening stock value of a particular company. We used Long Short Term Memory (LSTM) models, Linear Regression, and Support Vector Machines (SVM) as individual models. Further, we analyzed 10 years of historical data of the most active 20 companies of the NASDAQ stock exchange for implementing ensemble models.

In conclusion, experimental results depict blending ensembles perform the best out of compared ensembles in financial forecasting. Further, they reveal SVM is under-performing, LSTM outputs are satisfactory, while linear regression produced promising results.

Index Terms—Support vector machines, Long short term memory, Time series analysis, Regression analysis, Stock markets, Financial forecasting, Ensemble methods

I. INTRODUCTION

The stock market is a collection of markets and exchanges. Regular activities of buying, selling, and issuing of shares of public companies take place at them. Modern stock exchanges use electronic communication to make trades, which is helpful in speed and to reduce the costs. The typical stock values of an exchange are highly volatile. Factors such as supply and demand, interest rates, dividends, management of a company, the economy of the country/region, and political climate affect the stock values [1].

Investors and researchers use various methods to predict stock prices. In the modern age, machine learning based technologies have emerged because of their performance and accuracy in predictions [2]. A model of machine learning is an entity that can identify patterns existing in data through learning. In machine learning techniques, we use different models for the process of prediction. An ensemble approach combines these individual machine learning models together. Thus, they yield more accurate results in most of the cases [3]. We can implement these ensemble prediction methods as a product. Hence, such a product allows a user to get a reasonably accurate prediction. This helps to profit for a user when trading through the stock market. Therefore, this will be useful for investors, regardless of their experience on the subject.

The primary aim of the research was to investigate about heterogeneous ensemble methods for improving the accuracy of the prediction of stock values.

II. RELATED WORK

Most researchers have proven that the ensemble methods outperform the individual component models in both regression and classification [4]. Lawrence, R [5] have concluded that Neural Networks provide more accurate predictions than statistical and regression techniques.

Support Vector Machines (SVMs) are widely being used for stock prediction. Hence, SVMs are being researched throughout the literature related to financial forecasting. Many researchers employed SVMs for financial forecasting in most research papers we referred [1], [3], [6], [7]. For an example, a set of N Support Vector Machines (SVM) comprises the proposed ensemble model in [6]. Further, SVMs have proven to surpass traditional Back Propagation Neural Networks (BPNs) and Case-Based Reasoning models (CBR) [8], [9].

Asad [3] and Weng et al. [7] has used random forest classifiers in ensembles for predicting the stock. Narayanan and Govindarajan [1] has used Naïve Bayes’ algorithm and SVMs to build ensembles for stock prediction. Patel et al. [10] have conducted a comparison between the SVMs, Naive Bayes, Artificial Neural Networks (ANNs), and Random Forests. They have stated that random forest outperforms other techniques. Araújo and Ferreira [11] had proposed an Evolutionary Morphological Linear Forecasting (EMRLF) for financial forecasting. They have compared Multi-Layer Perceptron (MLP) networks and Time-delay Added Evolutionary Forecasting (TAEF) with their proposed method, and proved that EMRLF outperforms both MLP and TAEF.

Deep Learning is attracting much attention currently in many research areas. Therefore, many researchers have explored the usage of deep learning for stock market prediction [12], [13].

Researchers commonly use econometric models, such as Generalized Auto Regressive Conditional Heteroscedasticity (GARCH) for stock forecasting. Hajizadeh et al. [14] have proposed such hybrid models comprising GARCH and ANN for stock prediction. Kim and Won [15] have explored about the combination of econometric models with Deep Neural
Networks. They integrated GARCH models with Long-Short Term Memory (LSTM) models in their research. Further, they have shown that usage of several GARCH models with LSTM models lead to higher performance in stock prediction.

