Continue building the stock price prediction model by

*Feature engineering

* Model training

*Evaluation.

Abstract

The stock market is a very important activity in the finance business. Its demand is consistently growing. Stock market prediction is the process of determining the future value of company stock or other financial instruments traded on a financial exchange. For some decades Artificial Neural Network (ANN), which is one intelligent data mining technique has been used for Stock Price Prediction. It has been trusted as the most accurate consideration. This paper surveys different machine learning models for stock price prediction. We have trained the available stock data of American Airlines for this project. The programming language that we have used in this paper is Python. The Machine Learning (ML) models used in this project are Decision Tree (DT), Support Vector Regression (SVR), Random Forest (RF), and ANN. The data here is split into 70% for training and 30% for testing. The dataset contains stock data for the last 5 years. From the simulation results, it is shown that Random Forest performs better as compared to others. Thus, it can be used in the real-time implementation. Abstract The stock market is a very important activity in the finance business. Its demand is consistently growing. Stock market prediction is the process of determining the future value of company stock or other financial instruments traded on a financial exchange. For some decades Artificial Neural Network (ANN), which is one intelligent data mining technique has been used for Stock Price Prediction. It has been trusted as the most

accurate consideration. This paper surveys different machine learning models for stock price prediction. We have trained the available stock data of American Airlines for this project. The programming language that we have used in this paper is Python. The Machine Learning (ML) models used in this project are Decision Tree (DT), Support Vector Regression (SVR), Random Forest (RF), and ANN. The data here is split into 70% for training and 30% for testing. The dataset contains stock data for the last 5 years. From the simulation results, it is shown that Random Forest performs better as compared to others. Thus, it can be used in the real-time implementation. Abstract The stock market is a very important activity in the finance business. Its demand is consistently growing. Stock market prediction is the process of determining the future value of company stock or other financial instruments traded on a financial exchange. For some decades Artificial Neural Network (ANN), which is one intelligent data mining technique has been used for Stock Price Prediction. It has been trusted as the most accurate consideration. This paper surveys different machine learning models for stock price prediction. We have trained the available stock data of American Airlines for this project. The programming language that we have used in this paper is Python. The Machine Learning (ML) models used in this project are Decision Tree (DT), Support Vector Regression (SVR), Random Forest (RF), and ANN. The data here is split into 70% for training and 30% for testing. The dataset contains stock data for the last 5 years. From the simulation results, it is shown that Random Forest performs better as compared to others. Thus, it can be used in the real-time implementation.

Introduction :

1. The Stock Market is the accumulation of stockbrokers, traders, and investors who sell buy or share trades. There are so many companies that provide their stock list on market, these make their stocks attractive to investors [1]. Because ever since the 16s investors are trying different techniques to get knowledge about different companies to improve their investment returns [2]. It plays a very important role in increasing a developing country’s economic status like India [3]. The demand for Stock Market is growing significantly. We all know that it has been in focus for many years because of the outstanding profits [4]. Lots of

wealth are traded daily through the stock market and so it is seen as one of the most profitable financial outlets [5]. Now, the stock market is one of the factors which shows a country's economy [6]. Many people invest a handsome amount of money in the share market but sometimes they tend to incur very huge losses because they depend upon the stockbrokers, who advise investors based on fundamental, technical, and time series [7]. Investors have been trying to find an intelligent idea to overcome such problems. This is where Stock Price Prediction comes into action because predicting stock prices is very necessary [2].

2. Stock Price Prediction's main idea is to accurately predict the future financial outcome [5]. In the past few years, Machine Learning algorithms are seen to give promising results in various industries, so many traders are applying these techniques to their respective fields [8]. ML can be applied as a game-changer [9]. In this paper, some experimentation is done by taking different ML algorithms to predict the opening price of American Airlines stocks. The Machine learning (ML) algorithms that we have used are Random Forest (RF), Decision Tree (DT), Support Vector Regressor (SVR), and Artificial Neural Network [6]. Prediction of Stocks is based on the opening price of the day for this paper. The remaining paper has been laid out in the following order. In section -2 literature survey has been reviewed followed by section -3 where various approaches or different machine learning algorithms used have been discussed. In section-4 the problems that occurred or that needed to be improved previously have been addressed. Section-5 represents all the information about the dataset. In section-6 the results and future works have been discussed and in section-7 the paper has been concluded.

2. Literature Survey:

1. Since the introduction of the Stock Market so many predictors are constantly trying to predict stock values using different Machine Learning algorithms such as Support Vector Regressor (SVR), Linear Regression (LR), Support Vector Machine (SVM), Neural Networks Genetic Algorithms, and many more [5] on stocks of various companies. There is a diversity in many papers based on different parameters. Many

different ML algorithms are used by different authors based on different parameters. Some authors believe that Neural Networks have given better performance as compared to other approaches [5]. Like, in paper [12] Hiransha M and GopalKrishnan E. A has trained four models Multi-Layer Perceptron (MLP), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) and it was observed that CNN has performed better than the other three networks. On the other hand, many authors believe that Support Vector Regression which is known to solve regression and prediction problems gives better performance as seen in paper [13] by Haiqin Yang, Laiwan Chan, and Irwin King. In paper [5] Paul d. Yoo has trained 3 models Support Vector Machine, CaseBased Reasoning classifier (CBR), and Neural Networks (NN) from which Neural has given the most appropriate prediction. Sumeet et al [18] has done an approach where they have combined two distinct fields for stock exchange analysis. It merges price prediction based on real time data as well as historical data with news analysis. In this paper LSTM(Long Short-Term Memory) is used for prediction. The datasets are collected from large sets of business news in which relevant and live data information is present. Then the results of both analyses are combined to form a response which helps visualize recommendation for future increases. So, in many papers, it has been seen that neural networks give the expected prediction value.

3. Approaches :

In this project, prediction is carried out by using these ML algorithms. These are Decision Tree, Support Vector Regression, Random Forest, and Artificial Neural Network.

3.1. Decision Tree Methodology:

It is a supervised ML, which is used for both regressions as well a classification. That is how it is also called CART Classification and Regression Trees. In this algorithm, two nodes are present namely Decision Node which is for making the decisions and can be divided into multiple branches and Leaf Node which gives the output of decisions and this node can't be further divided into many nodes. The following is the formula for Leaf Node: $Information\ Gain = Class\ Entropy - Entropy\ Attribute$ (1) Branches-Here decision rules are set by which nodes can be divided further. For Prediction, it starts from the root node, compares values of the real attribute with the root attribute, and based on that comparison it follows the branch and jumps to the next node. This process continues until it reaches the leaf node of the tree. Entropy-It is a metric that helps in measuring error in a given attribute. The formula to find entropy is: - Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no) (2) Here, (S) implies the Total number of samples. P (yes) refers to the Probability of S and P (no) means the Probability of no.

