iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IV **Month of publication:** April 2022

DOI: https://doi.org/10.22214/ijraset.2022.41331

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

Stock Prediction using Neural Networks and Evolution Algorithm

Shekhar Paliwal[1], Shivang Sharma[2]
[1, 2]*Computer Science and Engineering Department, Inderprastha Engineering College*

Abstract: ***Various researches and studies have shown that machine learning techniques like neural network have the ability to learn and predict the general trend of stock market. Artificial intelligence and different machine learning techniques have been widely implemented in the field of forecasting stock prices for a long time. However, selecting the best model and best hyper-parameters for these models is highly necessary for better accuracy in prediction. Given the huge number of architecture types and hyper-parameters for each model, it is not practical to find the best combination for best accuracy. Therefore, in this research we used evolution algorithm to optimize model architecture and hyper-parameters. Promising results are found in stock prediction.***

Keywords: Neural Network, Long Short-Term Memory, Recurrent Neural Network, Dense Neural Network, Gated Recurrent Unit, Stock Prediction, Evolution Algorithm.

I. INTRODUCTION

Millions of people invest in stock market everyday. An individual may spend a lot of time in finding investment opportunities. They have to figure out the market themselves and make decisions on their own. However many times these decision may be irrational and without a tool to help them they may get swayed by personal feelings, thus getting unnecessary losses. This project was developed to find the best neural network model to predict stock trend with high accuracy and help these individuals. The evolution algorithm used in this project helps us to find the best model with best set of hyper-parameters to predict the stock prices accurately. Different models: Dense Neural Network, Recurrent Neural Network, Long-Short Term Memory, and Gated Recurrent Unit Neural Networks are used in this project for stock prediction. Each of these models are passed through various iteration of evolution to select the best hyper-parameters. Over time the errors in these models are reduced and they become more and more accurate in forecasting.

II. LITERATURE SURVEY

From the paper, “Machine Learning in Stock Price Trend Forecasting”[1], they used various features like PE ratio, PX ratio, PX EBITDA, 10-day volatility, etc. to predict the stock price for the next day. The algorithms used here were Logistic Regression, Gaussian Discriminant Analysis, Quadratic Discriminant Analysis, and SVM. Here, the accuracy was defined as the number of days that model correctly predicts the value. For short-term the results had very low accuracy, with Quadratic Discriminant Analysis being the best among them with 58.2% accuracy. In the long term they had better accuracy with SVM performing the best: 79.3 accuracy in 44 days period.

The paper, “Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques”[2], showed a way to use deterministic data to predict stock price. They took 10 technical indicators as input, then used prediction models to forecast the trend for next 10 days. The results showed an accuracy of 83.56%.

In the paper, “ANN Model to Predict Stock Prices at Stock Exchange Markets”[3], they used 70% of training data to predict next 60 days stock prices. The results had 0.71% mean absolute percentage error.

Various researches have shown that large-scale evolution can generate neural network model architecture with very high accuracy results. In the research [4], a large-scale evolution algorithm was applied for discovering image classification neural networks. The algorithm showed to slowly discard the poor accuracy models and mutating hyper-parameters to generate new models. The paper showed possibility of finding better models using this method.

III.METHODOLOGY

The main objective of the project is to predict the stock price of a company for next 10 days. This time frame is chosen because stock price in short term tend to depend upon historical price pattern. Whereas, long term stock prices depend upon several other fundamental factors such as: company management, income model, demand in market, etc.

Here, we have chosen mean square error as the loss function of the models we train. The main task of evolution algorithm will be to minimize the mean square error and it will also serve as a metric to compare different models.

For the project we have taken random historical stock prices of 5 different companies to test on. These companies are listed below:-

1) Alphabet Inc., GOOGL
2) Facebook Inc., FB
3) General Electronics Co, GE
4) Microsoft Inc., MSFT
5) Tesla Inc., TSLA

A. Data Pre-processing

Raw stock price data is first pre-processed before giving it as an input into the models. The process includes converting the data into the format that models can operate on.

We have taken a fixed length of period that the model will look back on from the day of prediction.

B. Prediction Output

The models will predict the out for next 10 days. These prediction will 10 output units in the form of 1D vector containing 10 stock prices. Here, the *i*-th element of the vector will represent the *i*-th day of stock price prediction.

NumPy and Pandas are used to build the datasets for model inputs. NumPy[5] is the library that provides effective n-D array data structure and function to manipulate these arrays.

Pandas[6] is a widely used framework for pre-processing. It can read data as raw .csv files and convert it into correct format. It also uses NumPy as the underlying data structure.

C. Model

Different neural network models used are listed below:-

1) Dense Neural Network
2) Recurrent Neural Network
3) Long Short-Term Memory Network
4) Gated Recurrent Unit Network

Each of these models are tested with different set of hyper-parameters such as: number of hidden layers, number of units in each hidden layer, activation function, etc.

All of these models have exactly the same high-level architecture. Therefore, we have used common interface for these models. They have their own class, with specific details to set the hyper-parameters.

To provide flexibility in changing these hyper-parameters these models are defined in JSON format.

Example of this is as below:-

```
"modelOptions": {
  "network_type": "dense",
  "net": {
    "layers": [
      {"units": 32, "activation": "relu", "is_input": true, "inputUnits": 10},
      {"units": 64, "activation": "relu"},
      {"is_output": true, "activation": null}
    ],
    "loss": "mse",
    "optimizer": "adam",
    "learning_rate": 0.001,
    "epochs": 20,
    "batch_size": 32,
    "metrics": ["accuracy"],
    "evaluation_criteria": {
      "minimize": true,
      "threshold": 10
    }
  },
  "predict_n": 10
}
```

Input for the mutations may also vary. The input options tells the hyper-parameters a model should have. They are also configured in JSON format.

Example of these are given below:-

```
"inputOptions": {
  "config": [
    {"type": "lookback", "n": 10, "stock_code": "GOOGL", "column": "adjusted_close"},
    {"type": "moving_avg", "n": 10, "stock_code": "GOOGL", "column": "adjusted_close"}
  ],
  "stock_codes": ["GOOGL"],
  "stock_code": "GOOGL",
  "column": "adjusted_close"
}
```

D. Training

A model with random hyper-parameters is generated and then trained for the population in evolution algorithm.

