Title

MSc Research Project

MSc in Data Analytics

Ganta Satish Kumar

Student ID: X22238409

School of Computing

National College of Ireland

Supervisor:Christian Horn

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Ganta Satish kumar		
Student ID:	x22238409		
Programme:	MSc in Data Analytics	**Year:**	2023-2024
Module:	Research Project ……………………………………………………………………………… …………………………………………		
Supervisor:	Christian Horn ……………………………………………………………………………………… …………………………………………		
Submission Due Date:	16-09-2024		
Project Title:	Enhancing Next-Day Stock Price Prediction Accuracy and Reliability: A Comparative Study of Bi-GRU, Transformer, and Hybrid Models.		
Word Count:	**7049 Page Count 26**		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	SATISH KUMAR GANTA
Date:	16-09-2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Next-Day Stock Price Prediction Accuracy and Reliability: A Comparative Study of Bi-GRU, Transformer, and Hybrid Models

Satish Kumar Ganta
x22238409
Research Project in MSc Data Analytics

Abstract

This research project is an attempt at enhancing the accuracy and performance of next-day stock price predictions using various machine learning models. In this paper, we look at how well Linear Regression, Bidirectional Gated Recurrent Unit, Neural Networks, and Transformer models do against stocks taken from Google historical data from January 2015 to December 2023. We did extensive experimentation and found out that a Linear Regression model was actually the most fitting one when compared to the large capacity models such as Bi-GRU and Transformer. Evaluations were based on the mean squared error and mean absolute error. The results indicated that, on the one hand, more complex models do have the ability to capture nonlinear trends very well, but on the other hand, less complicated models perform much better if there is a strong presence of linearity in data. This research contributes to the current body of literature and continuing research on financial forecasting since it gives insights into when some of these different machine-learning approaches have relative strengths and not only that—this work also emphasises model selection based on the characteristic of data.

1 Introduction

1.1 Background

Stock price prediction is very important in financial industries, since it helps investors to make very viable and quite effective decisions that will go a long way in determining their financial results. Traditional approaches to forecasting, based on linear assumptions, usually fail to capture the complex nonlinear dynamic behavior inherent in financial time series data. This has raised interest in more sophisticated machine learning models that can handle such complexities.

1.2 Research Objective

This study will be used to estimate and compare the performance of various machine learning models in predicting the next day's stock prices. Linear Regression, Bidirectional Gated Recurrent Unit, Neural Networks, and Transformer models are used in this research. All these above-mentioned models have different time series forecasting styles; therefore, through this research, it will find which model gives the best predictions based on historical stock data taken from GOOGL.The historical stock data was sourced from Yahoo Finance covering the period from January 2015 to December 2023.

1.3 Research Questions

These are then the research questions that this study hopes to answer:

1. How well do the simple models such as Linear Regression perform compared to complex models like Bi-GRU, Neural Networks, and Transformers in predicting next-day stock prices?
2. What are the conditions under which simpler models would be more welcomed than the complex ones in the prediction of stock prices?
3. How do different types of model architectures handle these non-linearities and temporal dependencies of the financial time series data?

1.4 Contribution to the Literature

The research work adds to the already existing literature in financial forecasting, with a relative comparison of various machine learning models within the realm of stock price prediction. While a great deal of the literature has bent over backward to tout the superiority of complex models, this research dwells on situations under which simpler ones may be better. Findings from this research shall aim to enlighten both the academic and professional in finance in respect of the pros and limitations of each of these approaches to modeling so that better and more effective choices and applications of models yield improved financial forecasting.

1.5 Report Structure

In this report, the subject has been organised as follows:

Section 2: Related Work - A review of the literature that already exists on stock price prediction using machine learning models.
Section 3: Research Methodology - The methods of data collection, model development, and evaluation criteria are expounded in fine details.
Section 4: Design Specification - This provides a design outline of the various architectures and frameworks used along with model designs.
Section 5: Implementation - It discusses the techniques of implementation with the corresponding tools and languages used.
Section 6: Evaluation - A performance appraisal of the models, supported by relevant statistical tools and representation through visual aids.
Section 7: Conclusion and Future Work - Summary of findings, discussion of implications, suggestions for future research directions.

2 Related Work

The research in the area of predicting stock prices has advanced with the aid of machine learning models. This section discusses some major studies in stock price prediction schemes using various models such as Bidirectional Gated Recurrent Units, Transformer models, hybrid models, or ensemble models to increase the level of accuracy and robustness.

2.1 Recurrent Neural Networks (RNNs) and Variants

Duan et al. (2023) enhanced the model to forecast stock prices, denoted as BiGRU, which outperformed the traditional models. These can be regarded as an illustration of the model's capacity to learn such complex time dependencies in data sequences of financial time series. Being able to process information from both the forward and backward directions, it correctly captures some of those hidden patterns in the data streams of stock price movements.

Xu et al. (2023) in this line pointed out a deep heuristic evolutionary regression model when they merged the architectures of BiGRU and BiLSTM to predict stock prices. This hybrid solution, hence, worked toward overall improvements in predictive accuracies coming from this model, due to the incorporation of both BiGRU and BiLSTM in the learning of long-term temporal dependencies. According to their research, the highly advanced variants of RNNs used to deal with the sequential nature of stock price data—which most of the time has a nonlinear and dynamic characteristic—provide high sensitivity when detecting changes.

In one other application, Karim and Ahmed (2021) used a deep learning-based BiGRU and BiLSTM method for forecasting stock prices. Their work tackled the main advantage of bidirectional models, able to capture both past and future contexts within the time series, enhancing the accuracy of the forecasting. In much recent work, the integration of techniques from these RNN variants with traditional methods has been considered with respect to this challenge, having issued that more promising results were achieved than it would if used with any strategy.

Victor (2024) infused an element of sentiment analysis in the model meant for the prediction of stock prices. He used a BERT-BiGRU-A model that integrated the power of BERT in understanding languages more efficiently and BiGRU for the sequence. The model shown below would be useful since it predicts the trend of stock price from the mood analysis of social media users together with other social influences, thus leaving the window open for the data source integration that would aid in further predictive accuracy.

The long memory approach to stock price indices prediction was used by Mao and Wu (2023) with the model SSA-BiGRU-GSCV. To improve its precision, the authors combined the techniques of Singular Spectrum Analysis, BiGRU, and Grid Search Cross Validation. Fusing these methods actually greatly improved BiGRU models for making long-term financial predictions.

2.2 Transformer Models

Costa and Machado (2023) researched the Transformer architecture in time series stock price forecasting. Results showed that the model was very effective at the tasks and, especially, is good at capturing long-range dependencies, which are very critical in financial markets. They also pointed out subtleties and raised a warning with the implementation of Transformers for the purpose at hand: it changes how this all is obtained through careful tuning and adaptation to the idiosyncrasies of financial data.

Even more interesting are some additional application experiments of Transformers through attention networks that are offered by Li et al. (2022) for stock movement prediction. Their research demonstrated the intuition that the attention mechanism underlying the Transformer is able to capture the complex dynamics in the market governing stock price changes. Through the use of the Transformer model on attention for various parts of the input sequence, more thorough and precise predictions could be made—thus being in its highest class in running financial forecasting tasks.

Findings of the efficiency of the transformers by Zeng et al. (2023), however, while testing on generated time series data, showed mixed results. The research has gone a long way in probing the appropriateness of Transformers within different forecasting scenarios, picking their strengths and weaknesses. The model did very well in some contexts but very poorly in others, almost implying further fine-tuning on or customisation of Transformers in certain specific financial forecasting applications.

Ramos-Pérez et al. (2021) elaborated on their Multi-Transformer architecture, designed for high-precision prediction for the volatility of S&P stocks. In their study, they showed that such fine-tuning could be applied to some financial tasks, especially the prediction of volatility, where the capture of intricate patterns is needed.