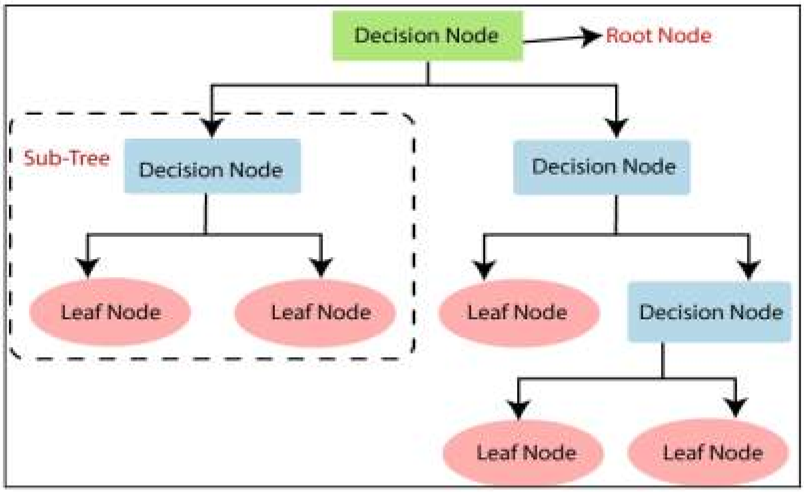
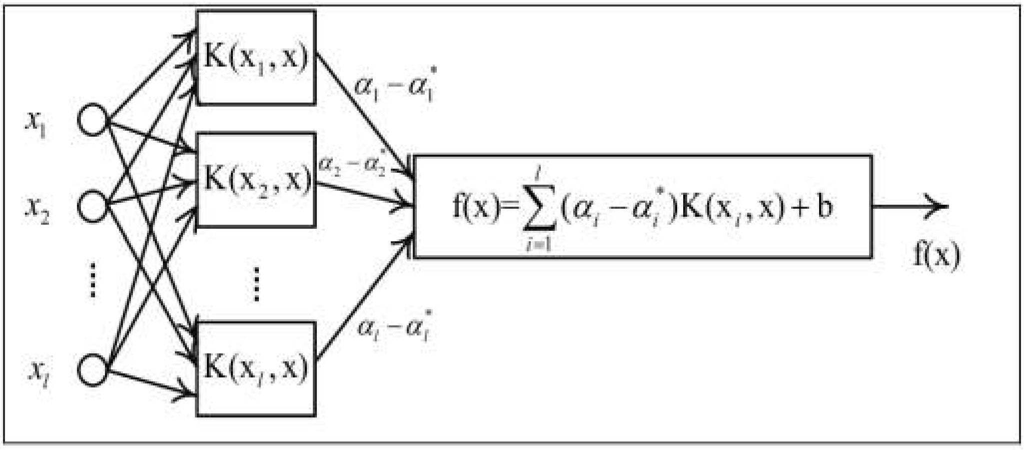


Figure 1: Decision Tree Classifier Process

3.2 Support Vector Regression Methodology :

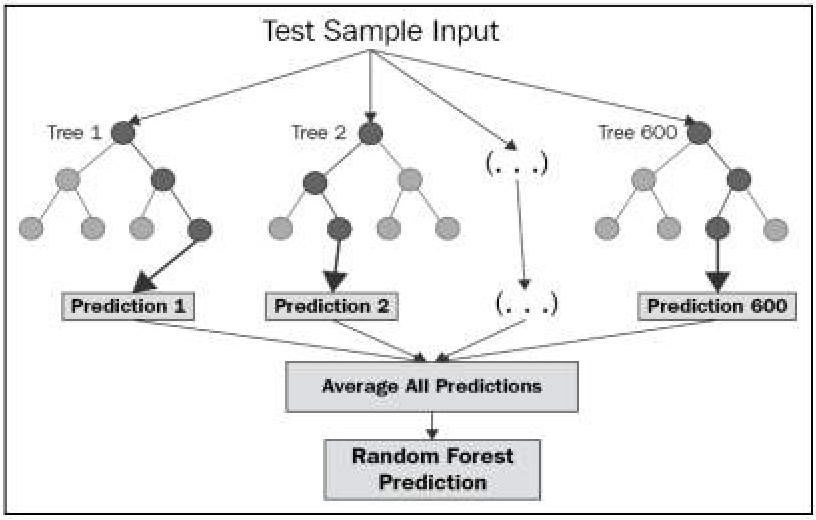
It is a Supervised Machine learning algorithm used for regression analysis. It finds the function that helps us approximate mapping based on the training sample from an input domain to real numbers. The Terminologies contained in this are Hyperplane -this is the line that is used to predict the continuous output. Kernel helps to find hyperplanes in higher dimensional space without increasing the computational cost of it and the decision boundary is a simplification line that differentiates positive examples and negative examples.



Support Vector Regression

3.3 Random Forest Methodology

Random forest is a supervised Machine Learning algorithm that is used for Regression analysis. This overcame the problem of overfitting as seen in the decision Tree [12]. It is an ensemble learning method. The steps for prediction are first a random k data point is picked from the training set then accordingly the decision tree is built. Then choose the number of trees we want to build and again follow the previous steps. From every new data point, make N tree Trees predict the value of Y for data points and assign new data points across all of y predicted Y values.



Random Forest Procedure

3.4 Artificial Neural Network Methodology:

An artificial Neural network is an interconnection of nodes that is like the biological neuron in our body but not similar. For the last few decades, ANN has been used for Stock Price Prediction [12]. It contains three layers, first is the Input Layer – this layer takes different inputs variable from the user then, the hidden layer-This layer is present between the input layer which identifies all hidden features and patterns and the last layer is the Output layer- This layer provides the final output. ANN takes different inputs and multiplies them with the specified weights for each with an activation function for the activation of neurons.

The formula of the transfer function is:

$$\sum Wi * Xi + b \; n \; i{=}1 \qquad (3)$$

Here, b is the threshold value. Xi is input and value and Wi is the weight.

