Market Trends Comparison of Models

Somesh Pushpkumar Bagadiya#1, Harshit Chouksey\*2, Arielle Shnayder#3

*#Engineering Department, San Jose State University University*

1Someshpushpkumar.Bagadiya@sjsu.edu

2Harshit.Chouksey@sjsu.edu

3Arielle.Shnayder@sjsu.edu

***Abstract*— We have utilized prominent machine-learning models for predicting stock market prices, random forest, Stacked long- and short-term memory (Stacked LSTM), and multilayer perceptron algorithms for this project. We explore these three models by doing a comparative analysis of the results. We will use a prominent finance dataset, i.e., the Yahoo finance dataset, to implement these models. This dataset is popular because it has many years of historical data and is complete. We will preprocess this data to be consumed by both of these models. The Random Forest model is famous for robustly handling diverse datasets. Stacked LSTM, a recurrent neural network, is known for proficiently capturing long-term dependencies in time-series datasets. We use Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) for the performance analysis. The study will illuminate the strengths and drawbacks of these models in stock market price predictions.**

1. INTRODUCTION

To understand and anticipate market trends, researchers and investors have explored advanced predictive models to decipher the intricacies of stock market dynamics. Our project aims to predict market trends using three machine-learning algorithms: Random Forest, Stacked Long- and Short-Term Memory (Stacked LSTM), and Multilayer Perceptron. Comparing these models in the context of stock market price predictions is the goal of this endeavor, which seeks to shed light on their respective strengths and shortcomings.

Our selection of machine-learning models represents a strategic decision to cover a range of methodologies. Due to its robustness and ability to adapt to the complexity inherent in financial data, the Random Forest algorithm is renowned for its versatility in handling diverse datasets. We incorporate stacked LSTMs, a variant of recurrent neural networks (RNNs) because they can capture long-term dependencies within time-series datasets. Multilayer Perceptrons are also introduced to provide insight into how they compare to the other models from a neural network perspective.

We conducted our study using the widely recognized Yahoo Finance dataset, a rich and comprehensive repository in the financial domain. The dataset consists of numerous years of historical data, which is crucial for training and evaluating the predictive models.

In order to assess the accuracy and reliability of predictive models, metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are used. Through this comparative analysis, we aim to highlight not only the predictive capabilities of each model but also their performance in the domain of stock market price predictions.

Essentially, this project explores how machine-learning algorithms can be applied to predict stock market trends, thereby contributing valuable insight to the ongoing debate about the efficacy of these models.

1. DATA SOURCES

Our choice was Yahoo Finance API, which contains a vast repository of over 100 million stocks dating back to 1962. For our data acquisition needs, this API provides essential data features such as open price, close price, daily high, daily low, adjusted price, and volume for each stock.

To extract this data, two methods are employed. First, we use the Yahoo Finance API directly to access real-time and historical stock information. As a result, the required financial data can be obtained efficiently and accurately.

Alternatively, you can download the data directly from Yahoo Finance's website in CSV format. It allows manual retrieval of data in a convenient format using a user-friendly approach.

1. ETHICAL CONSIDERATION
2. *Economic Impact*

If our models work well, we could impact the stock market globally. It can cause prices to swing either way. Also, this could cause a disadvantage to normal people who are trying to invest. If the model does predict prices correctly, people using the model will be at an advantage over everyone else, which is unfair.

1. *Environmental Impact*

All the computers required to run the models use a ton of electricity. This has a big impact on the environment. To reduce the environmental impact, maybe we should consider using green energy to run the computers.

1. LITERATURE STUDIES

There have been various studies utilizing machine learning models to predict stock market prices. A few of them are Linear regression, Time series models, Recurrent Neural networks, Support vector machines, Random forest, Gradient Boosting Models, Neural Networks, and Ensemble Methods. We are considered a few of the leading models for our paper.

