Title
Predicting the Returns of Progressive Corporation Stock

Author
Xu, Amanda

Publication Date
2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Predicting the Returns of

Progressive Corporation Stock

A thesis submitted in partial satisfaction

of the requirements for the degree

Master in Applied Statistics

by

Amanda Xu

2023

ABSTRACT OF THE THESIS

Predicting the Returns of
Progressive Corporation Stock

by

Amanda Xu
Master in Applied Statistics
University of California, Los Angeles, 2023
Professor Yingnian Wu, Chair

In this analysis, the objective is to forecast the stock prices of property and casualty insurance in 2022. This industry is known to be relatively stable and resilient to economic downturns. The data utilizes weekly adjusted closing prices of Progressive Insurance from 2019 to 2021 to form the training set. Three different models were created to predict weekly adjusted closing prices for 2022. The methods used were the LSTM and GRU recurrent neural network models, as well as the ARIMA time series analysis. Based on the results, the GRU method achieved the lowest RMSE due to its ability to avoid overfitting and does not rely on the assumption of stationarity.

The thesis of Amanda Xu is approved.

Nicolas Christou

Frederic Schoenberg

Yingnian Wu, Committee Chair

University of California, Los Angeles

2023

To my parents . . .

who have supported me in every stage of life

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

The stock market is a key indicator of economic welfare, and fluctuations in stock prices can bring bull or bear markets. Many factors affect stock prices including market sentiment, industry popularity, and overall confidence in the economy. The more potential that investors see in stocks, the more likely the stock price will increase, and conversely, a lack of confidence in the stock can lead investors to sell, which lowers the price. There are industries that are more resilient to fluctuations in stock prices due to necessity. The insurance industry is a leading example of one of these industries. Consumers are inclined to continue to or further protect their assets during a recession which secures long-term returns for insurers. Insurance carriers from health insurance to property and casualty insurance tend to perform steadily during recessionary times. They also invest in their premiums and have many diversified streams of income that decrease their volatility. However, when the market is down, the invested premiums are at risk of taking a hit. [SLC22] The year 2022 was a challenging year for the global economy as it faced slow economic growth and layoffs, ultimately leading to warnings of a potential recession. In June 2022, inflation hit a year-over-year high, and insurance companies rushed to increase their premiums to cover the rising costs of claims. Insurance is known to be a stable and recession-resilient industry, making it a reliable choice for investors during turbulent economic times. Therefore, it is essential to understand which machine learning methods can accurately predict price trends during inflationary and recessionary times. This paper aims to analyze the stock prices of the top property and casualty insurance companies to monitor their market performance during the tumultuous year of 2022. Progressive Insurance (PGR), the leading property and casualty insurance company in terms of profitability, had a successful year in 2022, with a steady rise in stock prices

[Yok21].

To test the predictive abilities of different machine learning models, Progressive was chosen as a baseline company to compare three different models: Long-Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and Autoregressive Integrated Moving Average (ARIMA). The selected models will predict the adjusted closed stock prices of Progressive Insurance and two other competitors to analyze the applicability of the models. If the models can accurately predict the stock price trends of a stable industry, then investing in their stocks can potentially be a safe choice. The ultimate goal of this analysis is to determine which model can accurately predict stock prices during turbulent economic times, making it a reliable tool for investors and insurance companies alike. We will use Root Mean-Squared Error (RMSE) as the metric to determine the top performing model. RMSE measures the average Euclidean distance of the predicted value to the true value, and it is a reliable metric to use because it takes into account the effect of large errors.

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(\hat{x}_i - x_i)^2}{n}} \tag{1.1}$$

CHAPTER 2

Data

The dataset used in this analysis consists of daily stock price data of Progressive Corporation from January 1, 2019 to December 31, 2022. The 4-year time frame captures data from a relatively stable economy to possible fluctuations due to COVID-19 and its aftermath. The data was collected using a Yahoo Finance API and contains a total of 1007 observations and 6 continuous variables, namely open price, high price, low price, close price, adjusted close price, and volume.

For this analysis, the adjusted closing price was chosen as the target variable for forecasting. The adjusted closing price reflects the stock price after adjusting for any dividends, stock splits, or new offerings that may have occurred during the time period.

To get a better understanding of the range of stock prices in the dataset, the maximum and minimum adjusted closing price for each year was calculated and summarized in the table below. The largest range was observed in 2020, which is not surprising given the significant impact of the COVID-19 pandemic on the global economy. The year 2022 also had a notable range in adjusted closing price, which may be attributed to overall market fluctuations during this period.

Overall, the dataset provides a comprehensive view of Progressive Corporation's stock price over a 4-year period that includes both stable and volatile market conditions. This information will be used to build and test time-series forecasting models, with the ultimate goal of predicting future stock prices for this company.

Year	2019	2020	2021	2022
Minimum Price	52.19	61.81	83.21	101.50
Maximum Price	75.83	94.51	104.80	132.32
Range	23.64	32.70	21.59	30.82

