COMP7409 FINAL REPORT
MACHINE LEARNING FOR STOCK PRICE FORECASTING

Group 32: Zeng Shuang, Cui Zeyang, Xie Zihan, Zhao Mingyang

ABSTRACT

The work aims to use the machine learning methods to forecast the next 15 days' stock price accurately. A range of data including numerical values and textures are gathered from APIs and processed. Both ML and DL models are implemented and improved to forecast the stock price. A series of experiments are taken to improve the model performance with the feature selection, the set hyperparameters and training process. Among the proposed fine tuned DL models, transformer achieves the highest performance with and without feature selection and the highest return from the implemented investment decision making algorithm. [1]

1. INTRODUCTION

Predicting stock market prices has long been one of the most challenging problems in financial field. Stock prices are difficult to predict because of their high volatility depending on different investor sentiment, political, economic and market factors, etc. Predicting stock prices based on historical data or textual information alone has proven to be insufficient. Existing research on sentiment analysis has found a correlation between stock price movements and news article publication.

To improve the accuracy of the algorithm, in the project, data collection, data exploration, data pre-processing, and model analysis are sequentially performed by collecting large amounts of time series data and using extensive machine learning as well as deep learning model analysis. The project collects stock price related data from websites and APIs as well as social media news, selects the top ten stocks of S&P100, and then uses natural language process (NLP) models to perform sentiment analysis and feature selection. Various deep learning models are built, using economic variables, market index, news score and stock prices of other companies as predictors to train the models and predict stock price changes.

Data preprocessing is first performed by crawling information using python, followed by transforming the data form in order to unify the analysis of different variables affecting stock prices. To perform sentiment analysis on linguistic texts such as news, two NLP models, FinBERT and GPT-3.5-Turbo, are used to determine whether news information has a positive or negative impact on stock prices. For numerical data such as stock prices and general market indicators, machine learning models are utilized to calculate the importance of features and analyze the correlation between different predictors to conduct feature selection. Also, data cleaning is performed to process outliers.

For higher prediction accuracy, the highlight of this project is that it updates the prediction output based on the latest stock prices, combined with the input features and parameters. In addition, multiple deep learning models are fine-tuned. Then multi-day stock price prediction and trading return calculation are performed with feature correlation, and the results of each model are compared to select the best model solution.

2. DATA DESCRIPTION

In this project, we have chosen Apple Inc. (AAPL) as our target stock for price prediction for the following reasons:

1. **Representativeness:** As one of the largest and most well-known companies worldwide, Apple's stock serves as a representative example for studying market trends and behavior, being a component of major stock market indices such as the S&P 500, NASDAQ, and Dow Jones.
2. **Market size:** Apple's substantial market capitalization implies that its stock price is primarily influenced by broader market trends, economic factors, and financial developments. This characteristic makes Apple an attractive subject for research, offering insights into larger market dynamics.

[1] The code is available here at GitHub and the data is available here at Google Drive.

3. **Stability:** In comparison to smaller companies that may be more susceptible to price manipulation or speculation by individual investors, Apple's stock is less likely to be affected by such factors. This stability renders it valuable for conducting research and developing prediction models.

2.1. Stock Price and General Market Indicators

The stock selected for the project is AAPL (Apple Inc.). To ensure the amount of training data for the model algorithm, the data from January 2015 to December 2022 was selected. The data set collected includes economic variables, market Index, news contents and stock prices of other companies. Historical stock price information is downloaded from Yahoo finance, including date, open, high, low. A total of 2013 rows of data are stored.

Regarding general market information, data including employment, inflation, business inventories, etc. are collected. Also, some market indexes are considered, such as import and export price index, consumer price index, consumer sentiment index, etc. In the project, the data of the top ten companies in Standard and Poor's 100 (S&P 100), a price averaged measurement stock market index of 100 prominent companies, are selected for specific feature correlation analysis. [1]

2.2. Macroeconomic State Variables

In our analysis of the impact of various market indicators on the target stock price prediction, we have categorized them into several types, each with its distinct characteristics and implications. Detailed descriptions and abbreviations for each market indicator are provided in the appendix section13. The main categories include:

1. **Real economic activity**

   The level of economic activity in a country, as measured by indicators such as gross domestic product (GDP), can provide important context for predicting stock prices. A strong economy may lead to higher stock prices, while a weak economy may lead to lower stock prices.

2. **Inflation**

   Inflation refers to the rate at which the prices of goods and services increase over time. High inflation can erode the purchasing power of consumers and reduce the profitability of businesses, while low inflation can stimulate economic growth. When predicting stock prices, it is important to consider the expected level of inflation and how changes in inflation may affect the economy and the stock market.

3. **Investment**

   Investment in the stock market is influenced by factors such as the availability of capital, investor sentiment, and the expected returns on investment. These factors can be difficult to predict, but may be important inputs to a machine learning or deep learning model.

4. **International crude oil prices**

   The price of oil can have a significant impact on the global economy, and may therefore influence stock prices. For example, high oil prices can lead to higher inflation and reduced economic growth, which may lead to lower stock prices.

5. **Trade openness**

   Trade openness indicates the extent to which a country engages in international trade, as measured by imports and exports as a percentage of GDP. Greater trade openness can provide access to larger markets and lower-cost inputs, which can improve the competitiveness of domestic firms. However, increased trade can also expose domestic firms to greater competition, which may reduce profitability.

6. **Money supply**

   The amount of money in circulation can affect the level of economic activity and inflation, which in turn can affect stock prices. For example, if the money supply increases rapidly, inflation may rise, which could reduce the profitability of businesses and lead to lower stock prices.

7. **Interest rate**

The interest rate refers to the cost of borrowing money, typically expressed as a percentage of the amount borrowed. Changes in interest rates can affect the availability of credit, the cost of borrowing, and the profitability of businesses, which in turn can affect stock prices.

8. **Exchange rate**

   Changes in exchange rates can affect the competitiveness of businesses and the value of foreign investments. For example, if a country's currency appreciates relative to other currencies, its exports may become more expensive, which could reduce the competitiveness of domestic firms and lead to lower stock prices.

9. **Stock market capitalization**

   The overall size of the stock market can provide important context for predicting stock prices. A large and active stock market may be more resilient to changes in economic conditions than a smaller, less active market.

10. **Market liquidity**

    Market liquidity shows the ease with which stocks can be bought and sold in a market. High market liquidity can help to ensure that stocks can be sold quickly and at a fair price, which can reduce the risk of large price movements and increase investor confidence. When predicting stock prices, it is considered since it may affect the volatility and stability of the stock market.

2.3. News

Investment-related news on social media platforms may affect stock price trends, so sentiment analysis of textual information is one of the main focuses of the project. In order to calculate the sentiment index of news, divided into bullish and bearish, the financial news related to the word 'AAPL' are acquired from the API of finance website SeekingAlpha to guarantee the turth and the influence of the information. The selected news titles are within the time range from the begining of 2015 to the end of 2022. A total of 4268 news items are stored as csv files after filtering and sorting used for sentiment analysis.

3. DATA EXPLORATION

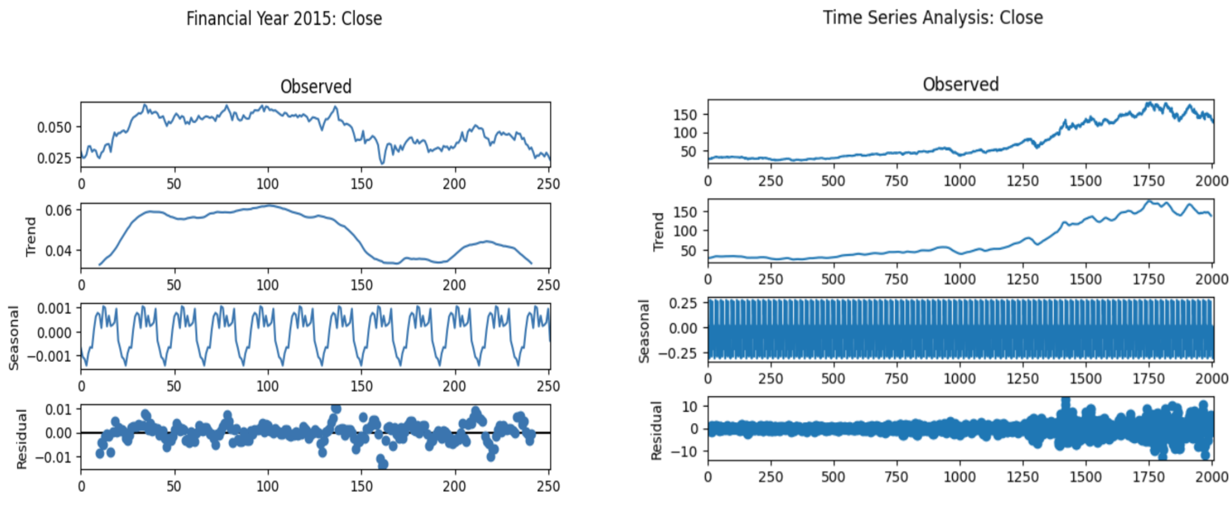
3.1. Time Series Analysis

The data analysis is discussed in the section: taking time series analysis, measuring the feature correlation, taking the hypothesis to evaluate the dependency between the factors. For the time series analysis, we measured the feature response to the time change by evaluating the trend, seasonality, and the residual.

$$Y(t) = T(t) + S(t) + e(t) \tag{1}$$

where $T(t)$ is the trend, $S(t)$ is the seasonality and $e(t)$ is the irregularity. The time series analysis measures the feature response to the time change by evaluating the trend, seasonality and irregularity. The trend indicates the non repetitive change of the feature. Seasonality relates to the phenomenon repeating itself after a regular period of time. The component not in trend or seasonality is the random noise. The addictive model with the period set to 21 is selected for the analyse:

The close price is analysed 1. Generally, the trend shows a steady fluctuation in the stock price in 2015. The amount of noise increases over time which might be caused by unexpected events. Since the stock price increased, the flotation in the price might also be higher which is not contained in the constant seasonality information. Therefore, in the project, the time series analysis is taken in financial years to further analyse the components.



(a) (b)

Fig. 1: Time series analysis (a) Time series analysis in the year 2015, (b) Time series analysis in all years

To facilitate data visualization, we created a line chart2 depicting the S&P 100 stock prices. We then segmented the data by year and illustrated the specific trends for each period. To maintain the clarity of the line chart, we only visualized a subset of the stocks. Upon examination, we noticed that many of these stocks share similar trends, which could be leveraged for further filtering and analysis to predict the target stock price (AAPL).

By observing the commonalities in these stock price trends, we can better understand the market dynamics that drive the performance of these companies. This insight allows us to make more informed decisions regarding which stocks to include in our feature selection process. Additionally, focusing on stocks with similar trends to AAPL can help us build a more accurate predictive model, as these companies are likely to be affected by similar external factors and market conditions.

In summary, data visualization plays a crucial role in identifying trends and patterns among S&P 100 stocks. By concentrating on stocks with comparable trends to AAPL, we can further refine our feature selection process and improve our model's predictive power for the target stock price. This approach not only enhances the model's overall performance but also aids in our understanding of the underlying market mechanisms that influence stock prices.

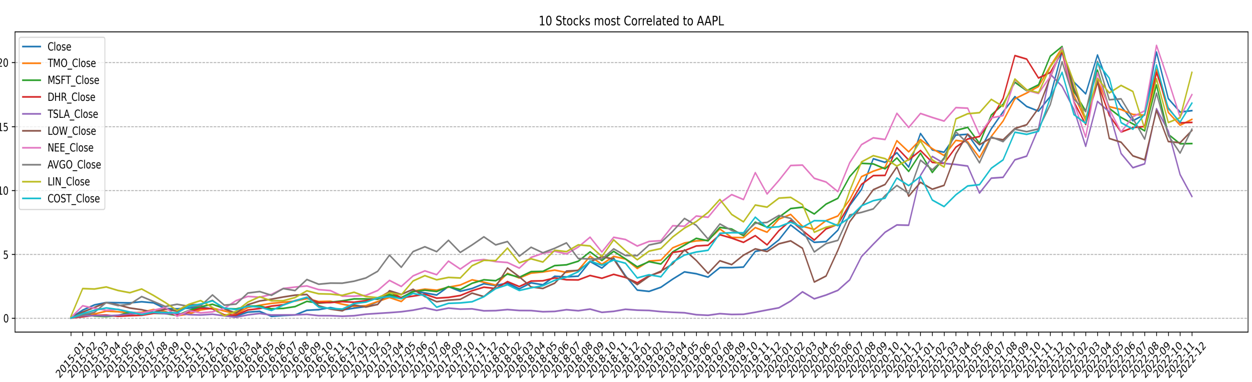


Fig. 2: Trend of a subset of S&P100 (a) trends in all years

In the following analysis, we explore general market indicators by dividing them into groups of ten and plotting their respective trends, as shown in Figure 3. It is evident that many factors exhibit similar trends to the target stock price, while others display considerable discrepancies in their trends, such as MICH (University of Michigan Consumer Sentiment Index, a survey-based measure of consumer confidence in the US economy) and CSCICP03USM665S (S&P/Case-Shiller US National Home Price Index, a measure of the average change in home prices across the United States over time).

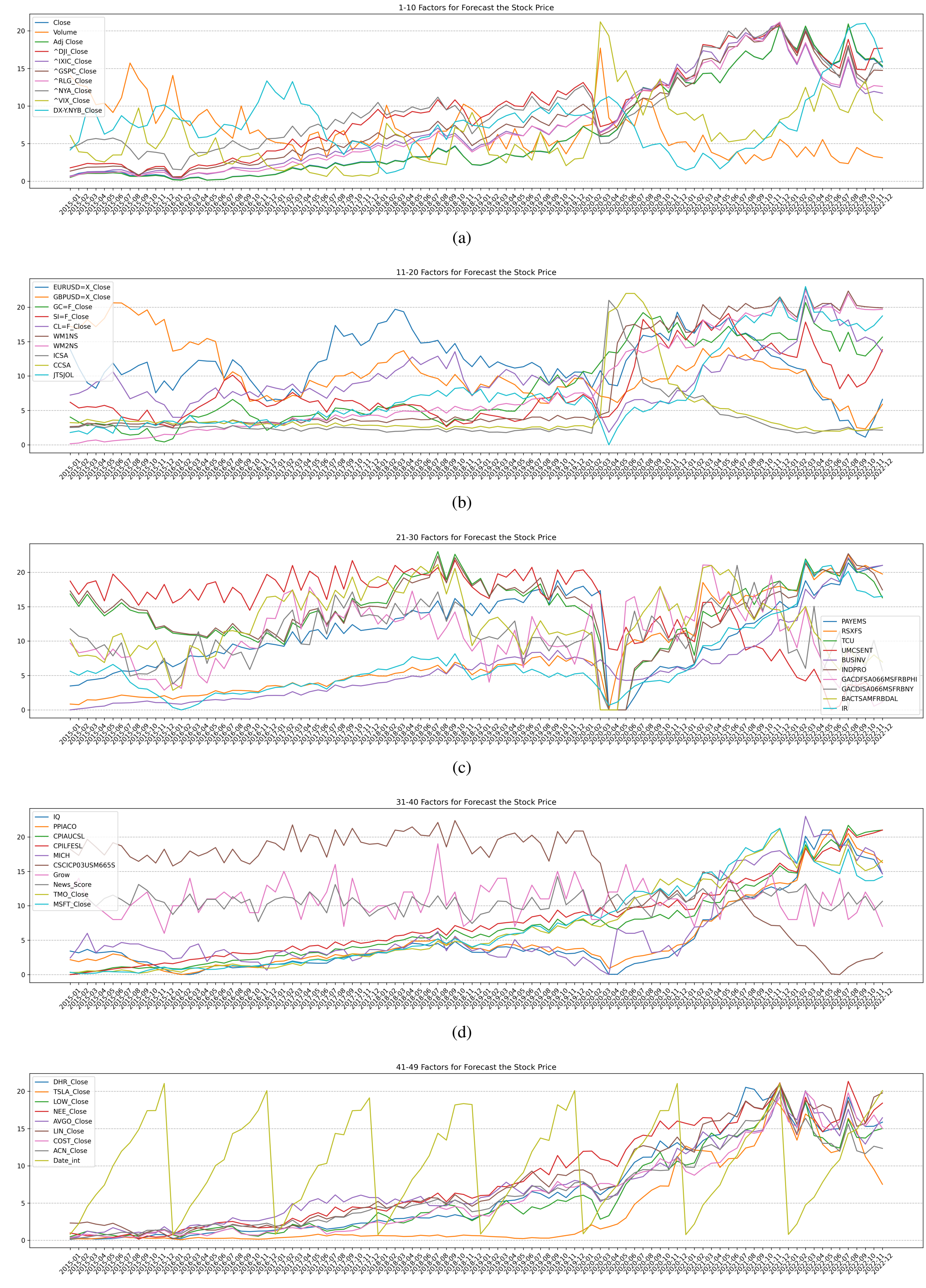


Fig. 3: The full features used for the stock price forecasting (a) the first 10 features, (b) the 11-20 features, (c) the 21-30 features, (d) the 31-40 features, (e) the 41-49 features

Given the varying trends among these market indicators, it is essential to further refine our feature selection process to focus on the most relevant and influential factors. By doing so, we can ensure that our predictive model accurately captures the relationships between these factors and the target stock price. In the upcoming chapter, we will delve deeper into the feature selection process, exploring various techniques to identify the most significant indicators and their impact on stock price predictions.