1. *Random Forest*

Random forests algorithm belongs to ensemble learning methods. The main idea behind ensemble learning methods is that a single classifier is sometimes not sufficient for correct classification of test data and that multiple classifiers can increase model correctness [1]. The random forest model is the combination of bagging, random feature selection, and majority voting. It operates by generating decision trees. And is known to handle high dimensional data. It can also handle unprocessed data. Manojlovic etc. used the random forest model to predict the stock market and ETF price for 5-day period and 10-day period with 75% to 80% accuracy respectively [1].

Sipie Du etc also followed a similar approach but they tried and tested different sampling methods to evaluate random forest effectiveness on stock market prediction and their execution times [2]. On a similar line Ji Sang Park etc. utilized 30 features and 10 fold cross validation methodology to predict stock prices for stocks and EFTs [3]. Yilin Ma etc explored an interesting idea of making a hybrid model with Random Forest and LSTM [4].

1. *Stacked Long Short-Term Memory*

In recent literature up to 2022, researchers have explored the application of stacked LSTM networks for stock market prediction using deep learning approaches. Stacked LSTMs, with multiple layers capturing complex temporal dependencies, have been employed to model historical stock prices, trading volumes, and other relevant features. Some studies focused on ensemble techniques, combining multiple LSTM models to enhance predictive accuracy. Hyperparameter tuning, data preprocessing, and careful consideration of evaluation metrics such as Mean Squared Error and financial metrics like Sharpe ratio were crucial aspects. Challenges included addressing noise, non-linearity in financial time series data, overfitting, and generalization to diverse market conditions. As the field evolves, newer publications may provide additional insights into this dynamic intersection of machine learning and finance. [5]

1. *Multi\_Layer Perceptron*

Multi-layer perceptrons (MLP) are the basic neural networks that are used in Machine learning. It consists of one input layer, one or more hidden layers, and one output layer. The layers are connected to each other in a fully connected network and these connections have weights assigned to them. While training the multi-layer perceptrons model utilizes backpropagation and adjusts weights to make the predicted output close to the actual output. It is proficient in identifying complex relations between the data.

Ecer etc. compared multi-layered perceptron models with hybrid models and found that multi-layered perceptrons had faster execution times but lower accuracy as compared to a few Hybrid models [6]. Pires etc. performed comparison of multi layer perceptron and support vector machines and concluded the multi layer perceptron model to be less proficient [7].

1. MODELS
   1. *Random Forest*

Random forest is an ensemble learning technique which uses decision trees. During the training cycle it generates multiple decision trees, the model predicts the outputs using the aggregation of decision trees constructed during training, The term random is justified at 2 levels first is during the bootstrapped sampling which selects subsets of the data at random. The other is selection of random features during evaluation of each node of the tree.

These selections help in mitigating the overfitting problem which is an issue for various Machine learning models. And helps in capturing complex relationships and makes the model robust. Random forest can also handle high dimensional data with ease and is versatile.

1. Data Preprocessing
   1. Data has been fetched for the last 5 years for any respective company using Yahoo API.
   2. The relevant fields considered for processing are the closing prices of the socks for a particular day. Additionally the date, month and year are considered.
   3. Data is split into 70% and 30% for training and testing.
2. Model
   1. For a few sample stocks are chosen for Hyperparameter tuning
   2. Using the random forest regressor the prediction model is trained for each combination of parameters mentioned in Table 1. A total of 72 combinations were tested.
   3. Hyperparameters with the best performance are chosen across the sample stocks, MSE is considered as a deciding factor.
   4. The model is run across multiple stocks using the optimum hyperparameters.
3. Testing of Model
4. the Mean Squared error is calculated showing the overall performance
5. The actual and predicted price graph is also created for the test set to evaluate the performance visually.

| **Hyper Parameter** | **Values Considered** |
| --- | --- |
| n\_estimators\_values | [100, 200] |
| max\_depth\_values | [None, 10, 20] |
| min\_samples\_split\_values | [2, 5] |
| min\_samples\_leaf\_values | [1, 2] |
| max\_features\_values | [ 'sqrt', 0.5,None] |