Temporal Fusion Transformer in stock price prediction was applied by Hu (2021). There is the ability of the architecture to capture temporal dependencies across all time scales. The TFT model, adopted as applied by Hu, turned out to be very decisive in holding both short-term and long-term temporal relationships, which constitute high ranks in constancy and are prerequisites in accurate forecasting of the movement of stock prices. This finding serves to highlight further the utility of advanced Transformer models in financial forecasting, particularly with relation to treacherous and highly complex time-series data.

2.3 Hybrid and Ensemble Models

Kwon and Moon (2007) were among the very first to consider hybrid modeling in stock prediction. They proposed one of the thorough attempts embodying a neuro-genetic approach—in essence, a hybrid of a neural network with genetic algorithms. They demonstrated that the structure of the neural network with regards to financial prediction tasks can be optimised by evolutionary techniques. Moreover, it opened up the way for further research in the coupling of multiple techniques with course modification to further enhance the prediction accuracy.

Wang et al. (2012) developed an ensemble model for the use of multiple indices in stock-based forecasting, which demonstrated that the prediction accuracy can be substantially improved by model ensembling, and in this way, potential errors in forecasting can be minimised. Their study noted the convenience of ensemble methods in combining the features of models to produce more robust and reliable forecasts.

In another related study, Araujo et al. (2015) designed several hybrid models explicitly for this purpose so as to design the high-frequency prediction of the stock market. The study yielded a model blending different artificial intelligence methods in a way that better results could be derived from it. The results showed that hybrid models were suited to handle some of the individual statistics for trading with those frequencies, which raised the importance dramatically high for speed and accuracy in prediction.

Chong et al. (2015) further developed the hybrid models by consecutive combinations of several artificial intelligence techniques in the forecasting of stock prices. Their study provided firm evidence that ensemble methods for drawing on the strengths of several models give way to improved robustness of predictions with accurate results, making them indispensable tools for financial forecasting purposes.

Introducing a fuzzy random auto-regression time series model in the prediction of stock markets, Efendi et al. (2018) indicated that this characteristic effectively captures both uncertainty and randomness that are typical of all financial markets. The research thus proved that this model is therefore much capable of pointing at subtle changes that take place in the financial stock markets, except in extreme situations, that are found in the financial markets.

Pai et al. (2010) on the capturing of seasonality within time series data fell on the usage of a seasonal support vector regression model. Their findings found that indeed correct capturing of seasonality is essential for the best financial forecasting since it might have very adverse effects on the performance of prediction models. Their finding emphasised that domain-specific knowledge, like seasonality, can and is required to be in-built into machine learning models to make them more perfect.

Summary

They together represent a significant thrust in the application of machine learning models for stock price prediction. The common theme lies within its potential through the use of engineered models typical, respectively, of the class of models that BiGRU and Transformers belong to, being in capturing the complex dependencies evident in the time dimension and in the market topography. It also points out that the hybrid and ensemble type of models—very similar to the architecture setup to those presented—led to improvements in accuracy in the predictions produced. Building on these therefore, the paper provides more insights into where some of these models have strength or limitation within the financial forecasting domain.

3 Research Methodology

3.1 Data Collection

This research is based on data collected for the historical daily stock prices of Google, GOOGL, covering the period from January 1, 2015, to December 31, 2023. The dataset was retrieved from Yahoo Finance, one of the dependable financial information websites. These variables are applied in recording the opening price, closing price, high price, low price, and trading volume for each day of trade.

Such variables, therefore, hold a holistic view of the stock's behavior during this period and are suitable for building predictive models.

3.2 Data Preprocessing

Data preprocessing in time series forecasting is of prime importance because the performance depends a great deal on the quality of input data. The following steps have been taken for this purpose:

Data Cleaning: Checked the dataset for missing, duplicated, or anomalous entries. Missing values, which may bias the prediction of models, are treated by interpolation, if possible. In cases of large gaps, affected rows were removed.

Feature Engineering: Lag features in a time series forecast are quite important as they give historical context to the model. In this research, the lag features were engineered for the past 20 days, so this would give the model a prediction of the price of the next day from that of the previous 20 days' stock prices. They help identify short-term patterns or trends that are very critical in financial markets.

Data Scaling: All of the input features should be on comparable scales and also aid in training efficiency. In this work, MinMaxScaler was used for normalisation. This will scale the data to a range between 0 and 1, making it more suitable, especially when neural networks are being used. Scaling makes the model converge faster and avoids potential problems arising from different feature scales.

Once model training and prediction were done, the results were rescaled back to the original scale by the inverse transformation from MinMaxScaler, which would lead to meaningful interpretation among the predictions.

Chronological Split: Due to the time series nature of the dataset, the data was chronologically split. This means using the first 80% of data points to train a model and the rest, namely the last 20%, for testing. This is basically a simulation of real-world scenarios in which models trained on historical data are used to predict future prices.

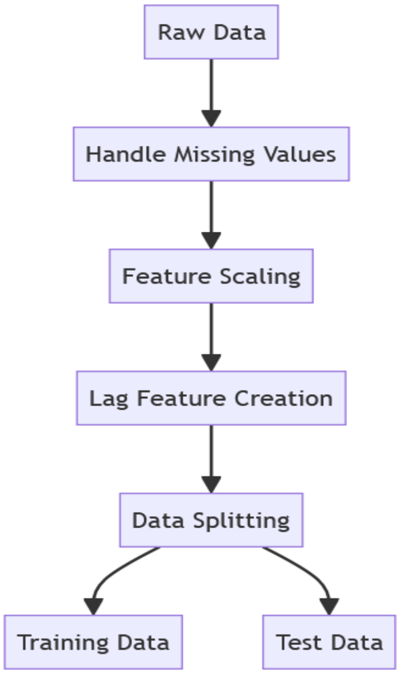

Fig1: Data Preprocessing Steps

3.3 Model Development

The research investigates the performance of various algorithms for time series forecasting, namely: Linear Regression, Bidirectional Gated Recurrent Unit, Neural Networks, and Transformer models. Each model processes time series data differently.

Linear Regression: It is a basic statistical technique that assumes linearity of the relationship between independent attributes (lagged prices) and the dependent variable (next day's price). It can be rather naive; still, it often offers a decent baseline for predictive modeling. In a time series setting, it will pick up all the direct effects that past prices have on the next day's price, though it does not do so well on nonlinear patterns or more extended dependencies.

Bi-GRU: A variant of RNN that was primarily designed for working on sequence data, it is an upgraded version of a simple RNN. Since Bi-GRU is bidirectional, the model looks into both past and future contexts before making any kind of prediction. This may turn very useful when working with time series forecasting, where the comprehension of the sequence and timing of events becomes paramount. On the other hand, GRU units deal more successfully with issues of the vanishing gradient compared to standard RNNs and hence are more proficient in learning long-term dependencies of data.

Neural Networks: In this work, the employed Neural Network model involves a multilayer perceptron with a number of hidden layers. What really makes Neural Networks very powerful is their ability to capture nonlinear relations between input features and the target variable. In time series forecasting, neural networks are capable of learning complex patterns in data that may otherwise not be immediately obvious. On the other hand, they have to be carefully tuned to prevent overfitting when there is limited training data available.

Transformer Model: One of the many strengths of the Transformer model is its use of self-attention mechanisms that weigh the relative importance of different time steps in making predictions. This model will be very good at capturing long-range dependencies like those contained in the data, hence very appropriate for time series forecasting where historical trends might have huge effects on future prices. Moreover, in the positional encoding used by Transformers, a sequential order-needed information is retained while dealing with time series data.