Through meticulous analysis of general market indicators, we can enhance our understanding of the factors influencing the target stock price and improve the overall performance of our predictive model. By focusing on the most relevant and influential indicators, we can create a more robust and accurate model that better reflects the intricacies of the stock market and its impact on AAPL's stock price.

3.2. Feature Correlation

By visualizing the correlation heatmap4 of the major general market indicators, we can draw conclusions similar to the previous analysis, which highlights the need for further feature selection. Many of the current features do not contribute significantly to the prediction of the target stock price and should be eliminated to enhance the accuracy and effectiveness of our predictive model. In the next chapter, we will provide a detailed description of the feature selection process, elaborating on the various techniques used to refine our feature set. Furthermore, we will present the results of this selection process, showcasing the reduced feature set and its impact on the predictive performance of our model. This comprehensive analysis will ensure that our model focuses on the most relevant and influential factors, leading to more accurate and reliable predictions of the target stock price.

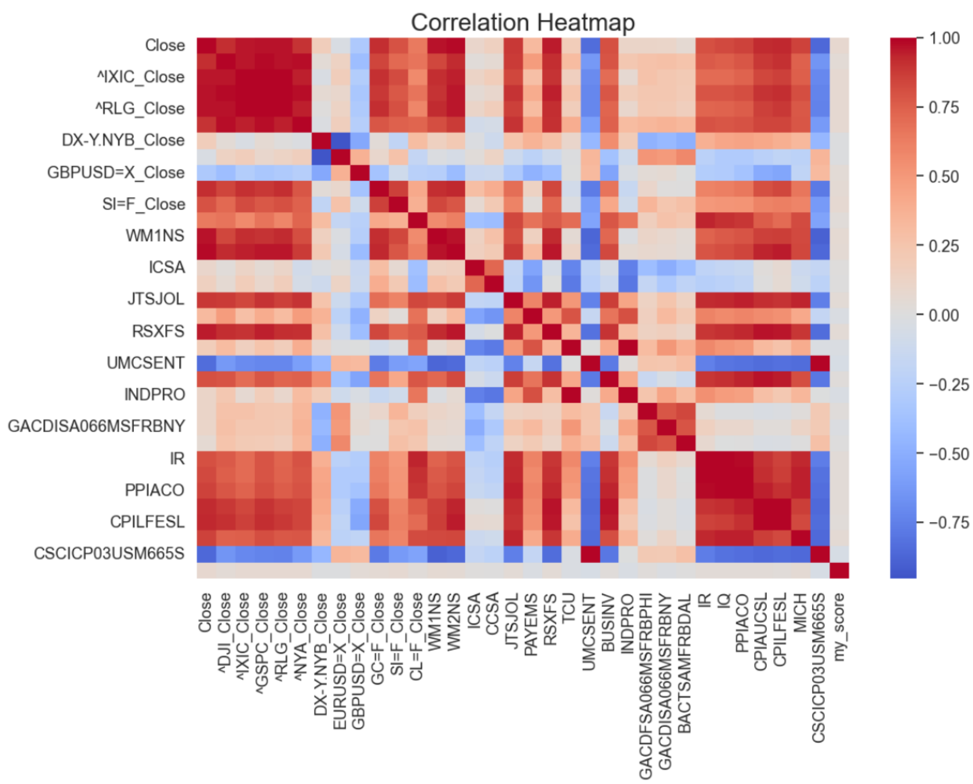


Fig. 4: The Correlation of the Major Features

4. DATA PREPROCESSING

In the Data Preprocessing section, we focus on processing both text and numerical data for further analysis. For text data, we utilize two cutting-edge natural language processing technologies: FinBERT[2] and GPT-3.5. FinBERT is a BERT-based pre-trained model specifically designed for financial sentiment analysis, while GPT-3.5, powered by OpenAI, is a state-of-the-art language model that can also perform sentiment scoring. We combine the sentiment scores obtained from both models to create a more robust representation of the text data. For numerical data, we analyze stock prices of S&P 100 companies and select the top 10 stocks with the highest correlation to our target stock, AAPL. In addition, we preprocess general market indicators and commodity prices by performing data cleaning and outlier handling. Outliers are replaced with the mean value, ensuring that our dataset is more representative of the underlying trends.

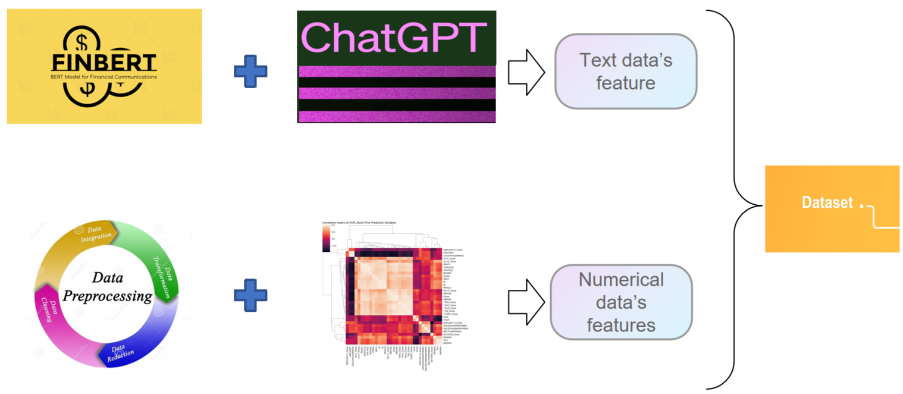


Fig. 5: Overview of whole datapreprocessing

Lastly, we perform feature selection on the numerical data to further eliminate irrelevant or redundant information from the features. By selecting the most important and informative attributes, we enhance the efficiency and effectiveness of our models, reducing the risk of overfitting and improving generalization to new, unseen data. This streamlined feature set allows us to build more robust and accurate models, ultimately leading to more reliable insights and predictions. We also compare the results of performing and not performing feature selection in the RESULT section. By carefully preprocessing both text and numerical data, our team lays a solid foundation for subsequent analyses and insights, enabling us to draw more accurate conclusions and make informed decisions.

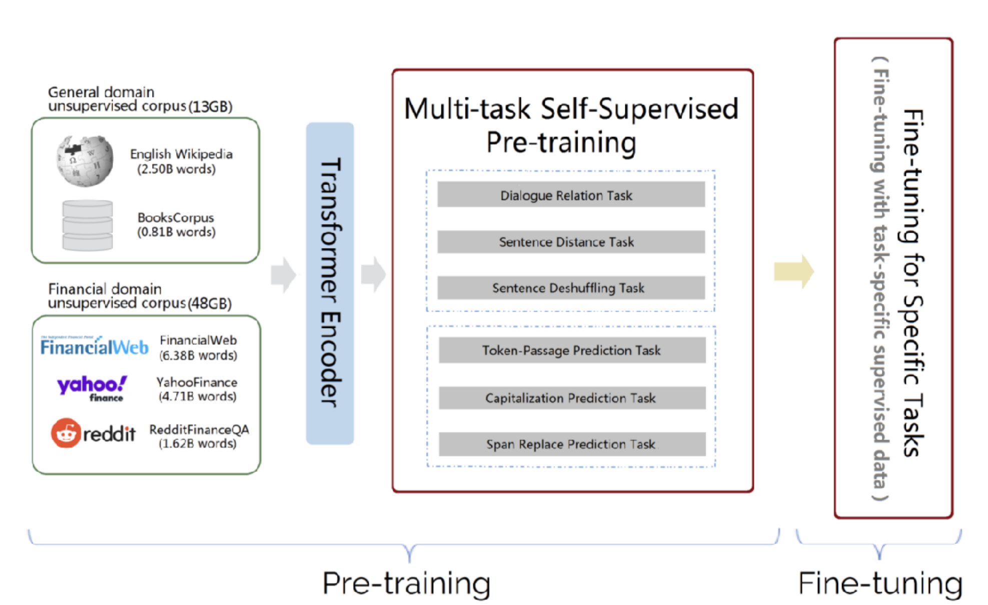


Fig. 6: Structure of FinBERT

4.1. Sentiment Analysis for News, Social Media and Financial reports

Two pretrained models are used for the sentiment analysis of the financial information: FinBert[2] and ChatGPT.

4.1.1. FineBERT

FinBERT[2] is an adaptation of the BERT[3] model specifically designed for the financial domain, aiming to enhance its performance on tasks like sentiment analysis and regression in finance-related contexts. To achieve this, FinBERT is further pre-trained on domain-specific corpora, such as financial news and reports, to better understand the nuances of the financial language. The implementation also focuses on mitigating the issue of catastrophic forgetting during fine-tuning, ensuring that the model retains its foundational language understanding while adapting to finance-specific tasks. Various training strategies are employed in FinBERT, including slanted triangular learning rates, which involve a learning rate schedule that first increases linearly and then decreases; discriminative fine-tuning, which applies lower learning rates to the lower layers of the network to preserve core language features; and gradual unfreezing, which initially freezes all layers except the classifier layer and progressively unfreezes them during training[4]. These techniques work together to maintain the model's language proficiency while fine-tuning it for the financial domain, leading to improved performance in tasks such as sentiment classification and financial regression analysis.

In this case the news titles are inputted to FinBert to provide three float values for positive, negative and neutral. The three values are summed as the index to show the influence on the stock market. We derive the final scoring of FinBERT from Equation 2

$$sentiment\ score = 1 * f(x)_{Positive} + 0 * f(x)_{Neutral} - 1 * f(x)_{Negative} \tag{2}$$

Hence, the sign of the value can indicate the outcome to stock price and a value close to zero is ambiguous to the price.

4.1.2. ChatGPT (GPT-3.5-Turbo)

Chatbot GPT, or ChatGPT, is an advanced language model developed by OpenAI, based on the powerful GPT architecture. GPT, or the Generative Pre-trained Transformer, leverages a transformer-based neural network structure, which allows for efficient parallelization during training and exceptional performance in natural language understanding and generation tasks. The model is pre-trained on a large corpus of text data, learning the structure and patterns of human language during this unsupervised training phase.

After pre-training, ChatGPT is fine-tuned using supervised learning on task-specific datasets, enabling it to better understand and generate contextually relevant responses in a conversational setting. With its ability to handle diverse conversational contexts and generate coherent, contextually appropriate responses, ChatGPT has proven to be a powerful tool for various applications, including question-answering, content generation, and conversational AI. The continuous advancements in the GPT architecture contribute to the ongoing improvement of ChatGPT's capabilities and performance.

Due to the corpus in the news title data that does not have very specialized financial terms, that is why we also use the more generalized chatGPT to score the news titles for sentiment in order to compensate with FinBERT which performs well in financial prediction. We mainly use OpenAI's API to get the sentiment scores, and below is the format of the message we transmit to the API.

```
message = {
 "role": "user",
 "content": f"Please rate the following article titles on a scale of -1 to 1,
 with higher scores being more positive and a score of 0 being emotionally neutral,
 you only need the score and no other information: \"{News_Text}\""
 }
response = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[message]
 )
 score = float(response.choices[0].message.content.strip())
```

In this case, after a basic cleaning of the collected news data, if multiple articles are found for the same date, we will calculate their average sentiment score. For non-trading days, the sentiment scores of the text data will be averaged and assigned to the next available trading day, ensuring that the sentiment analysis results are aligned with the corresponding market activity.

This approach allows us to obtain a comprehensive and meaningful understanding of the market sentiment on specific trading days, which can be valuable for financial analysis and decision-making processes.

The final sentiment score can be expressed as

$$sentiment\ score = 0.5 * FinBERT\ score + 0.5 * ChatGPT\ score \tag{3}$$

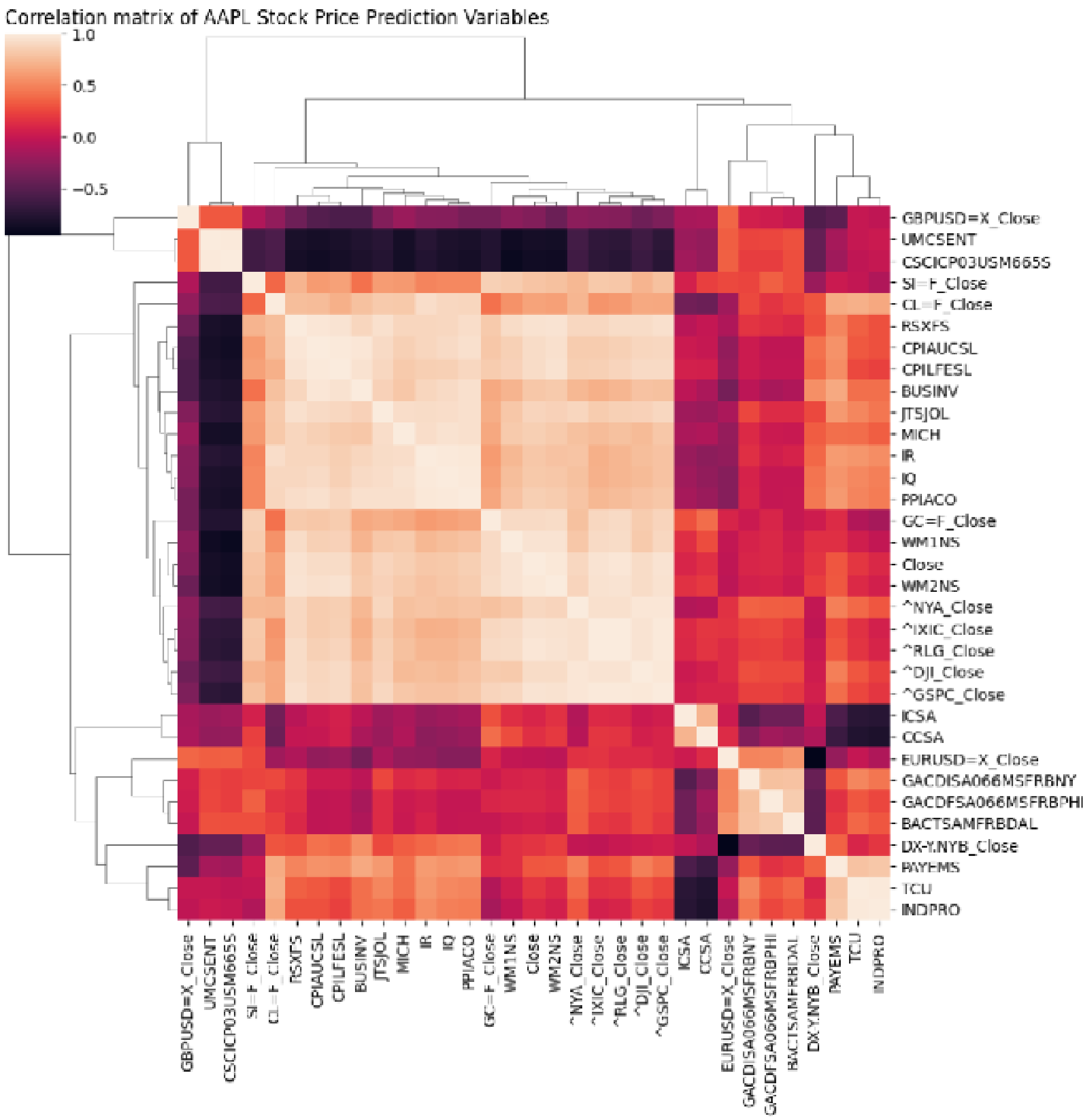


Fig. 7: The Correlation of the Selected Features

4.2. Numerical data processing

In this case, the numerical data primarily consists of stock prices from the S&P 100 index and general market indicators. they have been dealt with separately.

4.2.1. stock prices from the S&P 100

Using all 100 stocks from the S&P 100 index as features would be excessive and might lead to overfitting. To address this issue, we calculated the correlation between the prices of these stocks and our target stock (AAPL Table 15). Based on the results, we selected the top 10 stocks with the highest correlation to be included as features in our subsequent dataset.

Table 15 presents the correlation coefficients for each stock with respect to AAPL. As we can see, many stocks exhibit weak correlations with the target stock, suggesting that their prices might not contribute significantly to the prediction of AAPL's price. Therefore, we decided to exclude these less relevant stocks to reduce the dimensionality of our dataset and focus on the most important features. The final stock price features we used after screening were the prices of the following stocks:TMO & MSFT & DHR & TSLA &LOW &NEE & AVGO &LIN & COST & ACN. The features of numerical data types can be represented by the following equation4.

$$Features_{numerical} = concat(Features_{S\&Ptop10}, Features_{generalmarketindicators)} \tag{4}$$

4.3. Data Transformation

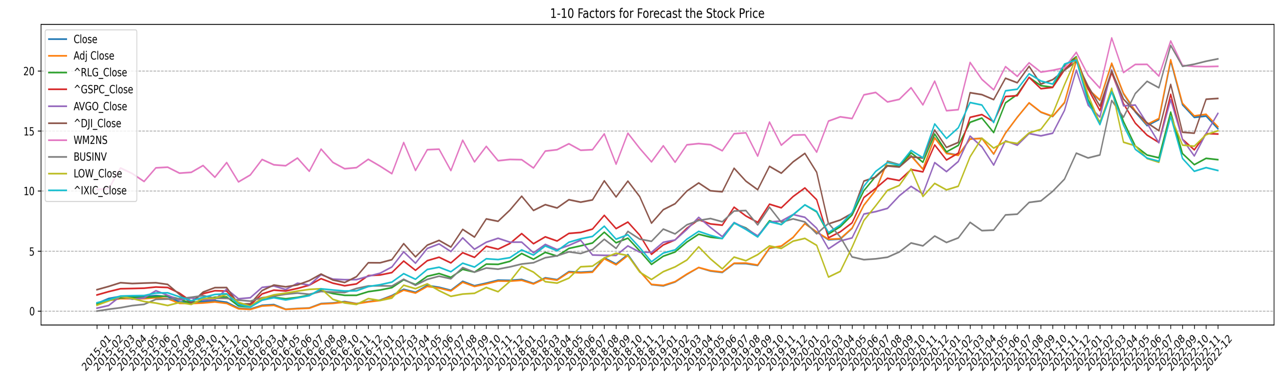
For the numerical data, we performed essential data-cleaning processes, including handling missing values and duplicates. In this particular case, our dataset exhibited very few missing values. To address these, we utilized mean imputation, replacing any missing data points with the mean value of the respective feature.