Table 2.1: Progressive Adjusted Closing Prices 2019-2022

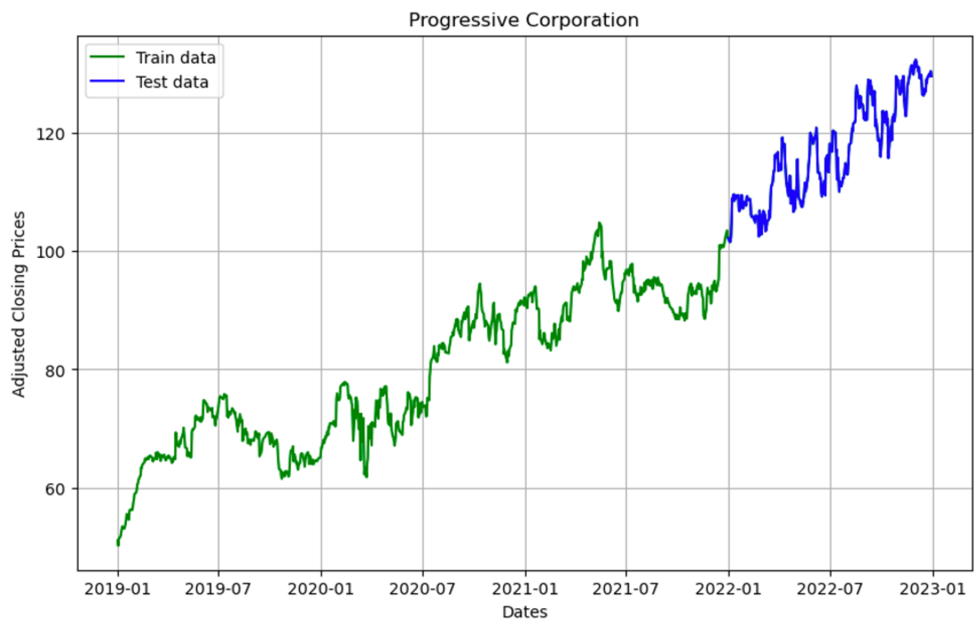

Figure 2.1: Progressive Corporation Adjusted Closing Prices 2019-2022

Because the goal is to predict stock prices for 2022, the training and testing datasets are split into 75% and 25% respectively. We train on years 2019-2021 and test on 2022 data. The training set has 756 observations and the test set has 251 observations. The overall increasing trend of the data shows promising for the insurance industry. There is a dip in the second half of 2021, and since 2022 has a steady increase in stock price, this can affect the predictions.

The two property and casualty insurance competitors that we will apply the most successful model to are Allstate Insurance Group and Travelers Insurance Group. Their adjusted closing price graphs are shown below, using the same time period and training and testing splits as that of Progressive. Both competitors' adjusted closing prices significantly dipped in early 2020, during the start of COVID-19 in the United States. They recovered in 2021 with the increase in prices but Allstate's prices dipped again at the end of 2021. In 2022, it was relatively stationary for Allstate but more of a volatile year for Travelers especially in the third fiscal quarter. Compared to Progressive, these two competitors do not have a consistent increasing trend throughout the past four years, but we will test out the best model in a later chapter.

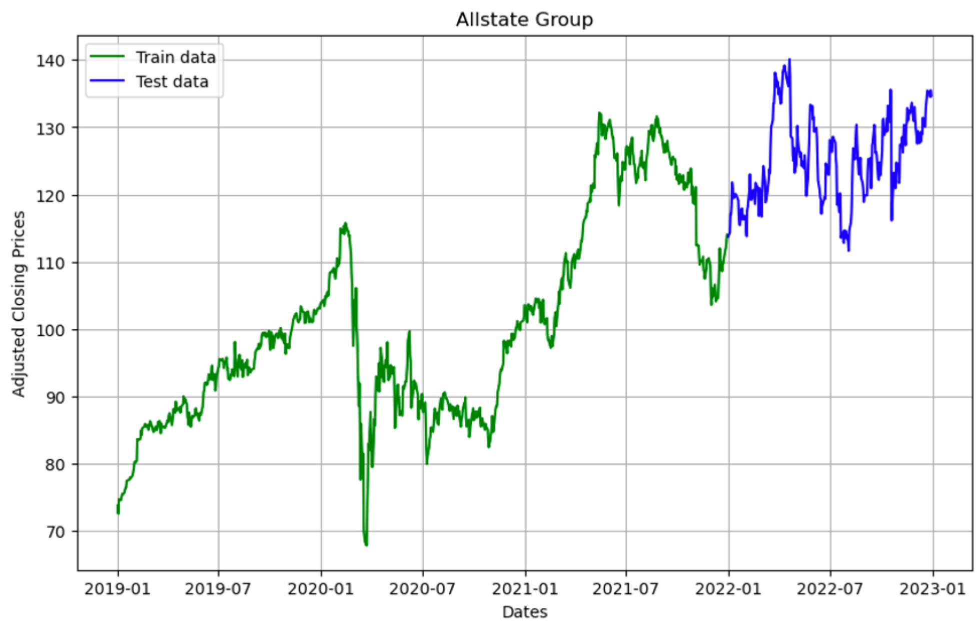

Figure 2.2: Allstate Corporation Adjusted Closing Prices 2019-2022

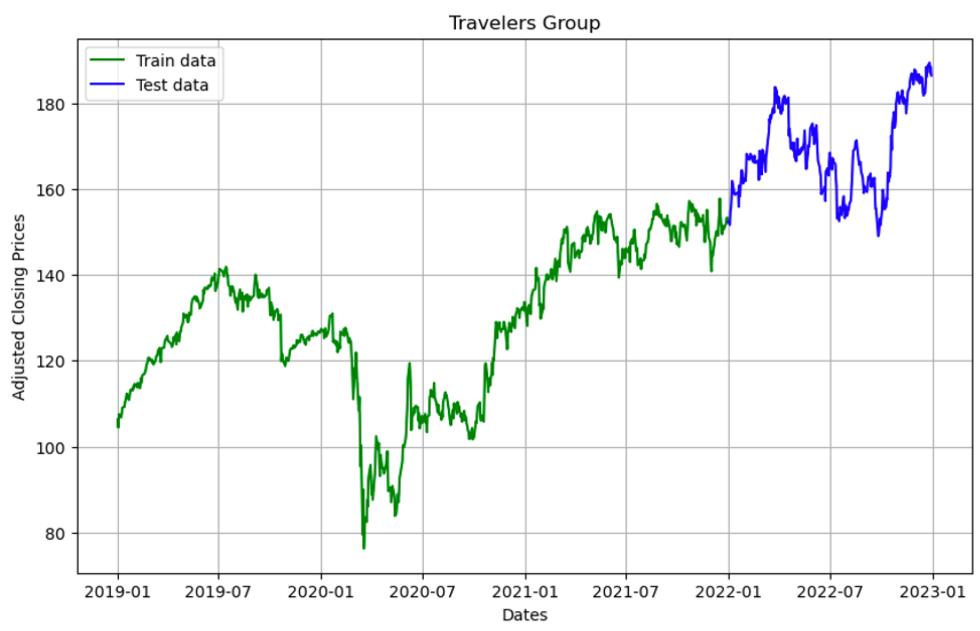

Figure 2.3: Travelers Group Adjusted Closing Prices 2019-2022

CHAPTER 3

Recurrent Neural Networks

A recurrent neural network (RNN) is a type of neural network architecture designed to work with sequential data. Unlike feed-forward neural networks that process a fixed input size, RNNs can handle inputs of variable length and are able to capture the temporal dependencies of sequential data. RNNs contain a feedback loop that allows the output of each step to be fed back into the input for the next step, creating a form of memory that allows the network to remember previous inputs and use them to make predictions about future inputs [RNN]. This makes RNNs particularly useful for applications like time series analysis, where the sequence of inputs is critical in understanding the meaning or making accurate predictions. The RNN methods used in this analysis are long short-term memory networks (LSTM) and gated recurrent networks (GRU).