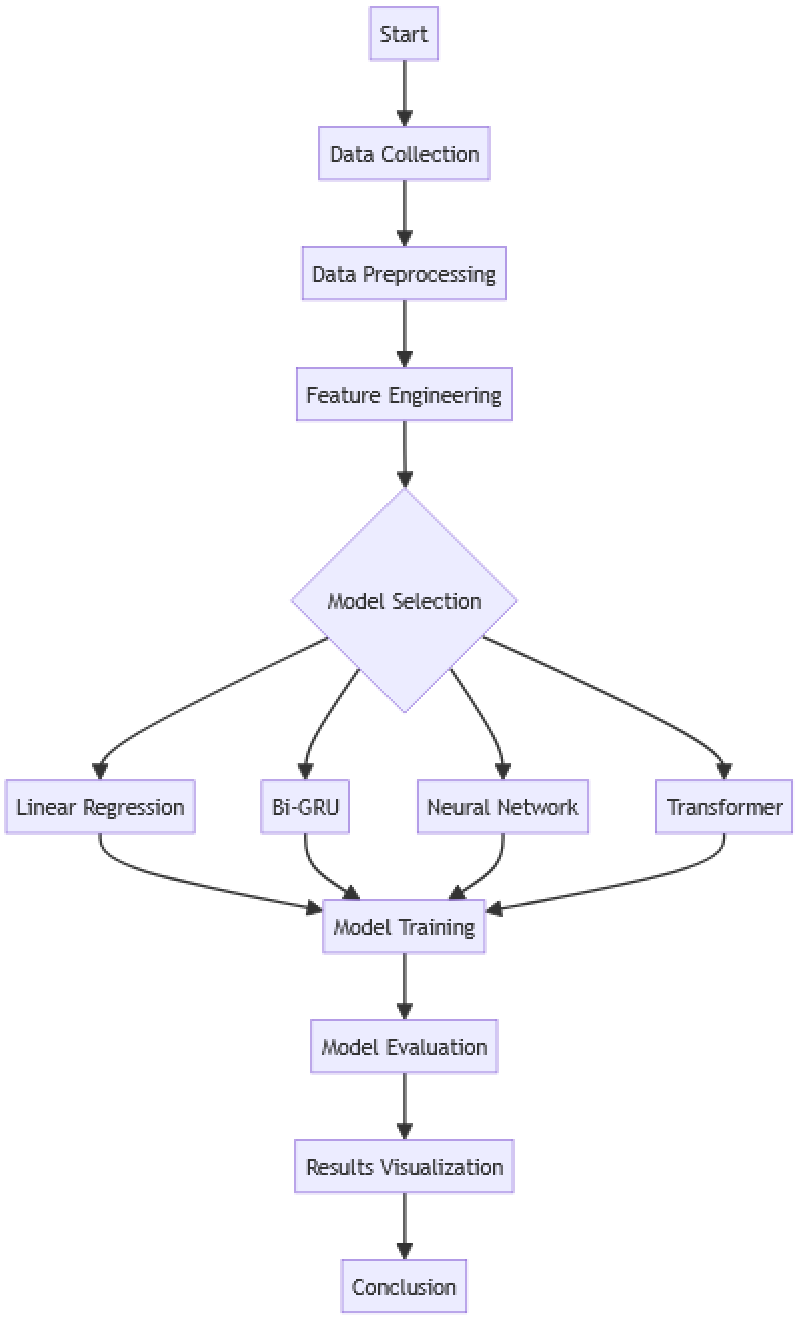

Fig2: Project Workflow Overview

3.4 Why MSE and Not RMSE?

One of the reasons to choose Mean Squared Error for this research as opposed to Root Mean Squared Error is that in this case, MSE will be able to quantify with greater precision the average of the squared differences between predicted and actual values; besides, it is extremely sensitive to large mistakes.

This sensitivity comes in handy in financial forecasting, where large deviations from the actual prices may meaningfully have substantial implications. Squaring the errors using MSE gives more weight to the larger errors, providing a clearer indication as to how well the model performed vis-à-vis avoiding large mistakes in prediction.

While RMSE is also useful because it is in the same units as the target variable, it does not give much weight to large errors, unlike MSE. Thus, MSE was preferred for emphasising the importance of minimising large prediction errors.

3.5 Model Training

Each model was trained on the preprocessed training data using the following configurations:

Linear Regression: The Linear Regression model was trained using the Ordinary Least Squares method, which minimises the sum of the squared differences between the actual and predicted stock prices. This approach finds the best-fit line that represents the relationship between the lagged features and the target variable.

Bi-GRU and Neural Networks: These models were trained using the Adam optimiser, known for its efficiency in handling non-stationary objectives and sparse gradients. The loss function used was Mean Squared Error. Early stopping was implemented to halt training once the validation loss stopped improving, helping prevent overfitting. Dropout layers were included in the Neural Network to further mitigate overfitting by randomly deactivating certain neurons during training.

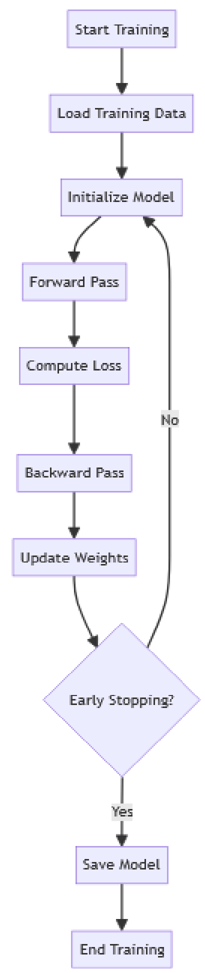

Fig3: Model Training Process

Transformer Model: The Transformer model was trained using the Adam optimiser and Mean Squared Error as the loss function. Positional encoding was added to the input sequence to maintain the temporal order of the data. Early stopping was applied similarly to the other models, ensuring that the model generalised well to the test data.

3.6 Model Evaluation

After the training, each of the models was tested on the test set. Predictions were inverse-transformed back to the original scale by MinMaxScaler. Their model performances will be measured using the following metrics:

MSE: Mean Squared Error, which, as explained earlier, is the average of the squares of differences between predictions and observed values. It especially shows models that let through large prediction errors, which in finance are very critical.

MAE: Mean Absolute Error provides, on average, the actual difference between predicted and real value, giving an intuitive measure for prediction accuracy. It evaluated the MSE and MAE for each model to come up with one that offered a better prediction regarding stock prices for the next day. This is an important evaluation for one to understand strengths and weaknesses of each modeling approach in financial time series forecasting.