The trained model is then saved for prediction. A dedicated saving format for the model is created, same models for different companies are saved in same directory.

The saving format is as below:-

```
/saved_models
  /dnn
    /<model_hash_1>
      /<stock_code_1>
      /<stock_code_2>
    /<model_hash_2>
      /<stock_code_1>
      /<stock_code_2>
    models_data.json
  /linear_regression
    /<model_hash_1>
      /<stock_code_1>
      /<stock_code_2>
    models_data.json
```

E. Predicting Stock Prices

For forecasting, the saved model is first loaded, then a feature vector is build specified by the input options. This vector is given as input to the model for prediction. The forecast is then appended to the raw dataset and shown as the out put along with historical data.

F. Performance Analysis

Each model is analyzed on test set. A test set contains 100 days of stock price data.

G. Evolution Algorithm

Designing a neural network is a difficult task. Given the huge amount of combinations of different hyper-parameters, it is not practical to find the best set which will give the most accurate forecasting.

The evolution algorithm used here was inspired by the paper [4]. The algorithm searches the best architecture type with most optimized hyper-parameters.

The steps of the evolution are listed below in detail:-

1) Create a population of random neural networks.
2) Train all the neural networks in population.
3) Calculate the mean square error on the test set of each neural network.
4) Randomly select 2 neural network from the population. From these two discard the one with more error.
5) Apply mutation on the other network and then re-insert it into the population.
6) Train the new mutated network.
7) Calculate the mean square error for the new mutated network on the test set.

8) Repeat steps 3 – 7 for a set number of time.

Table 1. Evolution Algorithm Mutations

Mutation	Input options
Add dense layer	
Remove dense layer	
Change units	8, 16, 32, 64, 128
Change activation function	ReLu, sigmoid, tanh, linear
Change learning rate	0.01, 0.001, 0.0001
Change batch size	16, 32, 64

Table 2. Evolution algorithm parameters

Parameter	Value
POPULATION_SIZE	10
ITERATIONS	>=100
Optimizer used in training	Adam optimizer
Epochs	20

IV. RESULTS AND DISCUSSION

The result of stock price prediction of the 5 different companies selected, without applying evolution algorithm are given as below:-