3.1 Long Short Term Memory

LSTM is a type of recurrent neural network that remembers information over long periods of time. Compared to traditional neural network, LSTMs use a more complex memory unit, called a cell, that is capable of selectively remembering and forgetting information over time. The LSTM cell has three gates: the input gate, the forget gate, and the output gate. These gates are used to control the flow of information into and out of the cell. At each time step, the LSTM cell takes in an input x(t) and a previous hidden state h(t-1). The input gate determines how much of the input should be allowed into the cell, while the forget gate determines how much of the previous cell state should be forgotten. The output gate controls how much of the current cell state should be output to the next layer or to the final output

[Hoc97]. The LSTM cell maintains a cell state, which acts as the long-term memory of the network. The cell state is updated at each time step by selectively adding new information and forgetting old information, based on the input, previous hidden state, and gates.

The input gate (Equation 3.1) has σ is the sigmoid function, W_i is the weight matrix, and $[h(t-1), x(t)]$ is the concatenation of the previous hidden state and current input. The forget gate (Equation 3.2) is similar to the input gate, but it decides which information to keep in the network. The information is stored in the prior cell state (Equation 3.3), and depending on the forget vector $f(t)$, the values will get dropped before reaching the candidate cell state (Equation 3.4). The output gate $o(t)$ determines the values entered into the hidden state, and finally, the hidden state (Equation 3.6) is is used for prediction.

$$i(t) = \sigma(W_i * [h(t-1), x(t)]) \tag{3.1}$$

$$f(t) = \sigma(W_f * [h(t-1), x(t)]) \tag{3.2}$$

$$C(t) = f(t) * C(t-1) + i(t) * g(t) \tag{3.3}$$

$$g(t) = tanh(W_g * [h(t-1), x(t)]) \tag{3.4}$$

$$o(t) = \sigma(W_o * [h(t-1), x(t)]) \tag{3.5}$$

$$h(t) = o(t) * tanh(C(t)) \tag{3.6}$$

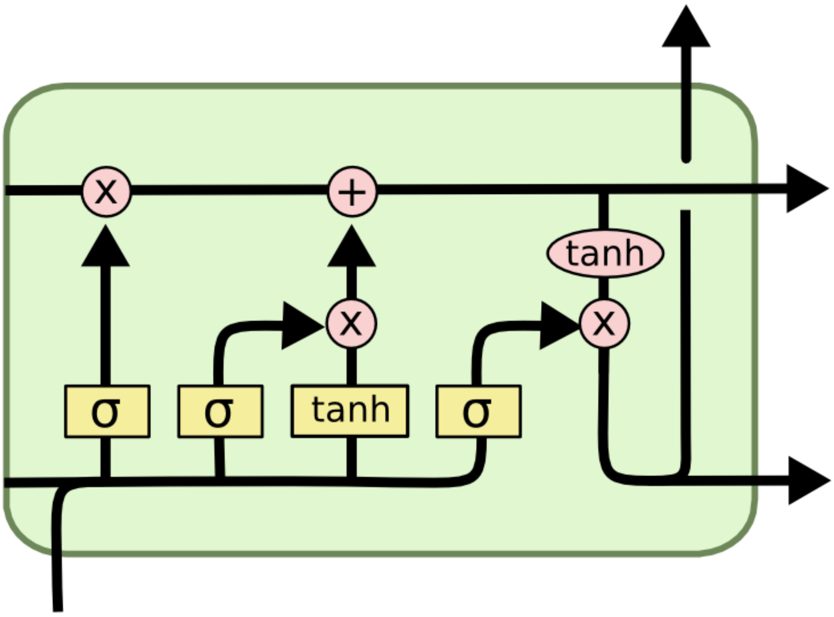

Figure 3.1: LSTM Diagram

In these equations, * represents matrix multiplication, and [ ] represents concatenation of vectors. The output of the LSTM at each time step is the hidden state, which can be fed into a final output layer for classification or regression tasks. A LSTM has been shown to be effective in modeling long-term dependencies and avoiding the vanishing gradient problem. In the context of stocks, the stock prices span over years, so LSTM method is suitable to predict adjusted closing prices.

3.2 Gated Recurrent Units

The second recurrent neural network used, the GRU network, is an alternative to the LSTM. A GRU network is similar to LSTM except it has 2 gates: a reset gate and an update gate. In addition to having fewer parameters, the GRU network also has a simpler architecture compared to the LSTM, making it easier to implement and understand. The reset gate and update gate in a GRU work together to selectively update the hidden state and control the flow of information throughout the network. By selectively retaining relevant information and discarding irrelevant information, the GRU can effectively capture long-term dependencies in time-series data while avoiding the vanishing gradient problem that can occur in traditional recurrent neural networks. Overall, the GRU network is a powerful tool for time-series prediction that can provide similar accuracy to the LSTM but with faster training times and simpler infrastructures[Dey17].

The update gate (Equation 3.7) includes σ as the sigmoid function. W_z is the weight matrix, $h(t-1)$ is the previous hidden state, and $x(t)$ is the current input. $x(t)$ is multiplied by its respective weight in W_z, and $h(t-1)$ stores information from the previous unit and is multiplied by its own weight in W_z. The reset gate (Equation 3.8) has the same equation as the update gate but the values of the weight matrix are different. The candidate hidden state (Equation 3.9) is an intermediate step to update the hidden state of the GRU. It is calculated based on the current input and previous hidden state and proposes a new hidden state at the current time. The candidate hidden state allows the GRU to update the final state and the hidden state more effectively. Lastly, the hidden state (Equation 3.10) is calculated by interpolating between the previous hidden state and the candidate hidden state.