We focus our research on investigating heterogeneous ensembles. It differs from the ensemble survey of homogeneous ensembles such as AdaBoost and Bagging outlined in [4]. Gonzalez et al. [6] have investigated an ensemble combining N SVMs by Majority Voting. Even though our research uses SVM as a base classifier, we do not use N base SVM classifiers. Zhang, G [15] had investigated a hybrid of an Auto-Regressive Integrated Moving Average (ARIMA) and an ANN model to gain advantages of linear and non-linear modeling. To some extent, this is like our research because it is a heterogeneous ensemble. However, we are combining a linear regression model, SVM, and an ANN. Tsai et al. [17] has investigated majority voting and bagging ensemble methods that comprise heterogeneous and homogeneous ensembles. They have compared those ensembles with single classifiers such as Multi-Layer Perceptron (MLP), Classification And Regression Tree (CART) decision trees, and Logistic Regression models. Further, they have compared homogeneous ensembles vs heterogeneous ensembles. Although their research is like our research, our focus is to compare heterogeneous ensembles against single classifiers only. We do not compare heterogeneous ensembles with homogeneous ensembles.

III. THEORY

A. Support Vector Machine (SVM)

Vapnik et al. introduced the SVM [18]–[20] which is used for binary classification. SVMs with hard margins [20] fails for problems with noisy data and outliers. Hence, Vapnik et al. introduced SVMs with soft margins [18], [19] to resolve such problems. SVMs with soft margins allow the classifier to misclassify some data points. In contrast to classification, SVMs have applications for regression as well. With Support Vector Regression (SVR) [21], instead of finding an optimal hyperplane that separates data points, SVR finds a function that best approximates target values at a specified level of error tolerance.

B. LSTM

Long Short Term Memory (LSTM) networks fall into the recurrent neural network category. Some designs in RNN were used to predict stock market in similar cases, like [22], [23]. Recurrent networks differ from traditional feed-forward networks providing connections bi-directionally while LSTMs differ from the traditional neural networks with the persistency. Schmidhuber and Hochreiter [24] introduced LSTM to tackle the vanishing gradient issue that recurrent networks have while dealing with long data sequences. Further, they proved LSTM is one of the most successful RNN architectures in [24].

Chen, Kai et al. [25] have shown that LSTM architectures have improved accuracy than random prediction. Henceforth, many researchers [26]–[28] used LSTM for stock prediction.

C. Ensemble

An ensemble model is a collective outcome of individual models, which are combined in such a way that the model outperforms each individual model in most cases.

There are a variety of ensemble techniques. Ensembles fall into two categories as homogeneous ensembles and heterogeneous ensembles based on the learning algorithms used. Homogeneous ensembles use single base learning algorithm, and popular techniques are bagging, boosting, etc. In contrast, heterogeneous ensemble techniques use different base learning algorithms. For example, popular heterogeneous ensembles include methods such as averaging ensembles, stacking ensembles, weighted ensembles, and blending ensembles.

The averaging ensembles takes the mean value of the prediction values of its individual models. The weighted ensemble linearly combines regression outputs of different models, where each model gets a weight based on their performances. Here, the best performing individual model gets the most weight, and we gave a lower weight to other two individual models. Still, these two models can overcome the prediction by the best model by combining. For example, assume we assign 3/7 weight for the best model and 2/7 weight for other two models. Then the weak individual models with less weight can collectively overrule the best model.

A stacking ensemble trains several individual models in parallel and combines them. There are two stages in stacking. In the first stage, we train a set of individual models on the raw data and make predictions on that raw data to generate more features. Then, another model at the second stage predicts the final test data by utilizing the generated features. Base models or level-0 learning models are the terms used to refer models trained at the first stage of the stacking. The terms meta-model or a level-1 learning model refer to models that train at the second stage of the stacking process. However, in practice there can be multiple layers of stacking. The blending ensembles are like the stacking ensembles, but blending ensembles use a hold-out data set to train the meta-model. Further, in blending this meta-model is a linear model. With regression, a meta-model can be a linear model, such as linear regression or logistic regression. Figure [1] shows the training process of a stacking ensemble model. We show only 3 base learners for the simplicity. Base learners train on the raw training data, as shown in Figure [1] We then feed the predictions from those base learners to the meta-learner as training data. Once we train the entire stacking ensemble model, we can make predictions on testing data.