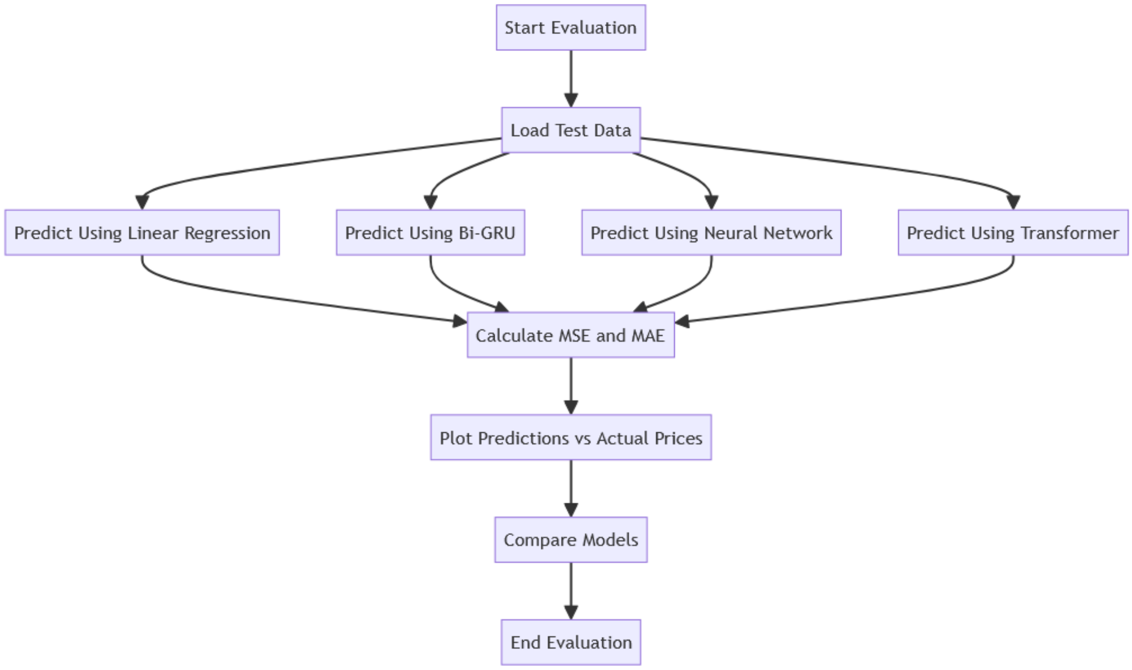

Fig4: Model Evaluation and Comparison

4 Design Specification

This section identifies, and describes, the techniques implemented, the architectures involved, and frameworks used to realise each model applied in this research. Each of these models differs in its design for handling time series data in a framework relevant to predicting stock prices, which are described herein by detailing how the models are structured and work.

4.1 Linear Regression Model

One of the simplest, yet most pervasive applied statistical techniques is linear regression. This research assumes a linear relationship between the features and the target variable, next day's stock price. The model will fit a linear equation in the data such that the sum of squared errors between the actual stock prices and that predicted by the model is minimised. This simple linear approach provides the baseline from which more complex models are compared. In the present research, Linear Regression will be implemented using the Scikit-learn library, due to its availability of efficient tools when it comes to linear modeling.

4.2 Bidirectional Gated Recurrent Unit (Bi-GRU)

Bidirectional Gated Recurrent Unit (Bi-GRU) is a special kind of Recurrent Neural Network that is designed to handle sequential information, for example, time series data. Compared to traditional RNNs, Bi-GRU processes the input data both forward and backward to capture dependencies from the context on either side of a given point. One way of bidirectional processing is when any time series forecasting methodology looks into the timing and sequence of events in a time series to any great degree. In this paper, the Bi-GRU model consists of an input layer fed by a sequence of previous lagging stock prices, which are then processed by a bidirectional GRU layer in two directions. The final output is produced with a dense layer that maps the processed sequence onto the next-day predicted stock price. This model will be implemented using TensorFlow and Keras, both of which offer structural tools in the construction and training of neural networks.

4.3 Neural Network Model

For this study, the neural network model to be adopted is an artificial feed-forward neural network of the multilayer perceptron category. This model would be right and appropriate to capture the nonlinearity inherent in data and turn out to be of help in tasks like stock price forecasting. The neural network is deep in architecture with several hidden layers learnt on complex patterns underlying the data. Rectified Linear Unit—ReLU has been used as the activation amongst these neuronal layers. Dropout layers are added to control overfitting—a quite common problem associated with deep learning models—by randomly deactivating a portion of neurons during training. This method of regularisation avoids biasing towards any neuron of the model, hence increasing the generalisation ability towards new data. Now, the last layer is included, which is the output layer. It makes a prediction for the stock price the next day. The neural network has been implemented using TensorFlow and Keras due to their stronger deep learning 份 abilities.

4.4 Transformer Model

The Transformer model is a more advanced method of modeling sequences; the methodology for treating time-series data incorporates attention this time. Self-attention mechanisms in the Transformer architecture are another innovation that allows the model to weigh different parts of the input sequence relative to one another in its predictions. This capability is especially useful in time series forecasting, as there are some data points from history more relevant for the future prices than others. Besides, positional encoding of the Transformer model allows one to retain the order of a sequence, an extremely important aspect of time series data.

For this research, the Transformer architecture will include an input layer, which will be followed by position encoding for the sequence of lagged stock prices. In addition, there are multihead attention layers focused on other segments of the series. After that, the processed sequence is fed forward until the last layer that makes a prediction for the stock price. This complex model will be implemented using TensorFlow and Keras with their advanced neural network functionalities.

5 Implementation

In the proposed solution, different machine learning models were applied against the study of its historical data to predict the next day's stock prices. This section explains how the results of this final stage of the implementation process were produced and which tools or technologies were used.

5.1 Data Transformation and Preparation

First, the data was transformed and prepared.These include the creation of lagging features, which are part of the time series forecasting module and were done by processing historical stock prices. The created lagging features will represent the closing prices of the stock during the last 20 days and will be given as input to each model. After the mentioned above features were created, the data was normalised using MinMaxScaler for ensuring that all input features were on a comparable scale.

This normalisation is important for neural network models in particular because it improves the efficiency of model training and helps create consistency across all of the models.

5.2 Model Development and Training

In this study, four kinds of models were initially developed and trained: Linear Regression, Bidirectional GRU, Neural Network, and Transformer models. Training of each model was done on the preprocessed training data according to its corresponding machine learning libraries.

Linear Regression: The model was implemented via Scikit-learn and trained by ordinary least squares. The idea was to learn the orders of squares differences between real stock prices and those that the model predicts.

Bi-GRU: The implementation of this model was in TensorFlow and Keras. For training, the Adam optimiser was used—the efficiency of which is high for stochastic problems with non-stationary objectives and sparse gradients. Besides that, early stopping was used to prevent overfitting, and dropout layers were added to enhance generalisation.

Neural Network: For the neural network model, also developed in TensorFlow and Keras, there were numerous hidden layers, which used an activation function ReLU. Dropout layers are added to avoid overfitting. An Adam optimiser is then trained on this model. Finally, early stopping is proposed in order not to fit too much over the training data.

Transformer model: for its level of complexity, an architecture that leveraged self-attention mechanisms and positional encoding to handle time series data. This model was realised with TensorFlow, Keras, and trained with the Adam optimiser. Early stopping has been used to guarantee the robustness when dealing with unseen data.

5.3 Model Outputs and Predictions

Each trained model was run on the test set to generate a stock price that is likely to exist on any next day. The inverse transform was applied using the same instance of MinMaxScaler initially used during training, hence taking these back to their original scale, so meaningful comparisons between the predicted and actual stock prices could be made. Afterwards, the performance of models was measured in terms of MSE and MAE assessments of accuracy and reliability.

5.4 Tools and Technologies

Implementation was mostly done in Python as the programming language, with a set of key libraries used playing important roles: Pandas: This package has been used in data manipulation, data preparation involving the generation of lagging features, and handling of missing values. NumPy: incorporated for efficient numerical computations and handling arrays. Sklearn: It is used for implementing a Linear Regression model and training it, and for scaling data. TensorFlow and Keras: Used for developing, training, and evaluating the Bi-GRU model, Neural Network, and Transformer. Matplotlib: Used for visualising the results, including performance metrics and prediction plots. These models were implemented with successes, giving a general overview of their relative validity for making stock price predictions and imparting valuable insights on their strengths or weaknesses vis-à-vis financial forecasting.

5.5 Visualisation

An important part of the model evaluation process was visualisation. Python, with its matplotlib library, was used to plot predicted stock prices against real prices and for monitoring model trainings.

For the Linear Regression model, we drew a line graph of predicted stock prices against the actual prices. In this case, the x-axis represents dates and the y-axis stock prices. The predictions are in green and the actual prices are in blue. This was very key to see how well our model captured the linear trends of the data.