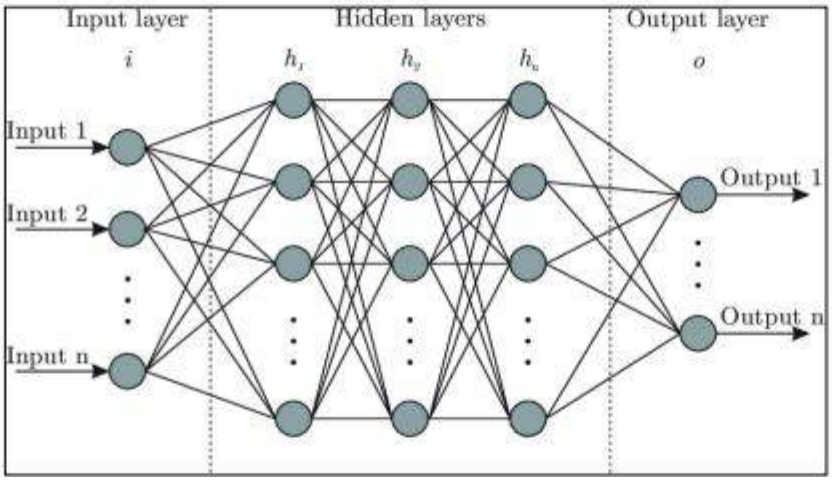


Figure 4. Artificial Neural Network Procedure

3.Problem Statement :

Now, stockbrokers who execute trading mainly depend on their experience, price trends, or fundamental analysis i.e. - buy or hold to select stocks. These methods may lead to great losses to investors if they make any wrong decisions because these are personalized and short-sighted due to their limited capacity. Lack of prominent results may lead to reluctance to participate in trading by investors. So, to overcome these drawbacks it is important to have a tool that can guide us on proper trading methods and consequences. Technical and fundamental analysis are the basis of future stock market Prediction. Here, Machine Learning methods come into action. These methods can help us analyze stock prices over time and create ideas about them and then help us in prediction and can be used to model a tool.

5. Stock Market Prediction Architecture :

Stock market data of American airlines from 2-08-2013 to 2-07-2018 has been used as a dataset in this project. This dataset has 1258 rows and 7 columns. Each row represents the information for a single day. For columns, the following are the feature description. 5.1 Data Preprocessing It includes searching for essential missing or null values and replacing them with mean values Searched for categorical value and if there is any unnecessary data then those values are dropped. 5.2 Data Splitting The processed data has been divided into 70% training data and 30% testing data using the train_test_split method. Here 881 data is taken as training data and the rest 377 is kept for testing. The training data values are taken from the date 2013-02-08 to 2016-08-09 and the testing data are from 2016-08-10 to 2018-02-06.

Table 1:
Dataset Feature Description Table

Sl. No	Feature	Description
1.	Date	It shows the date in the format: yy-mm-dd.
2.	Open	It shows the price of the stock at market opening.
3.	High	It shows the highest price reached on that day.
4.	Low	It shows the lowest price reached on that day.
5.	Close	It shows the lowest price reached on that day.
6.	Volume	It shows the number of shares traded on that day.
7.	Name	This is the name of the stock's ticker.

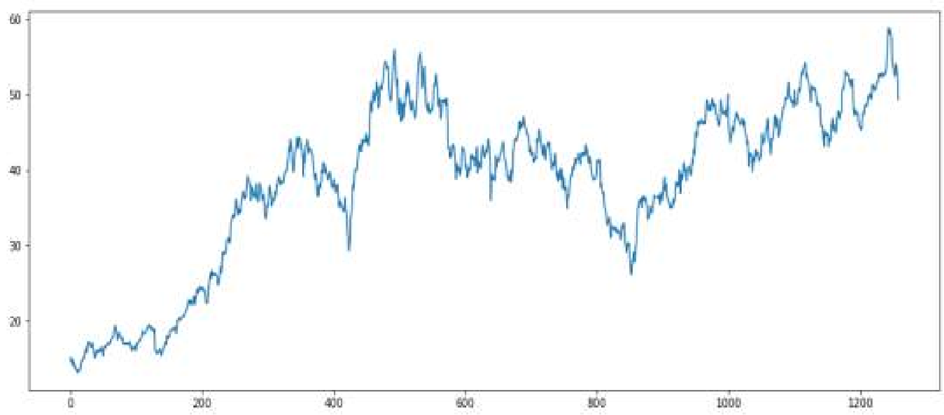


Figure 5: Opening Price Graph

5.3 Data Scaling

Standardization and Normalization are done on the data using Minmax Scaler and Standard Scaler to limit the ranges of variables to make them comparable on common grounds using ML methods.

5.4 Feature Selection

The selection of features is a very important task to predict future values. If we consider the worst features then the prediction can go wrong. In this paper, the attribute or feature used for feature extraction is the opening price or the ‘open’ column of American Airlines stocks. A data structure has been created with 7 timesteps and 1 output.

5.5 Prediction

We have adapted Machine Learning Approaches to find the prediction. In this case, training the model is very necessary. Random Forest, Decision Tree, and Support Vector Regression models have been used to do the prediction work.

5.6 Error Calculation

There are 4 types of error calculations present for evaluation. In this paper, we have used the MAPE method to find the error. Performance evaluation is done using MAPE values of all the models. Following are the formulae to find the MAPE

(Mean Absolute Percentage Error), MAE (Mean Absolute Error), rRMSE (Root Mean Squared Error), and MSE (Mean Squared Error) value

$MAPE = 1\ n \sum (\ |Ai - Pi \ | \ |Ai \ | \ ) \ n \ i=1 \times 100$ (4)

$minus - endMAE = 1\ n \sum (\ |Ai - Pi \ | \ |Ai \ | \ )$ (5)

$n\ i=1\ rRMSE = sqrt \ (\ 1\ n \sum (\ Ai - Pi\ Ai \ ) \ 2\ n\ i=1 \ )$ (6)

$MSE = 1/n \sum (\ i - Pi\ Ai \ ) \ n\ i=1 {}^{2})$ (7)

Here, n is the sample size, Ai is the predicted value and Pi is the Predicted value.