![Fig. 1. Training process of an stacking ensemble model](image-url)
IV. Methodology

We carried out stock prediction as a classification and a regression separately. In classification, we classified target classes as 'HIGH' or 'LOW'. We derived these 'HIGH' and 'LOW' labels by comparing the closing stock price for a particular company with the opening stock price for a particular day. For example, if the closing stock price is higher than the opening stock price, we labeled it as 'HIGH' and as 'LOW' if it is not. Further, in classification, we predicted these class labels for the next day. In regression, we predicted the actual closing stock price of the next day.

Poon and Granger [29] extensively reviewed volatility forecasting in financial markets. In conclusion, they reveal that volatility measured over many past years gives correct forecasts. Further, John Hanke and Dean Wichern [30] show there should be at least 2T to 6T amount of data where T refers to the length of the seasonality. In addition, the number of data points to be used depends on the variability of the data and the model being built [31].

We used train test split as the data splitting technique to preserve temporal data patterns. Cerqueira et al. [32] reveal that out-of-sample methods that preserve the temporal order of observations during data splits give better results.

Shynkevich et al. [33] illustrate that forecasting step affects the prediction accuracy of SVMs. They assert SVMs gain the highest prediction accuracy when the forecasting step is one step. Further, they [34] suggest that SVMs give better performance when the forecasting horizon approximates the input window length. In addition, Chan et al. [35] reveal that shorter forecast horizons are easier to predict and give better results. Therefore, considering these facts, we are forecasting only one day forward from the current date. Hence, this research focuses on short-term stock prediction.

We considered several heterogeneous ensemble techniques. Some simple ensembles are averaging ensemble and the weighted ensemble. In addition, we implemented more advanced ensembles such as stacking ensembles and blending ensembles. For the stacking ensemble, we combined only linear regression and SVR models. However, all other ensembles used all three models namely, the linear regression model, SVR model, and the LSTM model. We used a random forest regressor ensemble as the baseline for regression ensembles.

In terms of classification, we considered accuracy, precision, recall and F1 score as the performance criteria. Further, we used a random forest classifier ensemble as the baseline for the classification. In addition, for regression we used Root Mean Squared Error (RMSE) and R2 Score performance metrics.

\[
RMSE = \sqrt{\frac{1}{n} \cdot \sum_{i=0}^{n} |Y_{predicted} - Y_{true}|^2}
\]

\[
R^2 \text{ score} = 1 - \frac{\text{Residual sum of squares}}{\text{Total sum of squares}}
\]

Similar to classification, we have used a Random Forest Regressor as the baseline comparator for regression models.

V. Experimental Setup

A. Raw data set

We collected raw data from the NASDAQ, which is an American stock exchange [36]. We selected only the most active 20 companies because of computational complexity. Based on previous work, we took 10 years of historical data of these most active 20 companies into consideration. They are 10 years backwards from 21st September 2019, depending on the availability. Raw data comprises open, close, high, and low attributes for stock price for a particular day in a specific company. The volume column shows the total number of transactions per day. There are 40044 data points in total. A summary of data is given in TABLE I.