For the Bi-GRU model, we reshaped the data to feed into this three-dimensional input required model. We then plotted the predictions against the actual prices. Using a learning rate of 0.001 and with the Adam optimiser, we trained it. Early stopping has also been implemented in order to prevent overfitting. We also plot training and validation loss over epochs for monitoring the progress of the learning by the model and checking whether it has started overfitting or not.

In the neural network model, predictions are once more plotted against actual prices; orange is for the predictions and blue represents that of actual prices. This model made several hidden layers that included ReLU activation in combination with dropout layers to avoid overfitting. During training, exactly as used by the Bi-GRU model, the Adam optimiser has been used to plot loss for both the training and validation datasets over the epochs using the same learning rate as.

Since this was an advanced model, more complex visualization was required. We plotted its predictions against the actual prices to depict the ability of learning by this model about these long-range dependencies in data. Multi-head attention mechanisms enhanced the Transformer model for maintaining the sequence order through positional encoding. Watched the training process through plotting Training and Validation Loss; Early Stopping and Learning Rate Reduction is optimized in performance.

Finally, a comparison plot was created for all model predictions in a single graph. It would clearly outline the performance, where different colors and line styles referring to each of the models are green for linear regression, red for Bi-GRU, orange for Neural Network, and purple for the Transformer. The x-axis held the timeline, while the y-axis showed the stock prices that could be directly compared to see how each performed against the actual stock prices.

6 Evaluation

The evaluation section provides in detail the analysis of the results and the primary findings of the study. It also discusses their implications at both the academic and practitioner levels. Results are critically examined using statistical tools for the assessment of experimental research outputs and levels of significance. Effective presentation of results has been supported with visual aids such as graphs and charts.

6.1 Experiment / Case Study 1: Linear Regression Model

In the base case applied in this study, there was a linear regression model. It is among the simplest models, assuming a linear relationship between the input features and the target variable. The model was trained on the training dataset and tested on the test dataset. Mean Squared Error and Mean Absolute Error were used to evaluate model performance.

Results showed that, indeed, the Linear Regression model did a good job in capturing linear trends in stock price data. At the same time, it was predetermined to have its performance limited by the predictability of complex nonlinear patterns typical for financial time series data. Values of Mean Squared Error and Mean Absolute Error were relatively low but with problems of larger errors when nonlinear behavior was manifested in the stock prices.

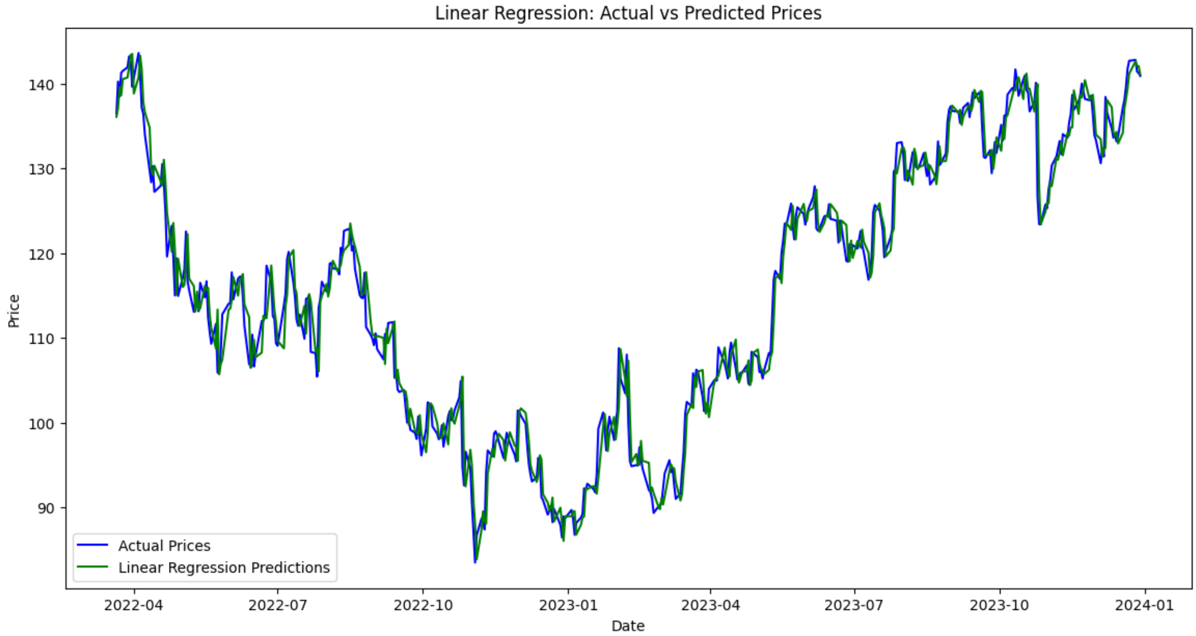

Fig 5. Linear Regression Actual vs Predicted prices

6.2 Experiment / Case Study 2: Bi-GRU Model

The Bidirectional Gated Recurrent Unit (Bi-GRU) model was designed to capture both past and future dependencies in the time series data. This model is particularly effective in handling sequential data, such as stock prices, where the order of events is crucial.

The Bi-GRU model will also be trained on the same dataset to be used with the Linear Regression model. In the work, it looks like an Adam Optimiser might work just fine, which minimises the mean squared error loss function. Early stopping and dropout layers should also be used to prevent overfitting.

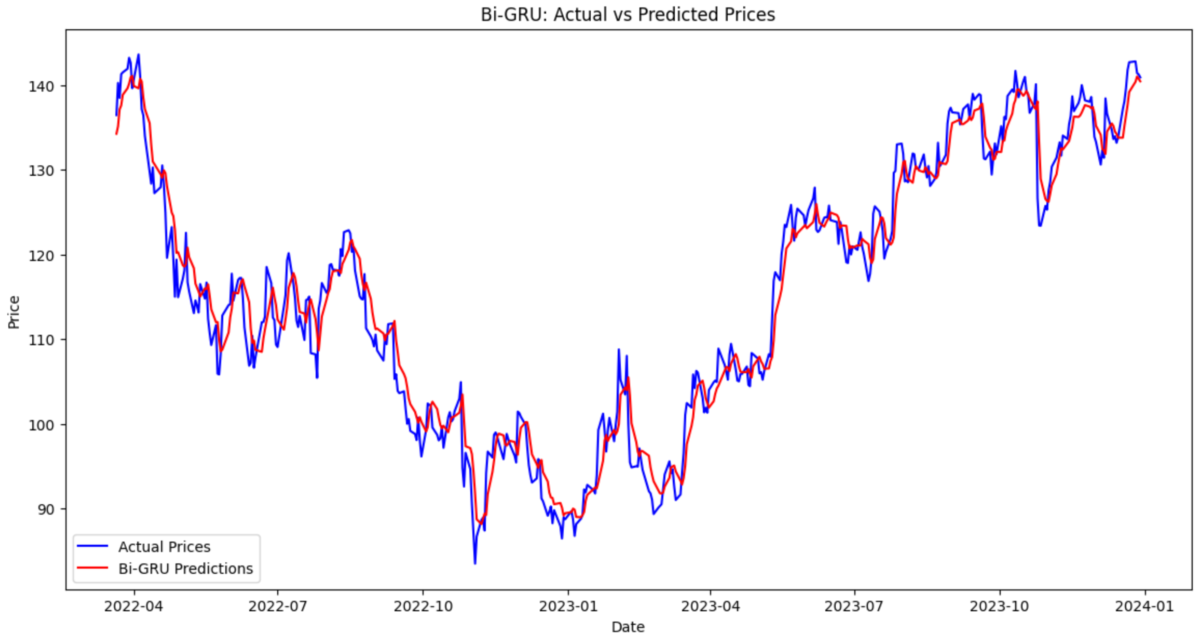

Fig 6. Bi-GRU Actual vs Predicted prices

The result of the evaluation was that, in comparison with the Linear Regression model, the Bi-GRU is much better in picking out complex patterns and dependencies in data. It also had lower MSE and MAE values, thus proving it is much better in predictive accuracy. However, this model required more computational resources and needed careful tuning to avoid overfitting.