A. Facebook (FB)

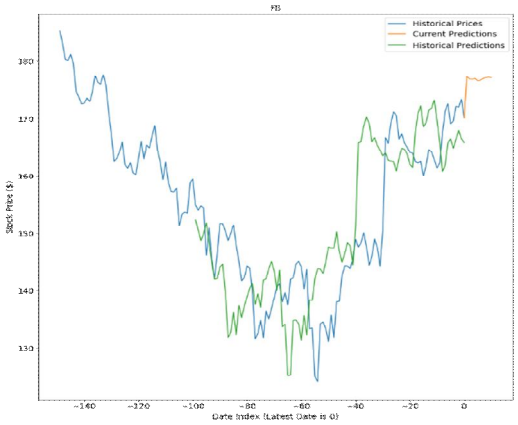


Figure 1.1: DNN model for FB

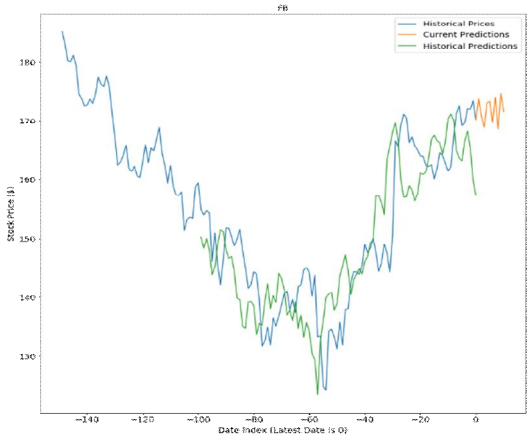


Figure 1.2: RNN model for FB

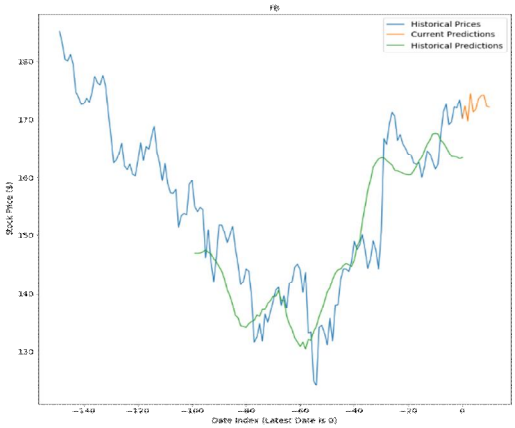


Figure 1.3: LSTM model for FB

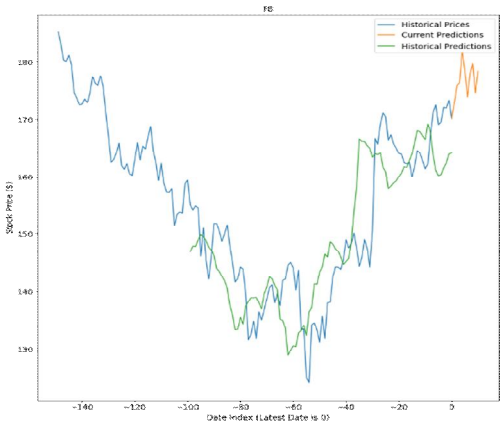


Figure 1.4: GRU model for FB

B. General Electronics (GE)

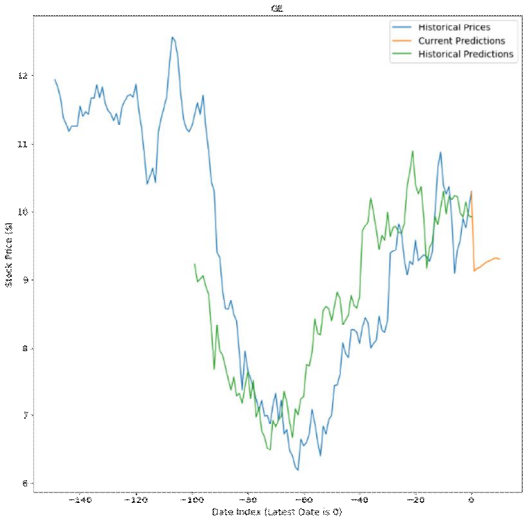


Figure 2.1: DNN model on GE

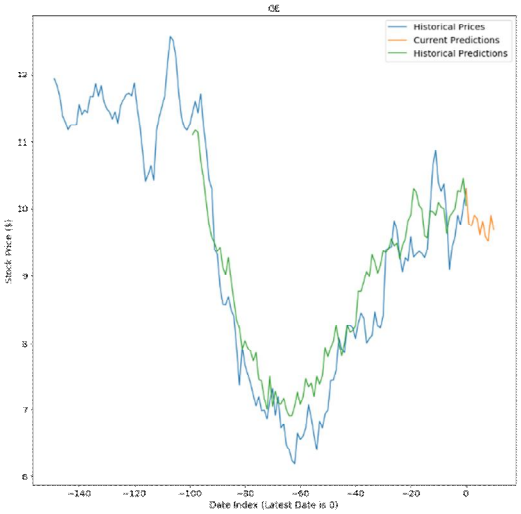


Figure 2.2: RNN model for GE

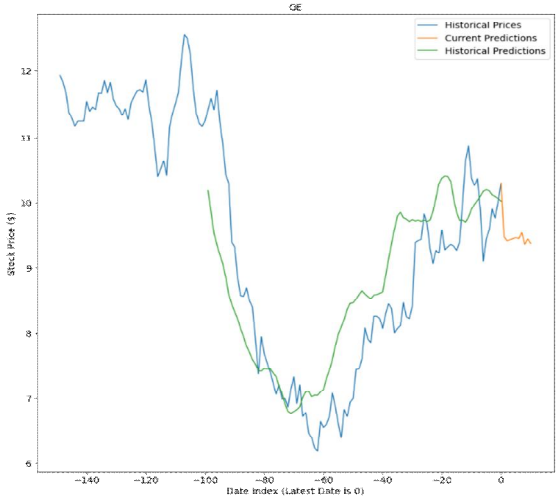


Figure 2.3: LSTM model for GE

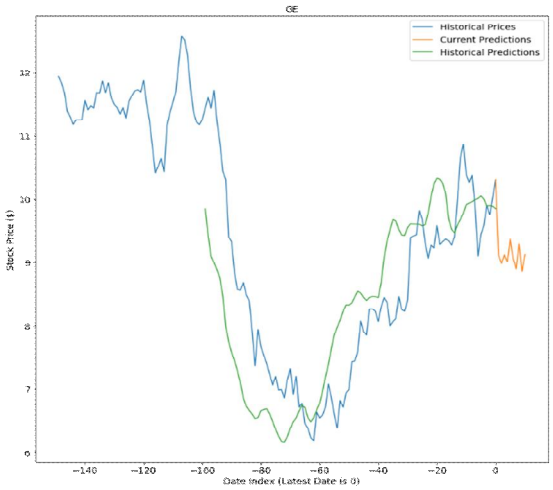


Figure 2.4: GRU model for GE

C. Google (GOOGL)

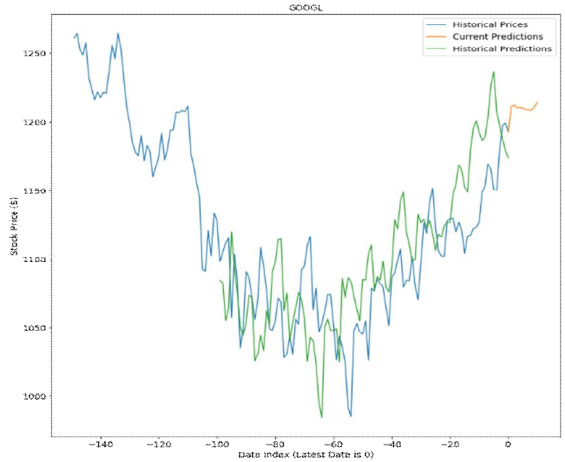


Figure 3.1: DNN model for GOOGL

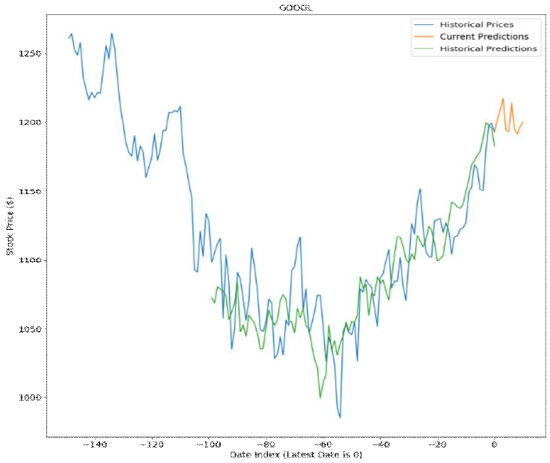


Figure 3.2: RNN model for GOOGL

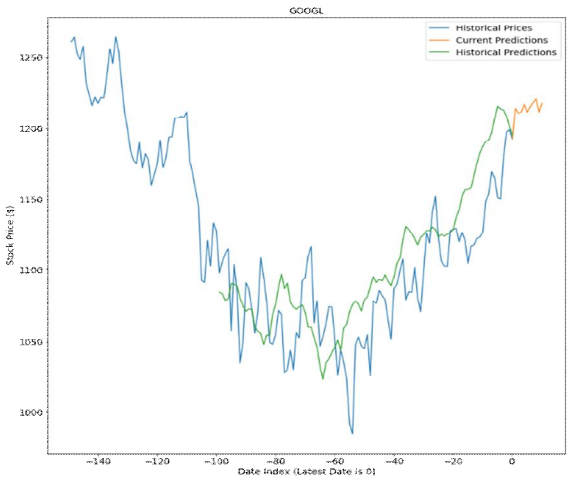