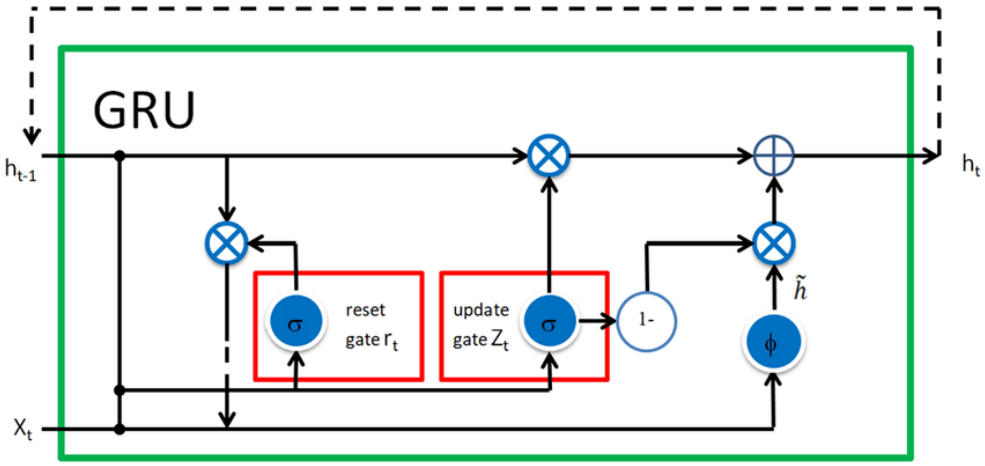

Figure 3.2: GRU Diagram

$$z(t) = \sigma(W_z * [h(t-1), x(t)]) \tag{3.7}$$

$$r(t) = \sigma(W_r * [h(t-1), x(t)]) \tag{3.8}$$

$$h'(t) = tanh(W * [r(t) * h(t-1), x(t)]) \tag{3.9}$$

$$h(t) = (1 - z(t)) * h(t-1) + z(t) * h'(t) \tag{3.10}$$

3.3 Model Results

To prepare the data for LSTM and GRU models, we adopt the MinMaxScaler technique, which scales the training stock prices to be between 0 and 1. We then run the training data for 50 epochs on both models, resulting in a root mean squared error (RMSE) of 5.26 for LSTM and 4.37 for GRU. Despite both methods showing slightly lower predicted stock prices than the actual prices, the predicted lines from both models are noticeably smoother than the actual lines. It is worth noting that the drop in actual stock price at the end of 2021 could be the reason for the lower predicted prices. Upon comparison, the GRU method slightly outperforms the LSTM method for this dataset, likely due to its architecture's ability to combine the benefits of the reset and update gates. However, further analysis is needed to determine if this is consistent across different datasets and time periods.

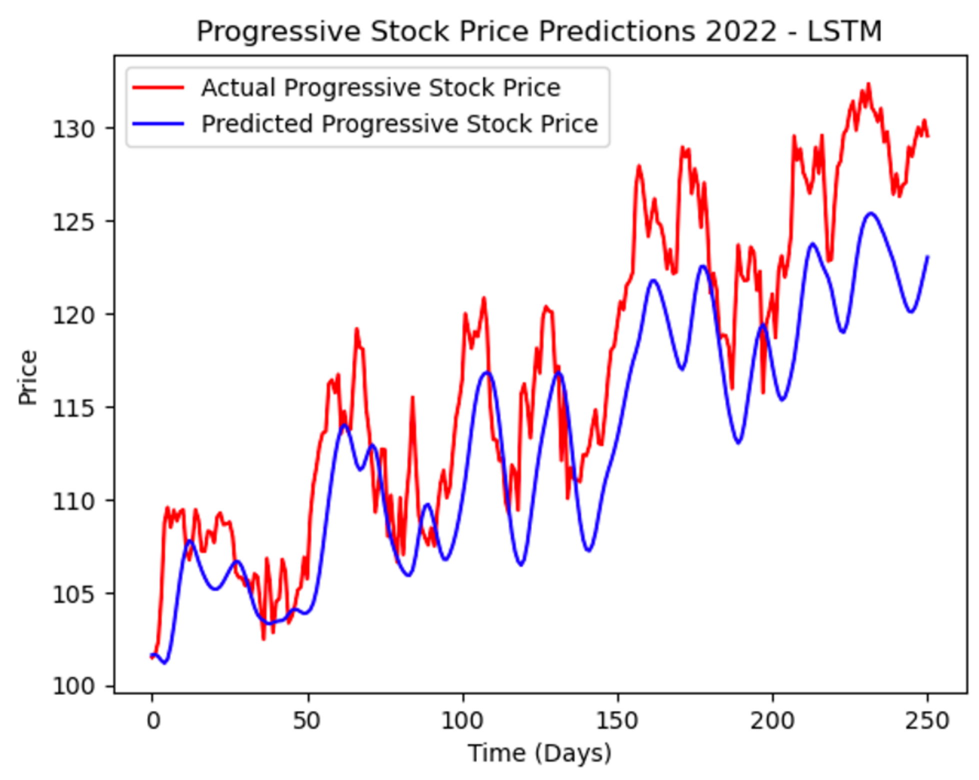

Figure 3.3: Actual vs. Predicted Returns - LSTM

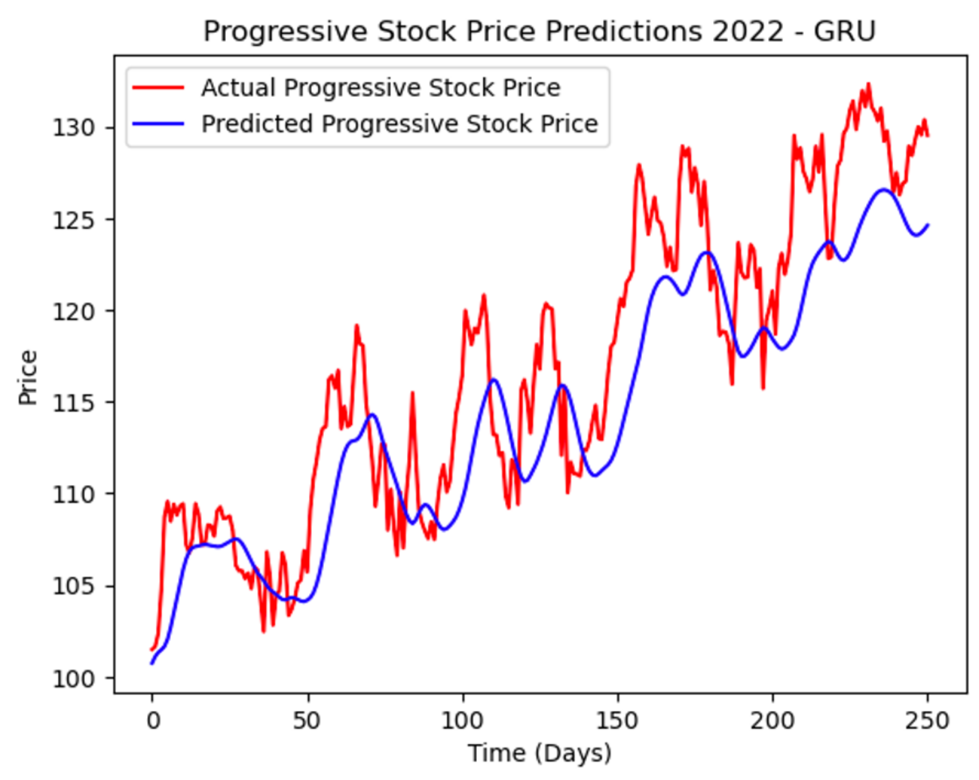

Figure 3.4: Actual vs. Predicted Returns - GRU

CHAPTER 4

Autoregressive Integrated Moving Average

The ARIMA model is a widely-used time-series forecasting method that is applied across many industries, including finance and economics. This method uses only prior data of the time series to make predictions, and is particularly popular for forecasting stock prices. However, there are limitations to the ARIMA model, such as its inability to extrapolate far into the future.