Additionally, we conducted an outlier analysis to identify data points that deviated significantly from the norm. As the Equation 5, Here x is the data value, μ is the mean and σ is the standard deviation of the data. The value is classified to be an outlier if the absolute value of the z-score is higher than 3 since the Z-score in normal distribution range from 3 to -3. The outliers were replaced by the maximum or minimum values after the removal of the outliers. Although in reality, extreme price fluctuations might indeed represent specific market conditions, we decided to handle outliers to prioritize the model's generalizability rather than focusing on predicting exceptional events (such as drastic price increases or decreases). This approach aims to strike a balance between capturing essential market trends and avoiding overfitting to rare events.

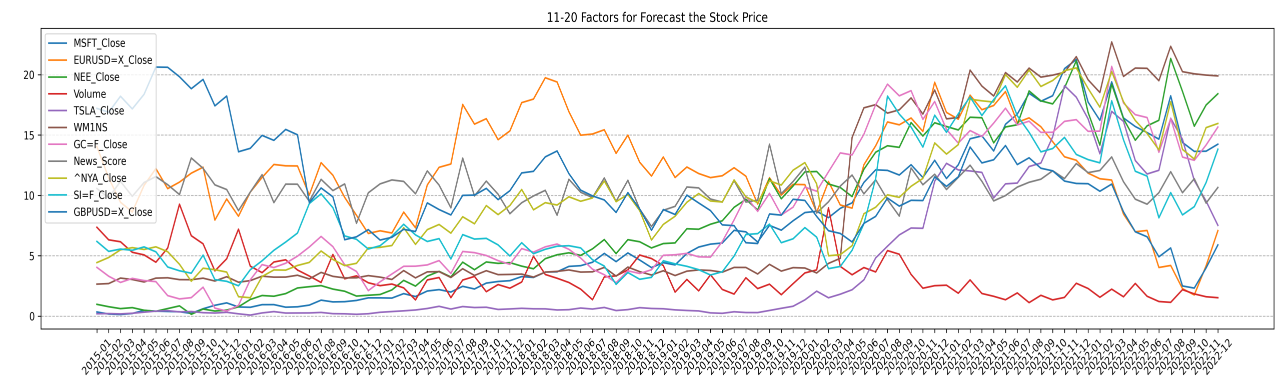
$$z = \frac{x - \mu}{\sigma} \tag{5}$$

Normalisation was carried out according to equation 6. The maximum minimum feature normalisation is used to scale the data into the range between 0 and 1.

$$x = \frac{x - X_{\min}}{X_{\max} - X_{\min}} \tag{6}$$



(a)



(b)

Fig. 8: The selected 20 Factors used to forecast the future stock price (a) the first 10 features, (b) the 11-20 features

4.4. Feature Selection

As observed in Fig 4, our current set of features includes many less important variables. To address this, we employed multiple methods, such as Gradient Boosting, Random Forest, Linear Regression, and Ridge Regression, to score the importance of each feature.

Gradient Boosting is an ensemble technique that builds multiple weak learners (typically decision trees) sequentially, with each new tree focusing on correcting the errors made by the previous one. Random Forest is another ensemble method that constructs numerous decision trees and combines their outputs to improve prediction accuracy and reduce overfitting. Linear Regression is a simple method that models the relationship between a dependent variable and one or more independent variables. Ridge Regression, a variation of Linear Regression, introduces regularization to prevent overfitting and reduce the impact of less important features.

After evaluating feature importance using these methods, we selected the top 20 features based on the Random Forest results as the final features for our dataset. This approach aims to retain the most influential variables while reducing dimensionality and noise caused by less important features.

Fig 7 below shows the final features and we can see that after feature selection all features can be more useful in predicting the target stock price.

Fig 8 shows the trend of the filtered features, and we can see that a lot of clutter has been removed compared to the EDA section3.

This concludes the data pre-processing section, where we combine textual(sentiment score) and numerical features to produce the final dataset features7.

$$Features_{finaldataset} = concat(Features_{sentimentscore}, Features_{processednumericalfeatures}) \tag{7}$$

5. DATA INFERENCE

In our study to forecast future stock prices, we propose a total of seven neural network models. We start with two base models utilizing LSTM and GRU for extracting general patterns from time sequences. Due to the numerous possible variables influencing economic phenomena, we aim for our models to focus on the appropriate features at the right time. To better fit the dataset, we integrate attention mechanisms and convolutional layers into our models. Additionally, we examine the performance of the Temporal Convolutional Neural Network (TCN) to minimize information loss from LSTM-based models. As feature representation predominantly relies on current observations, resulting in a lack of global perspective and long-term time dependencies, we propose the Transformer and the Sample Convolution and Interaction Network (SCINet) to enhance global information extraction.

To improve model performance in forecasting, we employ a technique where the model's output is added to the latest stock price. Instead of predicting stock prices directly, we aim to forecast changes in stock prices relative to the latest price:

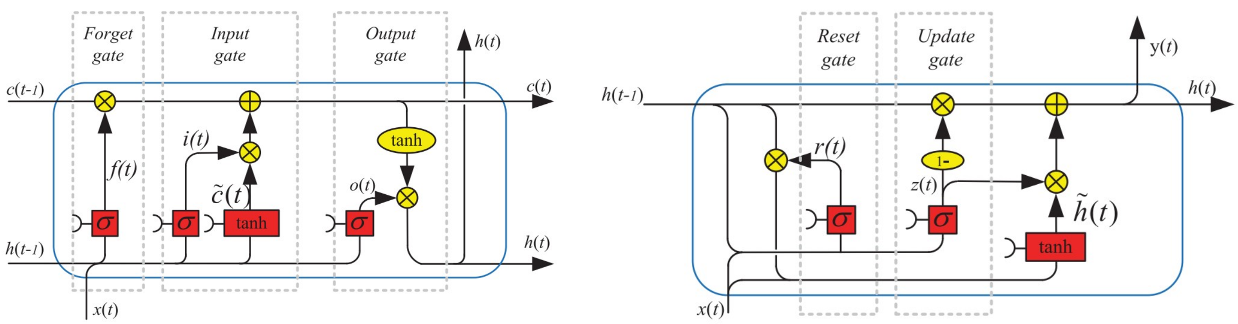
$$output = tanh(f_\theta(x)) + Latest\ Stock\ Price \tag{8}$$

Where x is the input features and θ is the model parameters. This approach is applied to all proposed deep learning models.

Two loss functions are utilised for training the forecasting models the mean absolute error (MAE) and mean square error (MSE):

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$$
$$MSE == \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

Due to the case sensitivity of the training results with various model structures, the training results with the two loss function are evaluated through the grid search to decide the more suitable loss function to train a specific model.



(a) (b)

Fig. 9: RNN Cell Architecture [5]: (a) LSTM and (b) GRU

5.1. LSTM & GRU

Long Short-Term Memory (LSTM) [6] is designed to handle the problem of vanishing gradient in traditional RNNs by using constant error carousels to enforce a constant error flow within special cells and the additive computation in the gates. It is accomplished by the architecture with forget gate $\mathbf{f}_t$, input gate $\mathbf{i}_t$ and output gate $\mathbf{o}_t$. The output of these gates is encoded based on the last hidden state h_{t-1} and the current input $\mathbf{x}_t$. The forget gate discards the irrelevant information from the past cell state c_{t-1}. The input gate decides the input features to encode to the cell state c_t by combining the weight h_{t-1} and $\mathbf{x}_t$. The cell state donated by the input and forget gate is mapped by the output gate to the hidden state of the cell.

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{I}_x\mathbf{x}_t + \mathbf{I}_h\mathbf{h}_{t-1} + \mathbf{b}_i)\\
\mathbf{f}_t &= \sigma(\mathbf{F}_x\mathbf{x}_t + \mathbf{F}_h\mathbf{h}_{t-1} + \mathbf{b}_f)\\
\mathbf{o}_t &= \sigma(\mathbf{O}_x\mathbf{x}_t + \mathbf{O}_h\mathbf{h}_{t-1} + \mathbf{b}_o)\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot g(\mathbf{C}_x\mathbf{x}_t + \mathbf{C}_h\mathbf{h}_{t-1} + \mathbf{b}_c)\\
\mathbf{h}_t &= \mathbf{o}_t \odot g(\mathbf{c}_t)
\end{aligned}$$

where $\mathbf{h}_t$ is the hidden information to the next cell, $\mathbf{c}_t$ is the current output of the cell, $\mathbf{i}_t$, $\mathbf{f}_t$ and $\mathbf{o}_t$ represents the input gate, forget gate and the output gate, σ is the element-wise logistic sigmoid function and g is an element-wise nonlinearity. An MLP layer is added to weigh the features before it is fed into the LSTM.

$$\begin{aligned}
\mathbf{r}_t &= \sigma(\mathbf{R}_x \mathbf{x}_t + \mathbf{R}_h \mathbf{h}_{t-1} + \mathbf{b}_i) \\
\mathbf{z}_t &= \sigma(\mathbf{Z}_x \mathbf{x}_t + \mathbf{Z}_h \mathbf{h}_{t-1} + \mathbf{b}_z) \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{r}_t \odot \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t
\end{aligned}$$

where $\mathbf{h}_{t-1}$ is the hidden information from the previous cell, $\mathbf{r}_t$, $\mathbf{z}_t$ and $\mathbf{o}_t$ represents the input gate and the output gate, $\mathbf{W}$ and $\mathbf{U}$ are the weights.

Based on the LSTM structure, Gated recurrent unit (GRU) [7] cell integrates the forget gate to the input gate to reduce the parameter and improve the training efficiency. The performance of the LSTM and GRU is case dependent and LSTM generally extracts information better for high complexity information. Hence, the results generated from LSTM and GRU are evaluated and compared.

A classical RNN model computes the time sequence through LSTM or GRU directly. In the model architecture, an MLP layer is added to weigh the features before it is fed into the LSTM or GRU as an embedding layer to weigh the relevance of and balance the importance of the features. Two MLP layers are used to estimate the price change from the latest price.

A dropout layer is added between the MLP layers to reduce overfitting. It is considered that not all of the output from the neuron in the first layer are correlated to the neuron in the second layer. The dropout layer is not used between the LSTM layers for a sufficient combination of the past and current features. The feature information can be better preserved with the dropout layer between the fully connected layers to remove the irrelevant information. The output is activated by the $tanh$ activation function.

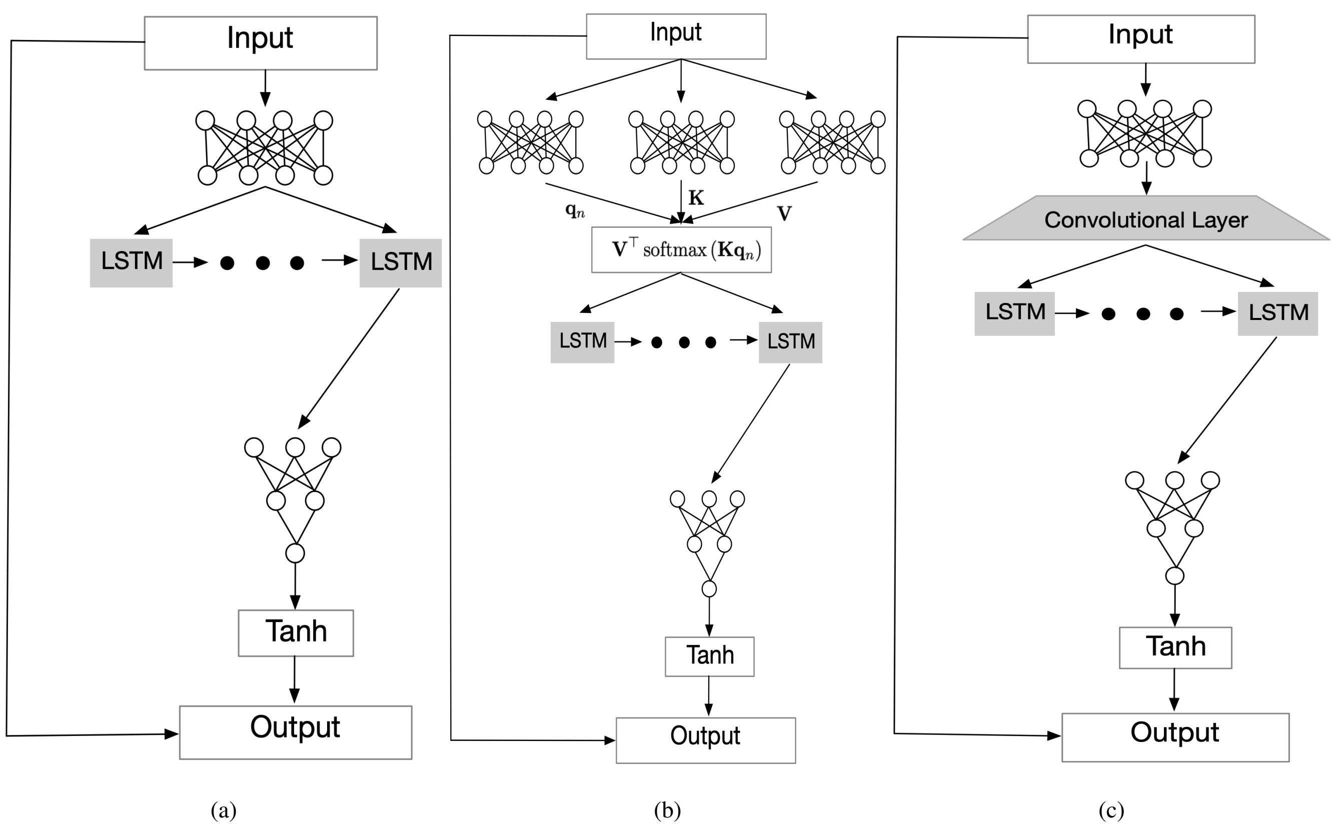


Fig. 10: Proposed Model Structures (a) LSTM, (b) AttLSTM, (c) CONVLSTM

5.2. LSTM with Attention and Convolutional Layer

The LSTM layer views the whole input feature to be equally important, while to forecast the stock price the model is required to focus more on the more suitable features at the correct time. Compared with the proposed LSTM with an MLP embedding layer, the attention mechanism helps the model to analyse the importance pattern of the input by imposing weight on the different features based on the query information which is extracted from the elements in the current state. Hence it improves the data fitting on the training set. A Query-Key-Value attention layer is added to the LSTM tagger to further weight and encode the features:

$$\begin{aligned} \operatorname{Attn}\left(\mathbf{q}_n, \mathbf{K}, \mathbf{V}\right) &= \sum_{m=1}^{M} \frac{\exp\left(\mathbf{q}_n^{\top} \mathbf{k}_m\right)}{\sum_{m'=1}^{M} \exp\left(\mathbf{q}_n^{\top} \mathbf{k}_{m'}\right)} \mathbf{v}_m^{\top} \\ &:= \mathbf{V}^{\top} \operatorname{softmax}\left(\mathbf{K} \mathbf{q}_n\right) \end{aligned}$$

Where Q, K, V are computed by three MLP layers. In the proposed LSTM with attention (AttLSTM), the model structure is similar to the GRU and LSTM models. The output feature from the LSTM is processed with MLPs and feature dropout.

Furthermore, we proposed an LSTM neural network with a convolutional layer for mapping the feature into a higher dimension. The convolution layer can help to extract the features using a learnable kernel sliding over the input sequence to find some general patterns. In our implementation, we have multiple features across the time series. Each convolution kernel analyses the features independently and maps them into a higher channel which improves the local and global feature representation.

The proposed convolutional LSTM (CONVLSTM) utilise a kernel with the size of 1×5 to capture two days of features before and after the current day and hence the padding size of 2 is assigned to the sequence. The feature is multiplied by the kernel matrix to perform cross-correlation and filtering information. A similar structure is used in the part after the feature is processed by the convolutional layer and the LSTM. Two layers of MLP with a ReLU activation function are used to compute the variation of the price. The output value is activated by the sink activation function to promote a more delicate change in the price.

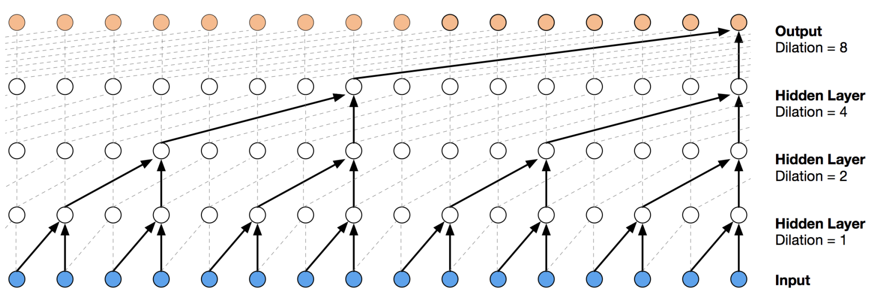


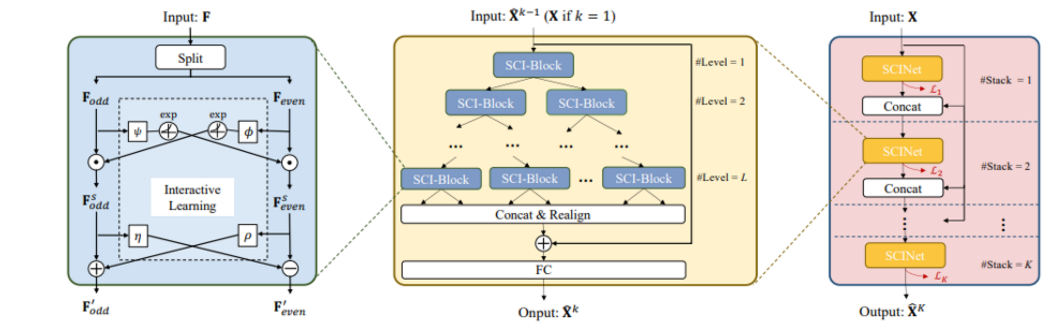
Fig. 11: TCN Architecture

5.3. TCN

A temporal convolutional network (TCN) [8] is a 1D convolutional neural network proposed to analyse the time sequence. It uses causal convolutions to combine both past and current information. Compared with the original kernel which combines both the past and future features into the current element, the causal convolutional kernel is proposed for the model to focus on the past elements. Compared to LSTM, TCN better represents long-term dependencies through dilated convolutions.

Due to the shifted kernel in classical convolutional networks for 1D sequence analysis, output representation tends to average features within the kernel size. In addition to the dilated convolutional kernel, TCN use the residual connection to reduce the vanishing gradient.

In the implemented model structure, the TCN consists of a stack with a convolutional layer followed by a ReLU activation and a drop out layer. Each block instead TCN has two stacks and a downsampling layer using convolution. Multiples of blocks which have different dilated values are used on the trained model (Fig11). The output of the final stack is mapped by a fully connected layer to generate the output.

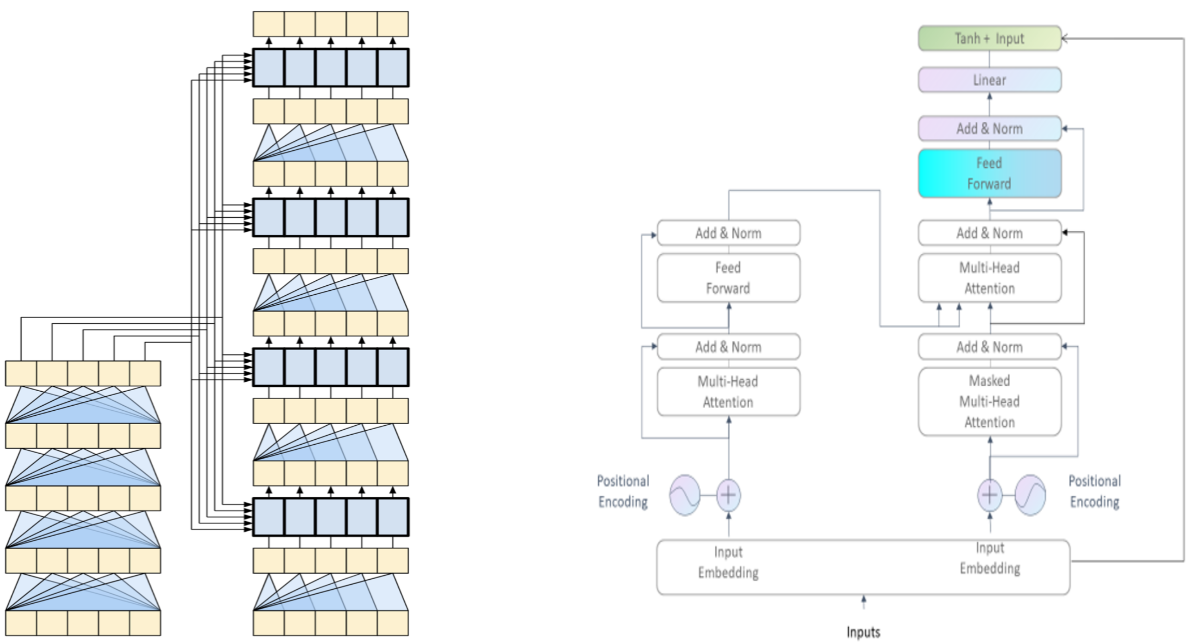


(a) SCI-Block (b) SCINet (c) Stacked SCINet

Fig. 12: SCINet Architecture

5.4. SCINet

Sample convolution and interaction for temporal modelling and forecasting (SCINet) [9] is proposed to improve kernel interaction and global feature representation. A tree structure is used in the SCINet to process the feature in multi-resolution and concatenated all of the nodes in the leaf layer. The split feature is processed by the SCI block which splits the features sequence into odd and even matrices which are projected into a latent space independently and then processed with summing and difference respectively to interact with the features. The ϕ, psi, η and ρ (Fig12) are four convolutional process consists of 1D convolution, LeakyReLU, dropout, 1D convolution and Tanh activation in the end. In the implementation of SCINet, one SCINet tree is used for encoding the feature followed by a 1D convolutional layer and a fully connected layer to project the forecast.



(a) (b)

Fig. 13: Proposed Model Structures (a) Transformer Block Structure, (b) Transformer Architecture

5.5. Transformer

Stock price forecasting can be treated by generating the next value with the highest possibility solved by the sequential value generation. The transformer is proposed to selectively focus on different parts of the input sequence, such as the most relevant financial indicators or past stock prices to make predictions.

Transformers[10] is a sequence-to-sequence model with an encoder-decoder architecture. The encoder and decoder are built by blocks connected with the residual connection. It uses multiple attention mechanisms using MLPs to analyse the sequence and generated answers. The input feature is firstly encoded using the positional embedding added to the feature map. The features are then fed into the encoder which used the multi-head self-attention and point-wise feed-forward layers to find the feature relevance in the sequence. The encoded representation from the multi-head attention from the side block is computed as the key and value for the cross attention to combine the feature map.

Compared with the classical transformer structure, the positional embedding is removed as it worsens model performance. Time patterns for stocks are primarily annual rather than monthly, making them unclear within the set window size. Features are then encoded by the Transformer's encoder with multi-head attention and fed to two MLP layers with ReLU for price forecasting.

5.6. Machine Learning Methods

Two machine learning techniques, linear regression and support vector machine are selected for the relatively high accuracy from the validation.

5.6.1. Linear Regression

Linear regression (LR) models the relationship between the dependent variables with the actual value represented by a straight line.:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n, \tag{9}$$

LR aims to minimise the sum of squared differences between the actual values and the predicted values.

5.6.2. Linear SVR

Support Vector Regression (SVR) compute the hyperplane best fits the data points which minimises the loss. It is optimised through the hinge loss which contains the L2 penalty:

$$\frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i(\mathbf{w}^T \mathbf{x} - b)\right) + \lambda \|\mathbf{w}\|^2 \tag{10}$$

To convert the non-linear regression to linear regression, SVR uses the kernel method, expressed from Mercer's Theorem $k(u, v) = \phi(u)\phi(v)$ to map the feature to a higher dimensional space. Linear, homogeneous polynomial (Poly), sigmoid and Gaussian radial basis function (RBF) are the kernels commonly used to separate the data margin. The linear kernel is used for the closest price forecasted: $k_{poly}(u, v)$=$(ak\,(u, v) + b)$..

6. RESULTS

This section begins with a thorough investigation of the impact of various hyper-parameters on the performance of seven deep learning models through a comprehensive grid search. Subsequently, we present quantitative and qualitative comparisons between these models and four traditional machine learning models in the context of time series forecasting, employing evaluation metrics and prediction curves. Additionally, we propose a novel metric, namely averaged trading return, to assess the efficacy of different models by utilizing the forecasted price.

6.1. Experiemntal Setup

6.1.1. Dataset Split

The dataset utilized in this study pertains to the stock prices of AAPL, spanning from January 2015 to December 2022. It is pertinent to note that the dataset was partitioned into three distinct subsets, namely training set (60%), validation set (20%), and test set (20%), in a chronological order.

6.1.2. Variables for Prediction

We conduct experiments under three different settings: using only the stock price variables, with initial extra data variables and with selected features from extra data variables.

6.1.3. *Predicted Length*

For the purpose of enhancing the comparability of various models, a consistent predicted length of 15 was adopted in all experiments. This length implies that the models were required to forecast closing prices for the ensuing 15 days.

6.1.4. *Metrics*

We used four conventional evaluation metrics defined as:
1. Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \sum_{(i,t)\in\Omega} \left| Y_{it} - \hat{Y}_{it} \right| \tag{11}$$

2. Root of Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{(i,t)\in\Omega} \left(Y_{it} - \hat{Y}_{it} \right)^2} \tag{12}$$

3. Absolute Error (AE)

$$AE = \sum_{(i,t)\in\Omega} \left| Y_{it} - \hat{Y}_{it} \right| \tag{13}$$

4. Correlation (CORR)

$$\mathrm{CORR} = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_t \left(Y_{it} - \mathrm{mean}\left(Y_i\right)\right)\left(\hat{Y}_{it} - \mathrm{mean}\left(\hat{Y}_i\right)\right)}{\sqrt{\sum_t \left(Y_{it} - \mathrm{mean}\left(Y_i\right)\right)^2 \left(\hat{Y}_{it} - \mathrm{mean}\left(\hat{Y}_i\right)\right)^2}} \tag{14}$$

For MAE, RMSE and AE, lower value is better, while for CORR higher value is better.

6.2. Grid Search

A grid search was employed to tune the hyper-parameters (HiddDim, Layers, Loss, and Optimiser) for each neural network model using a held-out validation set. The range of tunable values for HiddDim was fixed across all models, with values ranging from {64,96,128,256}. Similarly, the range of tunable values for Loss was limited to two choices: L1 loss (MAE loss) and L2 loss (RMSE loss). The optimiser was chosen from either Adam or SGD. For GRU, LSTM, and TCN models, the number of layers was selected from a range of {1,3,4}. For AttLSTM and SCINet models, the number of layers was chosen from a range of {1,2,3}. Finally, for ConvLSTM and Transformer models, the number of layers was tuned from a range of {1,3,5}. The grid search was performed under the experimental setting that included extra data. Results of the grid search were reported in Table 1-7 for each model, indicating the optimal hyper-parameters for each.

6.2.1. *GRU*

.

Table 1: The grid search results evaluated on the test set with GRU and measured on the root mean square error (RMSE) and mean absolute error (MAE), where ¶ marks the adam optimiser, § marks the SGD optimiser, L1 represents the MAE loss and L2 represents the MSE loss function

Hidden Dim	**64**		**96**		**128**		**256**	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
1 Layer L1 ¶	0.05143	0.04061	2.64642	2.59030	1.43083	1.40799	0.05353	0.04291
1 Layer L2 ¶	1.42236	1.41004	0.08125	0.06860	3.35596	3.33121	0.09947	0.08094
1 Layer L1 §	0.48209	0.47636	0.50241	0.49662	0.54659	0.54182	0.59206	0.58700
1 Layer L2 §	0.19700	0.19052	0.36318	0.35753	0.18541	0.17657	0.24744	0.24175
3 Layer L1 ¶	0.05657	0.04339	1.90180	1.86357	0.05365	0.04171	0.10064	0.09008
3 Layer L2 ¶	0.10513	0.09424	0.07153	0.06066	4.00639	3.97564	0.11199	0.09053
3 Layer L1 §	nan	nan	nan	nan	nan	nan	nan	nan
3 Layer L2 §	nan	nan	0.17036	0.16241	0.11733	0.10636	0.11757	0.10722
4 Layer L1 ¶	0.05309	0.04172	1.78844	1.76535	0.17775	0.17101	0.16035	0.13503
4 Layer L2 ¶	0.06269	0.05220	0.34372	0.33652	0.10633	0.09169	0.09306	0.07777
4 Layer L1 §	nan	nan	nan	nan	nan	nan	nan	nan
4 Layer L2 §	0.16028	0.15246	0.16707	0.15928	0.16548	0.15726	0.14390	0.13480

6.2.2. LSTM

Table 2: The grid search results evaluated on the test set with LSTM and measured on the root mean square error (RMSE) and mean absolute error (MAE), where ¶ marks the adam optimiser, § marks the SGD optimiser, L1 represents the MAE loss and L2 represents the MSE loss function

Hidden Dim	**64**		**96**		**128**		**256**	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
1 Layer L1 ¶	0.12141	0.11167	0.06249	0.05236	0.05812	0.04758	0.12880	0.11952
1 Layer L2 ¶	0.07649	0.06487	0.05451	0.04354	0.05515	0.04351	0.06968	0.05315
1 Layer L1 §	0.49709	0.49203	0.57353	0.56855	0.51778	0.51178	0.65041	0.64497
1 Layer L2 §	0.17848	0.17173	0.24310	0.23752	0.18606	0.17938	0.25913	0.25273
3 Layer L1 ¶	0.09327	0.07756	0.08680	0.07480	0.08838	0.07727	0.13742	0.12337
3 Layer L2 ¶	0.05176	0.04078	0.14471	0.13659	0.36947	0.36454	0.06818	0.05413
3 Layer L1 §	0.38135	0.37508	0.54424	0.53128	0.70489	0.69693	nan	nan
3 Layer L2 §	nan	nan	nan	nan	nan	nan	nan	nan
4 Layer L1 ¶	0.06951	0.05562	0.04933	0.03842	0.06405	0.05325	0.05083	0.03964
4 Layer L2 ¶	0.05431	0.04301	0.09277	0.07810	0.05681	0.04632	0.05065	0.03954
4 Layer L1 §	0.64627	0.64008	0.72366	0.71571	nan	nan	0.74761	0.73580
4 Layer L2 §	nan	nan	nan	nan	nan	nan	nan	nan

6.2.3. AttLSTM

Table 3: The grid search results evaluated on the test set with AttLSTM and measured on the root mean square error (RMSE) and mean absolute error (MAE), where ¶ marks the adam optimiser, § marks the SGD optimiser, L1 represents the MAE loss and L2 represents the MSE loss function

Hidden Dim	**64**		**96**		**128**		**256**	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
1 Layer L1 ¶	0.12000	0.05694	0.04920	0.03836	0.82380	0.23189	0.04920	0.03835
1 Layer L2 ¶	0.07582	0.04818	0.05000	0.03872	5.18593	3.17961	0.16500	0.11956
1 Layer L1 §	0.05298	0.04035	0.04987	0.03865	0.05203	0.03971	0.04937	0.03849
1 Layer L2 §	0.04919	0.03836	0.04920	0.03836	0.04919	0.03836	0.04919	0.03836
2 Layer L1 ¶	0.04921	0.03837	0.04919	0.03836	0.04923	0.03833	0.04920	0.03834
2 Layer L2 ¶	0.04918	0.03833	0.04920	0.03835	0.04922	0.03837	0.05194	0.04025
2 Layer L1 §	0.04922	0.03833	0.04922	0.03834	0.04924	0.03834	0.04923	0.03833
2 Layer L2 §	0.04919	0.03835	0.04920	0.03836	0.04920	0.03836	0.04920	0.03836
3 Layer L1 ¶	0.04922	0.03834	0.04919	0.03835	0.04919	0.03835	1.02942	0.52969
3 Layer L2 ¶	0.04919	0.03836	0.04919	0.03834	0.04920	0.03836	0.04918	0.03835
3 Layer L1 §	0.04919	0.03832	0.04919	0.03833	0.04919	0.03833	0.04919	0.03830
3 Layer L2 §	0.04920	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836

6.2.4. ConvLSTM

Table 4: The grid search results evaluated on the test set with ConvLSTM and measured on the root mean square error (RMSE) and mean absolute error (MAE), where ¶ marks the adam optimiser, § marks the SGD optimiser, L1 represents the MAE loss and L2 represents the MSE loss function

Hidden Dim	**64**		**96**		**128**		**256**	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
1 Layer L1 ¶	0.04919	0.03830	0.04919	0.03830	0.04919	0.03831	0.04919	0.03833
1 Layer L2 ¶	0.04917	0.03830	0.04919	0.03836	0.04917	0.03830	0.04917	0.03829
1 Layer L1 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
1 Layer L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
3 Layer L1 ¶	0.04919	0.03833	0.04918	0.03831	0.04919	0.03831	0.04919	0.03832
3 Layer L2 ¶	0.04917	0.03829	0.04917	0.03830	0.04917	0.03830	0.05106	0.03998
3 Layer L1 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
3 Layer L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
5 Layer L1 ¶	0.04919	0.03829	0.04919	0.03833	0.04919	0.03831	0.04919	0.03831
5 Layer L2 ¶	0.04918	0.03832	0.04919	0.03834	0.04918	0.03834	0.04917	0.03829
5 Layer L1 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
5 Layer L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836

6.2.5. TCN

Table 5: The grid search results evaluated on the test set with TCN and measured on the root mean square error (RMSE) and mean absolute error (MAE), where ¶ marks the adam optimiser, § marks the SGD optimiser, L1 represents the MAE loss and L2 represents the MSE loss function

Hidden Dim	**64**		**96**		**128**		**256**	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
1 Layer L1 ¶	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
1 Layer L2 ¶	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
1 Layer L1 §	0.14709	0.12644	0.09777	0.07650	0.15062	0.13134	0.08391	0.06411
1 Layer L2 §	0.04919	0.03834	0.04918	0.03834	0.04922	0.03837	0.04921	0.03835
3 Layer L1 ¶	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
3 Layer L2 ¶	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
3 Layer L1 §	0.09721	0.07801	0.10834	0.08891	0.09416	0.07559	0.11509	0.09576
3 Layer L2 §	0.04922	0.03836	0.04918	0.03833	0.04920	0.03833	0.04919	0.03834
4 Layer L1 ¶	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
4 Layer L2 ¶	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
4 Layer L1 §	0.08570	0.06723	0.12135	0.10256	0.09684	0.07833	0.08966	0.07013
4 Layer L2 §	0.04919	0.03833	0.04924	0.03837	0.04918	0.03833	0.04919	0.03834

6.2.6. SCINet

Table 6: The grid search results evaluated on the test set with SCINet and measured on the root mean square error (RMSE) and mean absolute error (MAE), where ¶ marks the adam optimiser, § marks the SGD optimiser, L1 represents the MAE loss and L2 represents the MSE loss function

Hidden Dim	**64**		**96**		**128**		**256**	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
1 Layer L1 ¶	0.15352	0.11719	0.24401	0.21123	0.08275	0.06703	0.06717	0.05288
1 Layer L2 ¶	0.07293	0.05912	0.10382	0.08341	0.08529	0.06834	0.06510	0.05276
1 Layer L1 §	0.09982	0.08620	0.08512	0.06821	0.07305	0.06111	0.05992	0.04908
1 Layer L2 §	0.15204	0.11977	0.07710	0.06134	0.14867	0.11399	0.12694	0.10516
2 Layer L1 ¶	0.09378	0.07118	0.21325	0.17122	0.42551	0.36837	0.06410	0.04924
2 Layer L2 ¶	0.06971	0.05552	0.12239	0.10657	0.10072	0.07992	0.11943	0.09653
2 Layer L1 §	0.07438	0.05829	0.07901	0.06276	0.06125	0.04825	0.09080	0.07569
2 Layer L2 §	0.31604	0.24932	0.15078	0.12590	0.21194	0.17816	0.11930	0.09442
3 Layer L1 ¶	0.07788	0.06442	0.07141	0.05725	0.35776	0.28610	0.13169	0.11101
3 Layer L2 ¶	0.65549	0.56910	0.18135	0.14687	0.16494	0.13148	0.12467	0.10013
3 Layer L1 §	0.11748	0.09932	0.09917	0.07973	0.11351	0.09621	0.14693	0.11549
3 Layer L2 §	0.17092	0.14129	0.17914	0.14581	0.17603	0.14310	0.13166	0.10282

6.2.7. *Transformer*

Table 7: The grid search results evaluated on the test set with Transformer and measured on the root mean square error (RMSE) and mean absolute error (MAE), where ¶ marks the adam optimiser, § marks the SGD optimiser, L1 represents the MAE loss and L2 represents the MSE loss function

Hidden Dim	**64**		**96**		**128**		**256**	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
1 Layer 3 Heads L1 ¶	0.04918	0.03829	0.04918	0.03828	0.04918	0.03829	0.04918	0.03832
1 Layer 3 Heads L2 ¶	0.04918	0.03832	0.04917	0.03830	0.04918	0.03832	0.04917	0.03830
1 Layer 3 Heads L1 §	0.04919	0.03833	0.04919	0.03836	0.04919	0.03834	0.04923	0.03833
1 Layer 3 Heads L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
1 Layer 5 Heads L1 ¶	0.04918	0.03830	0.04918	0.03829	0.04918	0.03829	0.04919	0.03832
1 Layer 5 Heads L2 ¶	0.04919	0.03834	0.04918	0.03831	0.04918	0.03834	0.04917	0.03829
1 Layer 5 Heads L1 §	0.04919	0.03836	0.04920	0.03834	0.04920	0.03834	0.04920	0.03828
1 Layer 5 Heads L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
1 Layer 15 Heads L1 ¶	0.04919	0.03833	0.04919	0.03833	0.04919	0.03831	0.04919	0.03833
1 Layer 15 Heads L2 ¶	0.04918	0.03834	0.04917	0.03830	0.04917	0.03829	0.04917	0.03831
1 Layer 15 Heads L1 §	0.04921	0.03830	0.04919	0.03835	0.04941	0.03845	0.04942	0.03845
1 Layer 15 Heads L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
3 Layer 3 Heads L1 ¶	0.04918	0.03828	0.04918	0.03830	0.04918	0.03828	0.04918	0.03828
3 Layer 3 Heads L2 ¶	0.04918	0.03832	0.04917	0.03830	0.04918	0.03833	0.04917	0.03829
3 Layer 3 Heads L1 §	0.04921	0.03834	0.04935	0.03833	0.05138	0.03957	0.04919	0.03832
3 Layer 3 Heads L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
3 Layer 5 Heads L1 ¶	0.04918	0.03828	0.04918	0.03831	0.04918	0.03831	0.04919	0.03831
3 Layer 5 Heads L2 ¶	0.04918	0.03833	0.04917	0.03830	0.04917	0.03830	0.04917	0.03829
3 Layer 5 Heads L1 §	0.04921	0.03834	0.04948	0.03842	0.04919	0.03836	0.04919	0.03836
3 Layer 5 Heads L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
3 Layer 15 Heads L1 ¶	0.04918	0.03830	0.04919	0.03831	0.04918	0.03828	0.04918	0.03828
3 Layer 15 Heads L2 ¶	0.04917	0.03829	0.04917	0.03829	0.04917	0.03829	0.04918	0.03831
3 Layer 15 Heads L1 §	0.04919	0.03832	0.04974	0.03856	0.04962	0.03862	0.04920	0.03835
3 Layer 15 Heads L2 §	0.04920	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
5 Layer 3 Heads L1 ¶	0.04918	0.03828	0.04918	0.03831	0.04918	0.03829	0.04919	0.03830
5 Layer 3 Heads L2 ¶	0.04917	0.03829	0.04917	0.03830	0.04917	0.03830	0.04918	0.03831
5 Layer 3 Heads L1 §	0.04919	0.03832	0.04919	0.03836	0.04921	0.03836	0.04919	0.03836
5 Layer 3 Heads L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
5 Layer 5 Heads L1 ¶	0.04919	0.03830	0.04918	0.03828	0.04919	0.03832	0.04918	0.03829
5 Layer 5 Heads L2 ¶	0.04917	0.03829	0.04917	0.03830	0.04917	0.03831	0.04917	0.03830
5 Layer 5 Heads L1 §	0.04923	0.03834	0.04919	0.03836	0.04919	0.03832	0.04927	0.03836
5 Layer 5 Heads L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836
5 Layer 15 Heads L1 ¶	0.04919	0.03831	0.04918	0.03828	0.04918	0.03828	0.04919	0.03831
5 Layer 15 Heads L2 ¶	0.04918	0.03833	0.04917	0.03831	0.04918	0.03831	0.04917	0.03830
5 Layer 15 Heads L1 §	0.04918	0.03832	0.04919	0.03836	0.04919	0.03832	0.04919	0.03834
5 Layer 15 Heads L2 §	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836	0.04919	0.03836

6.3. Fine Tuned Model Parameters

Through the use of grid search, the hyper-parameters (HiddDim, Layers, Loss, and Optimiser) of seven neural network models were fine-tuned. The resulting optimal parameters are presented in Table 8. Additionally, the optimal learning rate (LR) was determined to be 0.0001 for SCINet and 0.0005 for the remaining models. The optimal number of heads for Transformer was evaluated to be 5.

Table 8: The Selected Hyper-parameters for Each Model

	HiddDim	Layers	Loss	Optimiser
GRU	128	2	L1	Adam
LSTM	128	2	L1	Adam
AttLSTM	128	2	L1	SGD
CONVLSTM	96	3	L2	SGD
TCN	96	3	L2	SGD
SCINet	128	2	L1	SGD
Transformer	96	5	L1	Adam

6.4. PCA Feature Reduction

Principal Component Analysis (PCA) is a widely used technique for feature extraction and dimensionality reduction. When dealing with high-dimensional data, models may become too complex and overfit the training data, resulting in poor generalization to new data. In order to address this issue, PCA was conducted to reduce the number of features used in each model. Initially, the dataset had 49 features, but PCA was utilized to extract different numbers of features, which were then used as extra data in the prediction process. The optimal number of features was selected based on the lowest Mean Absolute Error (MAE) observed on the validation dataset.

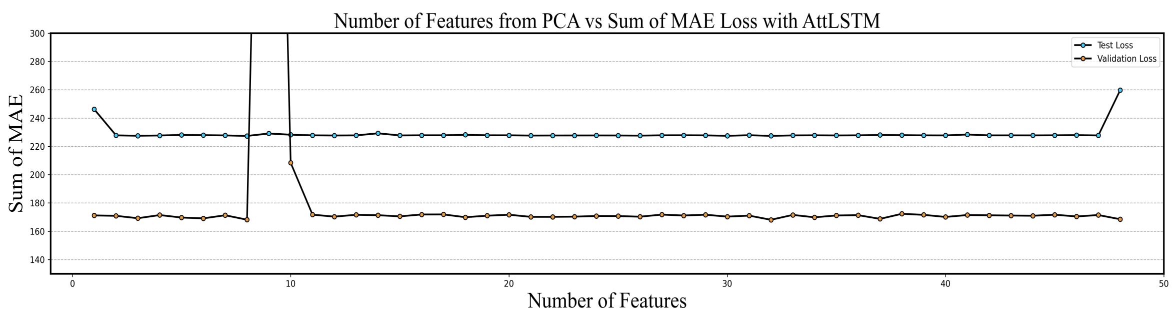


Fig. 14: Sum of the loss of the model with the input from PCA to reduce the number of features

6.5. Time Window Size

In order to assess the influence of time window size on the performance of each model, a series of experiments were conducted. The results revealed that increasing the window size did not lead to a significant improvement in model performance. This suggests that the most recent data has a greater impact on stock price prediction than historical data. Additional experimentation was carried out to investigate the effect of different time window sizes on model performance using machine learning techniques. Further details regarding the impact of various time window sizes on the sum of MAE Loss with AttLSTM can be found in Table 17, where performance was evaluated on both the validation and testing datasets.

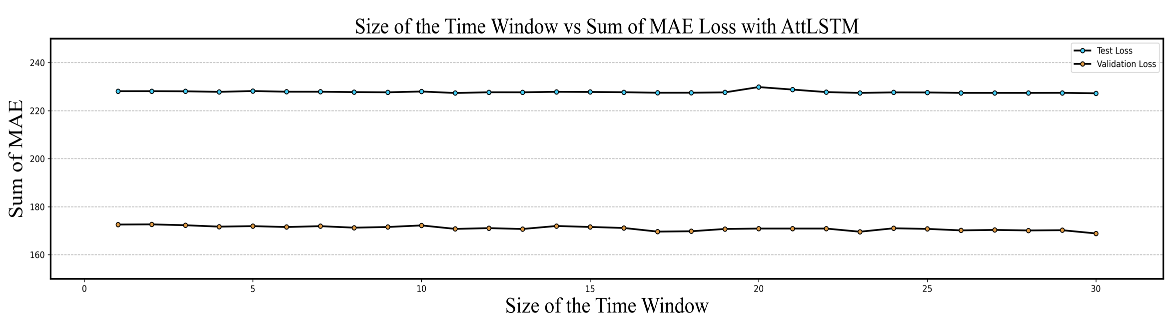


Fig. 15: Sum of the loss of the model with different input window size

In our study, we selected a window size of 15, which corresponds to the prediction length. To evaluate the performance of different models, we utilized four metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Absolute Error (AE), and Correlation (CORR). We assessed the models under three distinct settings: using only the stock price variables, with additional initial data variables, and with selected features from the extra data variables. The results of our model comparison are presented in Table 9 and Table 10. These tables display the performance metrics of each model under the three settings.

6.6. Results of Neural Network Models

The quantitive results of our study on the performance of seven neural network models in forecasting stock prices are presented in Table 9 and Table 10. For the fine-tuned models without feature selection shown in Table 9, we observed that incorporating additional data improved forecasting performance with smaller errors and larger correlations. Among these models, Transformer exhibited the best performance. In contrast, Table 10 displays that the application of feature selection enhanced model performance. We found that models utilizing selected features outperformed those without feature selection. The TCN achieved the highest accuracy among the models employing selected features, while Transformer obtained the highest accuracy using solely stock price. Our findings indicate that the TCN model using extra selected features achieved the best performance on the test dataset, with an MAE of 0.03815, RMSE of 0.04903, AE of 226.58755, and CORR of 0.99125. This outperformance of the strong neural baseline LSTM by 5.3% in the MAE metric with extra data underscores the superior predictive capabilities of TCN in stock price forecasting. Moreover, Transformer outperforms LSTM by a notable margin of 30.8% in the MAE metric when exclusively using stock price data, which highlight the potential of Transformer architecture for accurate and effective stock price forecasting.

The qualitative results of our study on seven neural network models that forecast stock prices are presented in Figures 16, 17, 18, 19, 20, 21, and 22. These figures depict the predicted stock prices on the training, validation, and test datasets, as well as the 15-day forecast results on the last test sample.

Our baseline models, GRU and LSTM, shown in Figures 16 and 17, respectively, demonstrate a rough trend of actual stock price changes across the training, validation, and testing datasets. However, there appears to be a bound on the precision between the predicted prices and actual prices. Moreover, these models perform poorly in forecasting the stock price of the last day of the 15 days, with some instances of incorrect forecasting. Improved LSTM models, AttLSTM and CONVLSTM, shown in Figures 18 and 19, respectively, perform better than their baseline counterparts. They achieve higher precision between the predicted prices and actual prices.The attention mechanism of AttLSTM helps in identifying the most informative features that should be considered while doing forecasting, thereby improving the model's ability to capture complex relationships between input and output. And the added convolutional layers of CONVLSTM can help capture temporal patterns and dependencies at multiple time scales.which can be useful for identifying important features for forecasting. However, they still do not perform well in predicting the stock price of the last day of the 15 days, often predicting it to be constant. Advanced model, TCN as shown in Figure 20, perform better than baseline models with its well designed architecture to extract feature information from external data Another advanced model, SCINet, as shown in Figure 21, does not perform much better than the baseline models. Furthermore, even though both of them use more sophisticated designs to extract relevant features from the sequence, they still struggle with predicting the stock price of the last day of the 15 days.

Our best-performing model, Transformer, shown in Figure 22, outperforms the previous models on all datasets. One of the strengths of the Transformer architecture is its ability to capture long-term dependencies in a sequence of data, which is particularly useful in time series forecasting tasks like stock price prediction. In stock price forecasting, there are often complex patterns and relationships between past and present stock prices that can be difficult for traditional neural network architectures to capture. Transformer overcomes this challenge by using self-attention mechanisms to learn the relationships between different parts of the input sequence. This allows it to identify important patterns and trends across longer time horizons than other neural network architectures.Additionally, the Transformer's multi-head attention mechanism allows it to attend to different aspects of the input sequence simultaneously, which can be especially useful when dealing with multiple stock prices and related financial metrics. Overall, these features make the Transformer architecture well-suited to the complex and dynamic nature of stock price prediction. As a result, the extracted trends and patterns are more accurate and precise compared to other models.

Table 9: The normalised normalised mean absolute error (MAE), root mean squared error (RMSE), absolute error (AE) and correlation (CORR) of the Proposed Fine Tuned DL Regression models trained with the **full features** in extra data and solely on the close price information: ⋆ marks the training set, † marks the validatioin set and ‡ marks the test set.

	MAE		RMSE		AE		CORR	
	Extra Data	Price	Extra Data	Price	Extra Data	Price	Extra Data	Price
GRU ⋆	0.01619	0.04686	0.01951	0.04940	287.95435	833.65735	0.94919	0.94919
LSTM ⋆	0.04470	0.03424	0.04672	0.04107	795.20386	609.09000	0.94919	0.94919
AttLSTM ⋆	0.00782	0.00782	0.01108	0.01108	139.04927	139.03130	0.94919	0.94919
CONVLSTM ⋆	0.00768	0.00771	0.01101	0.01103	136.66353	137.17902	0.94919	0.94919
TCN ⋆	0.00782	0.00782	0.01108	0.01108	139.13135	139.13135	0.94919	0.94919
SCINet ⋆	0.01608	0.08388	0.01954	0.17458	286.32086	1493.46240	0.94971	0.94971
Transformer ⋆	0.00765	0.00764	0.01098	0.01098	136.01611	135.85527	0.94919	0.94919
GRU †	0.03540	0.03635	0.04759	0.04802	210.27367	215.91948	0.98923	0.98923
LSTM †	0.02984	0.11347	0.04026	0.12948	177.22620	673.98285	0.98923	0.98923
AttLSTM †	0.02895	0.02895	0.03886	0.03886	171.96661	171.95328	0.98923	0.98923
CONVLSTM †	0.02860	0.02868	0.03858	0.03866	169.86945	170.33490	0.98923	0.98923
TCN †	0.02896	0.02896	0.03886	0.03886	172.01373	172.01373	0.98923	0.98923
SCINet †	0.03994	0.11527	0.05089	0.18972	236.63171	682.97186	0.98923	0.98923
Transformer †	**0.02850**	**0.02848**	**0.03849**	**0.03847**	**169.29773**	**169.14279**	**0.98923**	**0.98923**
GRU ‡	0.05557	0.07390	0.06909	0.08494	330.06372	438.96300	0.99113	0.99113
LSTM ‡	0.05054	0.22629	0.06115	0.23199	300.23325	1344.18740	0.99113	0.99113
AttLSTM ‡	0.03836	0.03836	0.04919	0.04919	227.85176	227.85144	0.99113	0.99113
CONVLSTM ‡	0.03831	0.03855	**0.04917**	0.04933	227.54874	228.97913	0.99113	0.99113
TCN ‡	0.03836	0.03836	0.04919	0.04919	227.86647	227.86647	0.99113	0.99113
SCINet ‡	0.04775	0.19034	0.05897	0.24035	282.89978	1127.79350	0.99113	0.99113
Transformer ‡	**0.03830**	**0.03829**	**0.04917**	**0.04917**	**227.49649**	**227.45349**	**0.99115**	**0.99115**

Table 10: The normalised normalised mean absolute error (MAE), root mean squared error (RMSE), absolute error (AE) and correlation (CORR) of the Proposed Fine Tuned DL Regression models trained with the **selected features** in extra data and solely on the close price information: ⋆ marks the training set, † marks the validation set and ‡ marks the test set.

	MAE		RMSE		AE		CORR	
	Extra Data	Price	Extra Data	Price	Extra Data	Price	Extra Data	Price
GRU ⋆	0.03666	0.03278	0.03911	0.03474	652.25490	583.11237	0.94919	0.94919
LSTM ⋆	0.07179	0.01938	0.07302	0.02303	1277.08130	344.78674	0.94919	0.94919
AttLSTM ⋆	0.00776	0.00776	0.01107	0.01107	138.05344	138.04976	0.94919	0.94919
CONVLSTM ⋆	0.00782	0.00782	0.01108	0.01108	139.13156	139.13156	0.94919	0.94919
TCN ⋆	0.00776	0.00826	0.01123	0.01189	138.01292	146.90071	0.94919	0.94919
SCINet ⋆	0.01884	0.06844	0.02382	0.07887	335.52510	1218.64400	0.94970	0.94970
Transformer ⋆	0.00765	0.00764	0.01104	0.01104	136.13525	135.98195	0.94919	0.94919
GRU †	0.02893	0.03656	0.03980	0.04766	171.84363	217.16066	0.98923	0.98923
LSTM †	0.04235	0.03425	0.05156	0.04445	251.57806	203.46384	0.98923	0.98923
AttLSTM †	0.02873	0.02874	0.03869	0.03870	170.68436	170.70135	0.98923	0.98923
CONVLSTM †	0.02896	0.02896	0.03886	0.03886	172.01385	172.01385	0.98923	0.98923
TCN †	**0.02798**	**0.02782**	0.03862	**0.03808**	**166.17517**	**165.25543**	**0.98923**	**0.98923**
SCINet †	0.04084	0.06731	0.05415	0.08667	241.97247	398.79086	0.99946	0.99946
Transformer †	0.02834	0.02831	**0.03836**	0.03833	168.35101	168.15358	**0.98923**	**0.98923**
GRU ‡	0.04022	0.04886	0.05118	0.06140	238.93150	290.22580	0.99125	0.99125
LSTM ‡	0.04029	0.05534	0.05113	0.06576	239.32971	328.70532	0.99125	0.99125
AttLSTM ‡	0.03833	0.03834	0.04919	0.04919	227.68640	227.71515	0.99125	0.99125
CONVLSTM ‡	0.03836	0.03836	0.04919	0.04919	227.86650	227.86649	0.99125	0.99125
TCN ‡	**0.03815**	0.03862	**0.04903**	0.04970	**226.58755**	229.42070	**0.99125**	**0.99125**
SCINet ‡	0.05165	0.08773	0.06509	0.10875	306.01984	519.78394	0.31687	0.31687
Transformer ‡	0.03828	**0.03829**	0.04918	**0.04918**	227.39383	**227.41720**	**0.99125**	**0.99125**

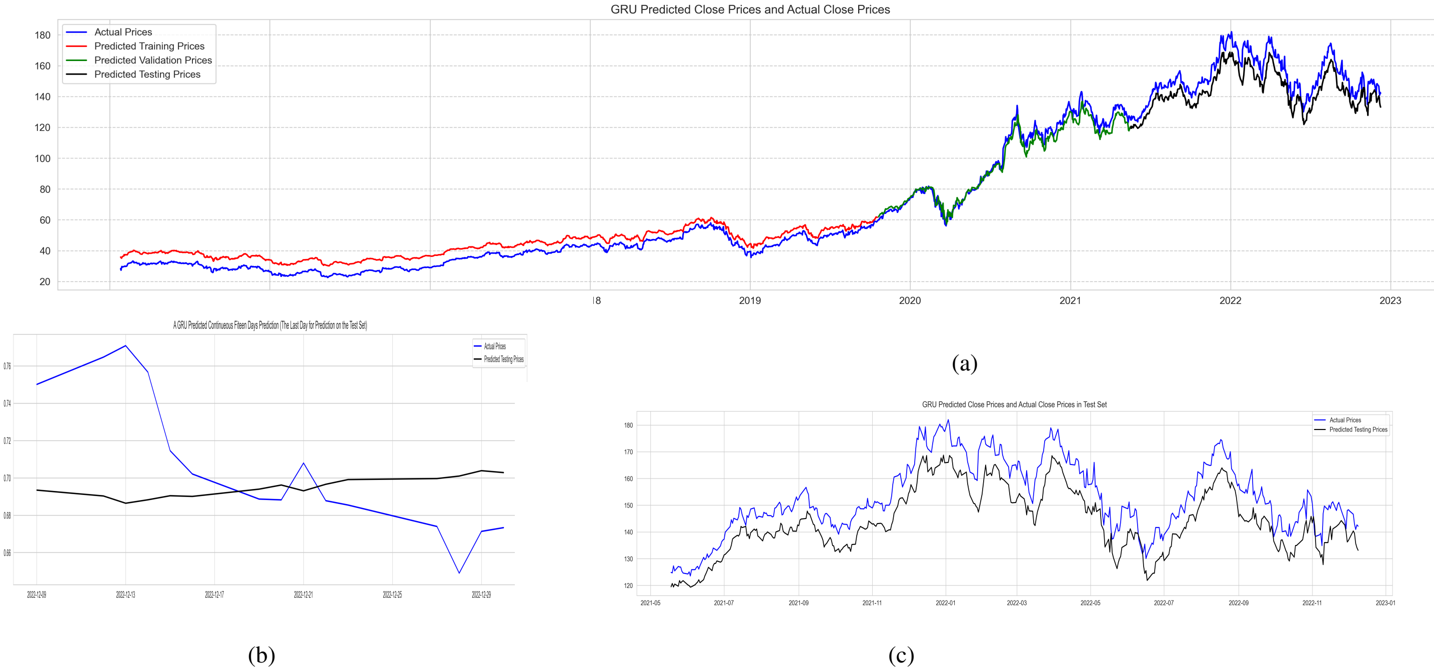


Fig. 16: **GRU** forecast results with selected features (a) stock price forecast in the training, validation and test set, (b) the 15 days forecast results on the last test sample, (c) the forecast results on the test set

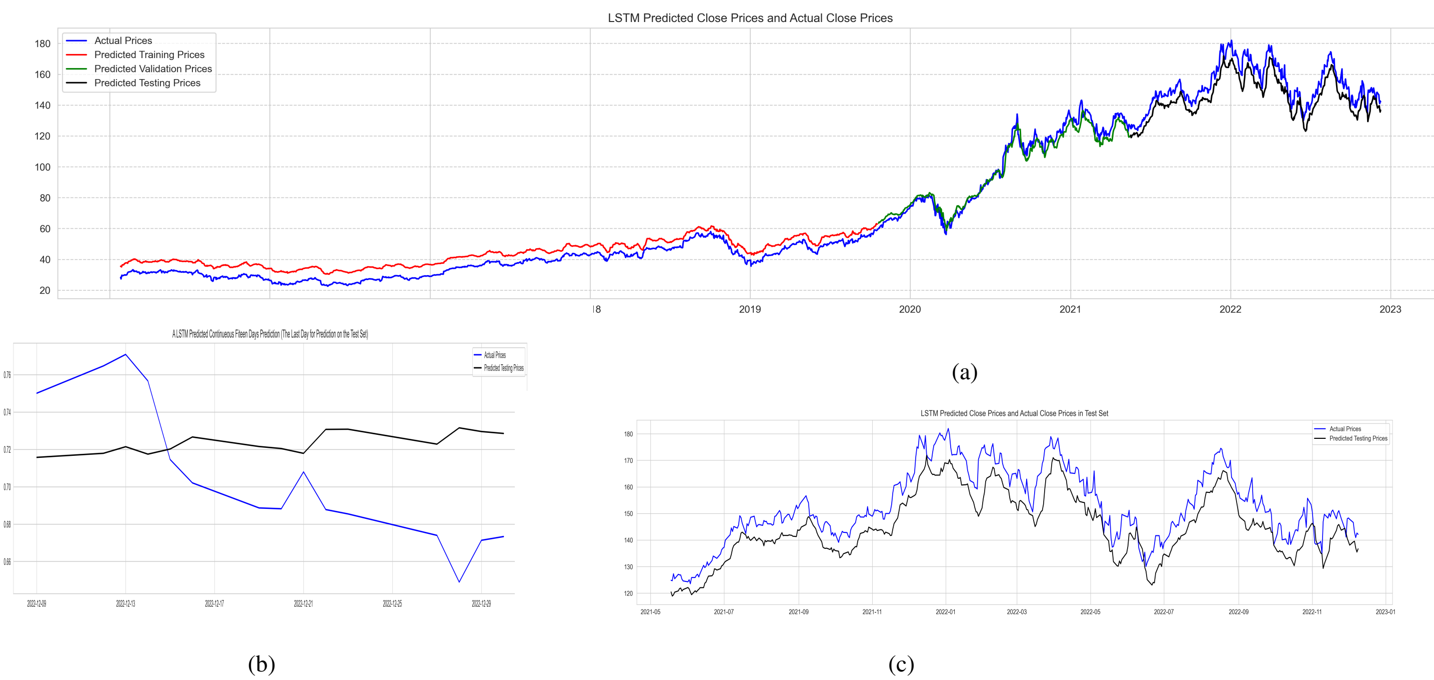
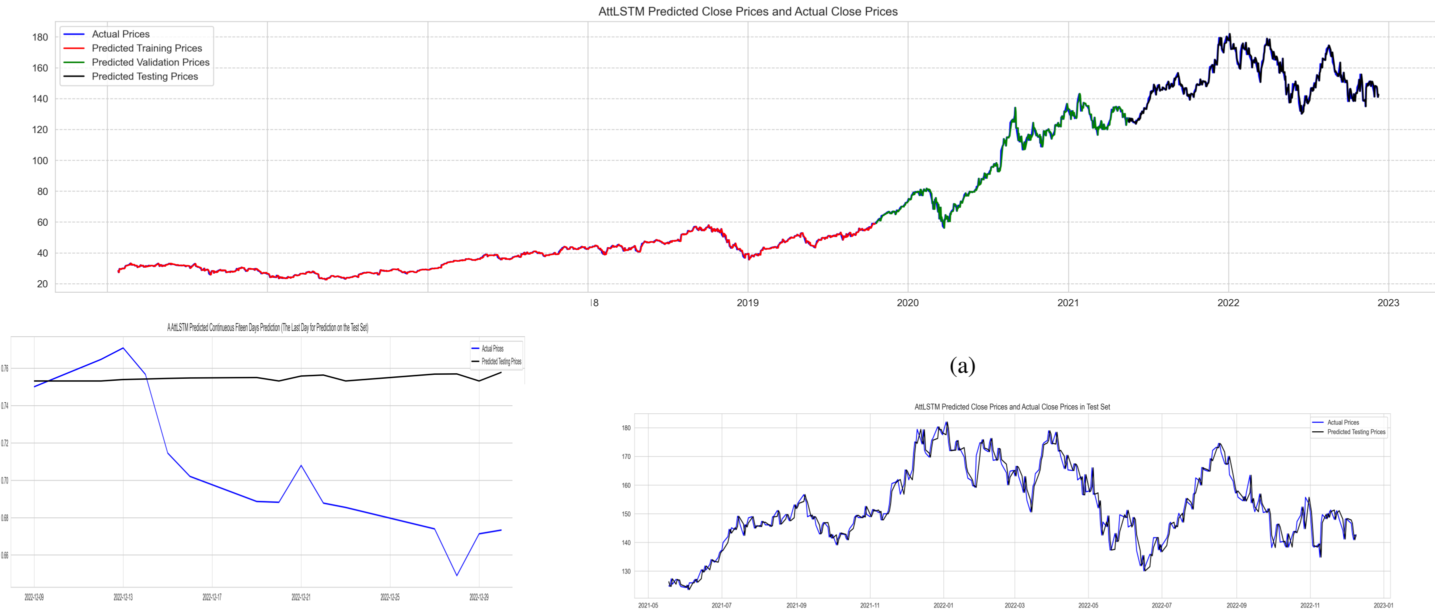


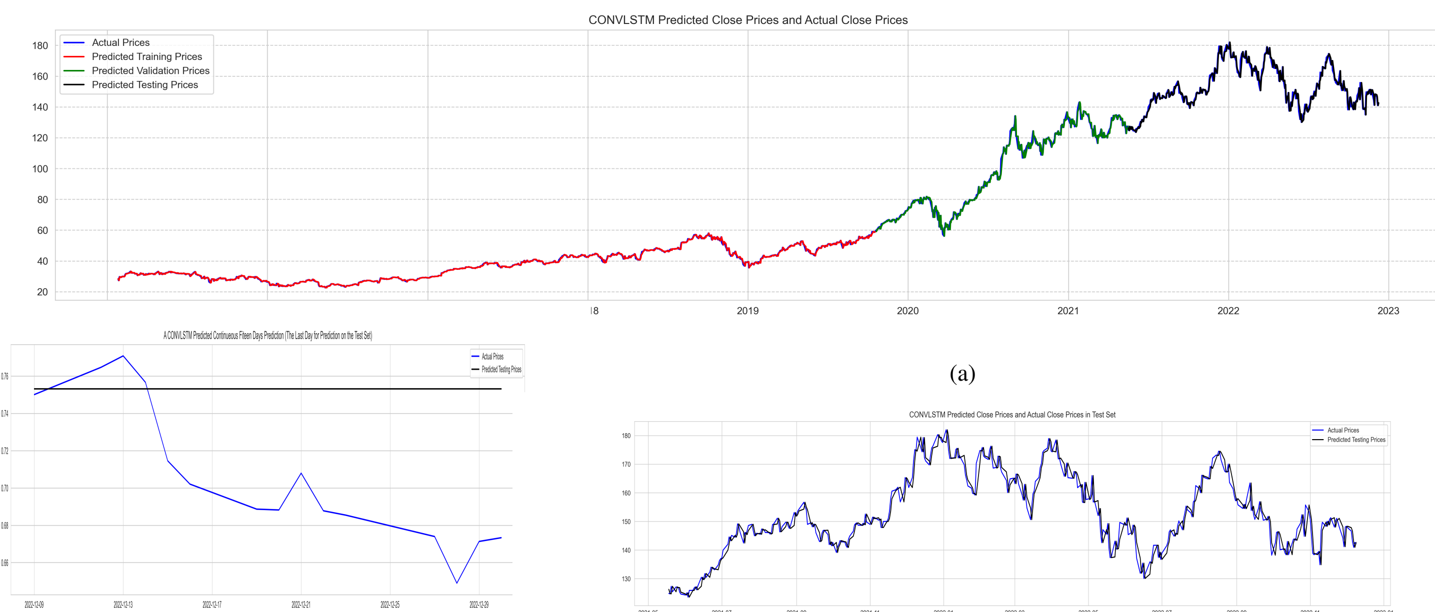
Fig. 17: **LSTM** forecast results with selected features (a) stock price forecast in the training, validation and test set, (b) the 15 days forecast results on the last test sample, (c) the forecast results on the test set



(a)

(b) (c)

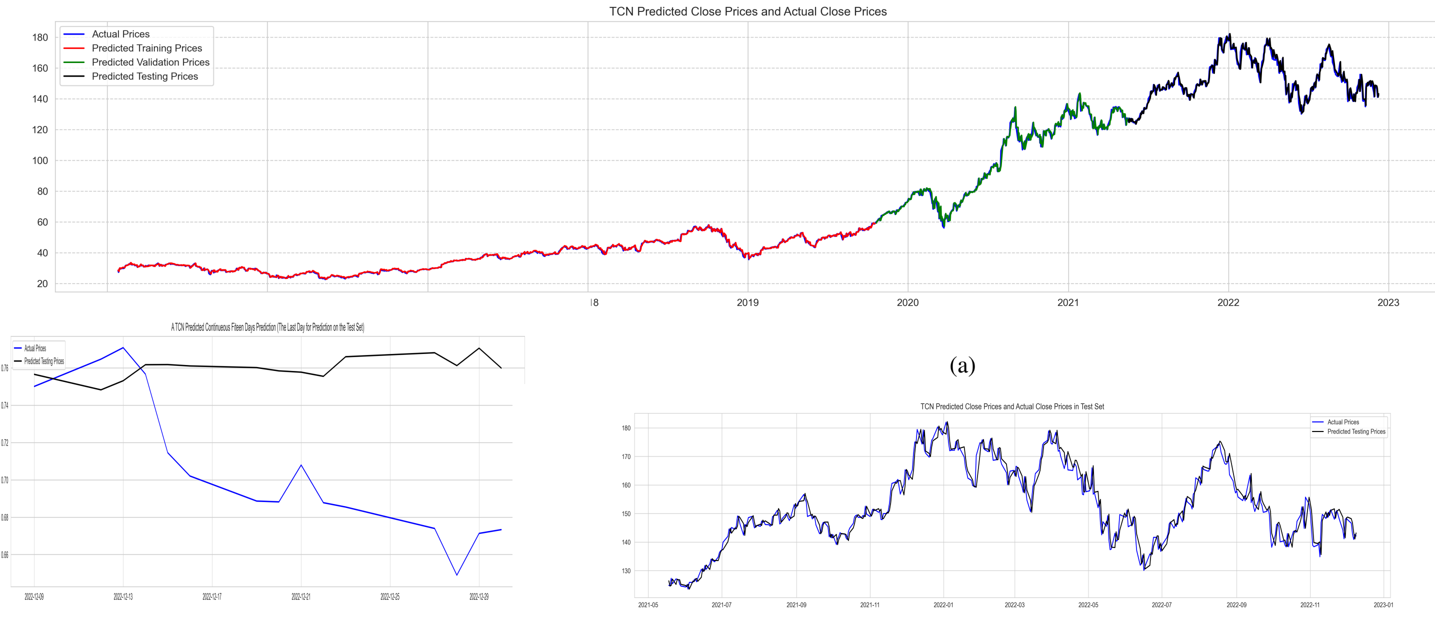
Fig. 18: **AttLSTM** forecast results with selected features (a) stock price forecast in the training, validation and test set, (b) the 15 days forecast results on the last test sample, (c) the forecast results on the test set



(a)

(b) (c)

Fig. 19: **CONVLSTM** forecast results with selected features (a) stock price forecast in the training, validation and test set, (b) the 15 days forecast results on the last test sample, (c) the forecast results on the test set

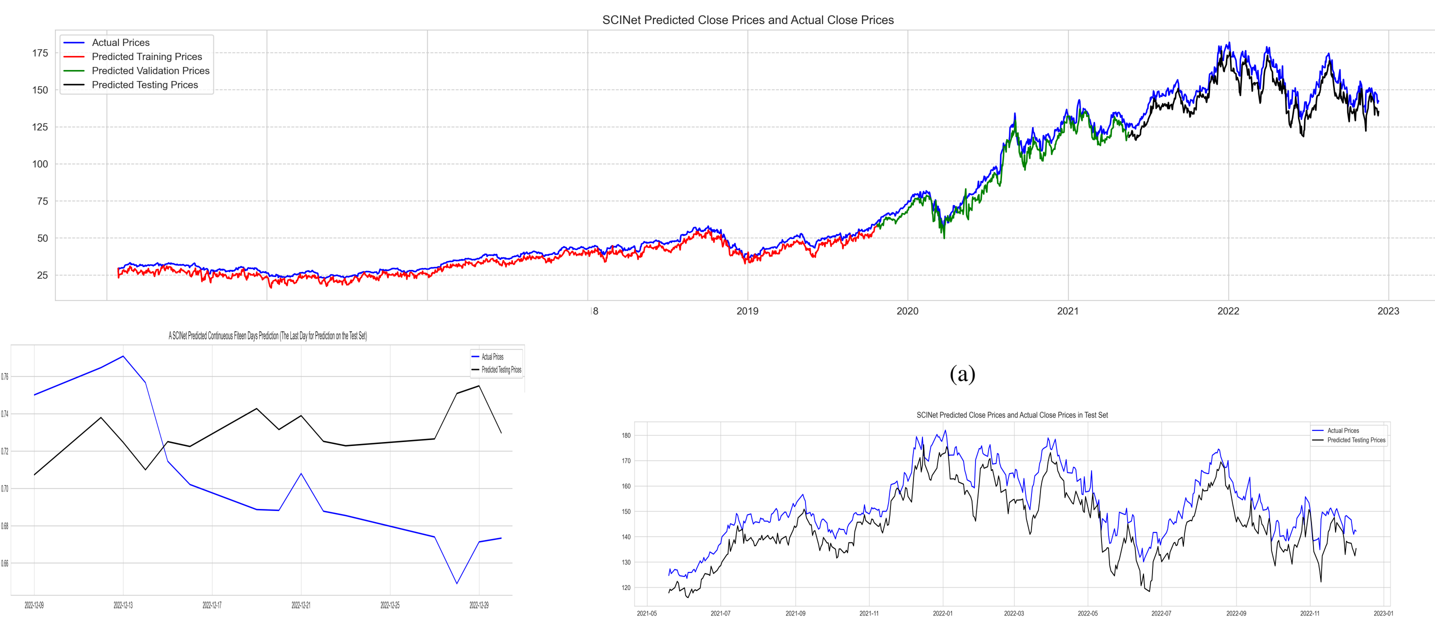


(a)

(b)

(c)

Fig. 20: **TCN** forecast results with selected features (a) stock price forecast in the training, validation and test set, (b) the 15 days forecast results on the last test sample, (c) the forecast results on the test set

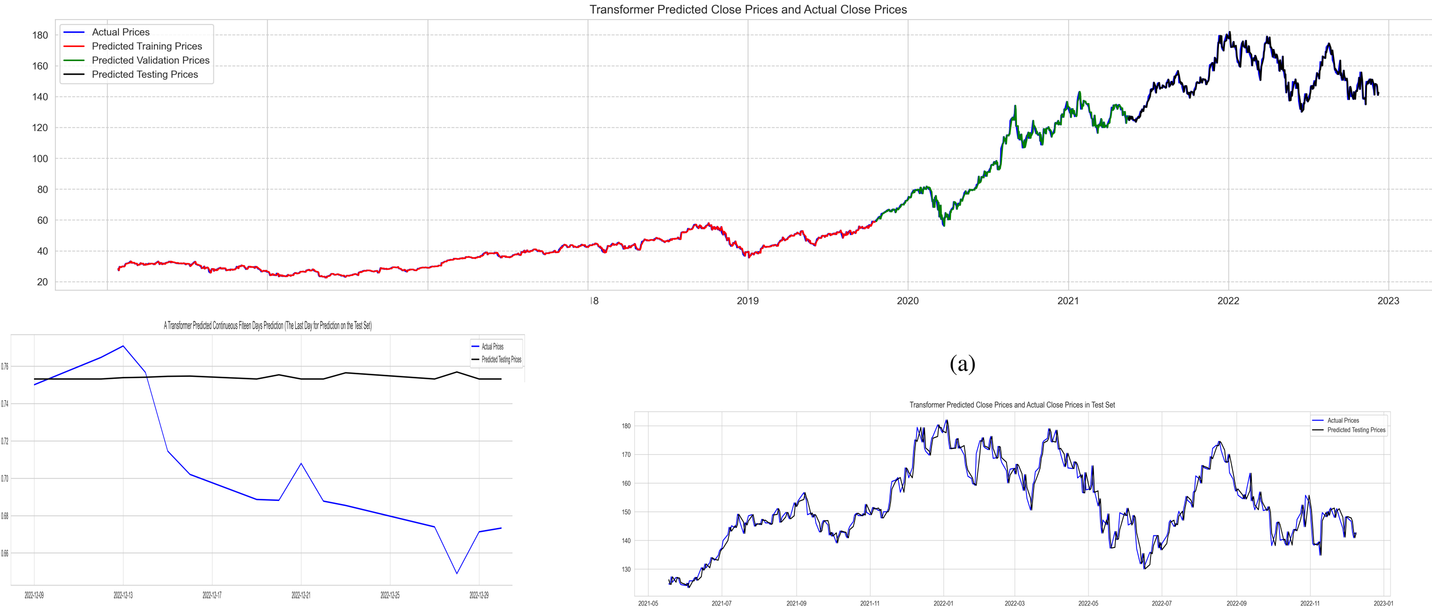


(a)

(b)

(c)

Fig. 21: **SCINet** forecast results with selected features (a) stock price forecast in the training, validation and test set, (b) the 15 days forecast results on the last test sample, (c) the forecast results on the test set



(a)

(b)

(c)

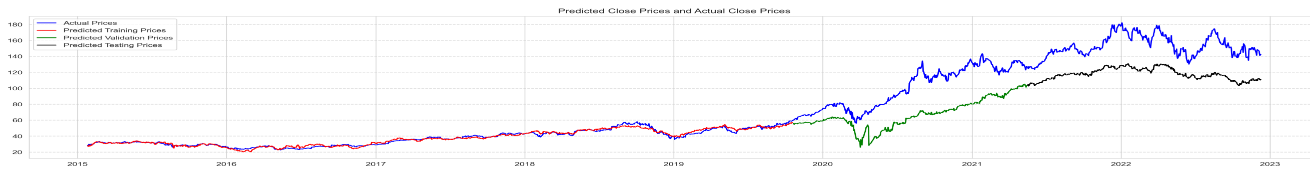
Fig. 22: **Transformer** forecast results with selected features (a) stock price forecast in the training, validation and test set, (b) the 15 days forecast results on the last test sample, (c) the forecast results on the test set

6.7. Results of Machine Learning Models

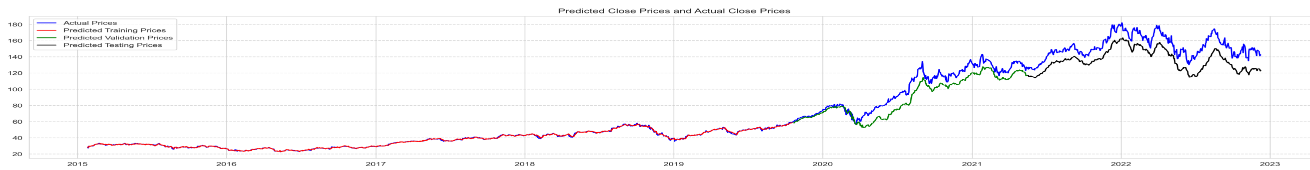
To facilitate a more comprehensive comparison between different models, we conducted a series of experiments using traditional machine learning models, including linear regression (LR), lasso regression (LassoR), ridge regression (RidgeR), and support vector regression (LinearSVR). The results presented in Table 11 and Figure 23 demonstrate that these conventional machine learning models perform significantly worse than neural network models. The lasso regression model achieved the lowest values for mean absolute error (MAE) at 0.12581, root mean squared error (RMSE) at 0.14371, and absolute error (AE) at 747.31. However, these performance metrics are still markedly worse than those obtained by the neural network models, which achieved the lowest MAE at 0.03815, RMSE at 0.04903, and AE at 226.58755. These findings suggest that neural network models outperform traditional machine learning models for predicting stock prices.

Table 11: The normalised normalised mean absolute error (MAE), root mean squared error (RMSE), absolute error (AE) and correlation (CORR) of the Classical ML Regression models: $\star$ marks the training set, † marks the validation set and ‡ marks the test set.

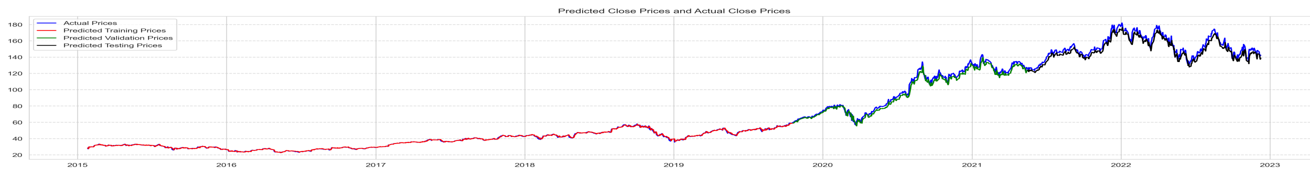
	MAE	RMSE	AE	CORR
LR $\star$	0.00402	0.00521	71.443	0.85282
LassoR $\star$	0.00672	0.089	119.51	0.93390
RidgeR $\star$	0.00554	0.00733	98.485	0.86079
LinearSVR $\star$	0.00823	0.01088	146.36	0.90168
LR †	0.85359	1.1194	5070.3	0.83308
LassoR †	**0.08420**	**0.089**	**500.13**	**0.97406**
RidgeR †	0.22962	0.27555	1363.9	0.87320
LinearSVR †	0.23925	0.29134	1421.2	0.96424
LR ‡	2.0442	2.1526	12142	0.99956
LassoR ‡	**0.12581**	**0.14371**	**747.31**	**0.99971**
RidgeR ‡	0.44048	0.46772	2616.5	0.99965
LinearSVR ‡	0.42507	0.45482	2524.9	0.99963



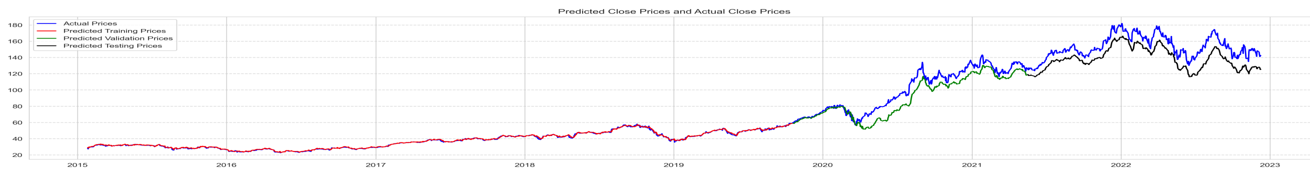
(a)



(b)



(c)



(d)

Fig. 23: Forecast results from the traditional machine learning model (a) Linear regression, (b) Ridge regression, (c) Lasso regression and (d) Linear SVR

6.8. Averaged 15 Days Return with the Models

In addition to traditional evaluation metrics, models predicting stock prices can also be evaluated based on their ability to generate profit. To this end, a simple trading algorithm was employed to compute the average 15-day return for each model. This algorithm utilized the curvature of the forecasted values, where a curvature of 2 indicated local maxima in stock prices, leading to a decision to sell stocks; a curvature of 0 indicated no change, resulting in a decision to hold stocks; and a curvature of -2 indicated local minima, prompting a decision to buy stocks.

The results of this evaluation are presented in Table 12, indicating that among the seven neural network models tested, Transformer achieved the highest profit of 418.49 with an initial trading budget of 10000. It is worth noting that this profit is significantly higher than what could be earned even with knowledge of the actual stock prices, which is typically around 663.50. Based on these findings, Transformer was selected as the best model according to this new metric.

Algorithm 1 Revised StockBot decision making algorithm [11]

Obtain the predicted trajectory, c (stock forecast for forward look days

Compute the nature of change: $\delta_i = \text{sign}\,(c_i - c_{i-1})$

Compute the curvature of the forecast: $\Delta_i = \delta_i - \delta_{i-1}$

$$\text{Make the decision: Decision} = \begin{cases} \Delta_i = 2 \rightarrow \text{ sell (Indicates local maxima)} \\ \Delta_i = 0 \rightarrow \text{ hold (Indicates no change)} \\ \Delta_i = -2 \rightarrow \text{ buy (Indicates local minima)} \end{cases}.$$

Table 12: The Averaged 15 Days Profit for Each Model with the initial trading budget of 10,000 where trading frequency indicates the number of days with buying or selling the stock within the 15 days [11]

	Profit	Trading Freq
GRU	401.47	8.93
LSTM	412.25	9.20
AttLSTM	386.89	8.46
CONVLSTM	350.16	5.03
TCN	387.85	8.81
SCINet	382.85	8.96
Transformer	**418.49**	**9.26**
Actual Price	663.50	10.29

7. CONCLUSION

To forecast the stock price, comprehensive work containing the data mining and model design has been conducted in the project. An extensive range of factors which may influence the stock price are considered in the project and data processing is conducted to the factors including the relevant data selection, feature importance analysis, time-series analysis, outlier removal and data normalisation. As an important part of the feature, the sentiment analysis is taken to the texture data using the pretrained language model FinBERT and GPT-3.5.

Four machine learning algorithms and seven neural networks are implemented and improved as the data inference in the project. The traditional machine learning methods are used to evaluate the feature importance and evaluate the neural network based model performance. The interpretable models including gradient boosting and random forest aiming to maximise the information gain and linear regression and ridge regression aiming to minimise the distance loss are used for measuring the importance of the features. Linear regression, lasso regression, ridge regression and linear SVM are used to donate the forecast of the stock price.

With the neural network as the data inference, GRU, LSTM, attentional LSTM, convolutional LSTM, TCN, SCINet are implemented and adjusted to improve the accuracy of the forecasted price. To improve the model performance, the model aims to forecast the change in the stock price compared with the latest stock price. Intensive experiments have been conducted to fine tuned the model parameters, and analyses the optimiser and the loss function. The actual trend in the stock price and the predicted price of the models are presented and analysed.

To evaluate the actual performance of the stock price forecast, the forecasted values are used for investing AAPL stock with buy and sell action with specific units based on the revised StockBot decision making [11]. Among the models, the adjusted transformer achieves the highest return from the strategy and the minimium difference in the forecasted price from the actual stock price.

8. REFERENCES

[1] Umer Mushtaq Lone, Mushtaq Ahmad Darzi, and Suhail Ahmad Bhat, "Macroeconomic state variables and stock market performance: A systematic review and future research agenda," *FIIB Business Review*, vol. 0, no. 0, pp. 2319714521995622, 0.