Figure 3.3: LSTM model for GOOGL

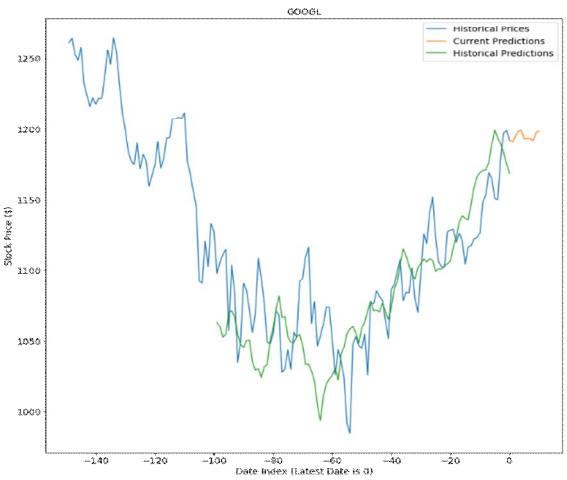


Figure 3.4: GRU model for GOOGL

D. Microsoft (MSFT)

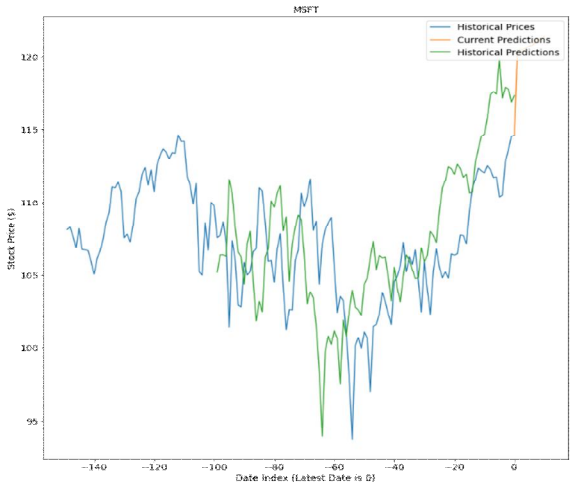


Figure 4.1: DNN model for MSFT

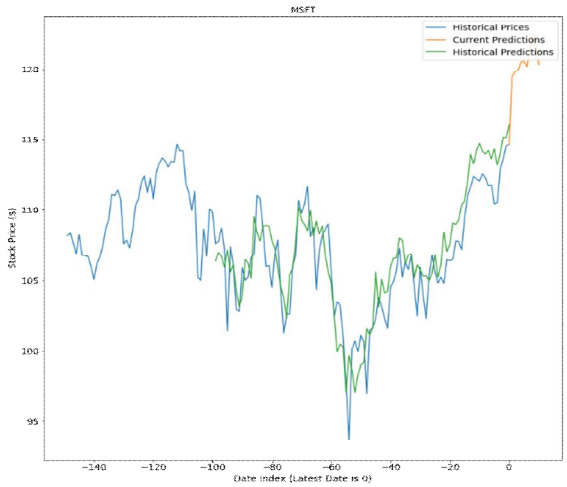


Figure 4.2: RNN model for MSFT

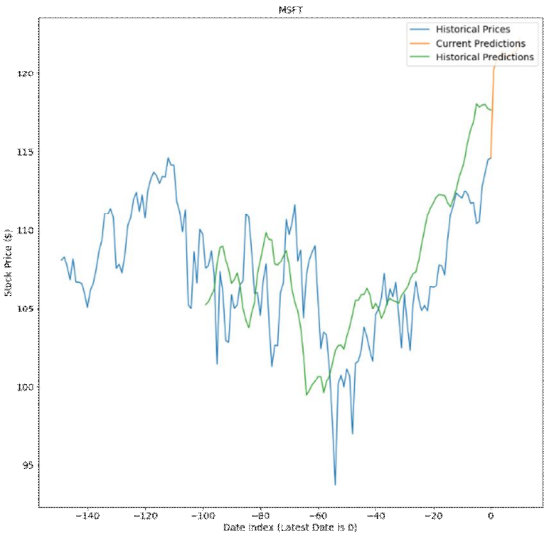


Figure 4.3: LSTM model for MSFT

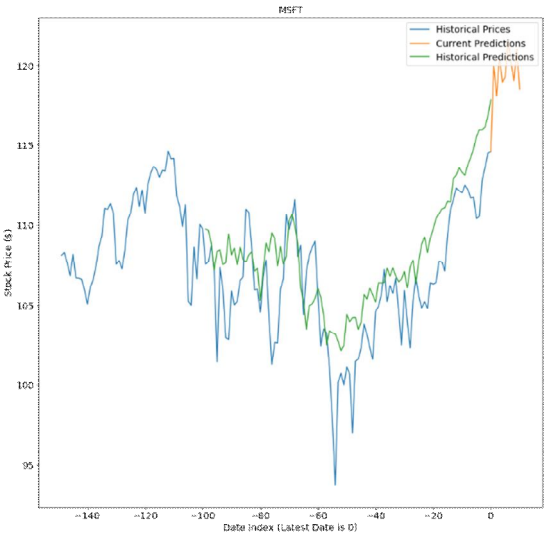


Figure 4.4: GRU model for MSFT

E. Tesla (TSLA)

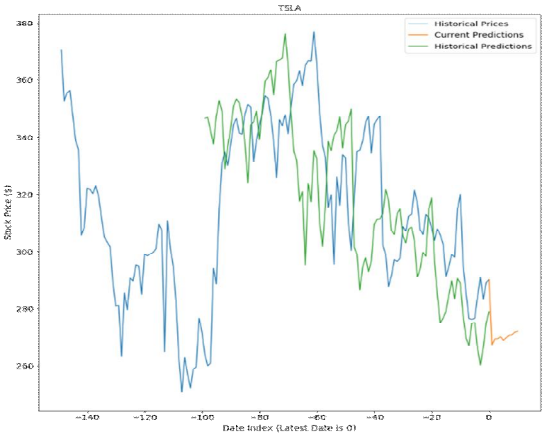


Figure 5.1: DNN model for TSLA

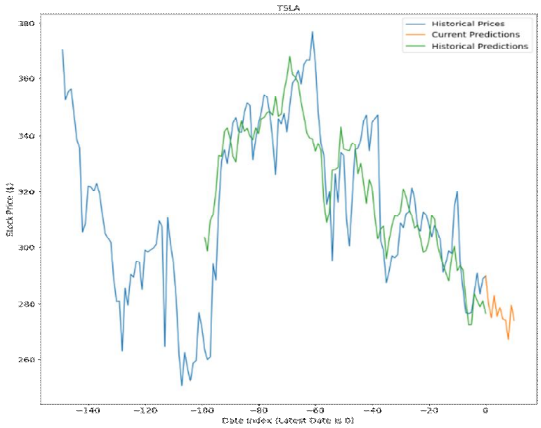


Figure 5.2: RNN model for TSLA

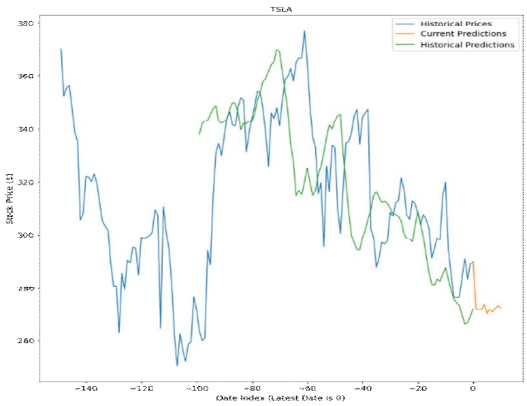


Figure 5.3: LSTM model for TSLA

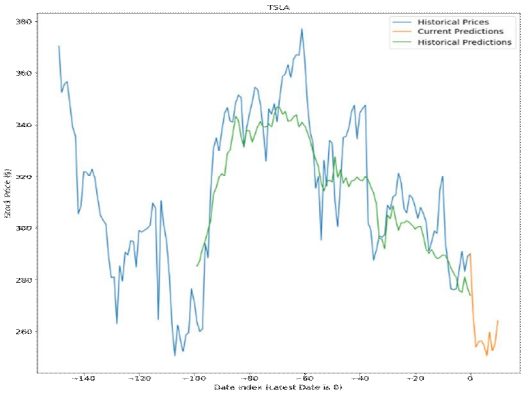


Figure 5.4: GRU model for TSLA