Another important consideration when using the ARIMA model is the stationarity of the time series data. Stationarity is a critical assumption in time series modeling, and refers to a situation where the statistical properties of the time series do not change over time. In other words, the time series is consistent and behaves similarly over all time periods [Abd19].

However, many real-world time series data are non-stationary, meaning that they exhibit trends or seasonality over time. In such cases, it is important to preprocess the data before applying the ARIMA model. This may involve decomposing the time series data into its underlying trend, seasonality, and residual components using techniques such as seasonal decomposition.

Once the data has been preprocessed to ensure stationarity, the ARIMA model can be applied to make predictions. It is important to note that the accuracy of the ARIMA model predictions can be impacted by the quality and quantity of the available data, as well as the choice of model parameters such as the number of lags and differences.

Overall, the ARIMA model is a useful tool for time-series forecasting that can provide valuable insights for businesses and investors. However, it is important to be aware of its limitations and to carefully preprocess the data to ensure stationarity before applying the model.

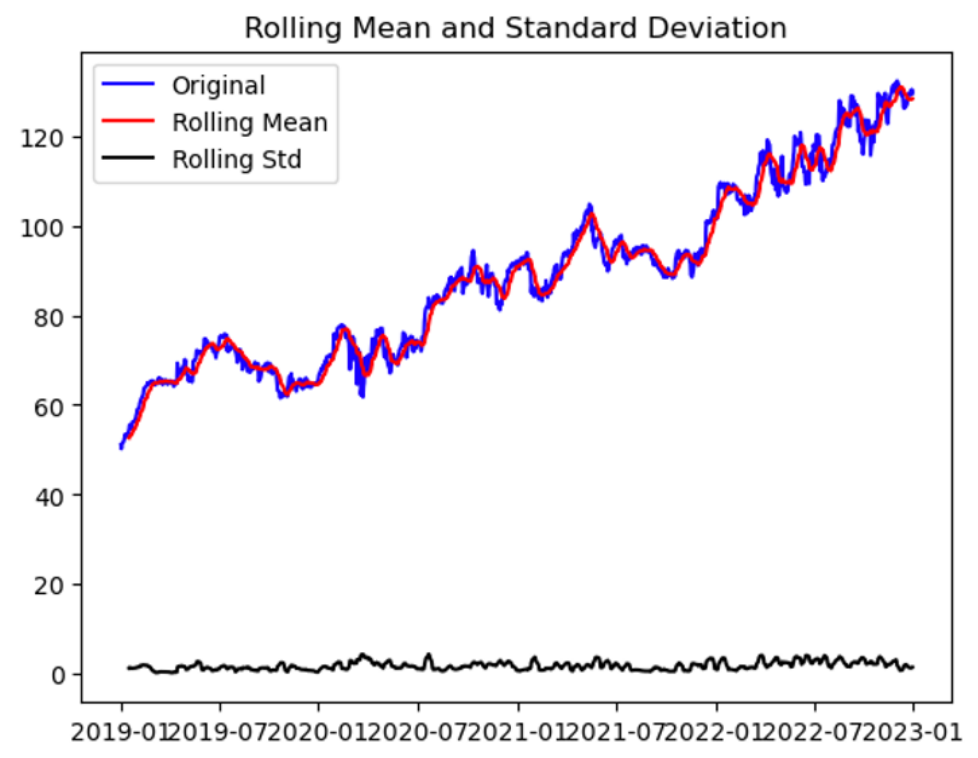

Figure 4.1: Rolling Mean and Standard Deviation of Data

To determine whether a time series is stationary or not, statistical tests like the Augmented Dickey Fuller Test (ADF) are commonly used. The ADF test is a hypothesis test that helps us determine if the series has a unit root, which indicates that the series is non-stationary. In this study, we performed an ADF test on Progressive's stock price data to check for stationarity. The null hypothesis of the test is that the series has a unit root in the autoregressive model, and the alternate hypothesis states that the series is stationary. The test yielded a p-value of 0.88, indicating that we cannot reject the null hypothesis, and hence, the series is non-stationary. Furthermore, we can visually see from Figure 4.1 that there is an increasing trend as time progresses, which further confirms the non-stationarity of the data. It is important to note that non-stationarity in the data can lead to inaccurate and unreliable predictions, which highlights the need to address this issue before proceeding with further analysis.