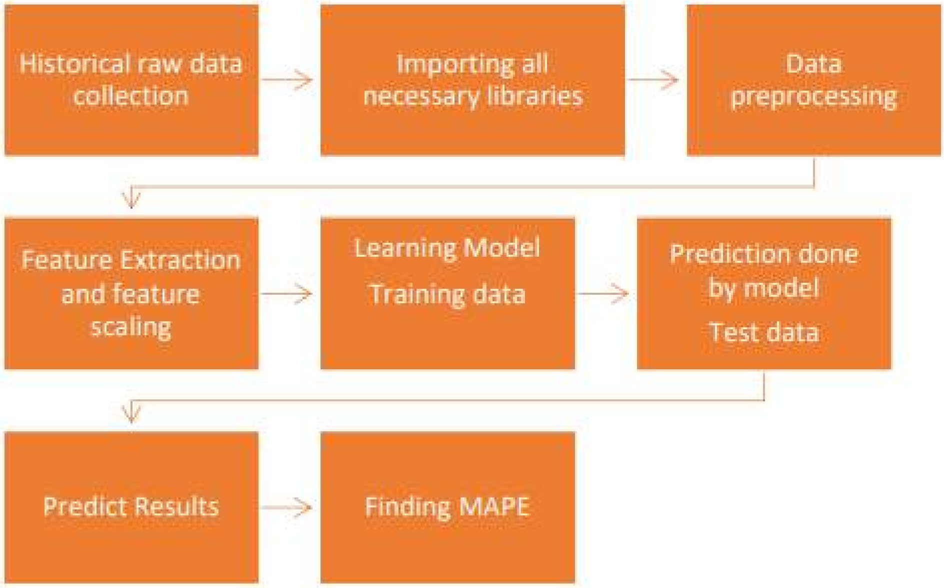


Figure 6: Architecture of Methodology

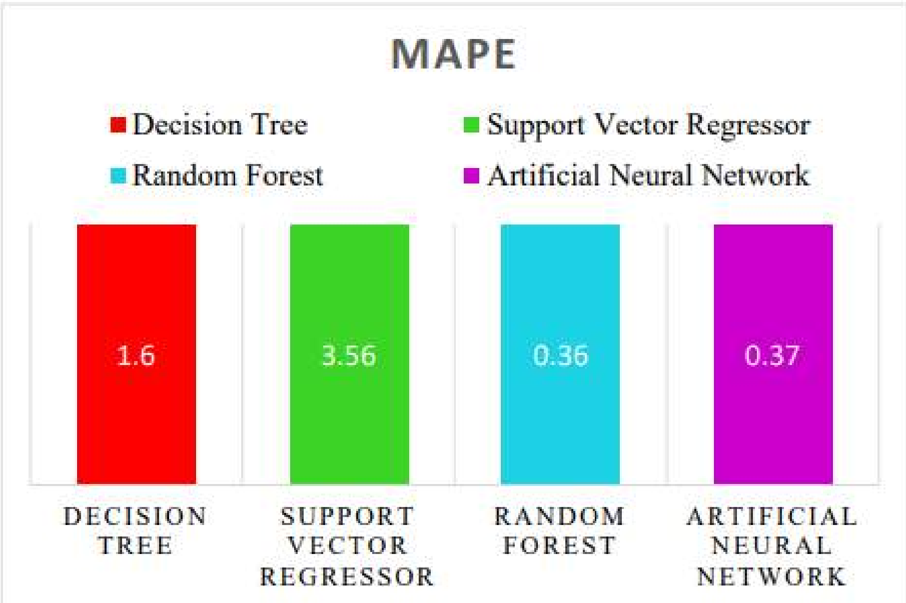
As shown in the figure above all the historical data were collected first and followed by the importation of all necessary libraries such as NumPy, Pandas, matplotlib, Seaborn, mean_squared_error, etc. In the next step, various data processing methods have been performed such as drop, isnull, etc. Then feature extraction and feature scaling techniques have been implemented using Min Max Scaler and sc.fit_transform. In the next step we have trained the data and learned the model required. In the next step various machine learning model which we have learned have been applied such as Decision Tree, Support Vector, Artificial Neural Networks, and Random Forest. Then we have got the prediction results. Out of all the 4 algorithms, Random Forest has the lowest MAPE value i.e.- 0.36

6.Results and Discussion :

The main objective of this project is to examine several different prediction techniques to predict future stock prices based on past returns. And here it is visible that Random Forest (6) Historical raw data collection Importing all necessary libraries Data preprocessing Feature Extraction and feature scaling Learning Model Training data Prediction done by model Test data Predict Results Finding MAPE (5) (7) is the best algorithm for this research giving a MAPE value of 0.36. This algorithm shall be used to predict opening prices shortly. The following is the table to show the MAPE values using Machine Learning Algorithms.

Table 2
MAPE Chart

S.No	Model	MAPE
01	Decision Tree	1.60
02	Support Vector Regression	3.56
03	Random Forest	0.36
04	Artificial Neural Network	0.37

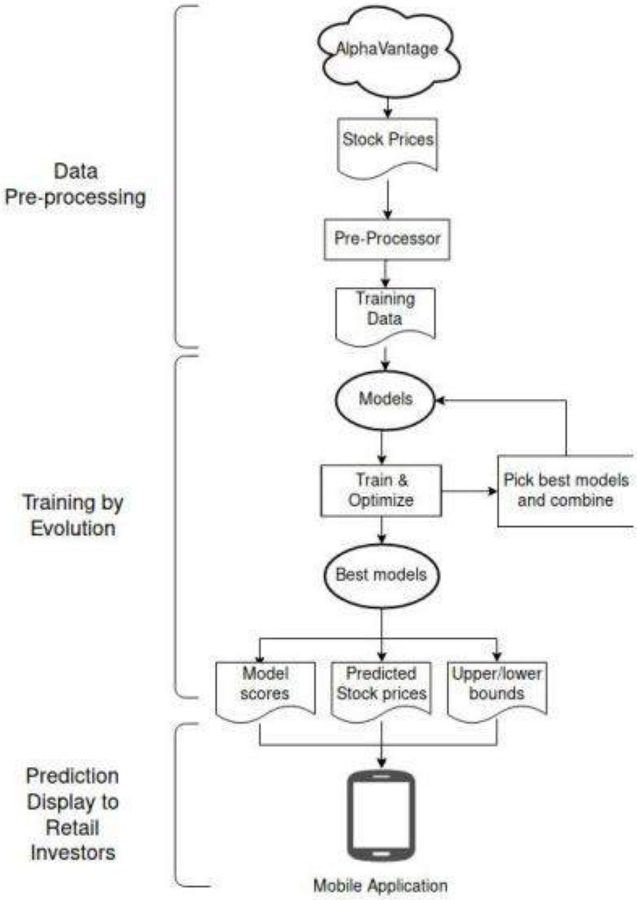


MAPE Comparison

8. Methodology:

Design 2.1 System

Architecture The architecture of the system follows a client-server model, where the server and the client are loosely coupled.