These graphs shows the stock price prediction of the 4 different neural network (DNN, RNN, LSTM, GRU) on the 150 days dataset of different companies. The result shows that while the forecasting follows general trend overall, but on day-to-day basis there are various wrong predictions by a large margin. This may result in huge loss for any individual who is investing in the stock market using prediction of these models.

Below are the stock prediction after applying evolution algorithm and again predicting on the same dataset:-

1) Facebook (FB)

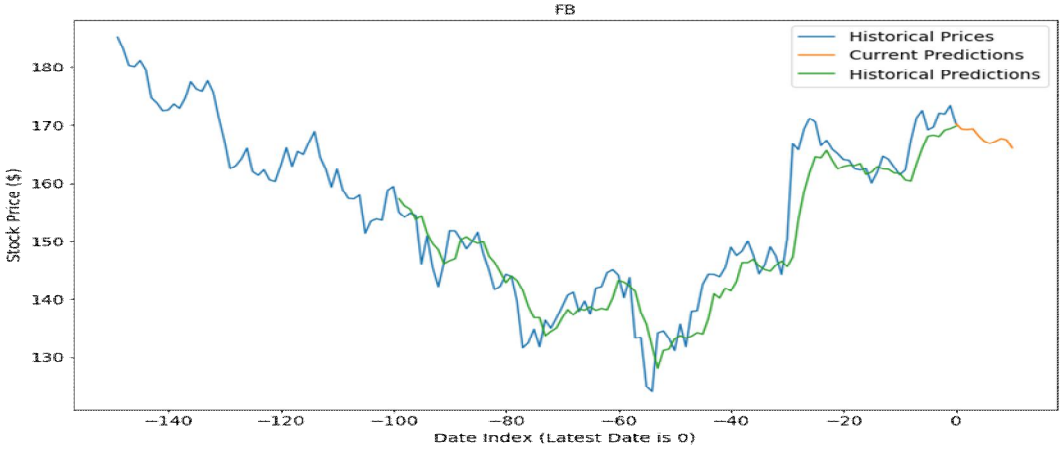


Figure 6.1: Stock prediction after applying evolution algorithm (FB)

2) General Electronics (GE)

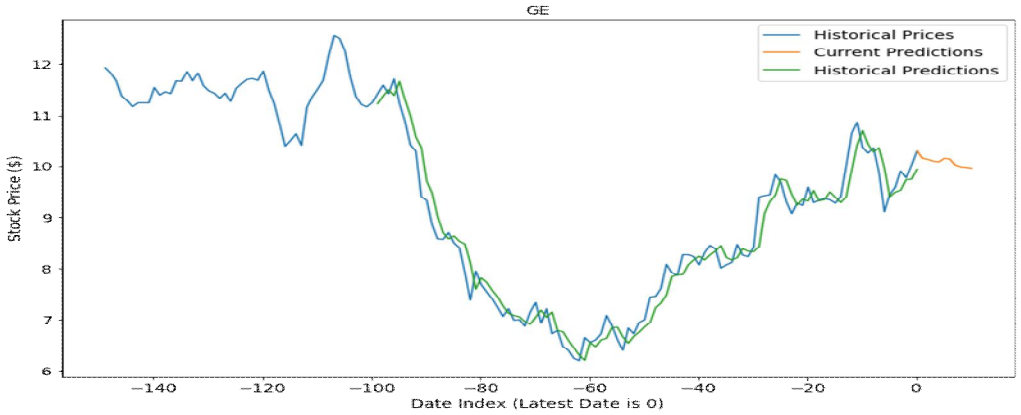


Figure 6.2: Stock prediction after applying evolution algorithm (GE)

3) Google (GOOGL)

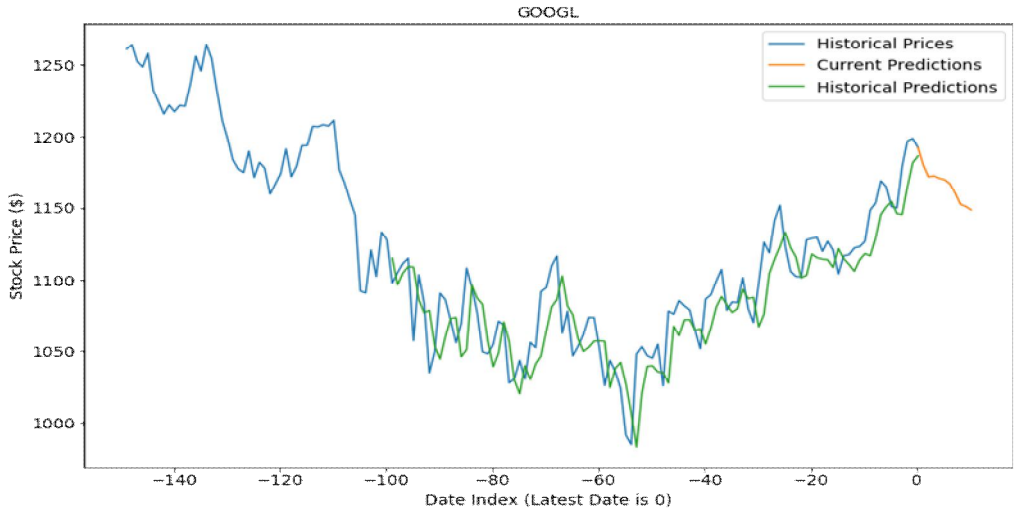


Figure 6.3: Stock prediction after applying evolution algorithm (GOOGL)