Table 1: Hyperparameter exploration options

* 1. *Stacked Long Short-Term Memory*

Long Short-Term Memory (LSTM) and Stacked Long- and Short-Term Memory (Stacked LSTM) are recurrent neural networks designed for sequential data analysis. LSTM features a single layer with memory cells and gates, effectively capturing short-term dependencies. In contrast, Stacked LSTM employs multiple layers, allowing for the hierarchical learning of complex features and improved performance on tasks involving long-range dependencies. The choice between the two depends on the depth of temporal understanding required and the available computational resources for training.

Stacked Long- and Short-Term Memory (Stacked LSTM) networks excel in market prediction because they can capture intricate temporal dependencies. With multiple layers, they automatically learn hierarchical representations, making them adept at discerning short-term fluctuations and long-term trends in financial data. The memory cells and gating mechanisms enhance adaptability to complex market patterns, while the sequential processing nature of LSTMs aligns well with the chronological structure of financial time series.

1. Data Preprocessing:
   1. Considering the stock's closing price to train the model.
   2. Split data into 70% train, 30% test
   3. Scaling the data using a min-max scalar in the range of 0-1.
2. Model:
   1. Using three stacks of LSTM.
   2. Using extended SGD for the gradient.
   3. Training the model over 100 epochs and the batch size of 32.
3. Testing on model
   1. Evaluate test loss “mean squared error.”
   2. *Multi-Layer Perceptron*

Multilayer perceptron is a class of feedforward artificial neural network which consists of at least three layers. Each neuron is connected with certain weights and uses an activation function like Tanh or RELU for example. This algorithm also uses backpropagation for training the network by adjusting the weights to minimize the error.

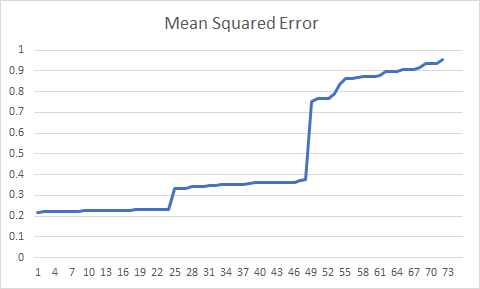
1. Data Preprocessing:
   1. We are using the following features: Open price, days high, days low, adjusted close, and volume. Our target feature is the close price.
   2. Split data into 80% train, 20% test
   3. Scale the train data

2. Model

* 1. Use 3 layers: the first layer has 50 neurons, second layer has 30 neurons, and the last layer has one neuron using the Relu activation function.
  2. Using 100 epochs, batch size of 32, and validation split of 0.1

1. Testing on model
   1. Evaluate test loss “mean squared error.”
2. RESULTS
   1. RANDOM FOREST

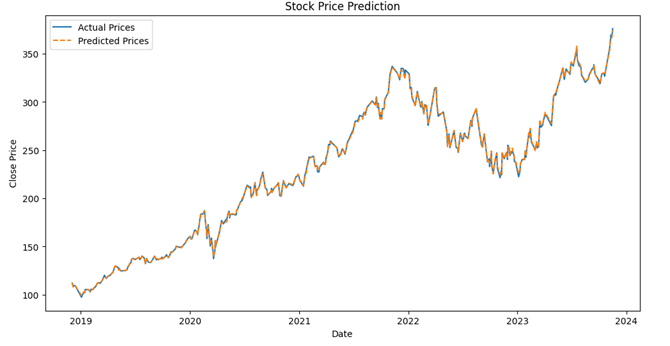
While hyperparameter tuning. For the stock market price prediction it was found that the performance has drastic change by the parameter max\_features. It is shown in the graph 1 below, the y axis shows MSE and x axis shows the permutation number.