| Symbol | Company                           | Data Points | No of years |
|--------|-----------------------------------|-------------|-------------|
| AAPL   | Apple Inc.                        | 2519        | 10          |
| AMD    | Advanced Micro Devices Inc.       | 2519        | 10          |
| CDW    | CDW Corporation                   | 1571        | 6           |
| CMCSA  | Comcast Corporation               | 2519        | 10          |
| CSCO   | Cisco Inc.                        | 2519        | 10          |
| ETSY   | Etsy Inc.                         | 1118        | 4           |
| FB     | Facebook Inc.                     | 1848        | 7           |
| INTC   | Intel Corporation                 | 2519        | 10          |
| JD     | JD.com Inc.                       | 1344        | 5           |
| MRNA   | Moderna Inc.                      | 199         | 1           |
| MSFT   | Microsoft Corporation             | 2519        | 10          |
| MU     | Micron Technology Inc.            | 2519        | 10          |
| NFLX   | Netfix Inc.                       | 2519        | 10          |
| OQQ    | InvescoQQQ Trust                  | 2519        | 10          |
| ROKU   | Roka Inc.                         | 499         | 2           |
| SIRI   | Sirius XM Holdings Inc.           | 2519        | 10          |
| TQQQ   | ProShares UltraPro QQQ            | 2420        | 10          |
| TVIX   | VelocityShares Daily 2x VIX       | 1526        | 6           |
| VEOY   | VEOY Ltd.                         | 2376        | 9           |
| ZNGA   | Zynga Inc.                        | 1953        | 8           |

B. Data Pre-processing

Data pre-processing transforms raw data to be compatible with machine learning models. Hence, in this research we used the following data pre-processing techniques on raw data.

- **Missing values**: In the raw data set, only the last data points (21st of September 2019) had missing values and improper formatting. Hence, we dropped it in each company.
- **Data type conversions**: We converted data types to proper data formats. For example, conversion of attributes to numerical values and date attributes which were string values in the original data.
- **Standardization**: We standardized data for each attribute so they have zero mean and unit variance.
- **Scaling of target values**: Scaling/standardization of target variables in terms of regression may sometimes depend on the model being chosen. However, in terms of neural networks, literature highlights that it is better to scale both input and target values [37], [38].

C. Implementation

We used Python language for the implementation. In addition, we have used python libraries such as Keras framework with TensorFlow as the backend, Sci-Kit Learn for machine learning. Further, we used Numpy and Pandas packages to
manage data. We built the neural network for stock prediction based on earlier research and trial and error. We selected the window size of the LSTM network too based on previous research. However, we used Bayesian optimization for selection of the number of neurons. We chose Adam (Adaptive Moment Estimation) as the optimization method because of its efficient nature, both memory-wise and computationally. The major parameters that need to be used for SVM in classification are the kernel, C, and gamma. Alexander Smola and Bernhard Schölkopf \cite{39} recommend using Gaussian kernels when a few extra information is available. Tay and Cao \cite{9} depict smaller C values in SVMs lead to under-fitting and larger C values in SVMs lead to over-fitting. They suggest an optimal range to test for C values is from 10 to 100 and the polynomial kernels give inferior results & take a long time to train. Hence, considering these facts, in this research, we used Radial Basis Function (RBF) as the kernel function and we tested different C values. We tuned the parameters using the Grid-Search-CV to find optimal parameters. In Linear Regression, we used the default values provided by Sci-Kit-library. Important attributes are fit_intercept = True and normalize = False, which means to calculate an intercept and to ignore scaling, respectively. We ignored scaling because we scaled data first and then fed to the Linear Regression model.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

A. Classification

TABLE II summarizes the experimental results for classification using SVM and baseline. Results shown in TABLE II are for C = 100, gamma = 0.01, and kernel = Radial Basis Function where these parameters were obtained by grid-search.