[2] Dogu Araci, "Finbert: Financial sentiment analysis with pre-trained language models," *arXiv preprint arXiv:1908.10063*, 2019.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[4] Jeremy Howard and Sebastian Ruder, "Fine-tuned language models for text classification," *CoRR*, vol. abs/1801.06146, 2018.

[5] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[6] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[8] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.

[9] Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu, "Scinet: time series modeling and forecasting with sample convolution and interaction," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5816–5828, 2022.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] Shaswat Mohanty, Anirudh Vijay, and Nandagopan Gopakumar, "Stockbot: Using lstms to predict stock prices," *arXiv preprint arXiv:2207.06605*, 2022.

A. ATTRIBUTE DESCRIPTION

Table 13: Attributes Description

Attribute	Description
Close	The final price at which a particular financial market or asset was traded on a specific trading day.
Volume	The total number of shares or contracts traded during a particular trading day.
Adj Close	The adjusted closing price of a stock, taking into account any dividends or stock splits that may have occurred.
∧DJI Close	The closing price of the Dow Jones Industrial Average, a stock market index that tracks the performance of 30 large publicly traded companies in the United States.
∧IXIC Close	The closing price of the Nasdaq Composite Index, a stock market index that tracks the performance of more than 3,000 companies primarily in the technology and biotechnology industries.
∧GSPC Close	The closing price of the S&P 500 Index, a stock market index that tracks the performance of 500 large companies listed on stock exchanges in the United States.
∧RLG Close	The closing price of the Russell 1000 Growth Index, a stock market index that tracks the performance of large-cap growth stocks in the United States.
∧NYA Close	The closing price of the NYSE Composite Index, a stock market index that tracks the performance of all common stocks listed on the New York Stock Exchange.
∧VIX Close	The closing price of the CBOE Volatility Index, a measure of the market's expectation of volatility in the next 30 days based on options prices for the S&P 500 Index.
DX-Y.NYB Close	The closing price of the US Dollar Index, a measure of the value of the US dollar relative to a basket of six major currencies.
EURUSD=X Close	The closing price of the EUR/USD currency pair, a measure of the value of the euro relative to the US dollar.
GBPUSD=X Close	The closing price of the GBP/USD currency pair, a measure of the value of the British pound relative to the US dollar.
GC=F Close	The closing price of gold futures, which are contracts that obligate the buyer to purchase a certain amount of gold at a specific price and time in the future.
SI=F Close	The closing price of silver futures, which are contracts that obligate the buyer to purchase a certain amount of silver at a specific price and time in the future.
CL=F Close	The closing price of crude oil futures, which are contracts that obligate the buyer to purchase a certain amount of crude oil at a specific price and time in the future.
WM1NS	The one-month London Interbank Offered Rate (LIBOR), a benchmark interest rate that banks use to lend money to each other in the international interbank market.
WM2NS	The two-month London Interbank Offered Rate (LIBOR).
ICSA	Initial Claims for Unemployment Insurance, a weekly report released by the US Department of Labor that provides information on the number of people who filed for unemployment insurance for the first time.
CCSA	Continuing Claims for Unemployment Insurance, a weekly report released by the US Department of Labor that provides information on the number of people who continue to receive unemployment insurance benefits.
JTSJOL	Job Openings and Labor Turnover Survey, a monthly report released by the US Bureau of Labor Statistics that provides information on the number of job openings, hires, and separations in the US labor market.
PAYEMS	Nonfarm Payrolls, a monthly report released by the US Bureau of Labor Statistics that provides information on the number of people employed in the US economy, excluding workers in agriculture, private households, and non-profit organizations.
RSXFS	The RTS Index, a stock market index that tracks the performance of Russian stocks traded on the Moscow Exchange.
TCU	Total Capacity Utilization, a measure of the percentage of a country's industrial capacity that is being used at any given time.
UMCSENT	University of Michigan Consumer Sentiment Index, a survey-based measure of consumer confidence in the US economy.
BUSINV	Business Inventories, a monthly report released by the US Census Bureau that provides information on the level of inventories held by US businesses.

Table 14: Attributes Description (Cond)

Attribute	Description
INDPRO	Industrial Production, a monthly report released by the US Federal Reserve that provides information on the output of US manufacturing, mining, and utilities sectors.
GACDFSA066MSFRBPHI	Gross Domestic Product for Federal Reserve Banks (FRB) Districts, a measure of the total economic output of each of the twelve Federal Reserve districts in the United States.
GACDISA066MSFRBNY	Gross Domestic Product for State (SA) and Local Areas (LA), a measure of the total economic output of each of the 50 US states and various metropolitan and non-metropolitan areas.
BACTSAMFRBDAL	Dallas Fed Bank Texas Manufacturing Outlook Survey, a monthly survey conducted by the Federal Reserve Bank of Dallas that provides information on the state of the manufacturing sector in Texas.
IR	Interest Rate, the amount charged by a lender to a borrower for the use of money.
IQ	Inflation Rate, the rate at which the general level of prices for goods and services is rising over time.
PPIACO	Producer Price Index for All Commodities, a monthly report released by the US Bureau of Labor Statistics that provides information on the average change in prices received by domestic producers of goods and services.
CPIAUCSL	Consumer Price Index for All Urban Consumers, a monthly report released by the US Bureau of Labor Statistics that provides information on the average change in prices paid by urban consumers for a basket of goods and services.
CPILFESL	Consumer Price Index for All Urban Consumers, All Items Less Food and Energy, a measure of the average change in prices paid by urban consumers for a basket of goods and services, excluding food and energy.
MICH	University of Michigan Consumer Sentiment Index, a survey-based measure of consumer confidence in the US economy.
CSCICP03USM665S	S&P/Case-Shiller US National Home Price Index, a measure of the average change in home prices across the United States over time.
Grow	The flag to indicate whether the stock price increases compared to the previous stock price.
News Score	The analysed news score for indicating the positve influence to the stock price.
TMO	Thermo Fisher Scientific Inc., a scientific research and instrumentation company that produces and sells a range of laboratory equipment, chemicals, and services to customers in the life sciences and healthcare industries. The company is headquartered in Waltham, Massachusetts.
MSFT	Microsoft Corporation, a multinational technology company that develops and licenses a range of software, hardware, and services. Its products include the Windows operating system, Office, Xbox gaming consoles, and the Microsoft Surface line of devices. The company is based in Redmond, Washington.
DHR	Danaher Corporation, a conglomerate that produces and sells a range of industrial and healthcare products, including analytical instruments, life sciences tools, dental equipment, and water treatment systems. The company is headquartered in Washington, D.C.
TSLA	Tesla, Inc., an automotive and energy company that designs and produces electric vehicles, energy storage systems, and solar products. The company is headquartered in Palo Alto, California.
LOW	Lowe's Companies, a home improvement retailer that operates a chain of retail stores in the United States and Canada. The company sells a range of products, including appliances, tools, building materials, and outdoor equipment. Lowe's is based in Mooresville, North Carolina.
NEE	NextEra Energy, Inc., an energy company that generates and distributes electricity through its subsidiaries. The company focuses on renewable energy sources such as wind and solar power, and it also operates natural gas pipelines. NextEra Energy is headquartered in Juno Beach, Florida.
AVGO	Broadcom Inc., a semiconductor company that designs and produces a range of semiconductor devices, including chips for networking, storage, wireless, and industrial applications. The company is headquartered in San Jose, California.
LIN	Linde plc, an industrial gases company that produces and distributes gases such as oxygen, nitrogen, and helium for use in various industries. The company also provides engineering and technology solutions for customers around the world. Linde is headquartered in Guildford, United Kingdom.
COST	Costco Wholesale Corporation, a membership-based retail company that operates a chain of warehouse stores selling a wide range of products, including groceries, electronics, and home goods. Costco is based in Issaquah, Washington.
ACN	Accenture plc, a multinational professional services company that provides consulting, technology, and outsourcing services to clients in various industries around the world. The company is headquartered in Dublin, Ireland.

B. STOCK PRICE CORRELATION

Table 15: Stock Price from S&P 100 Correlation with AAPL

Feature	Correlation	Feature	Correlation
TMO	0.9749621815	PM	0.6934372072
MSFT	0.9584408917	AMZN	0.6862779841
DHR	0.9575826520	DUK	0.6835256992
TSLA	0.9563119144	ADBE	0.6792954588
LOW	0.9524922058	BAC	0.6726946972
NEE	0.9393401279	SO	0.6633826895
AVGO	0.9365944174	HON	0.6576001985
LIN	0.9346751859	COF	0.6440110017
COST	0.9272982142	KO	0.6246900536
ACN	0.9243416519	FDX	0.6234879526
UPS	0.9230814088	AMGN	0.5961522488
QCOM	0.9196617265	GD	0.5849544780
NVDA	0.9136831784	BKNG	0.5813012557
UNH	0.9070783206	AMT	0.5681555440
GOOGL	0.9052657636	CRM	0.5436839686
PG	0.9051200401	JPM	0.5414507863
GOOG	0.9037163331	DOW	0.5401116025
MS	0.9008001948	EXC	0.5365882916
AMD	0.8998259206	GM	0.5230489450
UNP	0.8988840152	COP	0.5124887619
ABT	0.8966819714	SBUX	0.4801978971
TXN	0.8953937081	CHTR	0.4794854954
HD	0.8952863815	META	0.3645625073
JNJ	0.8909305233	BK	0.3627192139
TMUS	0.8746989764	GE	0.3566194590
CVS	0.8628491743	CVX	0.3534525711
ABBV	0.8599524132	LMT	0.3461452276
TGT	0.8515787196	AIG	0.3382245815
BRK-B	0.8514021816	PYPL	0.3032253371
MDLZ	0.8384152888	RTX	0.3030762949
SCHW	0.8368359914	CMCSA	0.2713778047
LLY	0.8336090133	MRK	0.2626351267
GS	0.8334006815	NFLX	0.2317437198
WMT	0.8319915533	XOM	0.2287230449
PEP	0.8317683817	MDT	0.1952537675
BLK	0.8315686863	CSCO	0.1849125088
MA	0.8309163962	DIS	0.1728127407
CAT	0.8265196374	USB	0.1519989806
ORCL	0.8243781254	IBM	0.1138601627
MCD	0.8144782173	WFC	0.1056573318
EMR	0.8081832582	MO	0.0972825441
V	0.7801973237	GILD	-0.0632311207
BMY	0.7583230404	MMM	-0.1435192760
NKE	0.7500122126	SPG	-0.1554576073
F	0.7442243825	C	-0.2203394997
AXP	0.7436420896	INTC	-0.3422279609
KHC	0.7402985101	WBA	-0.4329831451
MET	0.7249106349	VZ	-0.5250465574
PFE	0.7183985571	BA	-0.6697111423
CL	0.7101033262	T	-0.8035719197

C. FEATURE IMPORTANCE

Table 16: Feature Importance across the Attributes

	Gradient Boosting	Random Forest	Linear Regression	Ridge Regression
Close	0.8148765742	1.0000000000	1.0000000000	1.0000000000
Volume	0.0000845350	0.0002979568	0.1843573326	0.1345676242
Adj Close	1.0000000000	0.7176544144	0.1273802019	0.9565499200
∧DJI Close	0.0000313460	0.0006909163	0.1597867886	0.1162358735
∧IXIC Close	0.0001490777	0.0004277574	0.1280363820	0.2682778032
∧GSPC Close	0.0021507523	0.0021513328	0.0000000000	0.2022348008
∧RLG Close	0.0000286072	0.0030106306	0.3599851692	0.2645905262
∧NYA Close	0.0151654181	0.0005241741	0.2655058895	0.1381065778
∧VIX Close	0.0001046064	0.0001835571	0.1906609075	0.1526967934
DX-Y.NYB Close	0.0000597831	0.0001645523	0.1840350940	0.1232140953
EURUSD=X Close	0.0001086735	0.0003417798	0.1814637622	0.1162071577
GBPUSD=X Close	0.0001560669	0.0001876828	0.1876521438	0.1363252259
GC=F Close	0.0001575011	0.0002429076	0.1954718885	0.1568078018
SI=F Close	0.0001036133	0.0002447766	0.1713665206	0.0712009966
CL=F Close	0.0000629691	0.0002151160	0.1740595965	0.1615753381
WM1NS	0.0001614785	0.0002770322	0.1882997982	0.1551312217
WM2NS	0.0000379141	0.0004123237	0.1797262974	0.1741847759
ICSA	0.0000189722	0.0000690007	0.1607338712	0.1505341892
CCSA	0.0002453051	0.0001777972	0.2507585161	0.0821174798
JTSJOL	0.0001010539	0.0000346658	0.1730198097	0.1127303228
PAYEMS	0.0000000000	0.0002316020	0.3316751916	0.1214442534
RSXFS	0.0000201351	0.0000234521	0.1519033050	0.1133500791
TCU	0.0000197155	0.0000073511	0.1823157246	0.0984717090
UMCSENT	0.0001242611	0.0000492677	0.1798395235	0.1056882635
BUSINV	0.0000000000	0.0000000000	0.1747417076	0.0884094171
INDPRO	0.0000144134	0.0000082601	0.1773658894	0.1566153247
GACDFSA066MSFRBPHI	0.0000012482	0.0000411875	0.1818758580	0.1364947805
GACDISA066MSFRBNY	0.0000004447	0.0000330417	0.1883345761	0.1291164756
BACTSAMFRBDAL	0.0000000437	0.0000148802	0.1942448372	0.1634857613
IR	0.0004658858	0.0000276568	0.1866371872	0.1725634508
IQ	0.0001986572	0.0001322712	0.2046654425	0.1529972948
PPIACO	0.0000000000	0.0000104214	0.1915273300	0.1606463860
CPIAUCSL	0.0000000000	0.0000009219	0.2011425503	0.1363165936
CPILFESL	0.0000000000	0.0000031401	0.0274211881	0.0377125769
MICH	0.0000000000	0.0000051602	0.1719668382	0.1221148178
CSCICP03USM665S	0.0001176507	0.0000381045	0.1947323990	0.1535155815
Grow	0.0000114853	0.0000234098	0.1849070896	0.1346808198
News Score	0.0001022266	0.0002528857	0.1854733678	0.1386844583
index	0.0088733208	0.0029442419	0.2284734606	0.1142102728
TMO Close	0.0000485162	0.0001602744	0.1862333437	0.1298411482
MSFT Close	0.0009835697	0.0002901803	0.1719084820	0.1894914507
DHR Close	0.0000311872	0.0001136806	0.1692979328	0.0000000000
TSLA Close	0.0001247731	0.0002360137	0.0882126136	0.1248049690
LOW Close	0.0000393681	0.0007126454	0.1821092688	0.1203008914
NEE Close	0.0181420241	0.0009772713	0.1864810568	0.2604098602
AVGO Close	0.0008702718	0.0007071331	0.1806098329	0.1678674108
LIN Close	0.0000537331	0.0001418339	0.1900109527	0.1044197913
COST Close	0.0001098856	0.0002281930	0.2312793504	0.2738132435
ACN Close	0.0000570513	0.0001322045	0.1721719402	0.0550123341

Table 17: Feature Importance across the Period with the Input Window of 30 Days

	Gradient Boosting	Random Forest	Linear Regression	Ridge Regression
Day 30	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Day 29	0.0034483506	0.0237021204	0.0623038796	0.6611325883
Day 28	0.0246406326	0.0021356757	0.1671438559	0.4628924597
Day 27	0.0022635140	0.0016440841	0.1675173304	0.3421448774
Day 26	0.0008995081	0.0011618099	0.1404405447	0.2560505522
Day 25	0.0026266030	0.0008382724	0.1123950970	0.1867944903
Day 24	0.0043875824	0.0005152926	0.2074712732	0.1465477793
Day 23	0.0000244557	0.0004708890	0.0000000000	0.0816325393
Day 22	0.0022977852	0.0006617854	0.2182076440	0.0711928063
Day 21	0.0003019483	0.0005988312	0.0814382835	0.0500727849
Day 20	0.0000303682	0.0005888415	0.1428198169	0.0442238501
Day 19	0.0001623672	0.0000541242	0.1475617587	0.0420140111
Day 18	0.0001501257	0.0000316520	0.0996010546	0.0307026506
Day 17	0.0000737156	0.0000729711	0.1383923695	0.0346646107
Day 16	0.0001956486	0.0001936993	0.1250741004	0.0399755784
Day 15	0.0000229078	0.0001908285	0.1703640333	0.0640958585
Day 14	0.0000339192	0.0000397014	0.1623591878	0.0640893561
Day 13	0.0000365988	0.0002413209	0.0715799317	0.0445826617
Day 12	0.0000234883	0.0003251239	0.1582885854	0.0442239775
Day 11	0.0010714912	0.0003863798	0.1582825515	0.0383196776
Day 10	0.0000270928	0.0000097594	0.0935957420	0.0208296313
Day 9	0.0003017066	0.0002246913	0.1038944812	0.0000000000
Day 8	0.0001249579	0.0002751017	0.1424660669	0.0030699331
Day 7	0.0000751124	0.0000165323	0.0979439924	0.0046626097
Day 6	0.0008395266	0.0000000000	0.1782099888	0.0221397045
Day 5	0.0000000000	0.0000144216	0.1269643267	0.0285070397
Day 4	0.0005081024	0.0000791078	0.0791292754	0.0247234253
Day 3	0.0000154613	0.0000257468	0.2446502092	0.0477544126
Day 2	0.0001664591	0.0000222142	0.0242677548	0.0259667574
Day 1	0.0000376197	0.0000545287	0.1719544636	0.0338996799