Graph 1: Hyperparameter Tuning

After testing for a few stocks the optimum parameters were decided as n\_estimators=200, max\_depth=10, min\_samples\_split=2, min\_samples\_leaf=1, max\_features=None.

The model was run on multiple stocks, the graphical representation of the predicted and actual size for microsoft is shown below in graph 2.



Graph 2: Performance of random Forest Model

The model was run on some major companies, the results and execution times are shown in table 2:

| **Companies** | **MSE** | **Training Time (s)** | **Prediction Time (s)** |
| --- | --- | --- | --- |
| Microsoft | 1.425 | 1.592 | 0.0205 |
| Apple | 0.219 | 1.048 | 0.0203 |
| Tesla | 0.784 | 1.384 | 0.047 |
| Google | 0.024 | 1.706 | 0.0319 |
| JP Morgan | 0.855 | 1.609 | 0.0316 |
| Johnson and Johnson | 0.782 | 1.447 | 0.044 |
| Alibaba | 0.322 | 0.875 | 0.0218 |
| Visa | 0.642 | 1.676 | 0.0588 |
| Amazon | 0.019 | 1.662 | 0.0698 |
| Pfizer | 0.096 | 1.389 | 0.0248 |

Table 2: Performance of tuned model for major companies

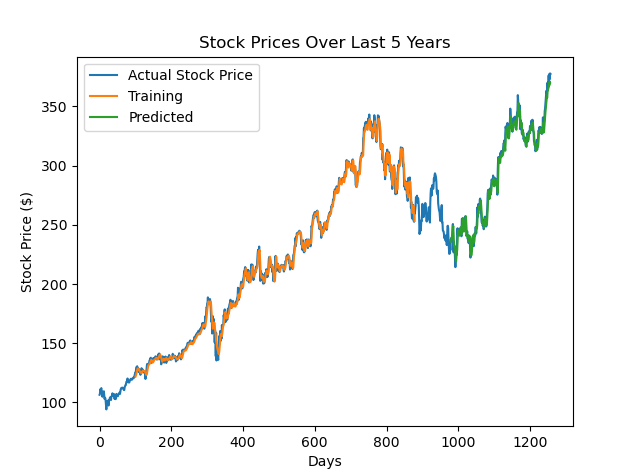
A specific test was also performed to predict the movement of the stocks based on the trained model. The tests were not exhaustive but from the test stocks and dates taken the random forest model also showed promising effects in predicting the general movement of the stock for the next day. As depicted in the below image.



Image 1: Test for general movement of the stock using random forest.

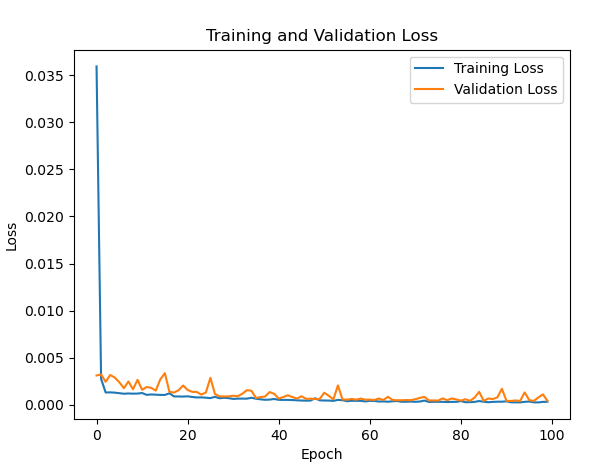
* 1. LSTM

The results are shown in the plot image below. The blue is the actual stock prices over a period of 1250 days, the orange line is the training phase of the model, and the green line is the predictions. The model captures the trend quite well, and the train and test root mean squared error are 228.05 and 295.44, respectively