After relevant stock data are retrieved from the third-party data provider through the cloud, the backend pre-processes the data and builds the models. After that, predictions are made and the prediction results will be stored on another cloud, which can be retrieved from the mobile application.

The advantages of the loosely coupled architecture include improved scalability and ease of collaboration. The workload for the cloud which serves the models and the one which serves the mobile application will be very different. One cloud serves the model prediction results, which are simple text files; another cloud serves the mobile application with a lot of rich user content such as images and large UI libraries.

Having two clouds to adapt to two different demand patterns is more efficient, especially since cloud providers these days usually serve content on demand.

Also, the separation allows different team members in the team to focus on different parts after agreeing on a common interface. It speeds up development as team members responsible for different parts of the system do not need to take care of the underlying implementation details. Also, it is easier to swap out different components, e.g. to replace the models the team could simply make changes to the backend, while the frontend remains unaffected.

8.2 Research Design

8.8.1 Problem Framing

The problem of the project is set to predict the stock price for the next 10 business days. “10 days” is chosen as the timeframe as short term price movements tend to depend more on trend momentum and price pattern, while long term price movements depend on the fundamentals of a stock (e.g. company management capabilities, revenue model, market demand, macroeconomic factors, etc.).

The loss function of the training algorithm is the mean squared error of the 10 predicted stock prices. The training algorithm or optimizer is set to minimize its value, and it serves as the basic performance metric for comparing different models.

Other scores are defined to provide more in-depth insights on a model predictability performance and finance-domain-based comparisons between models for investors.

Two different prediction approaches are mainly tested, predicting the stock prices for the next 10 days directly and predicting the stock price of the next day 1 at a time. It is suspected that the two different problem framing approaches will result in different abstractions learned hence performance for different use-cases.

As different stocks have very different characteristics and the stock prices exhibit different trends, individual models will be built for separate stocks.

For the project, S&P 500 stocks from different industries are selected. Multiple factors are considered when picking the stocks, including stock price volatility, the absolute magnitude of the price, the respective industries, company size, etc., and stocks exhibiting different characteristics are picked. The stocks are listed as below:

- Alphabet Inc., GOOGL (Technology)
- Amazon.com Inc., AMZN (Technology)
- Apple Inc., APPL (Technology)
- AT&T Inc., T (Telecom Services)
- Boeing Co., BA (Industrials)
- Caterpillar Inc., CAT (Industrials)
- Facebook Inc., FB (Technology)
- General Electric Co, GE (Industrials)
- Harley-Davidson, Inc., HOG (Consumer Cyclical)
- Microsoft Inc., MSFT (Technology)
- Procter & Gamble Co, PG (Consumer Defensive)
- Tesla Inc., TSLA (Consumer Durables)
- Walmart Inc., WMT (Consumer Defensive)

8.8.2 Robust Design

For the research side, the system is designed to be as robust as possible to facilitate model testing. Each model can be defined by a pair of model options and input options, specifying the model configurations and the inputs it takes. This accelerates the process of testing out different model and/or input configuration combinations.

8.8.3 Data Pre-processing

Raw stock price data is pre-processed before inputting into machine learning models. Pre-processing includes transforming the raw data into a format that models can take from and operate on, most likely feature matrix. It also attempts to extract some features, financial-domain-specific especially, manually to improve results, allowing the model to learn more abstractions. Two key features are selected as the input. First is a fixed-length list of some raw historical data like stock price and daily percentage change. The fixed length chosen specifies the length of the historical period to look back from today when predicting future stock prices. Referring to the principle of technical analysis, as the stock price reflects all relevant information, a technical analyst would focus on the trading pattern of the stock rather than the economic fundamentals and company fundamentals. Therefore, by getting a period of historical stock prices as the input for the training model, it could be a piece of useful information in finding the trading patterns and hence predicting the trend of future stock prices. Given a set lookback period, it is assumed that the price movement patterns that are predictive would occur in the specified historical period.

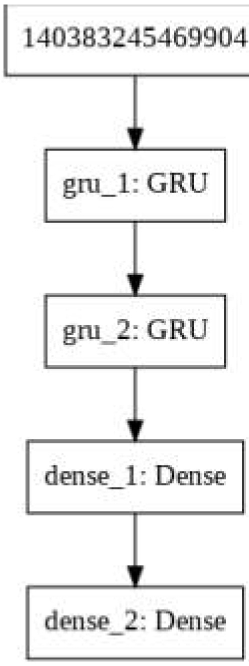
The second feature input is arithmetic moving averages. As mentioned in 1.1, one of the obvious approaches for retail investors to identify the trend of the market is through moving averages. With the robust system design, different period of moving averages could be used as the input into the model for stock price prediction, for example, a set of 22, 50, 100, 200 days moving averages, which are commonly used by investors [13].

8.8.4 Prediction

Output As mentioned in 8.8.1, 2 different prediction approaches are tested, which will have different outputs. For 10-day predictions, there will be 10 output units, resulting in a one-dimensional vector with 10 stock prices, where the i-th element represents the i-th day stock price prediction. For 1-day prediction, there will be 1 output unit which is the stock price in the following day. The predicted stock price of will then be the input of the next prediction, to predict the stock price in the second day, the process repeats until all 10 predictions are generated

8.8.5 Model Different

common neural network models are tested, including dense neural network, simple recurrent neural networks (RNNs), long short-term memory networks (LSTMs) and gated recurrent unit networks (GRUs);Different model architectures are tested by changing the number of hidden layers, the number of hidden units per hidden layer, and the activation function or recurrent activation function used in each hidden layer. All recurrent neural networks, RNNs, LSTMs, and GRUs, are set to have the same high-level architecture , a stack of recurrent layers by passing the full output sequence to the next layer, followed by a stack of dense layers.



Example of the common high-level architecture of recurrent neural networks

Linear Regression on features, as well as trendlines which interpolate the stock prices next 10 days linearly, are also tested.

8.8.6 Model Architecture and Hyperparameter Search With Evolution:

Algorithm Designing neural network architecture is challenging, even for computer scientists, researchers and machine learning experts. The team does not have the expertise in designing innovative and suitable architectures that will fit the requirement

. Given the huge number of architecture types and hyper-parameters for each model, the search space is basically infinite, so a brute-force approach with grid search would not be practical. Inspired by the paper [12], this project replicates the evolution algorithm in the context of stock price prediction.