4) Microsoft (MSFT)

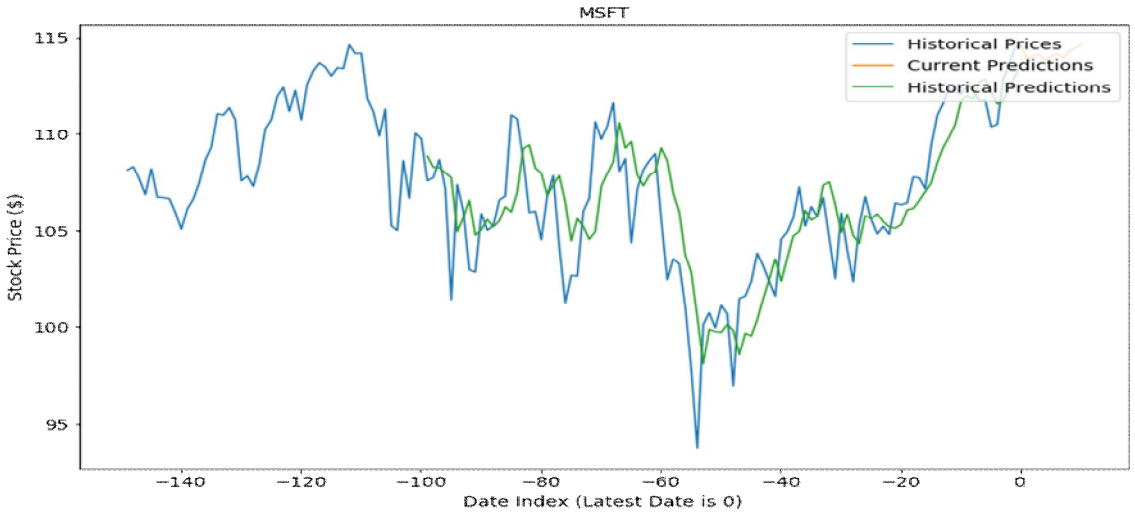


Figure 6.4: Stock prediction after applying evolution algorithm (MSFT)

5) Tesla (TSLA)

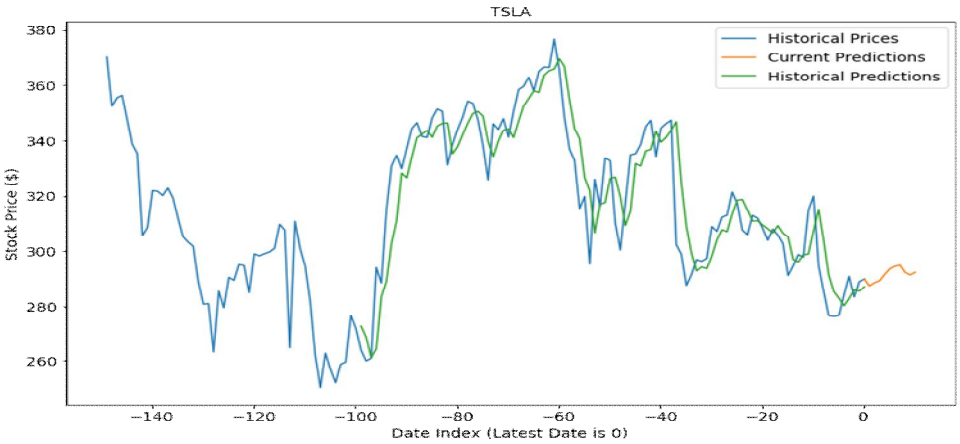


Figure 6.5: Stock prediction after applying evolution algorithm (TSLA)

As seen in the graphs above after applying evolution algorithm the model's stock predictions are much more accurate and follows the stock price pattern much closely as compared to before. Although the prediction reflects all data accurately, the magnitude of price change, especially in long-term, will depend upon various other real world circumstances such as, market demand for the company, new products launched, management of the company, etc.

Below are shown the error changes while implementing evolution algorithm:-

a) Facebook (FB)

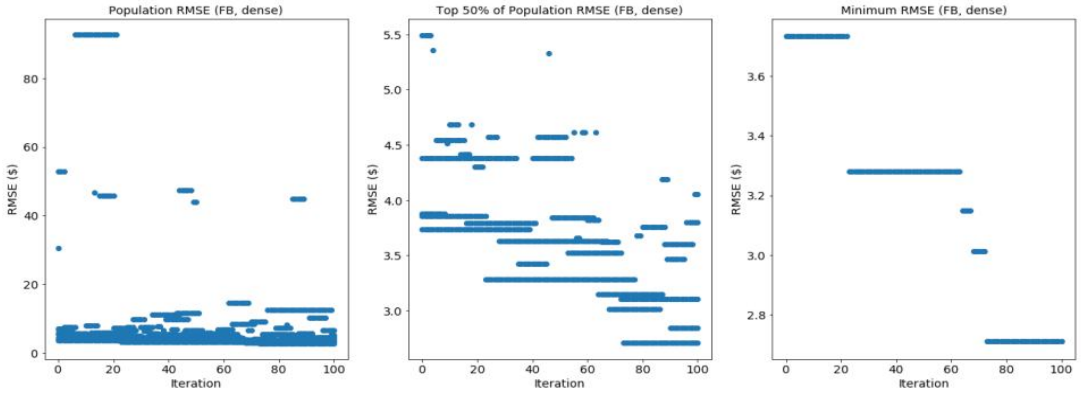


Figure 7.1: Root Mean Square Error with each iteration of evolution algorithm (FB)

b) General Electronics (GE)