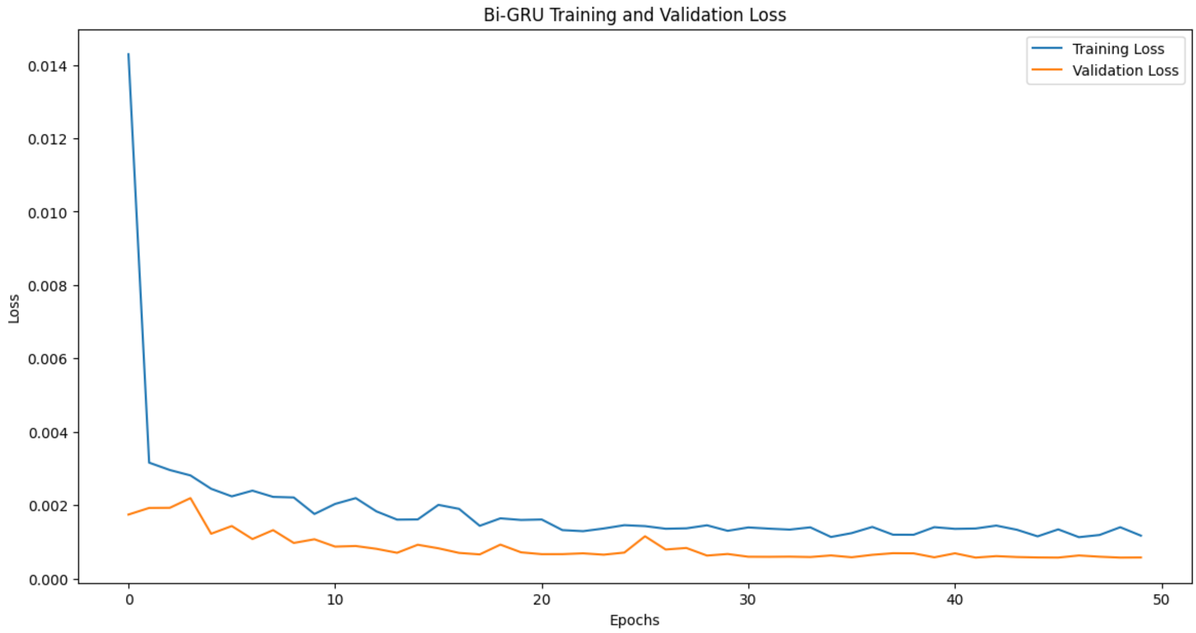

Fig 7. Bi-GRU Training and Validation Loss

6.3 Experiment / Case Study 3: Neural Network Model

This neural network, in this work, was a deep learning model with the use of several hidden layers to capture the non-linear relationship between input features and variables of interest.

We use the Adam optimiser with early stopping to prevent overfitting of the neural network.

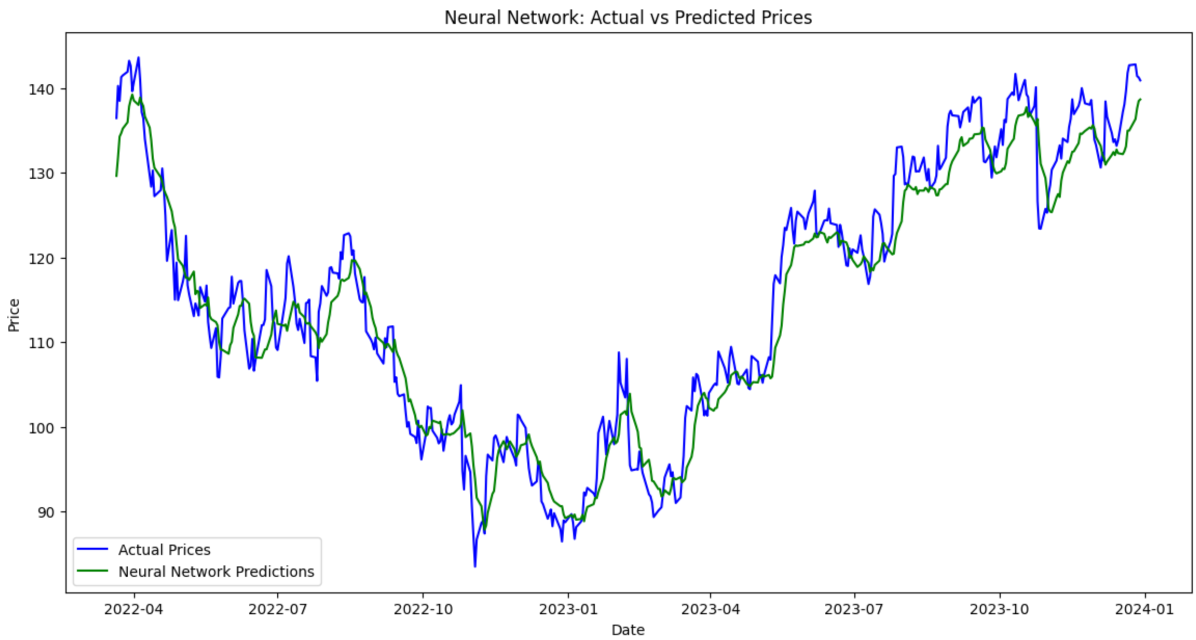

Fig 8. Neural Network Actual vs Predicted prices

In most places, dropout layers are also included to help its generalisation. Then it was found, though the neural network model is good at catching complex nonlinear patterns, it was also prone to overfitting as compared to the Bi-GRU model. Again, its mean squared error and mean absolute error are a little higher than that of the Bi-GRU model; however, it will perform pretty well in predicting stock prices.