The algorithm serves as a heuristic for architecture search, using reasonable and affordable computing power to search for ideal architectures. The same evolution algorithm was used in the paper [12] to train large-scale image classifiers.

The following is the evolution algorithm used, and the corresponding algorithm parameters are defined in Appendix E.

1. Create a population of size POPULATION_SIZE of random simple neural networks.

2. Train all neural networks in the population.

3. Calculate the mean squared error on the test set for each trained neural network.

4. Randomly select 2 networks. Select the one with better performance (lower error) as the parent network, and remove the one with a worse performance from the population.

5. Mutate the parent network to generate a new network and add it to the population.

6. Train the new network.

7. Calculate the mean squared error of the new network on the test set.

8. Repeat steps 3 - 7 for ITERATIONS number of iterations.

Different mutations are used at each step to slowly evolve the population, for example adding a dense layer, changing the number of units in a certain layer or changing the learning rate. For a full mutation list, see Appendix D. In theory, it is also possible to put the model inputs as a variable into the evolution algorithm, using the algorithm to find the optimal inputs. However, this would increase the search space significantly, and with limited resources, only a certain number of fixed inputs are tried.

8.8.7.1 Motivation

As mentioned apart from the mean squared error that a model tries to minimize, different finance-specific scores are introduced to evaluate and compare performance of different models, namely model accuracy score, model trend score and stock buy/sell score.

The scores are also designed to convey useful and meaningful messages to help investors understand a stock and make investment decisions.

8.8.7.2 Definitions In this project, the test set is defined as the last 100 days stock price.

To clearly explain the performance evaluation rationale, the following symbols are defined.

$$P_i : \text{actual price} \quad \hat{P}_i : \text{predicted price} \quad \sigma = stdev(\frac{P_{t+1}}{P_t} - 1) \quad S : \text{set of } Snakes$$

"Snakes" is defined as 10-day disjoint prediction segments in the test set, which will be a set of 10 "snake". It includes the actual prices and the predicted prices for the last 100 days.

Specifically, Snakes are defined below:

$$\begin{aligned} Snakes &= \{Snake_i \mid i \in [0,9] \ \} \\ Snake_i &= \{(\hat{P}_{10i+j}, P_{10i+j}) \mid j \in [1,10] \ \} \end{aligned}$$

It is named as "Snakes" because intuitively the 10-day disjoint segments look like snakes when being plotted on a graph of historical prices.

8.8.7.3 Model Accuracy Score

he first indicator of the performance is the Model Accuracy Score (MAS). It describes the accuracy of the price prediction regarding the actual price. It is a weighted sum of Model Prediction Score (MPS) and Model Direction Score (MDS), ranging in [0,1]. A variable α is declared to adjust the weighting between MPS and MDS contributing to MAS. Its formula is defined below:

$$\text{Model Accuracy Score (MAS)} = (1-\alpha) \cdot MPS \ + \ \alpha \cdot MDS$$

MPS is the average of Snake Prediction Scores (SPS). Each SPS is calculated by the prediction error in each of the 10-day disjoint segments, where the error is basically an average of the absolute relative change between the predicted prices and the actual prices over the 10 days. It is defined that SPS is 0 If the error is larger than the standard deviation of the stock, as the prediction would have no reference value under this circumstance. If otherwise, a scoring concave upward function is applied to scale the error to a range of [0,1] based on the standard deviation. A concave upward function is applied because the marginal usefulness of the model decreases with a marginal increase in error.

$$\text{Model Prediction Score (MPS)} = \frac{1}{|S|}\sum_{i=1}^{|S|} SPS_i$$

$$\text{Snake Prediction Score (SPS)} = \begin{cases} \left(\frac{e-\sigma}{\sigma}\right)^4 & \text{if } e \leq \sigma \\ 0 & \text{otherwise} \end{cases}, \text{ where } e = \frac{1}{10}\sum_{i=1}^{10}\left|\frac{\hat{P}_i}{P_i} - 1\right|$$

Meanwhile, MDS is the average of Snake Direction Scores (SDS). Each SDS is evaluated by the alignment of the prediction direction and the actual direction of the stock trend in each of the 10-day disjoint segments. If the prediction has a different direction with the actual direction, it means the prediction is giving a false trend signal to the users. Thus, SDS is 0. Otherwise, SDS would be evaluated based on the direction of the estimation error. In other words, if the prediction is overestimated, SDS is 0.8. Otherwise, it is 1. It is because it is assumed that an underestimated prediction means the model is more reserved and is better off than an overestimating model.

$$\text{Model Direction Score (MDS)} = \frac{1}{|S|}\sum_{i=1}^{|S|} SDS_i$$

$$\text{Snake Direction Score (SDS)} = \begin{cases} 1 & \text{if } sgn(\hat{P}_{10} - \hat{P}_0) = sgn(P_{10} - P_0) & \text{and} & |\hat{P}_{10} - \hat{P}_0| \leq |P_{10} - P_0| \\ 0.8 & \text{if } sgn(\hat{P}_{10} - \hat{P}_0) = sgn(P_{10} - P_0) & \text{and} & |\hat{P}_{10} - \hat{P}_0| > |P_{10} - P_0| \\ 0 & & \textit{otherwise} & \end{cases}$$

8.8.7.4 Model Trend Score

Another indicator of the performance is the Model Trend Score (MTS). It describes the correctness of the trend predicted by the models regarding the actual price, ranging in [0,1]. Since an accurate model in terms of the degree of price changes is difficult to obtain, sometimes the Model Accuracy Score might not be intuitive. As a result, instead of observing the exact changes in prices using MAS, we could look at the trend of the predictions which is easier to be accurate. With Model Trend Score (MTS), the users could gain accuracy insight on the future price change of the stock. It is defined as

$$\text{Model Trend Score (MTS)} = \frac{1}{10}\sum_{i=1}^{10} TS_i$$

Where TS is the Trend Score for i-day Prediction. It is the percentage of having a correct trend prediction of price i days later.

$$\text{Trend Score for i-day predict } (TS_i) = \frac{\sum_{j=1}^{100-i} f(\hat{P_{i+j}} - \hat{P}_j\ ,\ P_{i+j} - P_j)}{100-i}, \text{ where}$$

$$f(\hat{D}, D) = \begin{cases} 1 & \text{if } sgn(\hat{D}) = sgn(D) \\ 0 & \text{otherwise} \end{cases}$$

9. Methodology

- Testing

Unit Test The unittest module from Python [37] is used to implement all unit tests, as it is available by default in Python and integrates well with existing Python codes. Unit tests are done for the build dataset script, which transforms the raw input data into feature vectors usable for training and testing, as well as model score calculations.