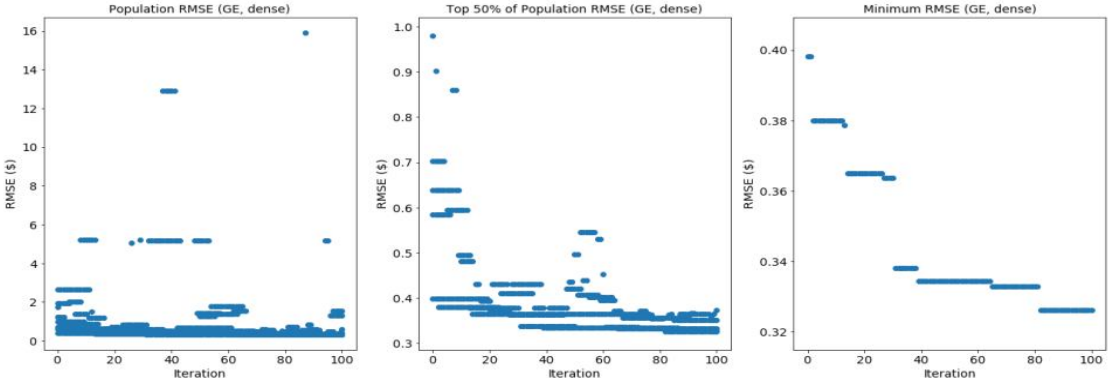


Figure 7.2: Root Mean Square Error with each iteration of evolution algorithm (GE)

c) Google (GOOGL)

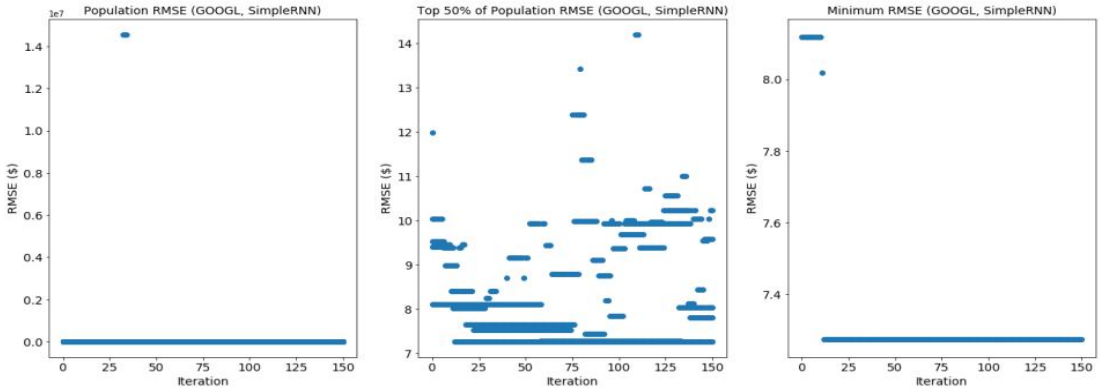


Figure 7.3: Root Mean Square Error with each iteration of evolution algorithm (GOOGL)

d) Microsoft (MSFT)

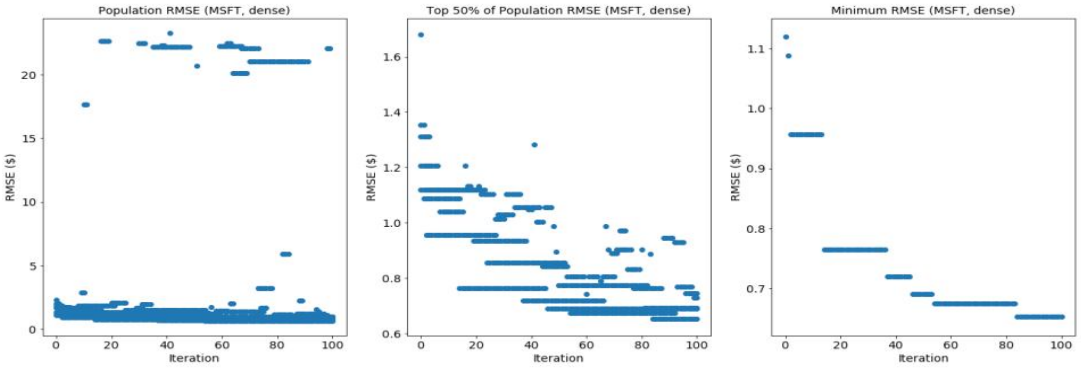


Figure 7.4: Root Mean Square Error with each iteration of evolution algorithm (MSFT)

e) Tesla (TSLA)

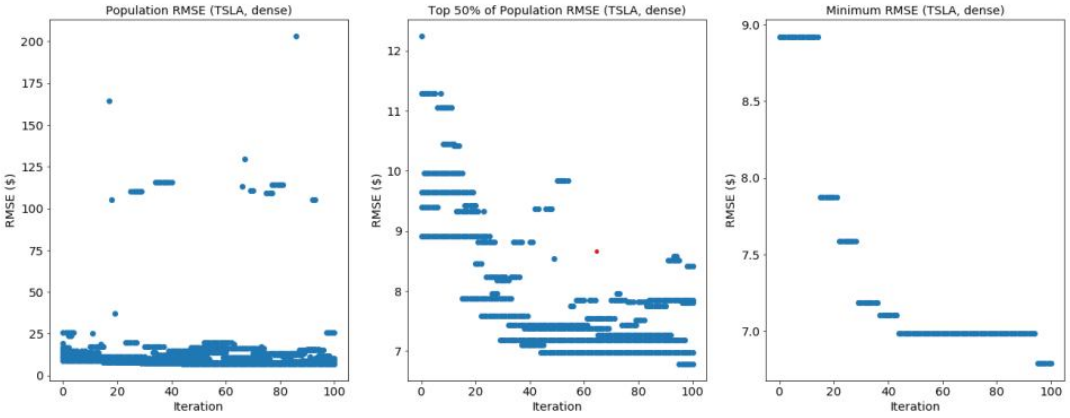


Figure 7.5: Root Mean Square Error with each iteration of evolution algorithm (TSLA)

All results have shown that evolution algorithm is exploring better and better model architecture over time. One observation from this experiment is that the best model architecture are fairy simple. The models are 1-2 layers deep with linear activation function. An explanation for such could be that in evolution algorithm parameters input used has limited the search space for possible models. Due to computational power restriction we only used population of 10 models at a time. Moreover, the no. of iterations here is 100 only. These low parameters may result in the algorithm not being able to perform deeper search.