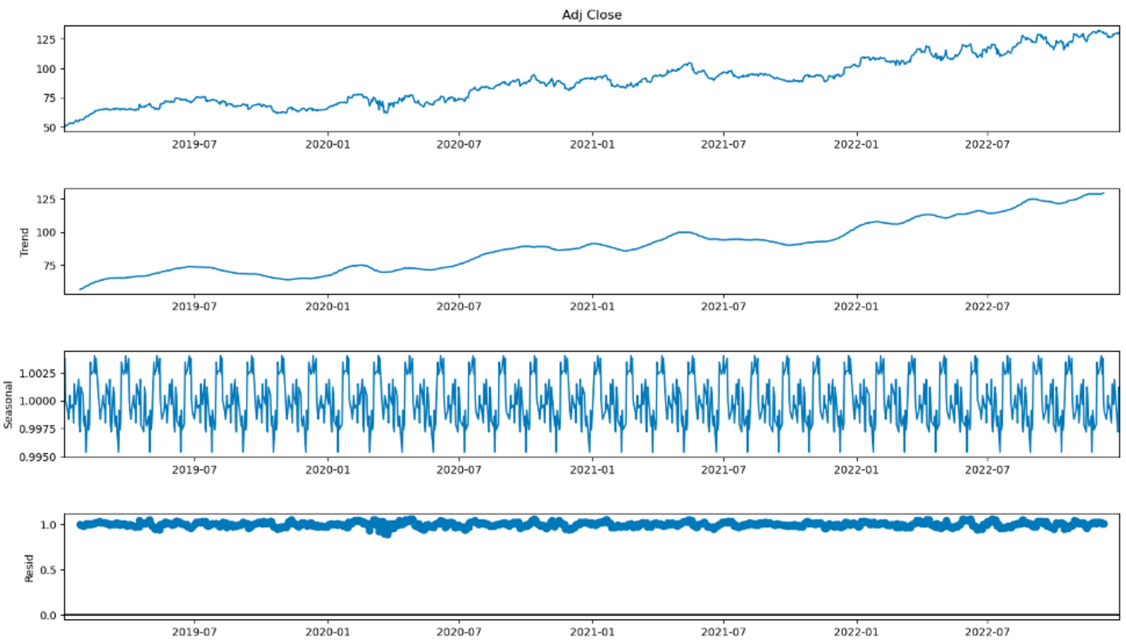

Figure 4.2: Decomposition Results

Before we conduct an ARIMA analysis, seasonality and trends in the data need to be separated. We use a seasonal decomposition approach which takes the log of the series to reduce the growing trend, and a rolling average is computed using data from the past 12 months. Figure 4.2 below shows the smoothed adjusted closing prices. After taking the log of the series, the moving average plot becomes relatively stationary and we can continue with the ARIMA analysis.

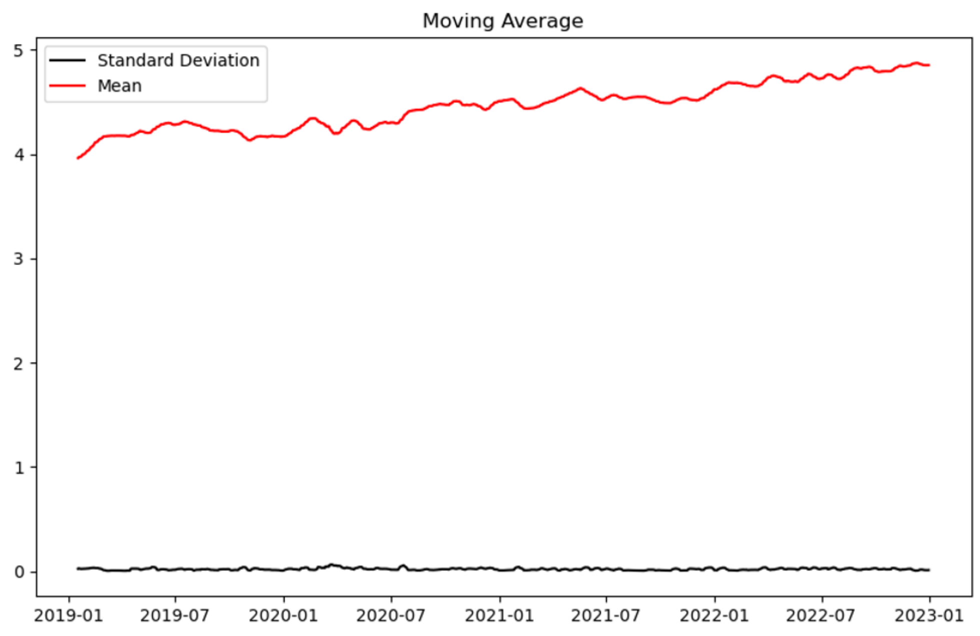

Figure 4.3: Rolling Mean and Standard Deviation of Data after Decomposition

After removing seasonality and trends, we split the data into the same train and test set as previously and use the auto_arima function in the pmdarima library in to compute the ARIMA model's parameters.

An ARIMA model has three parameters, labeled as ARIMA(p, q, d). The first parameter, p, represents the number of autoregressive terms in the model, or simply the lag. It determines how many past values of the time series will be used to predict future values. The second parameter, d, represents the number of times the time series needs to be differenced to achieve stationarity. Stationarity means that the statistical properties of the time series remain constant over time. The third parameter, q, represents the order of the moving averages. It determines the number of lagged forecast errors in the prediction equation. By choosing the appropriate values for p, d, and q, an ARIMA model can accurately capture the patterns in the time series and make reliable predictions for the future.

```
                               SARIMAX Results
==============================================================================
Dep. Variable:              Adj Close   No. Observations:                  753
Model:                 ARIMA(3, 0, 2)   Log Likelihood               -1265.617
Date:                Sun, 16 Apr 2023   AIC                           2545.234
Time:                        13:02:44   BIC                           2577.602
Sample:                             0   HQIC                          2557.704
                                - 753
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         79.6039     13.480      5.905      0.000      53.184     106.024
ar.L1          0.6357      0.218      2.917      0.004       0.209       1.063
ar.L2         -0.0354      0.180     -0.197      0.844      -0.388       0.317
ar.L3          0.3955      0.113      3.491      0.000       0.173       0.618
ma.L1          0.2231      0.213      1.048      0.294      -0.194       0.640
ma.L2          0.4763      0.100      4.742      0.000       0.279       0.673
sigma2         1.6762      0.051     32.905      0.000       1.576       1.776
===================================================================================
Ljung-Box (L1) (Q):                   0.04   Jarque-Bera (JB):               684.19
Prob(Q):                              0.85   Prob(JB):                         0.00
Heteroskedasticity (H):               1.64   Skew:                            -0.27
Prob(H) (two-sided):                  0.00   Kurtosis:                         7.64
===================================================================================
```

Figure 4.4: ARIMA Results

4.1 Model Results

From the model result (Figure 4.4) and using lowest AIC as the metric, the best result is ARIMA(3, 0, 2) model. From the residual plots (Figure 4.5), the standardized residual plot shows the no clear pattern and the residuals fluctuate around 0. The histogram of density shows a normal distribution with a mean centered at 0. The normal qq plot is linear with a few exceptions at the upper and lower tails. The correlogram plot shows no autocorrelation between residuals. Based off of the plots, the ARIMA assumptions are met. The RMSE from this model is 6.03.

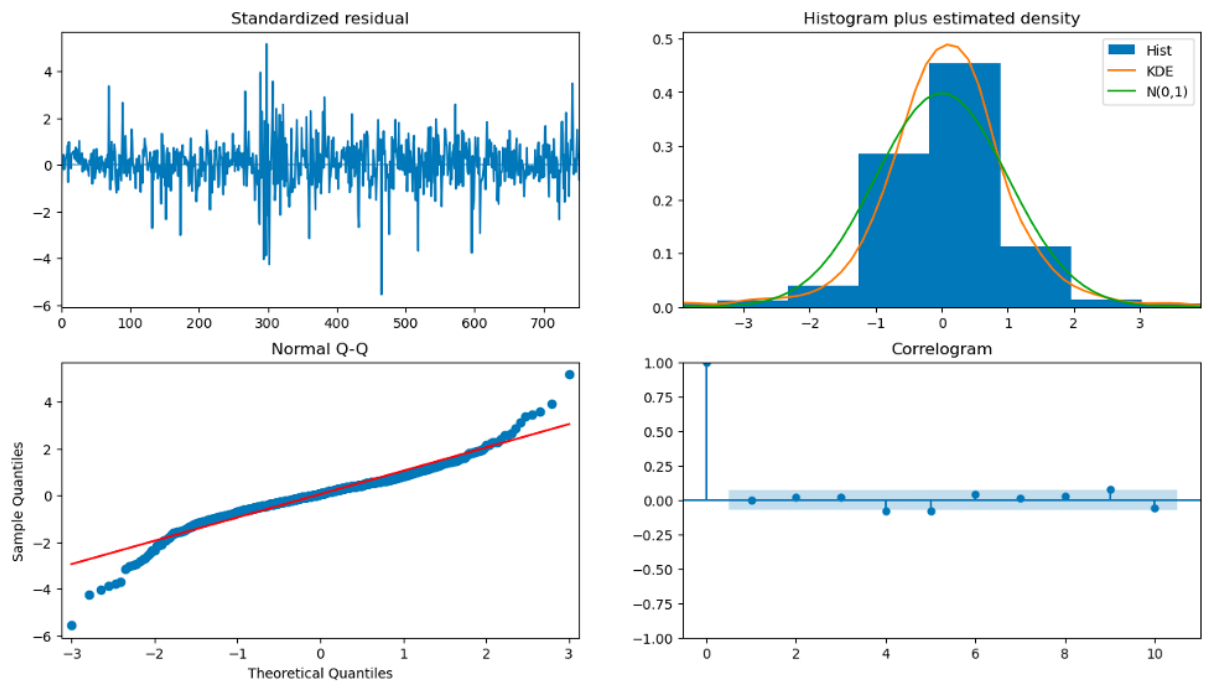

Figure 4.5: ARIMA Residual Plots

CHAPTER 5

Model Comparison

After testing out 2 recurrent neural networks and one time-series method, we can compare the RMSE. The table below shows the RMSE of all 3 methods for Progressive.

LSTM models have certain limitations. They require large datasets to train effectively, which can make them computationally expensive. Additionally, LSTMs can be prone to overfitting, and their black box nature can make it challenging to understand how they are making their predictions[Sin17]. The ARIMA model has certain disadvantages that make it less effective compared to modern deep learning models such as LSTM and GRU. One of the main assumptions of the ARIMA model is the stationarity and normality of data, which means that the mean, variance, and autocovariance of the data remain constant over time. In reality, many real-world datasets are non-stationary, which means that their statistical properties change over time. To use ARIMA on non-stationary data, it is often necessary to transform or decompose the original data. ARIMA is also better suited for short-term predictions and may not perform as well as LSTM or GRU for longer-term forecasts. From this analysis, the GRU method gives the lowest RMSE. The GRU model has the advantage of faster training due to its fewer parameters compared to LSTM. It is also less prone to overfitting, which is a common issue in complex models. Furthermore, GRU can be a good choice for smaller datasets where avoiding overfitting is challenging.

Method	LSTM	GRU	ARIMA
RMSE	5.26	4.37	6.03

Table 5.1: RMSE Comparison

CHAPTER 6

Competitor Analysis

To further evaluate the effectiveness of the GRU model on competitor data, we conducted a comparative analysis of the RMSE values for both Allstate and Travelers. We found that the RMSE values for both competitors were similar to the value we obtained for Progressive Corporation. This suggests that the GRU model can be applied to different companies in the insurance industry and still provide accurate predictions. Additionally, we observed that the predicted stock prices for both Allstate and Travelers were smoother than the actual stock prices, indicating that the model is able to reduce the noise in the data and provide more stable predictions. Overall, our research has demonstrated that the GRU model can be effectively applied to the insurance industry to predict stock prices during recessionary and inflationary times, providing investors with valuable insights for making informed investment decisions.

The RMSE is found in Table 6.1 below, and Figures 6.1 and 6.2 show the predicted returns.