Unit tests are conducted because the components are error-prone, calculation intensive. Also, they exhibit garbage-in-garbage-out properties, that the model will be completely wrong if it receives the wrong input, and if the model scores are wrong, the final buy-sell recommendation will be totally incorrect.

In particular, unit tests are written for the functions to build the dataset for training and prediction and the function to build the snakes. Combinations of input options are tested, including n-day stock price lookback as well as n-day moving average.

Correctness is ensured by asserting the feature vectors' shapes, as well as starting and ending elements. For model score calculation unit tests, different scenarios are emulated, including the case when the model accurately predicts all the stock prices, the case when the model predicts all the stock prices wrongly by a very large magnitude, the case when the model predicts the trend correctly but underestimates the trend, as well as the case when the model predicts the trend correctly but overestimates the trend.

9.2 Tools Used for Testing

Various tools have been used to assist in the development of the mobile application. In particular, Chrome Mobile Emulator is used to simulate the mobile view while developing the mobile application on desktop/laptop computers. After the application is deployed to the cloud, mobile phones with different operating systems and browsers, including Google Pixel running Android 9 (Google Chrome) and iPhone 7 running iOS 12.1 (Safari), are used to verify the user experience is consistent across different devices with different resolutions.

10.Methodology – Evaluation

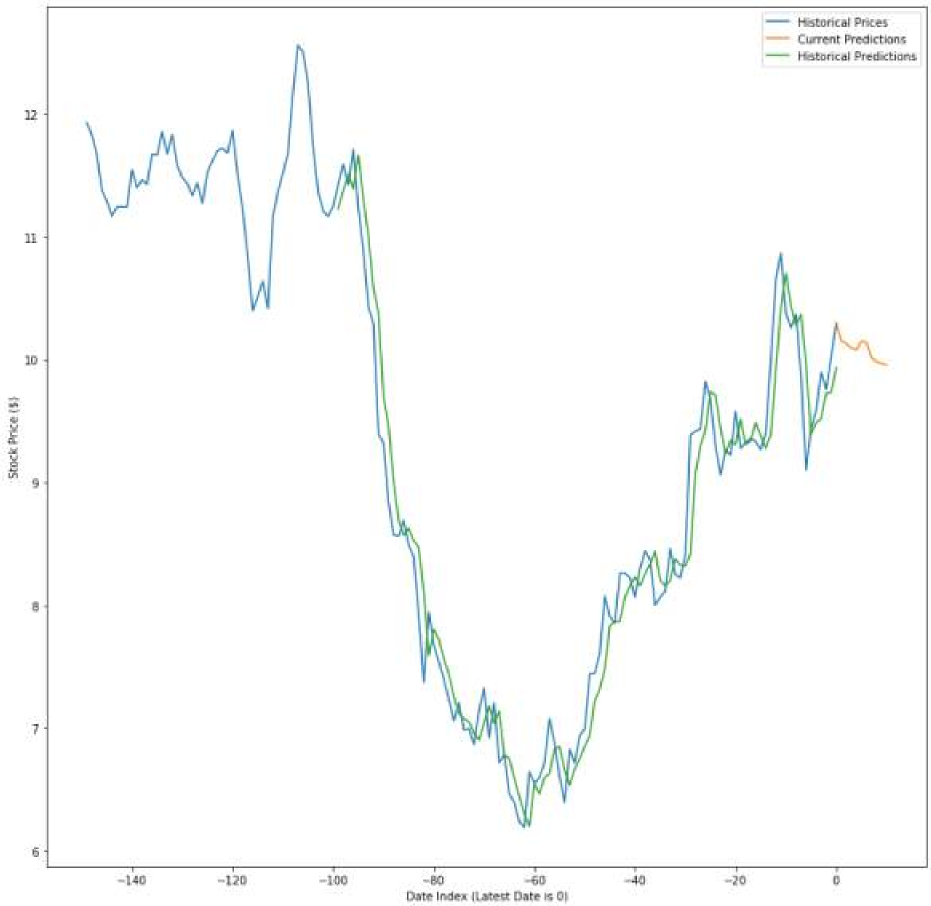
The project's objective is to provide a third-party investment tool to investors with democratized machine learning technologies. The success of the project is primarily determined by two factors, namely, whether the investment tool provides useful, accurate stock price predictions to investors, and whether investors can use and understand the predictive information provided by the machine learning technologies. The first factor is evaluated by the model scores described .

However, the evaluation of the second factor is based on user experience. External users have to be involved in the evaluation. For this purpose, hallway testing is used. Hallway testing involves allowing users who have not been involved in the development of the project to test the application and give constructive feedbacks about users feel about the application.

Users participating in the tests are asked a set of questions about the usability and whether they understand what the information presented by the mobile application. This would give indications about whether the democratization of the machine learning technologies succeeds.