This can be rectified with higher value of parameters, but that would require much higher computational power which we did not have. Another explanation for this may be that the dataset used in this experiment was small (150 days of data). Larger and deeper neural network may require a bigger dataset to work on.

The success of this project depend upon whether an individual will reliably be able to use this tool for investment purposes. In that regard, this project is partially successful as the prediction are reliable for at least short-term. For long-term, the investor also has to take into account various fundamental analysis and real-world circumstances, which simply can't be incorporated here.

V. CONCLUSION

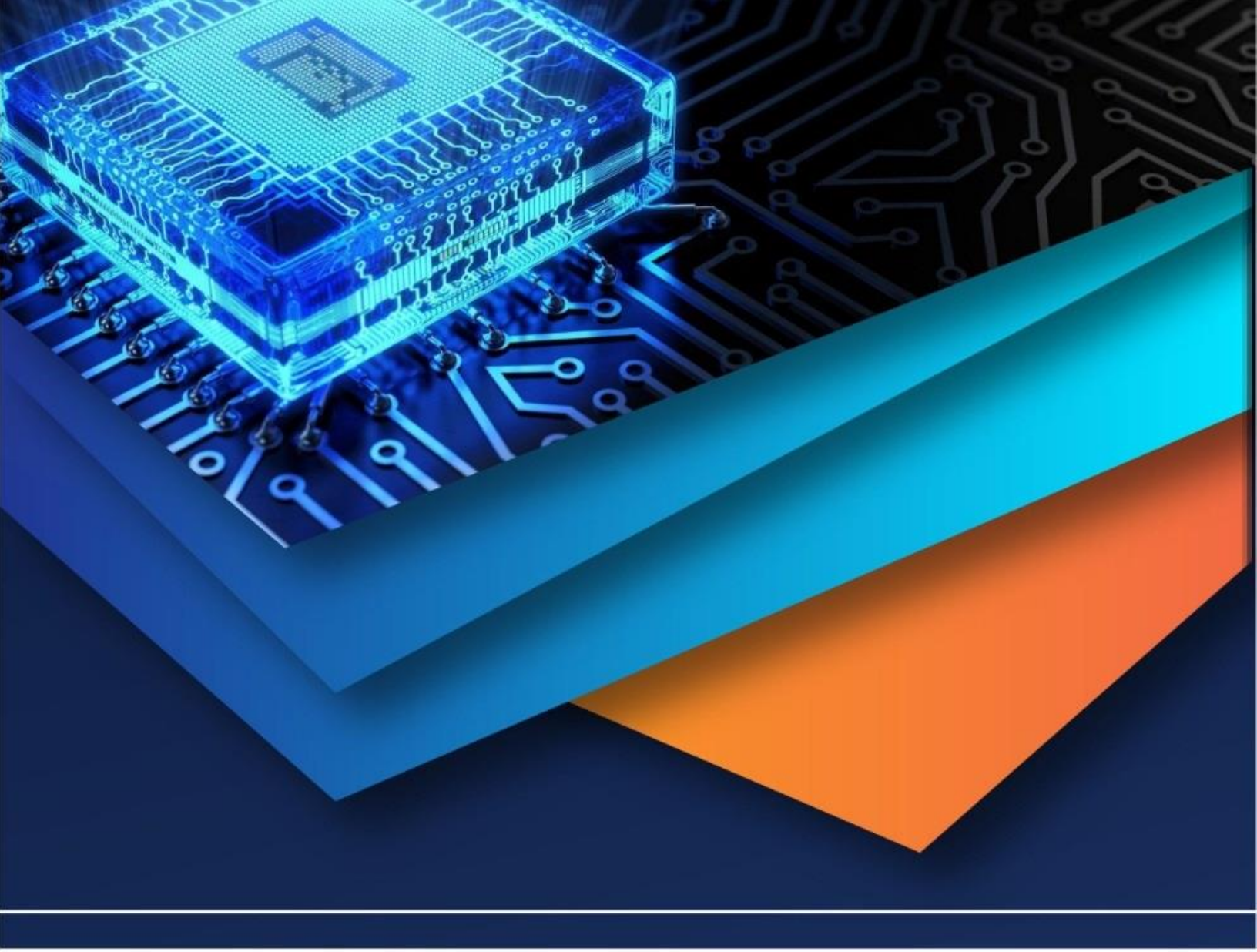
The project lays foundation of using evolution algorithm in prediction of not only stock prices but any time-series dataset. The result have shown that models after evolution algorithm show significant improvement in accuracy of prediction.

The evolution algorithm has proved to be successful in reducing the mean square error over time thus increasing the efficiency of the model and accuracy even further with more iterations.

There are various further improvement which can implemented in this project in the future. First, Incorporating market news, and sentiment analysis of the public regarding the company. Second, increasing the computational resources and providing higher values of parameters in evolution algorithm, which will result in deeper search and better optimization.

REFERENCES

[1] Y. Dai and Y. Zhang, "Machine Learning in Stock Price Trend Forecasting," Stanford University; http://cs229.stanford.edu/proj2013/DaiZhang-MachineLearningInStockPriceTrendForecasting.pdf.
[2] 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: An International Journal, Vol. 42, Jan. 2015, pp. 259-268
[3] B. Wanjawa and L. Muchemi, "ANN Model to Predict Stock Prices at Stock Exchange Markets," arXiv:1502.06434 [q-fin.ST], 2014
[4] E. Real, et al., "Large-Scale Evolution of Image Classifiers," arXiv:1703.01041 [cs.NE]. Jun 2017
[5] NumPy v1.14 Manual, The SciPy community. Available: https://docs.scipy.org/doc/numpy-1.14.5/
[6] Pandas: powerful Python data analysis toolkit, Pandas. Available: http://pandas.pydata.org/pandas-docs/version/0.23/.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY