FORECASTING COMPUTER PROCESSOR MANUFACTURES COMPANIES BY USING STATISTICAL, MACHINE LEARNING AND DEEP LEARNING MODEL WITH NADAM OPTIMIZER

NGO QUOC HUY[1], LY GIA HIEU[2], MAI TRAN PHUONG NHI[3], TRAN THANH PHONG[4]

[1]Faculty of Information Systems, University of Information Technology, (e-mail: 21522148@gm.uit.edu.vn)
[2]Faculty of Information Systems, University of Information Technology, (e-mail: 21522074@gm.uit.edu.vn)
[3]Faculty of Information Systems, University of Information Technology, (e-mail: 21522428@gm.uit.edu.vn)
[4]Faculty of Information Systems, University of Information Technology, (e-mail: 20521750@gm.uit.edu.vn)

ABSTRACT The article presents an analysis and stock price forecasts of QCOM, NVDA, and AMD, key companies in the processor manufacturing industry, which are indispensable for the advancement of Artificial Intelligence (AI). Utilizing models such as Linear Regression, ARIMA, RNN, GRU, LSTM, Holt-Winters, Boosting Models, and MLP, we have predicted the stock prices of these companies. Additionally, by applying the Nadam optimization algorithm, we enhanced the performance of Deep Learning models, providing deep insights into stock price fluctuations and advanced forecasting methods.

INDEX TERMS **Stock Price Forecasting, Machine Learning, Deep Learning, Computer Processor Manufactures Companies.**

I. INTRODUCTION

In today's modern technological landscape, the significance of processor manufacturing companies has grown exponentially due to their close ties with Artificial Intelligence (AI). Leading enterprises such as NVDA, AMD, and QCOM have swiftly ascended to dominance in the global stock market, propelled by the ongoing advancements in AI.

This study is centered on the statistical analysis and stock prediction of NVDA, AMD, and QCOM. By employing an array of forecasting models, including Linear Regression, ARIMA, RNN, GRU, LSTM, Holt-Winters, Boosted Models, and MLP, our objective is to grasp the intricate dynamics steering stock price fluctuations. Furthermore, harnessing the Nadam optimization algorithm will augment the precision of our forecasts

Through analyzing historical trends and utilizing various forecasting techniques and models, this research aims to forecast stock prices. Furthermore, it provides valuable insights for investors, analysts, and stakeholders navigating the convergence of technology and finance, particularly in predicting stock prices.

II. RELATED WORKS

In recent years, there has been a substantial amount of research dedicated to predicting stock prices using various machine learning and statistical models. The study [1] conducted by Bhawna Panwar et al… covers two models, Linear Regression and Support Vector Regression (SVR) to forecast Amazon's stock price (AMZN) in October 2019. The results showed that the accuracy of Linear Regression was 98.76%, which was higher than the accuracy of SVR, which was 94.32% .

Aishwariya Subakkar, S.Graceline Jasmine… used the ARIMA model to predict stock prices, focusing on the Bombay Stock Exchange (BSE) and National Stock Exchange [2]. Their findings reveal the ARIMA model's effectiveness in short-term and daily stock price predictions, demonstrating its strong performance compared to other models.

Vivek Varadharajan, Nathan Smith… conducted the work with Long Short-Term Memory (LSTM-RNN) and Recurrent Neural Networks (RNN) [3] to predict the daily closing price of the Amazon Inc. stock (AMZN). Showing strong performance in the datasets. After optimization, the model achieved a really good performance with RMSE of 2.51 and MAPE of 1.84% on the training set. This study [4] by Nrusingha

Tripathy, Surabi Parida… presents a hybrid LSTM-GRU model that outperforms existing models (LSTM, RNN,...) in stock price prediction. By combining LSTM and GRU algorithms, the model delivers superior performance with an MSE of 3.041 and RMSE of 1.744.

In the study [5] CS Agustina, T Asfihani, RR Ginting, and S Subchan have shown Holt Winters method offers good stock price predictions with low MAPE errors for PT. X Tbk(1.6%), PT. Y Tbk(1.29%), and PT. Z Tbk(1.38%). They have proved that Holt Winters method can be effective for stock portfolio management, MPC optimizes portfolio capital based on stock price predictions, providing satisfactory results.

Xinwen Xu has provided a comparison between Linear Regression model with Gradient Boosting model in the study [6].The linear regression and gradient boost models effectively predict Netflix stock returns before COVID-19, as the data lacks major fluctuations. Linear regression reveals a linear relationship with stock data over time, while gradient boost, based on weak learners, offers notable accuracy even without parameter tuning.

Haixu Wu, Jiehui Xu, Jianmin Wang… presents Autoformer [7], a model for long-term time series forecasting. It uses a series decomposition block to aggregate long-term trends and an Auto-Correlation mechanism for dependency discovery. Autoformer achieves *O(L log L)* complexity and consistently outperforms state-of-the-art methods on real-world datasets.

In the study [8], Jatin Sharma, Sameer Soni… Have shown the strength of Nadam optimizer through the combination with LSTM model compared with the original LSTM and other optimizers. Nadam was the key step in improving the forecast accuracy. Using Nadam optimizer has enhanced the model's accuracy hence reducing RMSE and MSE.

III. MATERIALS

A. DATASET

The historical stock price of QUALCOMM (QCOM), Advanced Micro Devices (AMD) and NVIDIA (NVDA) from 01/03/2019 to 01/06/2024 will be applied. The data contains column such as Date, Open, High, Low, Close, Volume. As the goal is to forecast close prices, only data relating to column "Close" (USD) will be processed.

B. DESCRIPTIVE STATISTICS

TABLE 1. QCOM, AMD, NVDA's Descriptive Statistics

	QCOM	AMD	NVDA
Count	1323	1323	1323
Mean	121,513	86,976	23,129
Std	32,511	39,188	21,009
Min	53,530	22,010	3,345
25%	91,205	55,405	9,248
50%	124,290	85,370	16,174
75%	143,175	108,360	27,507
Max	213,080	211,380	114,824

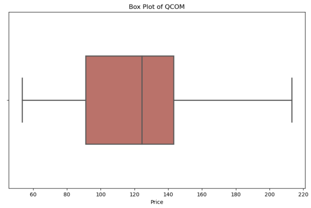

FIGURE 1. QCOM stock price's boxplot

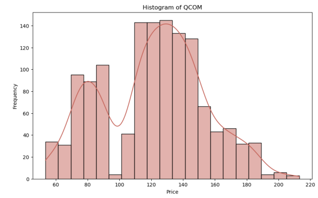

FIGURE 2. QCOM stock price's histogram

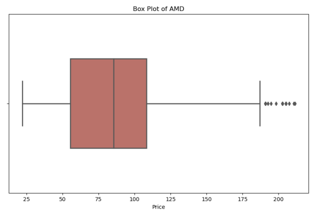

FIGURE 3. AMD stock price's boxplot

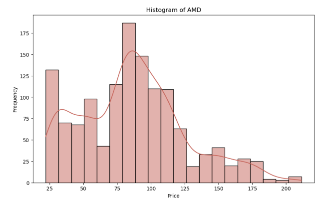

FIGURE 4. AMD stock price's histogram

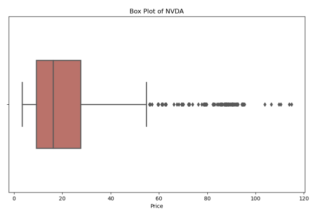

FIGURE 5. NVDA stock price's boxplot

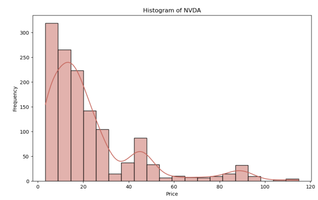

FIGURE 6. NVDA stock price's histogram

IV. METHODOLOGY

A. LINEAR REGRESSION

Regression analysis is a tool for building mathematical and statistical models that characterize relationships between a dependent variable and one or more independent, or explanatory, variables, all of which are numerical. This statistical technique is used to find an equation that best predicts the y variable as a linear function of the x variables. A multiple linear regression model has the form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon \tag{1}$$

Where [11]:

- Y is the dependent variable (Target Variable).
- $X_1, X_2, \ldots, X_k$ are the independent (explanatory) variables.
- β_0 is the intercept term.
- $\beta_1, ..., \beta_k$ are the regression coefficients for the independent variables.
- ε is the error term.

B. ARIMA

ARIMA stands for Autoregressive Integrated Moving Average Model. ARIMA is a forecasting algorithm based on the idea that the information in the past values of the time series can alone be used to predict the future values. The general model of ARIMA is written as follows:

$$\Delta(Y_t) = \phi_1\Delta(Y_{t-1}) + \phi_2\Delta(Y_{t-2}) + \cdots + \phi_p\Delta(Y_{t-p}) + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2}) + \cdots + \theta_q\epsilon_{t-q} \tag{2}$$

Where:
• $\Delta(Y_{t-i})$: is the difference value.
• ϵ_{t-i} : are white noise.

ARIMA is essentially a combination of two models:

• AutoRegression (AR) - Automatically regress a time-series value based on its lag value. The largest lag here (also known as the autoregressive order) is p.

• Integrated (I) - Is the order for the difference series to be stationary. We denote it as d.

• Moving Average(MA) - Degree of moving average. We denote it as q.

C. HOLT-WINTERS

The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations: one for a typical value (average) , one for the trend (slope) , and one for the seasonal component (a cyclical repeating pattern) , with corresponding smoothing parameters.

The Holt Winters' model further extends Holt's linear trend method by adding seasonality to the forecast. The addition of seasonality gives rise to two different Holt Winters' models, additive and multiplicative.

1) Holt-Winter's Additive Method:

This technique involves calculating the seasonal factor by comparing observed values with the average value of the crop cycle. It's applicable when the seasonal cycle's amplitude remains consistent over time.

$$\hat{y}_{t+h} = l_t + hb_t + S_{s+h-m} \tag{3}$$

$$l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \tag{4}$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \tag{5}$$

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \tag{6}$$

Where [9]:
• y_t :is the observed value at time T
• l_t :is the level estimate at time t
• b_t :is the trend estimate at time t
• s_t :is the seasonality estimate at time t
• $h = 1, 2, 3, ...$:h=1,2,3,... is the value at the time $t + h$ to be forecast
• m :is the length of the crop cycle
• h :is the number of time periods to forecast
• k :is the integer part of $(h-1)/m$, this index helps to determine the crop value at time h based on the previous crop value
• α :is smoothing parameter for level $(0 \leq \alpha \leq 1)$
• β :is smoothing parameter for trend $(0 \leq \beta \leq 1)$
• γ :is smoothing parameter for seasonality $(0 \leq \gamma \leq 1)$

2) Holt-Winter's Multiplicative Method:

This approach determines the seasonal factor by multiplying the observed value with the mean value of the crop cycle. It's effective for time series where the seasonal cycle's amplitude fluctuates over time.

$$\hat{y}_{t+h} = l_t + hb_t + S_{s+h-m} \tag{7}$$

$$l_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1} + b_{t-1} \tag{8}$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \tag{9}$$

$$s_t = \gamma\frac{y_t}{l_{t-1} + b_{t_1}} + (1-\gamma)s_{t-m} \tag{10}$$

D. RECURRENT NEURAL NETWORK (RNN)

Designed to process sequential data and time series data, a Recurrent Neural Network (RNN) maintains a memory of previous inputs to inform its current decisions.

An RNN is a type of artificial neural network characterized by its looped feedback connections. RNNs are called "repetitive" because they perform the same task for each sequence element by leveraging previously captured information sequentially to predict future unknown sequential data. [10]

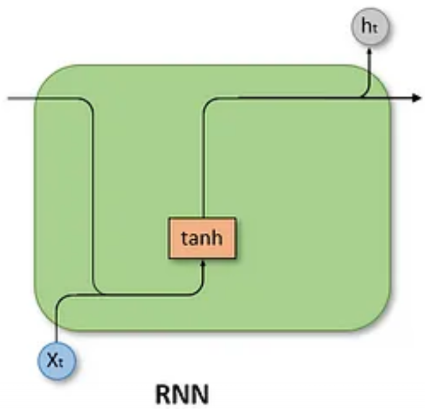

FIGURE 7. Structure of RNN

In RNN network systems, different time step lengths have the same one weight and can be cyclically linked across time. Due to the sharing of weights, the temporal parameters in the RNN network system are greatly reduced. [10]

One significant drawback of RNNs is their limited capacity to retain information from earlier steps in a sequence. This constraint makes them ineffective for managing and storing longer sequences of data.

E. GATED RECURRENT UNIT (GRU)

Building upon the LSTM model, the GRU model simplifies the architecture by combining the forget and input gates into a single update gate. It also forgoes the cell state, leading to a reduction in the number of parameters. This design addresses the vanishing gradient problem.

A GRU unit is composed of reset gate and update gate, due to the simpler architecture, it is contributes to train faster and search optimal solution easily. [14]

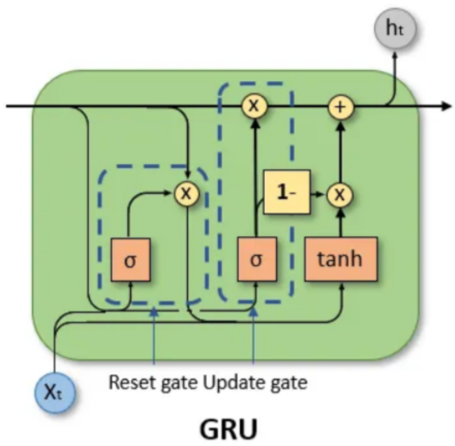

FIGURE 8. Structure of GRU

In the first step, reset gate is calculated using both the hidden state from the previous time step and the input data at the current time step, it be reserved by applying a sigmoid function σ, as expressed in the equation [14]

$$r_t = \sigma(W_r \times x_t + U_t \times h_{t-1}) \tag{11}$$

Where:

- x_t is input data at the current time step.
- h_{t-1} is the hidden state from the previous time step.
- W_r and U_r are the weighting vectors respectively.

Next, determine the data to retain from prior time steps along with the new inputs. This can be represented mathematically as follows:

$$\tilde{h_t} = tanh(W_h \times x_t + U_h \times (r_t \times h_{t-1})) \tag{12}$$

Second, the update gate is derived by combining the preceding hidden state with the current input data, utilizing a similar formula to the one used for the reset gate. However, the weights applied to the input and hidden state are distinct and specific to each gate. As a result, the final vectors for the update gate are unique and differ from those of the reset gate, as demonstrated in the following equation:

$$z_t = \sigma(W_r \times x_t + U_t \times h_{t-1}) \tag{13}$$

Next, summed with the output, which is from the update gate multiplied by the candidate hidden state, as expressed in equation:

$$h_t = z_t \times h_{t-1} + (1 - z_t) \times \tilde{h_t} \tag{14}$$

F. LONG SHORT-TERM MEMORY (LSTM)

LSTM is an improved version of RNN designed to address the issue of long-term memory. It avoids long-term dependency problems and inherently retains long-term information without requiring additional training.

The LSTM model filters information through the gate structure to maintain and update the state of memory cells. Its door structure includes input, forgotten, and output gates. Each memory cell has three sigmoid layers and one tanh layer [13]

LSTM marks a major step forward in RNN usage by allowing each step to access broader contextual information

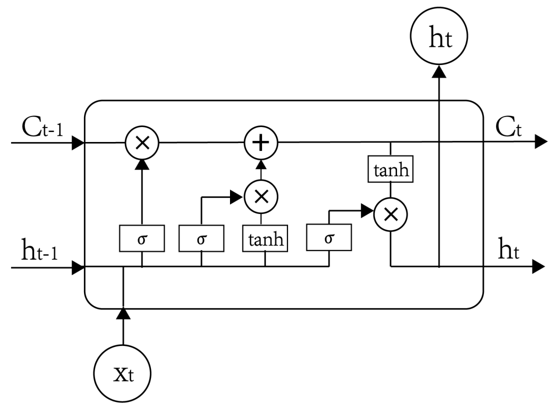

FIGURE 9. Structure of LSTM

- **Sigmoid function**:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{15}$$

- **Tanh activation function**:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{16}$$

- **Forget gate:**:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{17}$$

- **Input gate**:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C} \tag{18}$$

- **Calculation methods of** :

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{19}$$

- **Calculation methods of**:

$$\tilde{C} = tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{20}$$

- **Output gate**:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{21}$$

$$h_t = O_t * tanh(C_t) \tag{22}$$

Where:

- h_{t-1} is the hidden state from the previous time step.
- x_t is the input at time step t
- C_t is the cell state at time step t

- $\tilde{C}$ is the new information proposed to be added to the cell state.
- f_t, i_t, o_t are the decision gates (forget gate, input gate, output gate).
- W_f, W_i, W_c, W_o and b_f, b_i, b_c, b_o are the parameters of the LSTM model trained to adjust how gates and information are computed.

G. GRU-LSTM BOOSTING

Ensemble learning refers to a group (or ensemble) of base learners or models, which work collectively to achieve a better final prediction.

Boosting method is a main type of ensemble learning methods. It is a combination of models, but models are not completely independent, they have a chain dependency. That is, a model is developed using information predicted from previously trained models.

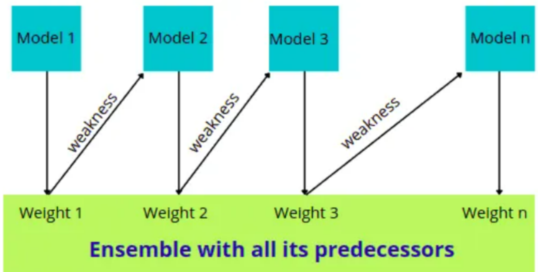

FIGURE 10. Boosting model

In this research, we use the GRU model and train this model with the train data set and let it predict this data set again. Then, we use these predictions as the training set for the LSTM model, the LSTM model will predict on the test data set. These predictions are compared with the original test data set to make an assessment.

H. MLP

A Neural Network's architecture is inspired by brain neurons, with layers interconnected by nodes called neurons [15]. Signals are transmitted between neurons similarly to synapses in the brain. A Multilayer Perceptron (MLP) consists of an input layer, one or more hidden layers, and an output layer. Neurons in each layer receive inputs from the previous layer, transformed by a weight matrix and bias, and apply nonlinear activation functions. The output of a three-layer MLP can be expressed mathematically by the equation [16]:

$$O_{MLP} = f^o[\textstyle\sum_{j=0}^{J} W_{pj} \times f^h(\sum_{i=0}^{I} W_{ji} X_i + \xi_j^h) + \xi_p^o] \tag{23}$$

Where:

- i, j, p are the indexes of input, hidden, and output layers neurons.
- $f^o \& f^h$ are the activation functions in output and hidden layers.
- $I \& J$ are the numbers of input layer and hidden layer neurons.
- X_i is the input matrix.
- W_{pj} is the weight matrix connecting the jth neuron of the hidden layer and the pth neuron of the output layer.
- W_{ji} is the weight matrix connecting the ith neuron of the input layer and the jth neuron of the hidden layer
- ξ_j^h is the bias matrix of the jth neuron in the hidden layer.
- ξ_p^o is the bias matrix of the pth neuron in the output layer.

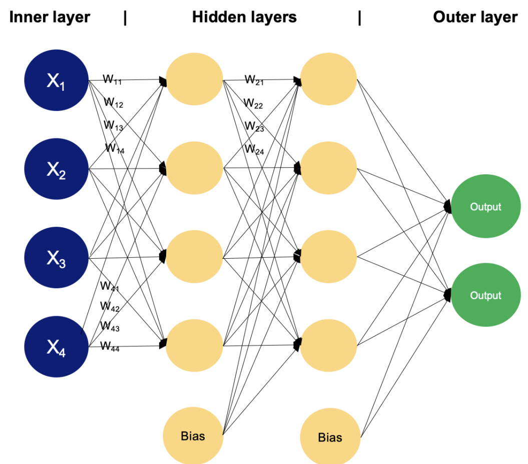

FIGURE 11. Structure of Multilayer Perceptron

I. NADAM

Nadam (Nesterov-accelerated Adaptive Moment Estimation) is an optimization algorithm that extends the Adam optimizer by incorporating the Nesterov momentum [12]. This combination aims to improve convergence speed and predictive accuracy, especially in complex neural network architectures.

- **Compute the moving averages of the gradient and the squared gradient**:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{24}$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{25}$$

- **Compute bias-corrected estimates**:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{26}$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{27}$$

- **Compute the predicted gradient (Nesterov)**:

$$\hat{m}_{t+1} = \beta_1 \hat{m}_t + \frac{(1 - \beta_1) g_t}{1 - \beta_1^{t+1}} \tag{28}$$

- **Update weights**:

$$\theta_t = \theta_{t-1} - \frac{\eta \hat{m}_{t+1}}{\sqrt{\hat{v}_t} + \epsilon} \tag{29}$$

Where:

- m_t and v_t are the moving averages of the gradient and the squared gradient, respectively.
- β_1 and β_2 are hyperparameters (typically 0.9 and 0.999).
- η is the learning rate.
- ϵ is a small value to avoid division by zero (typically $1e-8$).
- g_t is the gradient at time step t.
- θ_t is the value of the weights at time step t.

V. RESULT

A. EVALUATION METHODS

Mean Percentage Absolute Error (MAPE): is the average percentage error in a set of predicted values.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - \hat{y}_i}{y_i}| \tag{30}$$

Root Mean Squared Error (RMSE): is the square root of average value of squared error in a set of predicted values.

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}} \tag{31}$$

Mean Absolute Error (MAE):measures the average of the errors' magnitude between the predicted and actual values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{32}$$

Where: • n is the number of observations in the dataset.
• y_i is the true value for the ith data point.
• $\hat{y}_i$: Predicted value for the ith data point

B. QCOM DATASET

TABLE 2. QCOM Dataset's Evaluation

QCOM Dataset's Evaluation				
Model	Proportion	RMSE	MAPE (%)	MAE
Linear Regression	7:3	58.01	45.09	55.48
	8:2	36.06	26.05	31.73
	9:1	20.59	9.52	15.98
Holt-Winters	7:3	18.76	10.67	14.38
	8:2	35.38	17.03	26.60
	9:1	31.07	14.88	25.44
ARIMA	7:3	81.16	55.50	72.73
	8:2	195.40	113.23	166.86
	9:1	54.46	28.61	47.53
RNN	7:3	3.01	1.70	2.27
	8:2	3.06	**1.46**	2.24
	9:1	4.37	1.64	3.02
GRU	**7:3**	**2.98**	1.64	**2.21**
	8:2	3.06	1.48	2.27
	9:1	4.19	1.71	3.12
LSTM	7:3	3.83	2.18	2.96
	8:2	3.73	1.90	2.88
	9:1	4.75	1.90	3.49
GRU_LST (Boosting)	7:3	3.20	1.82	2.40
	8:2	3.58	1.76	2.69
	9:1	4.35	1.66	3.05
MLP	7:3	5.13	3.00	4.07
	8:2	5.49	3.32	4.56
	9:1	4.10	1.96	3.16
MLP (Nadam optimization)	7:3	4.75	2.71	3.68
	8:2	4.18	2.29	3.19
	9:1	3.90	1.86	2.96

Based on the above results table, the lowest RMSE and MAE evaluation coefficients at the GRU model and the ratio 7:3. The lowest MAPE evaluation coefficient at the RNN model and the ratio 8:2. We will use these models to forecast the next 90 days of the dataset.

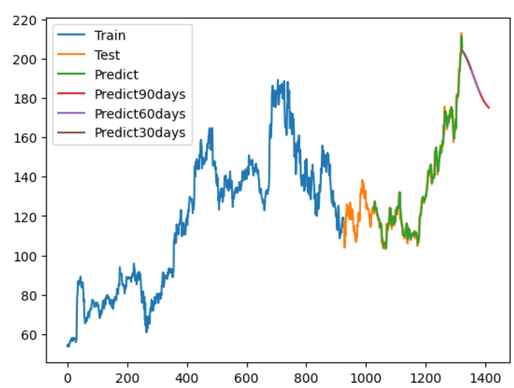

FIGURE 12. GRU model's result with 7:3 splitting proportion

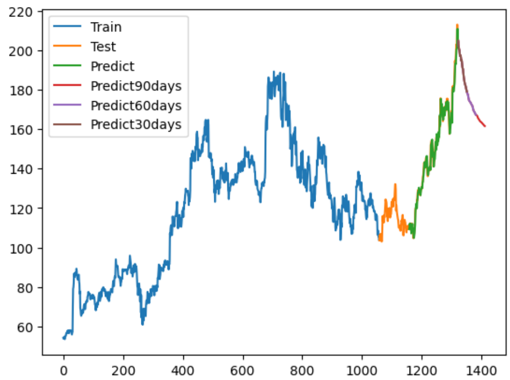

FIGURE 13. RNN model's result with 8:2 splitting proportion

C. AMD DATASET

TABLE 3. AMD Dataset's Evaluation

AMD Dataset's Evaluation				
Model	Proportion	RMSE	MAPE (%)	MAE
Linear Regression	7:3	34.67	32.66	30.79
	8:2	30.39	14.80	22.60
	9:1	43.98	22.82	38.87
Holt-Winters	7:3	58.76	36.98	48.77
	8:2	39.29	19.26	29.97
	9:1	40.57	20.58	35.24
ARIMA	7:3	52.22	32.43	42.92
	8:2	39.45	25.49	33.33
	9:1	59.58	26.68	42.73
RNN	7:3	4.19	2.30	3.05
	8:2	4.70	2.19	3.33
	9:1	4.53	**2.09**	3.27
GRU	**7:3**	**4.12**	2.26	**2.95**
	8:2	4.89	2.28	3.47
	9:1	4.73	2.22	3.44
LSTM	7:3	5.22	2.96	3.92
	8:2	5.26	2.65	3.94
	9:1	4.84	2.34	3.63
GRU_LSTM (Boosting)	7:3	4.40	2.41	3.19
	8:2	5.00	2.47	3.67
	9:1	4.75	2.27	3.53
MLP	7:3	6.66	4.04	4.98
	8:2	5.29	2.98	4.06
	9:1	10.23	5.30	8.71
MLP (Nadam optimization)	7:3	5.99	3.72	4.50
	8:2	5.19	2.91	3.97
	9:1	7.19	3.58	5.84

Based on the above results table, the lowest RMSE and MAE evaluation coefficients at the GRU model and the ratio 7:3. The lowest MAPE evaluation coefficient at the RNN model and the ratio 9:1. We will use these models to forecast the next 90 days of the dataset.

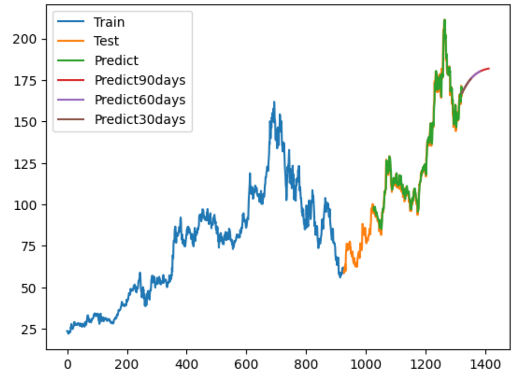

FIGURE 14. GRU model's result with 7:3 splitting proportion

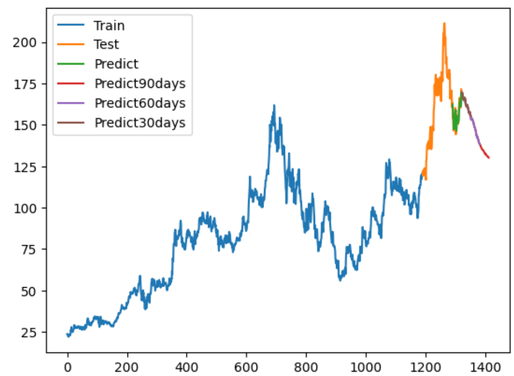

FIGURE 15. RNN model's result with 9:1 splitting proportion

D. NVDA DATASET

TABLE 4. NVDA Dataset's Evaluation

NVDA Dataset's Evaluation				
Model	Proportion	RMSE	MAPE (%)	MAE
Linear Regression	7:3	28.02	38.80	20.51
	8:2	36.01	47.83	30.52
	9:1	41.04	47.02	36.99
Holt-Winters	7:3	38.24	55.85	30.31
	8:2	32.64	40.49	26.61
	9:1	27.83	26.51	22.51
ARIMA	7:3	17.75	19.15	11.70
	8:2	22.63	25.65	17.37
	9:1	7.50	9.97	6.60
RNN	7:3	2.21	2.94	1.60
	8:2	2.49	2.42	1.71
	9:1	3.56	2.99	2.74
GRU	**7:3**	**2.00**	2.66	**1.43**
	8:2	2.31	**2.25**	1.60
	9:1	3.57	3.00	2.76
LSTM	7:3	2.53	3.58	1.93
	8:2	2.47	2.56	1.76
	9:1	3.38	2.78	2.53
GRU_LSTM (Boosting)	7:3	2.28	2.86	1.61
	8:2	2.43	2.54	1.75
	9:1	3.65	3.00	2.77
MLP	7:3	2.79	4.28	1.92
	8:2	4.79	5.62	3.65
	9:1	3.58	3.61	2.72
MLP (Nadam optimization)	7:3	2.18	3.26	1.47
	8:2	4.14	5.09	3.18
	9:1	3.36	3.32	2.46

Based on the above results table, the lowest RMSE and MAE evaluation coefficients at the GRU model and the ratio 7:3. The lowest MAPE evaluation coefficient at the GRU model and the ratio 8:2. We will use GRU model to forecast the next 90 days of the dataset.

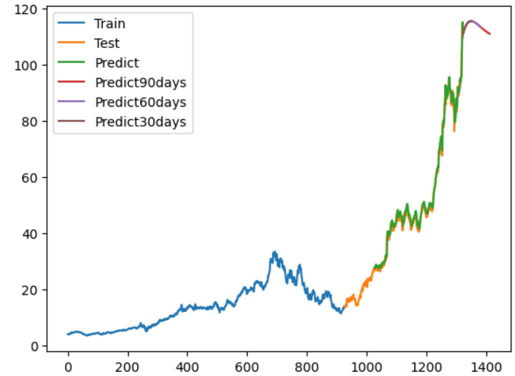

FIGURE 16. GRU model's result with 7:3 splitting proportion

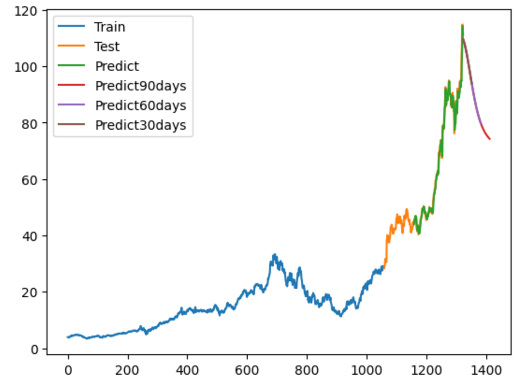

FIGURE 17. GRU model's result with 8:2 splitting proportion

E. LOSS FUNCTION

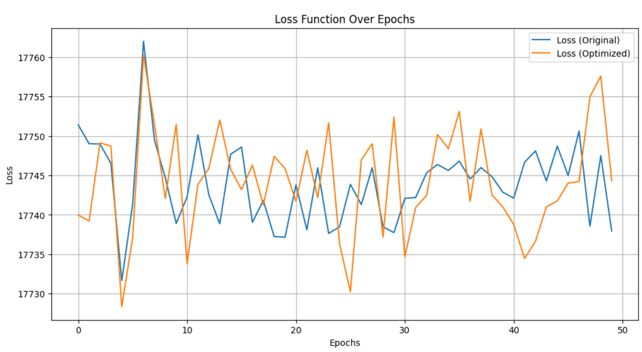

FIGURE 18. Loss function proportion 7:3 QCOM dataset

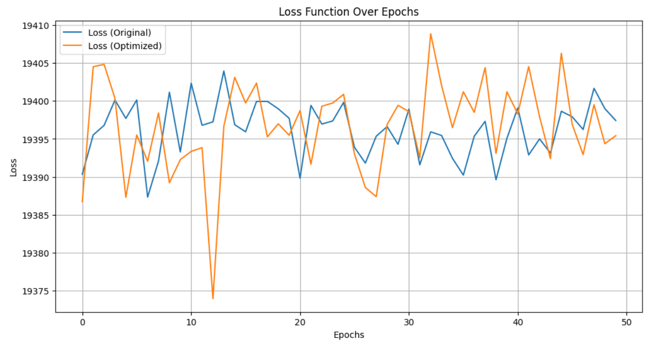

FIGURE 19. Loss function proportion 8:2 QCOM dataset

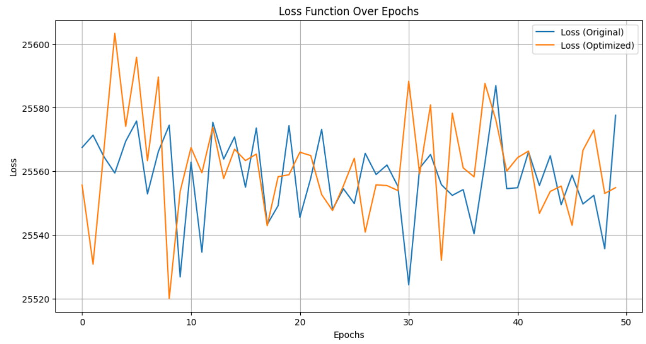

FIGURE 20. Loss function proportion 9:1 QCOM dataset

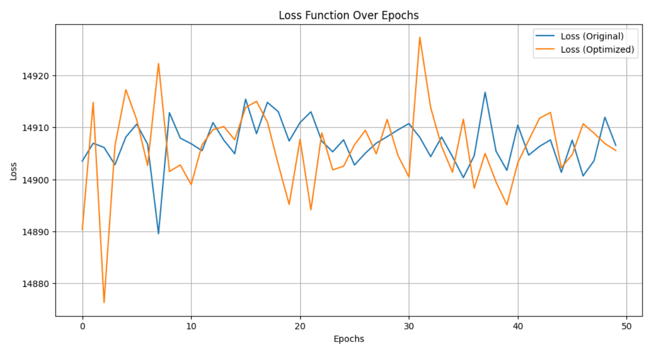

FIGURE 21. MLP loss function proportion 7:3 AMD dataset

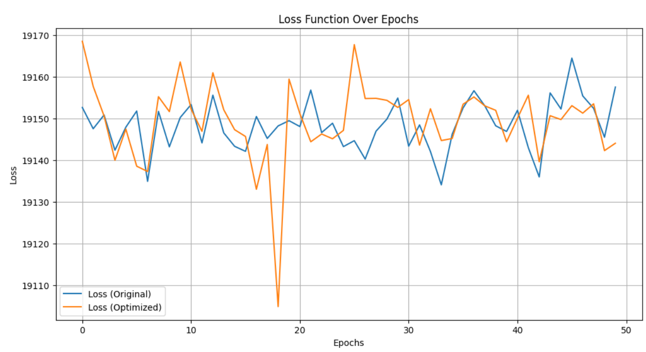

FIGURE 22. Loss function proportion 8:2 AMD dataset

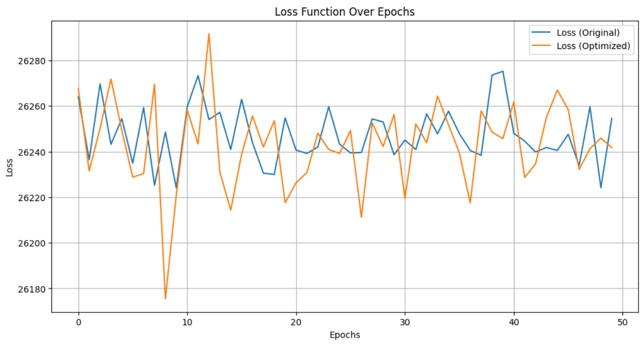

FIGURE 23. Loss function proportion 9:1 AMD dataset

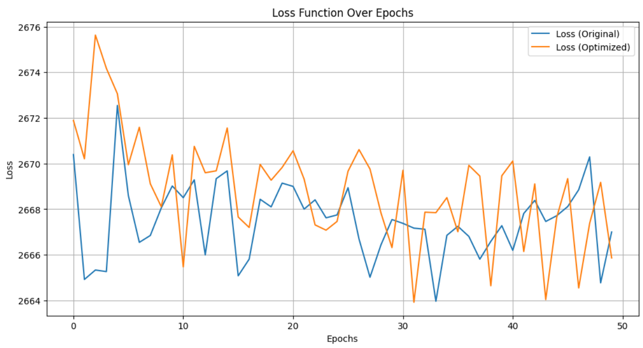

FIGURE 24. Loss function proportion 7:3 NVDA dataset

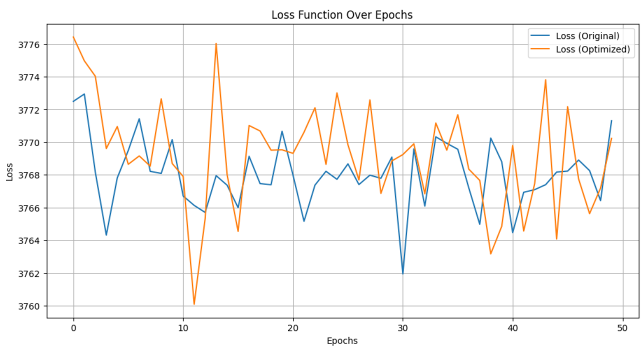

FIGURE 25. Loss function proportion 8:2 NVDA dataset

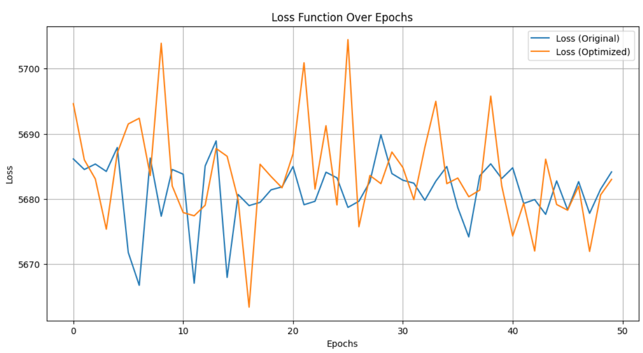

FIGURE 26. Loss function proportion 9:1 NVDA dataset

Overall, the loss functions of the MLP model before and after optimization still exhibit fluctuations, with the models tending to fluctuate more in the initial epochs before gradually stabilizing.

Furthermore, datasets split with a 9:1 ratio generally exhibit higher loss functions compared to splits with 7:3 and 8:2 ratios. Among the datasets, the NVDA dataset demonstrates the lowest loss function, fluctuating between 2000 and 6000. In comparison, the AMD dataset fluctuates between 14000 and 27000, while the QCOM dataset fluctuates between 17000 and 26000.

Based on the above observations, it is evident that the model has significant potential for further development through better parameter tuning and increased training iterations.

VI. CONCLUSION

A. SUMMARY

The stock market experiences significant volatility, with prices changing frequently due to various external and internal factors. As a result, creating an accurate stock price forecasting system using machine learning models has gained popularity and importance. In this study, we employed eight models, ranging from statistical models and machine learning to deep learning, to forecast the stock prices of Computer processor manufacturing companies. Additionally, we applied the Nadam optimization model to the MLP deep learning model and used a boosting method with LSTM-GRU to enhance prediction accuracy. To evaluate the performance of each model, we used the evaluation metrics Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). Our findings indicate that while traditional statistical models offer a foundational understanding of trends, machine learning and deep learning models excel at identifying intricate patterns and temporal dependencies in stock price data. By combining the strengths of each approach, the ensemble of models creates a more robust and accurate prediction framework.