| Algorithm | APL | AMD | CDW | CMCSA | CSKO | ETSY | FB | INTC | JD | JR | MSFT | MU | NFLX | SIRI | TQQQ | TVIX | VEON | ZNGA |
|-----------|-----|-----|-----|-------|------|------|----|------|----|----|------|-----|------|-----|-----|-----|-----|-----|
| Accuracy  | 0.80| 0.68| 0.60| 0.55 | 0.62 | 0.52 | 0.60| 0.69 | 0.55| 0.55| 0.55 | 0.65 | 0.52 | 0.56 | 0.55 | 0.57 | 0.52 | 0.56 |
| Precision | 0.49| 0.55| 0.46| 0.55 | 0.49 | 0.51 | 0.50| 0.50 | 0.47| 0.66 | 0.47 | 0.66 | 0.62 | 0.57 | 0.62 | 0.47 | 0.67 | 0.62 |
| Recall    | 0.60| 0.55| 0.46| 0.55 | 0.49 | 0.51 | 0.50| 0.50 | 0.47| 0.66 | 0.47 | 0.66 | 0.62 | 0.57 | 0.62 | 0.47 | 0.67 | 0.62 |
| F1 Score  | 0.61| 0.55| 0.46| 0.55 | 0.49 | 0.51 | 0.50| 0.50 | 0.47| 0.66 | 0.47 | 0.66 | 0.62 | 0.57 | 0.62 | 0.47 | 0.67 | 0.62 |

B. Regression

We used neural network, linear regression, and Support Vector Regression (SVR) approaches as individual approaches.

C. Ensemble results

TABLE IV shows the Root Mean Squared Error (RMSE) for averaging, weighted, stacking, and blending ensembles along with the baseline random forest regressor ensemble.

| Algorithm | Random Forest | Averaging ensemble | Weighted ensemble | Blending ensemble |
|-----------|---------------|-------------------|------------------|------------------|
| Accuracy  | 0.80          | 0.76              | 0.80             | 0.80             |
| Precision | 0.55          | 0.66              | 0.66             | 0.66             |
| Recall    | 0.61          | 0.55              | 0.61             | 0.61             |
| F1 Score  | 0.62          | 0.55              | 0.62             | 0.62             |

Tay and Cao \cite{9} suggest that improper kernel selection in SVMs can cause longer training times and lead to poor performance. In fact, through our experimental setup, we got similar results validating the findings of \cite{9} in terms of training times. Further, we observed using certain parameter combinations, lead to longer training times.

We found the following best hyper-parameters for SVM using RBF kernel, C = 100, gamma = 0.01 in this experimental
setup. However, even using those parameters, the performance was low for all the companies. In contrast, linear regression has produced promising results. Further, we can conclude that a higher positive correlation exists between the target values and the training features, which has resulted in better performance of the linear regression model. The opening stock prices and closing stock prices have a relationship that is almost linear. Still, linear regression fails over several companies as illustrated in experimental results (See TVIX). The probable reasons behind the failure of linear regression for TVIX is that they have non-linear trends or because of outliers. As a result, the linear model cannot fit to those companies properly. Based on our experimental setup, we establish that scaling of training features and target values in terms of regression is essential for training of neural networks.

VII. CONCLUSION

In this research, we compared several heterogeneous ensembles by combining results from an ANN, an SVM, and a Linear Regression for stock prediction. We approached the stock prediction problem as a classification and as a regression. We observed SVMs could not capture the underlying patterns in the time series data in the classification. Therefore, this experimental setting was not suitable for time-series classification. A reason for such failure could be the time-axis distortion problem when using classical kernels, like the Gaussian Radial Basis Function (GRBF). In particular, GRBF kernels cannot identify patterns in time series data that contain patterns shifted in time, distorted & scaled. However, Gudmundsson et al. [37] suggests that using kernels with Dynamic Time Warping (DTW) provides better performance for such time-series data.

We identified common pitfalls and mistakes to be avoided during financial forecasting via this research. For example, there could be mistakes when scaling the target values of financial time-series data. Scaling/standardization of target values may depend on the model being chosen. It is better to scale target values for neural networks. The reason for such suggestion is that neural networks may become unstable if large weight variations occur while training the neural networks. Even the literature related to neural network suggests doing so [37], [38]. In terms of classification of time-series problems, when using SVMs, it might be tricky to choose proper hyper-parameters for SVMs. Even the most commonly used kernels for SVMs, such as the Gaussian Kernels, might not capture the time series data patterns when we use SVMs for time-series classification. In fact, we found that using the Gaussian Radial Basis Function (GRBF) kernel, we can not capture the time-series pattern in our experimental setup. Sometimes, improper data splitting of time series data impacts the prediction performance. Since the time series data has temporal patterns and seasonalties, incorrect data splits may break out those hidden patterns. Hence, usual techniques used for data splitting, such as cross-validation, may not be suitable for time series data.

VIII. FUTURE WORK

We investigated only 20 companies and 10 years of data points due to scope and time limitations. In addition, future research could explore the validity of the results with more data from different stock markets and time ranges. We have focused on using only SVM for the classification approach. As subsequent improvements, we propose looking further into other classification techniques. Further, researchers can look into non-linear regression models because we only researched simple linear regression in this study. In terms of neural networks, we have considered only Long Short Term Memory (LSTM) neural networks with a few configurations. As future work, one could start with more hidden layers, more neurons, various optimizers, activation functions, and see whether results change. Interested researchers can even try to apply different neural networks for financial forecasting. In terms of the ensembles, we examined only a few heterogeneous ensembles. However, we have mostly used only the default parameters provided by the libraries that we used, in those ensemble models. Further research can extend these to the hyper-parameter tuning of individual models and building better ensembles.

ACKNOWLEDGMENT

The authors would like to extend their heartfelt regards to Dr. Damayanthi Herath for her immense support in making this research a success and guiding us whenever necessary.

REFERENCES

[1] B. Narayanan and M. Govindarajan, “Prediction of stock market using ensemble model,” International Journal of Computer Applications, vol. 128, no. 1, pp. 18–21, 2015. [Online]. Available: https://doi.org/10.5120/jca201506412

[2] S. K. Sharma and V. Sharma, “Comparative analysis of machine learning techniques in stock forecasting,” International Journal of Computer Applications, vol. 53, no. 6, pp. 51–54, 2012. [Online]. Available: https://doi.org/10.5120/8429-2198

[3] M. Asad, “Optimized stock market prediction using ensemble learning,” in 2015 9th international conference on application of information and communication technologies (AICT). IEEE, 2015, pp. 263–268. [Online]. Available: https://doi.org/10.1109/ICAICT.2015.7338559

[4] T. G. Dietterich, “Ensemble methods in machine learning,” in International workshop on multiple classifier systems. Springer, 2000, pp. 1–15. [Online]. Available: https://doi.org/10.1007/3-540-45014-9_1

[5] R. Lawrence, “Using neural networks to forecast stock market prices,” University of Manitoba, vol. 333, pp. 2006–2013, 1997. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.460.8845

[6] R. T. Gonzalez, C. A. Padilha, and D. A. C. Barone, “Ensemble system based on genetic algorithm for stock market forecasting,” in 2015 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2015, pp. 3102–3108. [Online]. Available: https://doi.org/10.1109/CEC.2015.7257276

[7] B. Weng, L. Lu, X. Wang, F. M. Megahed, and W. Martinez, “Predicting short-term stock prices using ensemble methods and online data sources,” Expert Systems with Applications, vol. 112, pp. 258–273, 2018. [Online]. Available: https://doi.org/10.1016/j.eswa.2018.06.016

[8] B. Gui, X. Wei, Q. Shen, J. Qi, and L. Guo, “Financial time series forecasting using support vector machine,” in 2014 Tenth International Conference on Computational Intelligence and Security. IEEE, 2014, pp. 39–43. [Online]. Available: https://doi.org/10.1109/CIS.2014.22

[9] F. E. Tay and L. Cao, “Application of support vector machines in financial time series forecasting,” omega, vol. 29, no. 4, pp. 309–317, 2001. [Online]. Available: https://doi.org/10.1016/S0305-0483(01)00026-3