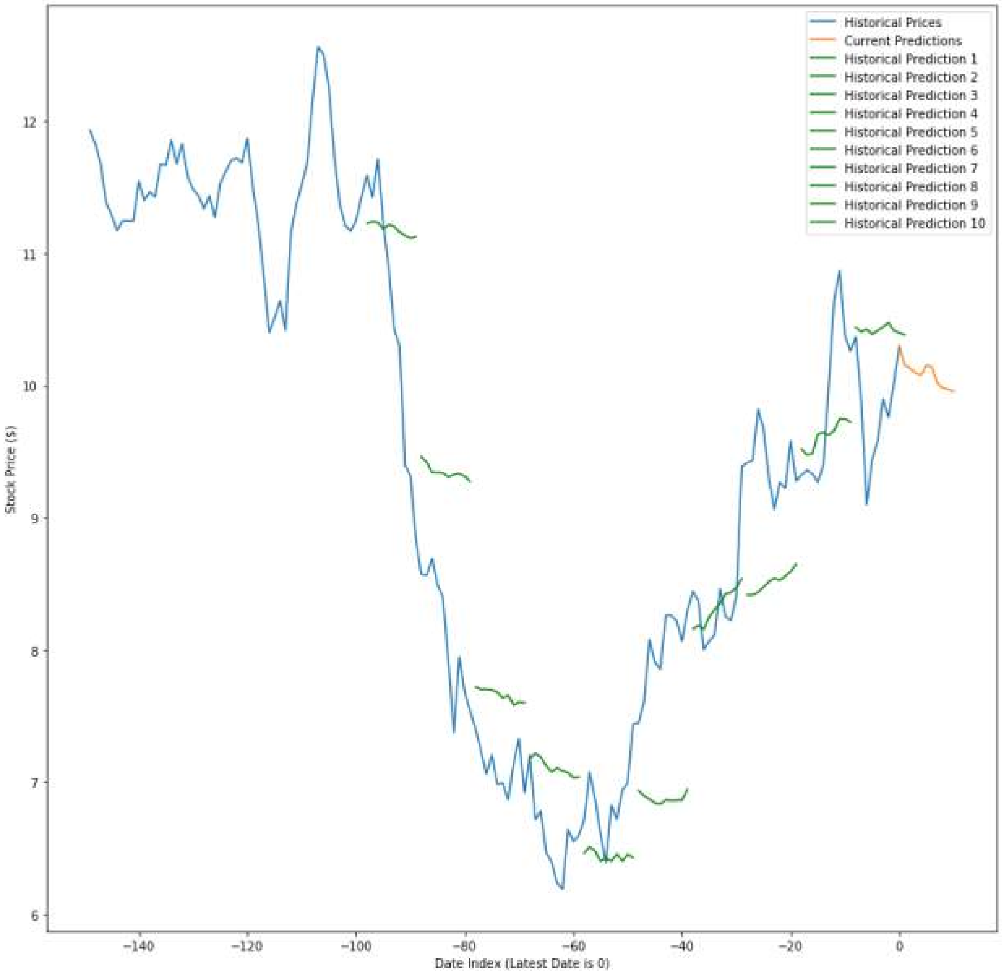
11. Findings

All results and findings graphs can be found in a Google Colaboratory notebook at https://colab.research.google.com/drive/1GYuxbYywhN8-_D3eycsiQ-iYLzv-YjXq. General Findings The following are some general findings from testing out different machine learning models.



1-day interval historical predictions (GE, Dense Neural Network)

From fig it shows that the 1-day interval historical predictions line follows closely with the historical prices. The graph looks like the prediction line is just 1 day shifting from the historical prices, similar to a shifted and smoothed out historical prices line. Therefore, the shape of the historical predictions line is similar to the shape of the exponential moving averages (EMA), where the price changes from t to t+1 heavily depends on the direction and magnitude of changes from t-1 to t, followed by decreasing importance from earlier historical prices. Other models in predicting stock prices of other stocks also show similar results.

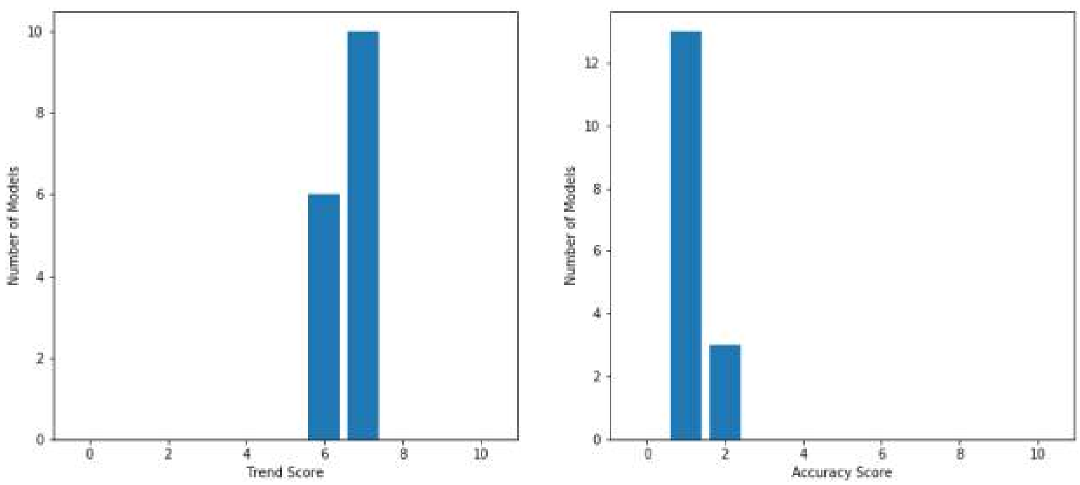


1b 10-day interval historical predictions (GE, Dense Neural Network)

From Fig 1b, it shows that the 10-day interval historical predictions line do not follow closely with the historical prices but could demonstrate the trend.

For example, historical predictions 1, 2, 3, 4, 7, 8, 9, 10 provided insights on the correct market direction, yet the magnitude did not match the actual price movements.

A possible reason for this error can be the 10-day interval prediction has to predict more values while having fewer data compared to the case of 1-day interval prediction, which for 1-day interval prediction, data of close prices until previous day are available. Therefore, a longer period of interval prediction could subject to greater changes in market fundamentals, including market news, macroeconomic factors, earning reports, etc. Other models in predicting stock prices of other stocks also show similar results. Although price reflects all available information, the magnitude of price changes in the future might need other data for forecasting purpose, such as market sentiment, company announcement, retail and institutional investors' attention on the company, etc. This is one of the possible explanation of why the 10-day interval prediction might have a large difference to actual values as there are potential shifts in market momentum. Therefore, the price migh too compact and other information is required to make a more accurate prediction.



1c Trend score and accuracy score distribution (16 best models from evolution)

2 scores are used to measure the performance of historical predictions, trend score and accuracy score, introduced in fig . The higher the trend score means that the model is more accurate in trend prediction and could provide more meaningful price movement direction insights.

The score representations used in this application could be useful for the user to interpret the errors of predictions in a quantifiable way. The higher the accuracy score means that the model could follow the actual stock prices more accurately.

From Fig .1c, it shows that all best models generated from the evolution algorithm experiment have a trend score ranging from 6-7 but have an accuracy score ranging from 1-2 on the test set. This finding matches the earlier findings, that the trend could be predictable, especially for less volatile stocks, but exact price, especially further into the future, could hardly be predicted accurately.

Despite common research findings that recurrent neural networks in general perform better than dense feedforward neural networks at predicting time-series data such as stock prices, in this project feedforward neural network outperforms recurrent neural networks.

One possible explanation is that training a recurrent neural network requires more data than the dense neural network in general, as recurrent neural networks have more parameters.

As the models are trained using only daily stock prices dating back 20 years (or less if the stock is listed fewer than 20 years), there might not be enough data for training the recurrent networks to a good performance.