Tokyo Stock Exchange Prediction

PAROUSIDOU VASILIKI CHRYSOVALANTO
Dept. of Informatics
Thessaloniki, Greece
vparousid@csd.auth.gr

CHATZICHRISTODOULOU ZOI
Dept. of Informatics
Thessaloniki, Greece
zchatzi@csd.auth.gr

KALIAKATSOS CHARILAOS
Dept of Informatics
Thessaloniki, Greece
kaliakac@csd.auth.gr

ABSTRACT

Keywords: stock market prediction; machine learning (ML); long short-term memory (LSTM); extreme gradient boosting (XGBoost); random forest

1 INTRODUCTION

Stock market prediction is a challenging real-world problem at the intersection of finance and computer science. A stock market is a public market where someone can purchase and sell shares for various companies. The stocks, which are also known as equities, represent ownership in the company. By issuing stocks, companies can raise capital for expanding or paying the dept. The investors can make profits by either the companies whose stocks pay regular dividends or by selling their stocks when their values reach a higher rate than the one at which these stocks were purchased. In order to succeed, someone has to identify solid investments. In general, if a stock is undervalued, someone should purchase it, while when it is overvalued, someone should sell it. However, investing funds involves various market risks since the stock market is an ever-changing field with many highs and lows as companies succeed or go under. Thus, predicting stocks is of great importance to those that engage in the stock market for detecting accurate profits and reducing potential market risks.

The attempt to forecast future stock prices is one of the most significant challenges because of the numerous imponderable factors that contribute to the fluctuation, such as demand and supply, government policies, general economic conditions, investors' expectations, etc. Over the years, traders, analysts, and researchers have been interested in predicting stocks, whereas some theories suggest that stock markets behave in a random walk. Initially, fundamental and technical analysis have been used, where the first one uses the intrinsic value (what an asset is worth) while the later relies on the basis of charts. Moreover, linear models, such as Linear Regression and statistical models, which include exponential smoothing and autoregressive integrated moving average (ARIMA), have also been utilized. Finally, the development of machine learning has led to the prediction of stocks using various algorithms and ensemble models, while deep learning techniques also yield really promising results.

The analysis in the current work is implemented in the context of a "Kaggle" competition (https://www.kaggle.com/competitions/jpx-tokyo-stock-exchange-prediction/overview). More specifically, it contains financial data from the Japanese market, such as stock information and historical stock prices. The aim is to predict the future returns of about 2000 stocks and rank them from highest to lowest expected returns. Then, the top 200 stocks are considered purchased and the bottom 200 shorted. The stocks are weighted based on their ranks and the total returns for the portfolio are calculated assuming the stocks were purchased the

next day and sold the day after that. The results are evaluated on the Sharpe Ratio [8]. The Sharpe Ratio (or Sharpe index) measures the return of an investment compared to its risk. The ratio is the average return earned in excess of the risk-free rate per unit of volatility (fluctuations of an asset). Generally, the greater the value of the Sharpe ratio, the more attractive the risk to take.

The remaining sections of this paper are organized as follows: In Section 2, the related work is presented. In Section 3, the dataset is described and analyzed, while Section 4 presents our approach and the methods that have been implemented. In Section 5, the metrics that will be used for the evaluation of our approach are discussed, and in Section 6, the experimental results of the various models used are compared. The conclusions and some possible future research directions are presented in Section 7.

2 RELATED WORK

In this section, we present some of the existing approaches when dealing with stock market prediction. First, Ballings, Michel, et al [7] compare some traditional machine learning classifiers to three ensemble methods using data from 5767 publicly listed European companies. They conclude that Random Forest is the best algorithm followed by Support Vector Machines, Kernel Factory, AdaBoost, Neural Networks, k-Nearest Neighbors, and Logistic Regression. Selvamuthu et al [2] train artificial neural networks (ANNs) based on three learning algorithms and specifically Levenberg-Marquardt, Scaled Conjugate Gradient, and Bayesian Regularization for stock market prediction and compare their results using both tick data and 15-min data of an Indian company. Jiang [1] describes the different data sources that can be used and their preprocessing and gives a review of deep learning techniques used in stock market prediction from 2017 to 2019. The deep learning methods that he analyzes include feedforward neural networks, convolutional neural networks, long short-term memory (LSTM) networks, transfer learning and reinforcement learning. Moreover, Nti, Isaac Kofi et al [4] analyze and compare various ensemble techniques, such as stacking, bagging, blending, and boosting using decision trees, support vector machines and neural networks as base classifiers. They evaluate these methods in stock data from Ghana Stock Exchange (GSE), Johannesburg Stock Exchange (JSE), Bombay Stock Exchange (BSE-SENSEX), and New York Stock Exchange (NYSE) and they conclude that stacking and blending methods yield better results in terms of accuracy and root means square error (RMSE). Finally, Nabipour, Mojtaba, et al. [6] implemented several machine learning algorithms to predict future values of four Iranian stock market groups. They compared the performance of Decision Tree, Random Forest, Adaboost, gradient boosting, XGBoost, ANNs, Recurrent Neural Networks (RNN) and LSTM and proved that LSTM yields better results in terms of Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Relative Root Mean Square Error (RRMSE) and Mean Squared Error (MSE).

3 THE DATA

The given data set is structured in a set of arrays with historical data concerning the Japanese stock market including the JPX target shares.

The data set includes a table that describes the structural information for each of 4.417 total shares such as its identification code, the Issued shares as well as a flag of the prediction target universe. Apart from the above array, the remaining tables are time-dependent and consist of numerous entries sorted in chronological order. In further detail the main arrays are:

- **Stock_prices**: The main table that includes the goal column time series and some extra features for each stock. The features it contains are further analyzed in Table 1 Stock_prices.
- **Options**: Based on the broader market, information on the status of a range of options. Even though many options are not assessed explicitly, they may be of importance

because they offer implicit projections of future stock market prices.

- **Secondary_stock_prices:** The main dataset includes the 2,000 most widely traded stocks, but the Tokyo market also trades numerous less liquid assets. This file provides information on securities that aren't rated but may be useful in evaluating the market as a whole.
- **Trades:** Trading volumes from the preceding business week are summarized.
- **Financials:** Quarterly profits reporting results.

Feature	Definition
Open	First traded price on a day
High	Highest traded price on a day
Low	Lowest traded price on a day
Close	Last traded price on a day
Volume	Number of traded stocks on a day
Adjustment Factor	Factor to calculate theoretical price/volume when split/reverse-split happens
Expected Dividend	Expected dividend value for ex-right date. This value is recorded 2 business days before ex-dividend date.
Supervision Flag	Flag of Securities Under Supervision & Securities to be Delisted

Table 1 Stock_prices

The object of the final forecast is the index TARGET which constitutes the change ratio of adjusted closing price of stock *k* between day *t+2* (two days after prediction) and day *t+1* (the day after prediction) where day *t+0* is TradeDate and is given by the following formula.

$$r_{(k,t)} = \frac{C_{(k,t+2)} - C_{(k,t+1)}}{C_{(k,t+1)}}$$

The main measurements that will contribute most to the prediction model are located in the first array and are related to the stock price values per day hence the goal value is inextricably related to those measurements.

Finally, in addition to the above static dataset, a time series API that fetches the time-sensitive stock prices is provided in order to feed the implemented prediction model with the newer data. It is worth mentioning that the daily given feed is referred to two days before because the prediction model should predict two days ahead. Moreover, not all 2000 stocks provide data daily so that should be taken into account in the prediction model.

4 APPROACH

This section presents the procedure that was followed to pre-process our data and implement machine learning techniques to predict the stock market.

4.1 DATA ANALYSIS

4.2 DATA PREPROCESSING

In the preprocessing phase, we start with the feature engineering. First, we group by the "SecuritiesCode" column which is the identity of the stock. After that, for each stock we use the rolling function, that creates a rolling window with a given size and calculates the mean value for a specific feature. In this way, we create four new features by using windows of sizes 5, 10, 15 and 20 days and calculating the mean value of the "Close" column. Moreover, we create four more features depending on the date. More specifically, we extract the quarter of the year since a stock may drop in specific quarters. Also, we extract the day, the month, and the year. This can add more information to the model as in general it's better to buy stocks on Monday, before the market opens, and sell on Friday, before the market closes. Other features that are created are the difference between the highest traded price on a day and the lowest traded price on a day and the difference between the closing and the opening price for a given stock.

Finally, MinMaxScaler was used for scaling the data so that we can bring all the price values to a common scale.

4.3 PREDICTION MODEL

As for the main part of our approach, various algorithms were implemented and compared. More specifically, Linear Regression was applied, which is a simple linear approach, to model the relationships between the target and the independent variables. K-Nearest Neighbor is another simple technique that has low computational cost, whose prediction is an interpolation of the targets of the most similar data to the needed estimation. The number of neighbors was selected to be 50 after testing a certain number of neighbors using a validation set.

In addition, linear Support Vector Regression (SVR) was implemented using the default values for parameters, where a straight line, referred as hyperplane, fits the data, and is used to predict the continuous output. Moreover, some ensemble algorithms were trained such as Random Forest, Adaboost and XGBoost. Ensemble regression combines various models in order to improve the prediction accuracy in learning problems with a numerical target value. Random Forest is a meta-estimator that trains some decision trees on different sub-samples of the data and averages their predictions. The number of decision trees was configured to be 80 and the maximum depth of each tree was selected to be 10. Adaboost is another meta-estimator that begins by fitting a regressor on the original data and, after that, it fits additional copies of the regressor, where the weight of each instance is adjusted according to the error of the current prediction. In this way, subsequent regressors focus on the difficult cases. The number of estimators was selected to be 100 and the learning rate was chosen to be 3. XGBoost stands for extreme Gradient Boosting and is an implementation of gradient boosting decision trees designed for speed and performance. The number of estimators was selected to be 800, the maximum depth of each tree 16 and the learning rate 0.01.

Finally, Long Short-Term Memory (LSTM) was implemented since it is considered to be state-of-the-art in predicting the stock market. LSTM is a deep learning algorithm and is extremely powerful for time series, since it is able to capture historical trend patterns and predict future values with high accuracy. The key component of LSTM is the cell, whose state is managed by three gates: the forget gate, which decides which information to throw away from the previous cell state, the input gate which chooses which new information gets added in the current cell state and the output gate which controls the output flowing to the next cell state.

In this project, we used four LSTM layers where each one has 256 units. Dropout layers were also added with drop rate being equal to 20%, while Adam optimizer was utilized with learning rate 0.0001. A validation set was also used (20% of the overall dataset) and the number of epochs were selected to be 10. Mean squared error is the loss function for optimizing the problem with Adam optimizer, whereas mean absolute error is the metric used in our LSTM network as it is associated with time-series data. The following diagrams depict how mean absolute error and loss changes across epochs.

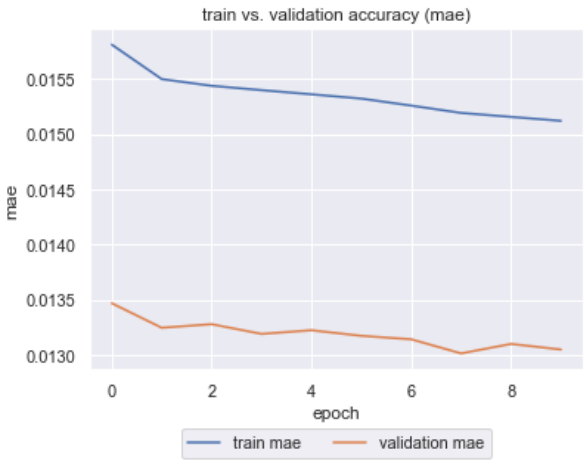

Figure 1 Mean Absolute Error

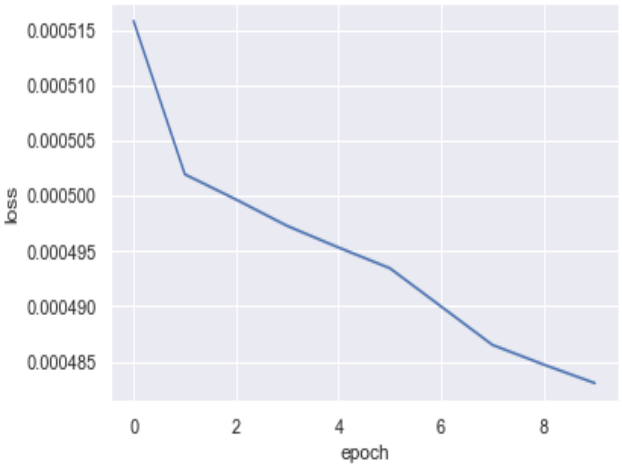

Figure 2 Loss

5 EVALUATION

There are various evaluation metrics available for measuring the performance of the applied methods. The most common metrics that are suitable for regression are Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Relative Root Mean Square Error (rRMSE) and Mean Squared Error (MSE). Moreover, when dealing with stock market prediction, maximizing the profit and minimizing the risk is the main goal. This is quantified by the Sharpe Ratio, which measures the return of an investment compared to its risk and is defined by the following formula:

$$Sharpe\_Ratio = \frac{R_p - R_f}{\sigma_p}$$

, where R_p is the return of the portfolio, R_f is the risk-free rate and σ_p is the standard deviation of the portfolio's excess return.