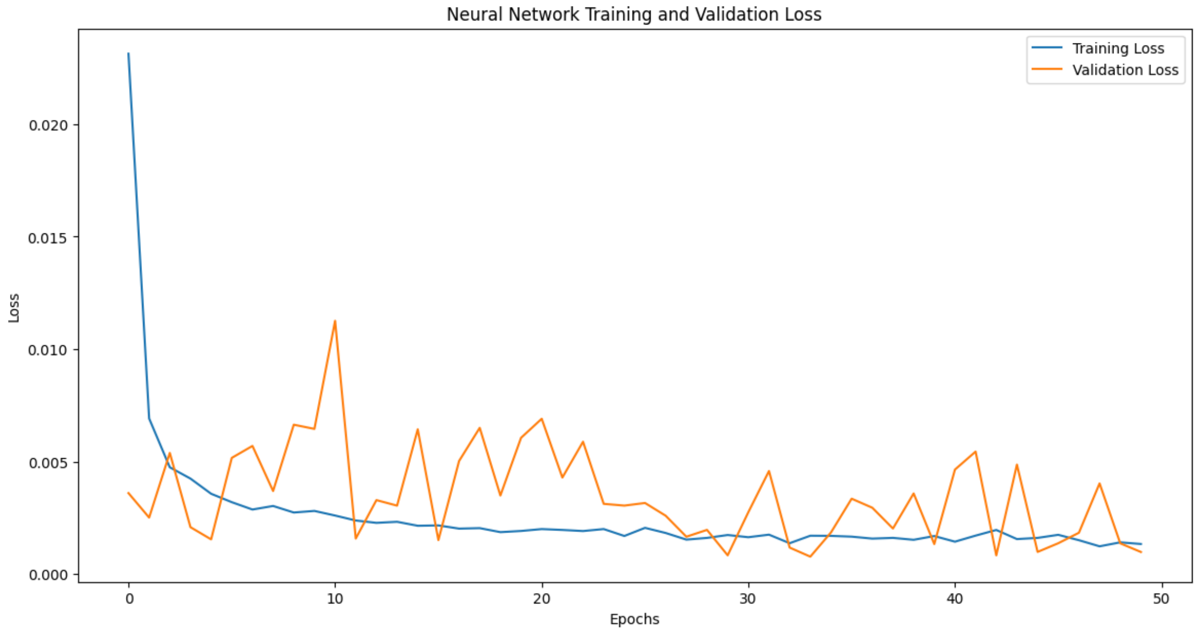

Fig 9. Neural Network Training and Validation Loss

6.4 Experiment / Case Study 4: Transformer Model

One of the advantages of the Transformer model was that it could handle long-range dependencies existing in sequential data with its self-attention mechanisms. Implementing a Transformer model with positional encoding in this research helped retain information regarding the order of items in a sequence in time series data. In the process of training, the Adam optimiser was considered for this Transformer model, while early stopping avoided overfitting. As mentioned, the performance metrics were MSE and MAE.

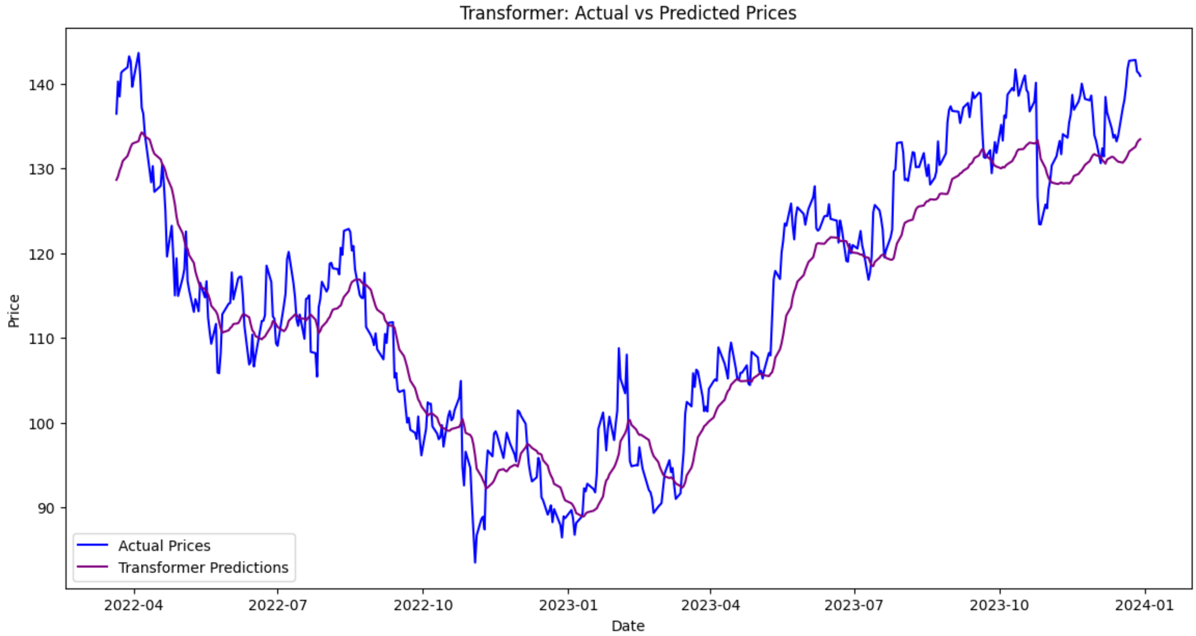

Fig 10. Transformer Actual vs Predicted prices

Mixed results from the Transformer model are to be shown in the evaluation, whereby this model performed rather perfectly in capturing long-range dependencies while still being far from short-term fluctuations in stock prices. The values of MSE and MAE turned out to be rather high themselves, thus showing that the model probably needs to be fine-tuned and adapted more for financial time series forecasting.

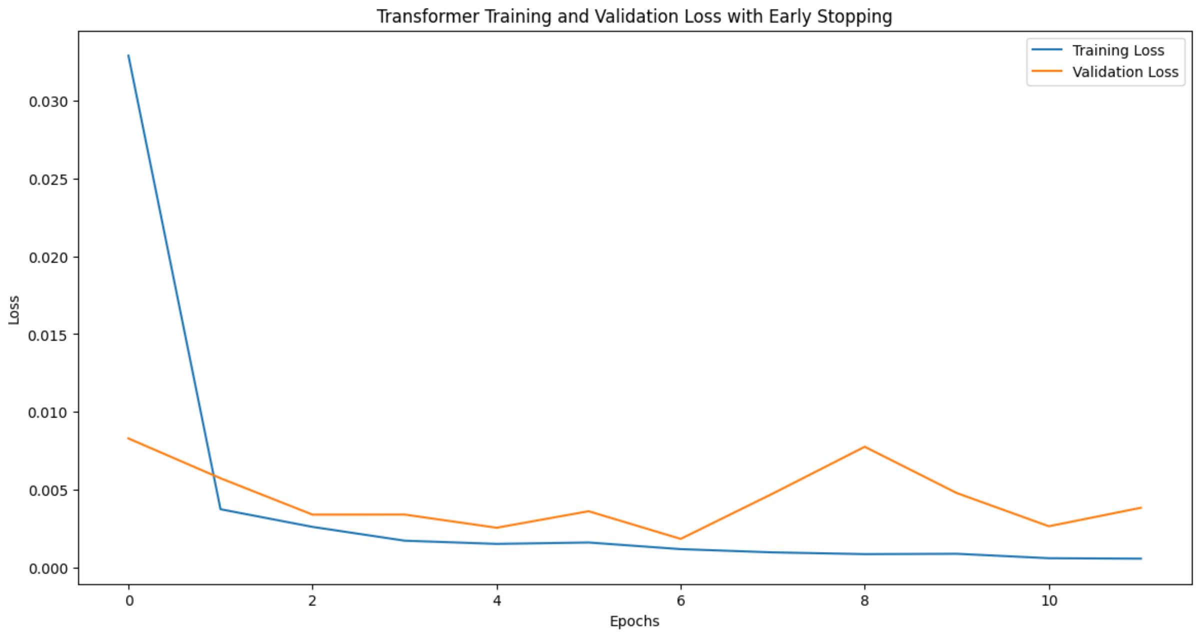

Fig 11. Transformer Training and Validation Loss

6.5 Discussion

These experiments insinuate how various machine learning models will work for the purpose of predicting stock prices. Linear Regression is a rather simple model that showed quite reasonable results in catching linearity and was easily confused by nonlinear trends. The Bi-

GRU model has turned out to be useful when trying to learn higher, complex dependencies within data, hence making it a very good candidate for time series forecasting. While the neural network model was able to capture nonlinear relationships, this came at a cost of being far more sensitive to overfitting and necessitating careful tuning. The Transformer model, though theoretically very promising, did not do as well in this application as was expected and needed further study and adaption. In other words, the findings of this research work place models like Bi-GRU and Neural Networks as offering better predictive accuracy, and simpler ones like Linear Regression not without their place in predicting linear trends in data. A potentially very powerful Transformer model still requires specialised tuning for successful financial forecasting.