While the above algorithms have demonstrated promising results, applying machine learning or deep learning in this domain presents several challenges. Factors influencing stock prices, including financial markets, political and social dynamics, and a company's financial performance, add complexity to the prediction task. To address these challenges, a practical approach involves capitalizing on each method's strengths while mitigating their weaknesses. Specifically, we fine-tuned the foundational parameters of statistical models, such as seasonality and trend, to better adapt to stock price fluctuations. Additionally, we employed a synthetic learning technique called boosting to integrate LSTM and GRU, enhancing the effectiveness of machine learning. Furthermore, for the MLP model, we implemented Nadam Optimization to optimize its performance, as evidenced by the evaluation coefficients.

B. FUTURE CONSIDERATIONS

For our upcoming plans, prioritizing further optimization of the previously mentioned models is crucial:

• Improving Model Accuracy: Despite promising results, continuous adjustment and refinement of model parameters are essential for enhancing accuracy. Exploring untested scenarios and meticulously documenting adjustments will provide a broader perspective and foster greater precision.

• Exploring Additional Machine Learning Techniques: Expanding our repertoire of machine learning models and techniques is pivotal. For instance, leveraging ensemble learning approaches, such as boosting techniques applied between LSTM and GRU, has demonstrated success. Future efforts will focus on testing and integrating other advanced techniques to further elevate predictive capabilities.

• Deepening Understanding of Stock Dataset Characteristics: Recognizing that relying solely on closing stock prices may limit practical applicability, we plan to delve deeper into understanding the nuances of the stock dataset. This involves investigating how external factors such as political events and economic indicators influence stock performance beyond closing values.

By exploring new features, data sources, and techniques, we aim to continually optimize our forecasting models for more precise and reliable stock price predictions.

Furthermore, we intend to deploy our current models in an online application to showcase a straightforward stock price forecasting project. This application will streamline the implementation process for our aforementioned upcoming plans.

ACKNOWLEDGMENT

First and foremost, we extend our sincere gratitude to **Assoc. Prof. Dr. Nguyen Dinh Thuan** and **Mr. Nguyen Minh Nhut** for their exceptional guidance, expertise, and invaluable feedback throughout the research process. Their mentorship and unwavering support have been pivotal in shaping the direction and quality of this study. Their profound knowledge, critical insights, and meticulous attention to detail have greatly contributed to the success of this research.

REFERENCES

[1] Dinesh Bhuriya, Girish Kaushal, Ashish Sharma, and Upendra Singh "Stock market predication using a linear regression" . In 2017 international conference of electronics, communication and aerospace technology (ICECA), volume 2, pages 510–513. IEEE, 2017.

[2] A. Subakkar, S. Graceline Jasmine, L. Jani Anbarasi, J. Ganesh, and C. Yuktha Sri. "An analysis on tesla's stock price forecasting using ARIMA model," in Proceedings of the International Conference on Cognitive and Intelligent Computing: ICCIC 2021, Volume 2, pp. 83–89. Springer, 2023.

[3] Vivek Varadharajan, Nathan Smith, Dinesh Kalla, Ganesh R Kumar , Fnu Samaah, Kiran Polimetla "Stock Closing Price and Trend Prediction with LSTM-RNN" Journal of Artificial Intelligence and Big Data, 2024.

[4] Nrusingha Tripathy, Surabi Parida, Subrat Kumar Nayak "Forecasting Stock Market Indices Using Gated Recurrent Unit (GRU) Based Ensemble Models: LSTM-GRU" International Journal of Computer and Communication Technology - IJCCT July 2023.

[5] C S Agustina , T Asfihani , R R Ginting , and S Subchan. "Model predictive control in optimizing stock portfolio based on stock prediction data using Holt Winters exponential smoothing" Journal of Physics: Conference Series - ICOMPAC 2020.

[6] Xinwen Xu. "Research on the Stock Price Forecasting of Netflix Based on Linear Regression, Decision Tree, and Gradient Boosting Models" Proceedings of the 2022 2nd International Conference on Business Administration and Data Science (BADS 2022).

[7] Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting" Part of Advances in Neural Information Processing Systems 34 (NeurIPS 2021)

[8] Jatin Sharma, Sameer Soni, Priyanka Paliwal, Shaik Saboor, Prem K. Chaurasiya, Mohsen Sharifpur, Nima Khalilpoor, Asif Afzal. "A novel long term solar photovoltaic power forecasting approach using LSTM with Nadam optimizer: A case study of India". Energy Science & Engineering Volume 10, Issue 8 Aug 2022 Pages 2577-3215.

[9] Rob J Hyndman, George Athanasopoulos. "Holt-Winters' seasonal method". Forecasting: principles and practice Chapter 7.3 Pages 257 - 263 Monash University, Australia Public May 2018.

[10] L. Lin, "Stock Prediction and Analysis based on RNN Neural Network," SHS Web Conf., vol. 151, p. 01002, Dec. 2022, doi: 10.1051/shsconf/202215101002.

[11] M. Tranmer and M. Elliot, "Multiple Linear Regression," The Cathie Marsh Centre for Census and Survey Research (CCSR) 2008.

[12] Sebastian Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016.

[13] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long- short term memory neural network based on attention mechanism," PLOS ONE, vol. 15, p. e0227222, Jan. 2020, doi: 10.1371/journal.pone.0227222.

[14] C.-J. Chen, F.-I. Chou, and J.-H. Chou, "Temperature Prediction for Reheating Furnace by Gated Recurrent Unit Approach," IEEE Access, vol. 10, pp. 33362–33369, 2022, doi: 10.1109/ACCESS.2022.3162424.

[15] Ecer, F. Artificial Neural Networks in Predicting Financial Performance: An Application for Turkey's Top 500 Companies. Econ. Comput. Econ. Cybern. Studies Res. 2013, 47, 103–114

[16] Ghorbani, M. A., Deo, R. C., Karimi, V., Kashani, M. H., Ghorbani, S.(2019). Design and implementation of a hybrid MLP-GSA model withmulti-layer perceptron-gravitational search algorithm for monthly lakewater level forecasting. Stochastic Environmental Research and RiskAssessment, 33, 125-147.