The evaluation in the competition will be based on the "score", which is the ratio between mean and standard deviation of the time series of daily spread return that is calculated every business day during a specific period.

$$Score = \frac{Average(R_{day_1-day_x})}{STD(R_{day_1-day_x})}$$

The daily spread return is the overall predicted return at a specific day if the proposed strategy is followed.

$$R_{day} = S_{top} - S_{down}$$

6 RESULTS

The initial experiments were run locally using an AMD Ryzen 7 5800H processor with 16GB RAM and the evaluation metrics for each algorithm, using a validation set, are presented in the following table. The last column represents the time (in seconds) needed to fit each algorithm using the training data and make predictions for the test data.

Algorithm	MAE	RMSE	MSE	Time (sec)
Linear Regression	0.016	0.026	0.0006	**2.922**
k-Nearest Neighbors	0.020	0.029	0.0008	1027
Support Vector Regressor	0.016	0.0263	0.0006	1277
Random Forest	0.017	0.0273	0.0007	713.9
AdaBoost	0.162	0.2201	0.0484	253.1
XGBoost	0.018	0.0277	0.0007	928.5
LSTM	**0.013**	**0.0232**	**0.0003**	31,115

Table 2 Evaluation Metrics

We observe the LSTM yields excellent results in terms of MAE, RMSE and MSE. However, it needs a lot of time for training compared to the other algorithms. All algorithms were submitted in the competition and the following figure depicts the scores that we received.

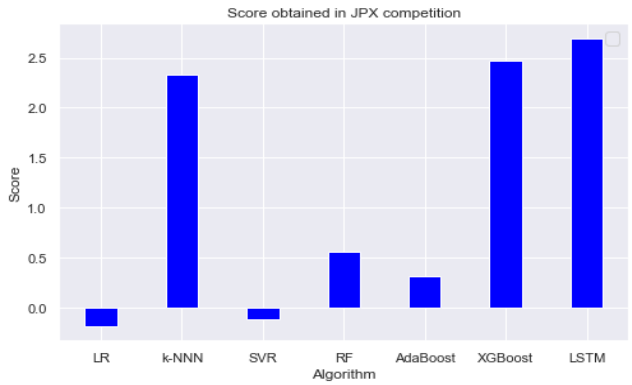

Figure 3 Score in Competition

As expected, the largest score was received for the LSTM model and as a result we were ranked 303rd out of 1476 teams. However, this is not the final ranking but the current one, which is calculated using the best score that has been achieved by the team. The range in the scores as observed in the corresponding "Leaderboard" section of the competition is between -6.025 and 7.814.

The final score in the context of the competition will take place a posteriori the final submission on 5th of July, using new financial data.

7 CONCLUSION

To sum up, we would say that stock market prediction is an interesting yet challenging task. In this project, we applied several machine learning techniques to rank 2,000 stocks in the Tokyo market based on the forecasted future returns. Long Short – Term Memory (LSTM) yields the most accurate predictions since deep neural network architecture is capable of capturing hidden dynamics. In the competition we managed to rank 303rd out of 1476 teams. However, it is hard to get accurate prediction results based only on the market data. Other data sources that would improve the results are "text data", which include social media, news, or web searches. These data could be analyzed, and a sentiment vector could be used to predict the direction of future stock prices. For sentiment analysis of text data, Transformers and pretrained BERT are widely used. Moreover, "macroeconomics data", which refers to the economic circumstances of a particular country and indicates how healthy the overall stock market is, could be also utilized.

REFERENCES

[1] Jiang, Weiwei. "Applications of deep learning in stock market prediction: recent progress." Expert Systems with Applications 184 (2021): 115537.

[2] Selvamuthu, Dharmaraja, Vineet Kumar, and Abhishek Mishra. "Indian stock market prediction using artificial neural networks on tick data." Financial Innovation 5.1 (2019): 1-12.

[3] Kumar, Deepak, Pradeepta Kumar Sarangi, and Rajit Verma. "A systematic review of stock market prediction using machine learning and statistical techniques." Materials Today: Proceedings (2021).

[4] Nti, Isaac Kofi, Adebayo Felix Adekoya, and Benjamin Asubam Weyori. "A comprehensive evaluation of ensemble learning for stock-market prediction." Journal of Big Data 7.1 (2020): 1-40.

[5] Shah, Dev, Haruna Isah, and Farhana Zulkernine. "Stock market analysis: A review and taxonomy of prediction techniques." International Journal of Financial Studies 7.2 (2019): 26.

[6] Nabipour, Mojtaba, et al. "Deep learning for stock market prediction." Entropy 22.8 (2020): 840.

[7] Ballings, Michel, et al. "Evaluating multiple classifiers for stock price direction prediction." Expert systems with Applications 42.20 (2015): 7046-7056.

[8] Sharpe, William F. "The Sharpe Ratio, the journal of Portfolio Management." Stanfold University, Fall (1994).