Graph 3: Model Performance LSTM

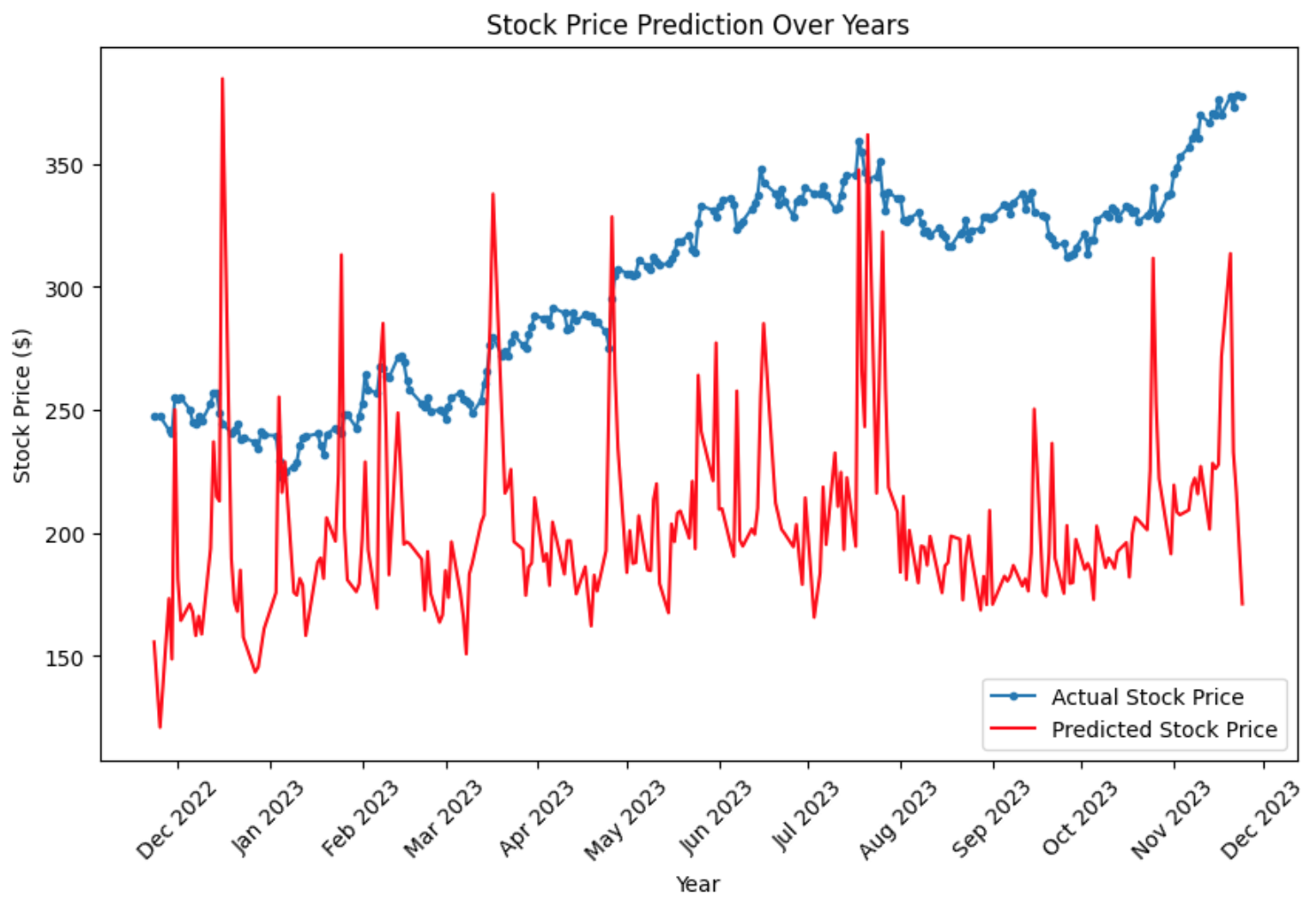
The results shown in the plot image below are a plot of training loss vs the validation loss of the model. The model took around 271 secs to train, and the predicting took around 4 secs.