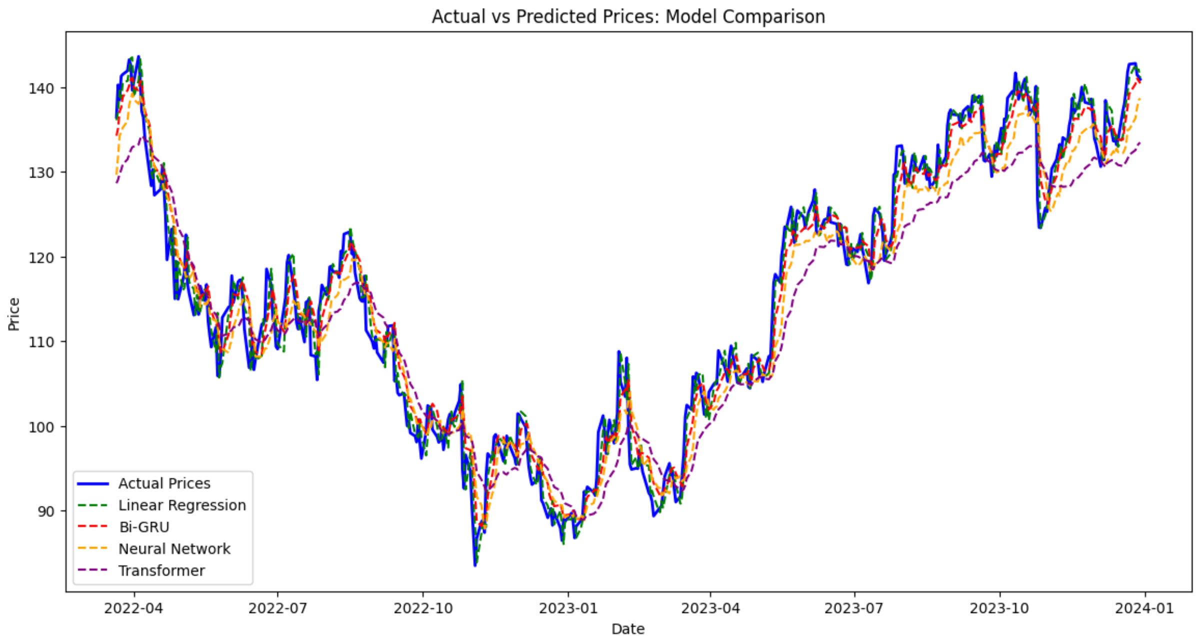

Fig 12. Actual vs Predicted prices of All Models Comparison

7 Conclusion and Future Work

In this research, the primary question posed was how well different machine learning models, including Linear Regression, Bidirectional Gated Recurrent Unit (Bi-GRU), Neural Networks, and Transformer models, could predict the next day's stock prices based on historical data.That means the different deep-learning models should be evaluated and compared based on their ability to predict the stock prices accurately from the real financial time-series data gathered from Google (GOOGL).

The question was well addressed, and the objectives were fulfilled by rigorous experimentation and analysis. Key findings demonstrated that while more complex models, such as Bi-GRU or Neural Networks, could capture complex patterns and nonlinear relationships in the data, surprisingly, it was proved by the Linear Regression model that in scenarios when linearity dominated, and thus required little of the complex architecture in the deep models, their strong ability was absolutely necessary. Although the Transformer model has a potential toward handling long-range dependencies, it has not turned in very promising results for this particular application, thus much room for further fine-tuning and adaption in financial time series forecasting.

The implications of the research are huge for academia and industry. This study contributes to the literature on fiscal forecasting by pointing out some strengths and limitations of various machine learning models in regard to stock price prediction. Model selection for practitioners, therefore, has to be carefully tailored to data characteristics or the nature of the forecasting task.

While complex models can perform better in certain scenarios, there are others where simpler ones—like Linear Regression—can be more efficient and show better performance. However, the research as a whole also had its weaknesses. A major weakness inherent in this study is that it used only one dataset from Google GOOGL; these results may not generalise to other stocks or other market conditions. The models were optimised and tuned for this particular dataset, but further refinement and experiments might give better results.

Several meaningful directions can be pursued in this area in the future.First, further tuning of the performance could be done with the Transformer model by trying out different configurations, such as the number of attention heads or the strength of the model. Second, features such as sentiment analysis from news articles or social media can be added, thus providing a richer context setting for the models to possibly improve their predictive accuracy. Third is the potential analysis of such ensemble methods that can combine the strengths of various models to create even more robust predictions.

Further research on other stocks or asset classes would actually confirm the findings and check the models for generalisability. Not least, checking the interpretability of these models, mainly with respect to financial forecasting, might add valuable insights to decision-makers and render the research more applicable in practice. In other words, this paper has demonstrated the potential of various machine learning models for predicting stock prices but has also hinted that models should always be chosen and optimised with caution. This will add to the efforts already made toward enhancing the techniques of financial forecasting and provide a stepping stone for further research in this area.

References

Duan, Y., Liu, Y., Wang, Y., Ren, S., & Wang, Y. (2023). Improved BIGRU Model and Its Application in Stock Price Forecasting. Electronics, 12(12), 2718.

Xu, L., Xu, W., Cui, Q., Li, M., Luo, B., & Tang, Y. (2023). Deep heuristic evolutionary regression model based on the fusion of BiGRU and BiLSTM. Cognitive Computation, 15(5), 1672-1686.

Karim, M. E., & Ahmed, S. (2021, October). A deep learning-based approach for stock price prediction using bidirectional gated recurrent unit and bidirectional long short term memory model. In 2021 2nd Global Conference for Advancement in Technology (GCAT) (pp. 1-8). IEEE.

Victor, S. (2024, June). Analyzing Stock Market Trends using Social Media user Moods and Social Influence using Lightweight BERT-BiGRU-A Model. In 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 1841-1847). IEEE.

Mao, Z., & Wu, C. (2023). Stock price index prediction based on SSA-BiGRU-GSCV model from the perspective of long memory. Kybernetes.

Costa, L. D., & Machado, A. M. (2023, August). Prediction of Stock Price Time Series using Transformers. In Anais do II Brazilian Workshop on Artificial Intelligence in Finance (pp. 85-95). SBC.

Li, Y., Lv, S., Liu, X., & Zhang, Q. (2022). Incorporating transformers and attention networks for stock movement prediction. Complexity, 2022(1), 7739087.

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023, June). Are transformers effective for time series forecasting?. In Proceedings of the AAAI conference on artificial intelligence (Vol. 37, No. 9, pp. 11121-11128).

Ramos-Pérez, E., Alonso-González, P. J., & Núñez-Velázquez, J. J. (2021). Multi-transformer: A new neural network-based architecture for forecasting S&P volatility. Mathematics, 9(15), 1794.

Hu, X. (2021, December). Stock price prediction based on temporal fusion transformer. In 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI) (pp. 60-66). IEEE.

Kwon, Y. K., & Moon, B. R. (2007). A hybrid neurogenetic approach for stock forecasting. IEEE transactions on neural networks, 18(3), 851-864.

Wang, J. J., Wang, J. Z., Zhang, Z. G., & Guo, S. P. (2012). Stock index forecasting based on a hybrid model. Omega, 40(6), 758-766.

Araújo, R. D. A., Oliveira, A. L., & Meira, S. (2015). A hybrid model for high-frequency stock market forecasting. Expert Systems with Applications, 42(8), 4081-4096.

Chong, W. U., Peng, L. U. O., Yongli, L. I., & Chen, K. (2015). Stock price forecasting: hybrid model of artificial intelligent methods. Engineering Economics, 26(1), 40-48.

Efendi, R., Arbaiy, N., & Deris, M. M. (2018). A new procedure in stock market forecasting based on fuzzy random auto-regression time series model. Information Sciences, 441, 113-132.

Pai, P. F., Lin, K. P., Lin, C. S., & Chang, P. T. (2010). Time series forecasting by a seasonal support vector regression model. Expert Systems with Applications, 37(6), 4261-4265.