[10] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques,” Expert systems with applications, vol. 42, no. 1, pp. 259–268, 2015. [Online]. Available: https://doi.org/10.1016/j.eswa.2014.07.040
D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, “Stock market’s price prediction,” *Information Sciences*, vol. 237, pp. 3–17, 2013. [Online]. Available: https://doi.org/10.1016/j.ins.2009.07.007

T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018. [Online]. Available: https://doi.org/10.1016/j.ejor.2017.11.054

A. Yoshihara, K. Fujikawa, K. Seki, and K. Uehara, “Predicting stock market trends by recurrent deep neural networks,” in *Pacific rim international conference on artificial intelligence*, vol. 8862, Springer, 2014, pp. 759–769. [Online]. Available: https://doi.org/10.1007/978-3-319-13560-1_60

E. Hajizadeh, A. Seti, M. F. Zarandi, and I. Turksen, “A hybrid modeling approach for forecasting the volatility of S&P 500 index return,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 431–436, 2012. [Online]. Available: https://doi.org/10.1016/j.eswa.2011.07.033

H. Y. Kim and C. H. Won, “Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple garch-type models,” *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018. [Online]. Available: https://doi.org/10.1016/j.eswa.2018.03.002

G. P. Zhang, “Time series forecasting using a hybrid arima and neural network model,” *Neurocomputing*, vol. 50, pp. 159–175, 2003. [Online]. Available: https://doi.org/10.1016/S0925-2312(01)00702-0

C.-F. Tsai, Y.-C. Lin, D. C. Yen, and Y.-M. Chen, “Predicting stock returns by classifier ensembles,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2452–2459, 2011. [Online]. Available: https://doi.org/10.1016/j.asoc.2010.10.001

C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: https://doi.org/10.1023/A:1007858810918

V. N. Vapnik, “An overview of statistical learning theory,” *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999. [Online]. Available: https://doi.org/10.1109/72.788640

B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152. [Online]. Available: https://doi.org/10.1145/130385.130401

H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” *Advances in neural information processing systems*, vol. 9, pp. 155–161, 1997.

E. W. Saad, D. V. Prokhorov, and D. C. Wunsch, “Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks,” *IEEE Transactions on neural networks*, vol. 9, no. 6, pp. 1456–1470, 1998. [Online]. Available: https://doi.org/10.1109/72.728395

A. M. Rather, A. Agarwal, and V. Sastry, “Recurrent neural network and a hybrid model for prediction of stock returns,” *Expert Systems with Applications*, vol. 42, no. 6, pp. 3234–3241, 2015. [Online]. Available: https://doi.org/10.1016/j.eswa.2014.12.003

S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/089976697300017352

K. Chen, Y. Zhou, and F. Dai, “A lstm-based method for stock returns prediction: A case study of china stock market,” in *2015 IEEE international conference on big data (big data)*. IEEE, 2015, pp. 2823–2824. [Online]. Available: https://doi.org/10.1109/BigData.2015.7364089

S. Liu, X. Liao, and Y. Ding, “Stock transaction prediction modeling and analysis based on lstm,” in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2018, pp. 2787–2790. [Online]. Available: https://doi.org/10.1109/ICIEA.2018.8398183

M. Roondiwala, H. Patel, and S. Varma, “Predicting stock prices using lstm,” *International Journal of Science and Research (IJSR)*, vol. 6, no. 4, pp. 1754–1756, 2017.

D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, “Stock market’s price movement prediction with lstm neural networks,” in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1419–1426. [Online]. Available: https://doi.org/10.1109/IJCNN.2017.7966019

S.-H. Poon and C. W. Granger, “Forecasting volatility in financial markets: A review,” *Journal of economic literature*, vol. 41, no. 2, pp. 478–539, 2003. [Online]. Available: https://doi.org/10.1257/jole.41.2.478

J. E. Hanke and D. Wichern, *Business Forecasting; Pearson New International Edition*, 9th ed. Harlow: Pearson Education Limited, 2014.