Graph 4 Training loss vs Validation Loss

* 1. Multi-Layer-Perceptron

The results are shown in the plot image below. The blue is the actual stock prices over the months and years while the red shows the predicted stock price. The variation between these two is significant because the stock market has no underlying pattern and is very volatile. The predicting time was around 0.2242 seconds while training time was around 21.24 seconds. The mean square error was around 11771.37.



Graph 5: Model performance MLP

1. Conclusions

By compiling the results from all the 3 models we found that the multilayer perceptron had the worst performance in accuracy and moderate execution time. In a comparative sense this was not a good model for the financial predictions.

The Stacked LSTM model had the best accuracy, but it had considerable training and execution times.

The random forest model had moderate accuracy, and very good execution and training time. This is depicted in table 3.

| **Models** | **MSE** | **Training Time (s)** | **Prediction Time (s)** |
| --- | --- | --- | --- |
| Stacked LSTM | 0.000485 | 410.25 | 4.86 |
| MultiLayer Perceptron | 5.3 - 10.6 | 11.63 - 21.16 | 0.118 - 0.162 |
| Random Forest | 0.5168 | 1.43 | 0.037 |

Table 3: Aggregated results

Since there are multiple kinds of trading strategies and scenarios, a common machine learning model cannot be recommended to fit all the trading strategies. Accuracy is equally important for all scenarios but there are some wherequick decision making is equally important.

Therefore, for the trading strategies where we have enough time to make the decisions, the Stacked LSTM model can be utilized; such strategies are Swing Trading and Position trading.

For other trading strategies which are very time sensitive, the Random forest model can be utilized; such strategies are Day Trading, Scalping and Algorithmic Trading.

| **Trading Type** | **Description** | **Preferred Model** |
| --- | --- | --- |
| Day Trading | Quick buying and selling within the same trading day. | Random Forest |
| Swing Trading | Positions held for several days to weeks to capture trends. | Stacked LSTM |
| Scalping | Making numerous small trades to exploit short-term inefficiencies. | Random Forest |
| Position Trading | Long-term trading with positions held for months or years. | Stacked LSTM |
| Algorithmic Trading | Automated trading based on predefined criteria and algorithms. | Random Forest |

Table 4: Recommended Models

There is always room for improvement when it comes to financial market price predictions. It can be on the accuracy or the execution times. For this hybrid models can be explored which can take the better parts of multiple models and provide an efficient solution.

References

1. T. Manojlović and I. Štajduhar, "Predicting stock market trends using random forests: A sample of the Zagreb stock exchange," 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2015, pp. 1189-1193, doi: 10.1109/MIPRO.2015.7160456.
2. S. Du, D. Hao and X. Li, "Research on stock forecasting based on random forest," 2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA), Dalian, China, 2022, pp. 301-305, doi: 10.1109/ICDSCA56264.2022.9987903.
3. J. S. Park, H. Sung Cho, J. Sung Lee, K. I. Chung, J. M. Kim and D. J. Kim, "Forecasting Daily Stock Trends Using Random Forest Optimization," 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2019, pp. 1152-1155, doi: 10.1109/ICTC46691.2019.8939729.
4. Y. Ma, R. Han and X. Fu, "Stock prediction based on random forest and LSTM neural network," 2019 19th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea (South), 2019, pp. 126-130, doi: 10.23919/ICCAS47443.2019.8971687.
5. H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. C. Khatri, “Predicting stock market index using LSTM,” *Machine Learning with Applications*, p. 100320, May 2022, doi: https://doi.org/10.1016/j.mlwa.2022.100320.
6. Ecer, F. et al. (2020) Training Multilayer Perceptron with Genetic Algorithms and Particle Swarm Optimization for Modeling Stock Price Index Prediction. *Entropy*. [Online] 22 (11), .
7. M. M. Pires and T. Marwala, "American option pricing using multi-layer perceptron and support vector machine," 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), The Hague, Netherlands, 2004, pp. 1279-1285 vol.2, doi: 10.1109/ICSMC.2004.1399801.