Company	Allstate	Travelers
RMSE	4.65	4.37

Table 6.1: GRU Method RMSE for Competitors

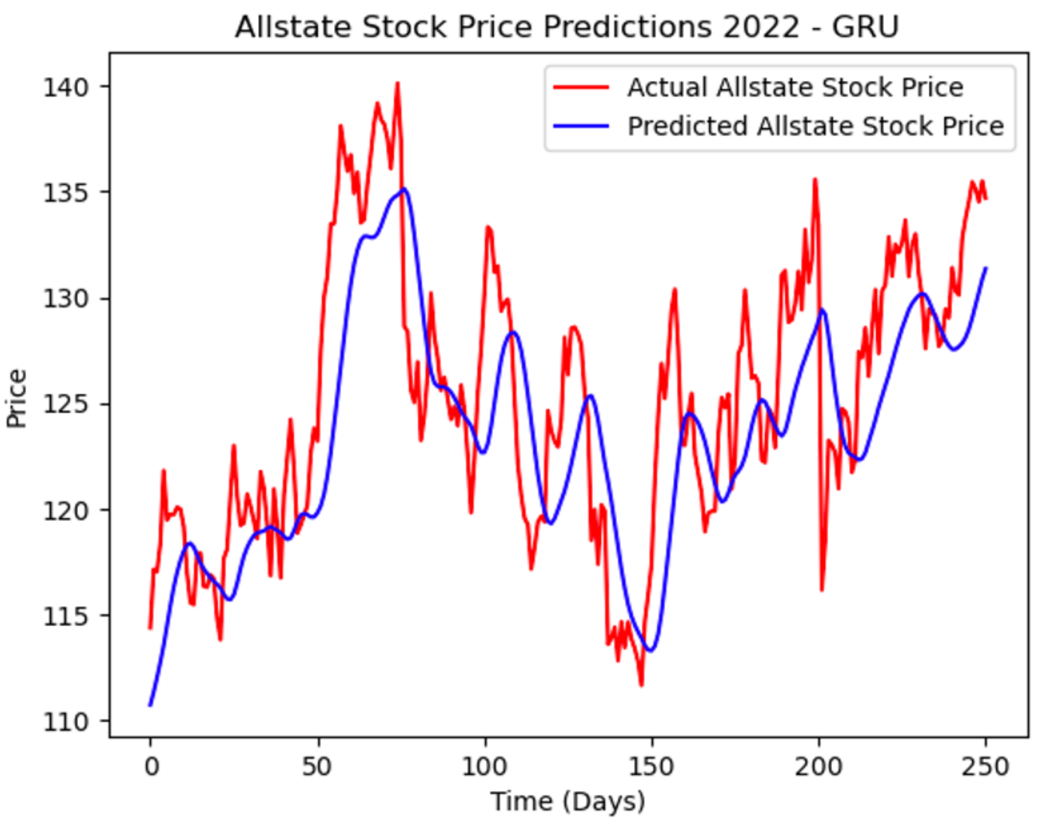

Figure 6.1: Allstate Actual vs. Predicted Returns - GRU

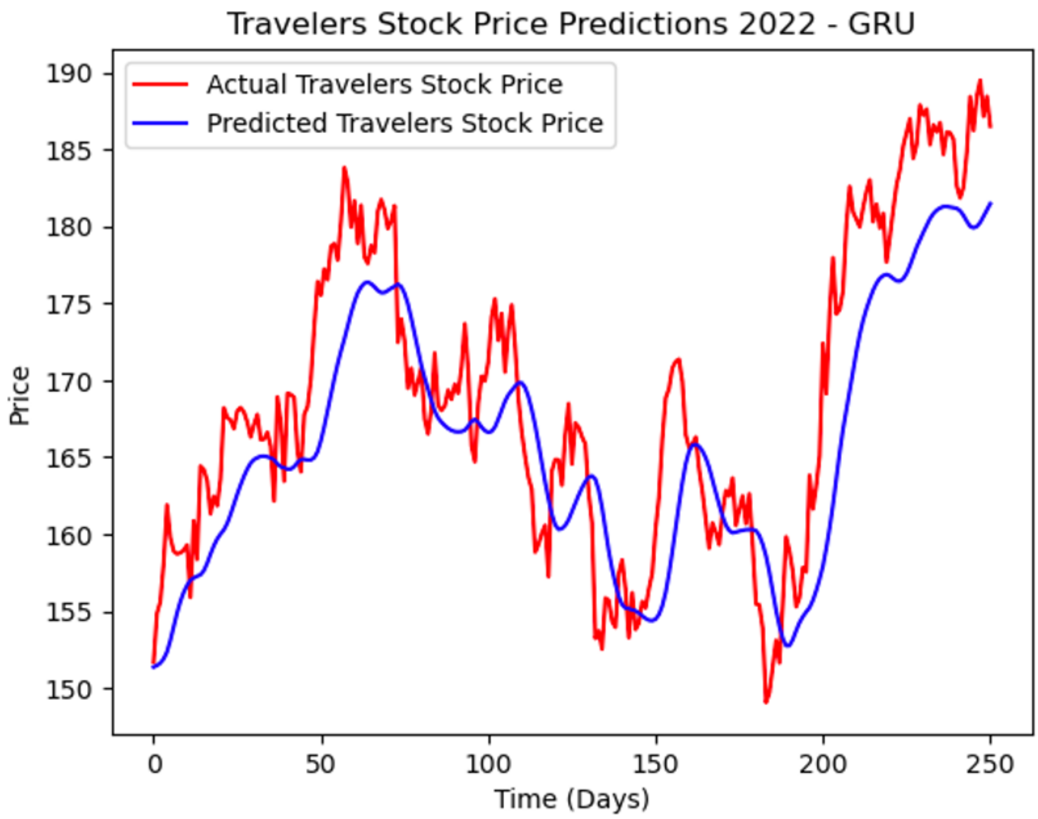

Figure 6.2: Travelers Actual vs. Predicted Returns - GRU

CHAPTER 7

Conclusion & Further Discussion

The aim of this analysis is to predict stock prices for property and casualty insurance in 2022, given that they are known to be relatively stable and recession-proof. We trained the data using weekly adjusted closing prices for Progressive Insurance from 2019 - 2021 and then used 3 different methods to predict the weekly adjusted closing prices for 2022. The two recurrent neural network models used were LSTM and GRU, and the ARIMA time series analysis was also used. The model with the lowest RMSE is the GRU method because it is least prone to overfitting and does not have a stationarity assumption.

The strength of the data is that the time series goes back many years, so we can increase the size of the data for future work. To improve an LSTM model, there are several strategies that can be employed. One approach is to increase the amount of training data, as LSTMs require large datasets to train effectively. Another strategy is to reduce the model's complexity, such as by decreasing the number of hidden layers or neurons, to avoid overfitting. Additionally, we can adjust the hyperparameters of the model. Hyperparameters include the learning rate or batch size, which can improve the model's performance. Regularization techniques, such as L1 or L2 regularization or dropout, can also help prevent overfitting. Preprocessing techniques, like feature scaling or normalization, can also improve the model's performance. Finally, experimenting with different architectures, such as bidirectional LSTMs or stacked LSTMs, can also lead to improvements in the model's accuracy.

To improve the ARIMA model, we should ensure that the data is stationary by transforming the data better. Additionally, using an extended version of the ARIMA model, such as SARIMA or ARIMAX, can capture more complex patterns in the data. Another way to improve an ARIMA model is to incorporate external factors, such as economic indicators or weather data, using an approach like dynamic regression. Furthermore, ensembling multiple ARIMA models or using a combination of ARIMA and other models, such as LSTM or GRU, can improve the accuracy and robustness of the model. Finally, using grid search or other optimization techniques to tune the hyperparameters of the model can further improve its performance.

REFERENCES

[Abd19] Nasir Hamid Rao Abdul Jalil. "Chapter 8 - Time Series Analysis (Stationarity, Cointegration, and Causality)." pp. 85–99, 2019.

[Dey17] Rahul Dey. "Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks." 2017.

[Hoc97] Sepp Hochreiter. "LONG SHORT-TERM MEMORY." 1997.

[RNN] "Recurrent Neural Network." https://deepai.org/machine-learning-glossary-and-terms/recurrent-neural-network. [Accessed 1-May-2023].

[Sin17] Gaurav Singhal. "LSTM versus GRU Units in RNN." 2017.

[SLC22] SLC. "Inflation: 2022's biggest threat to insurance investment portfolios?" 2022.

[Yok21] Mamiko Yokoi-Arai. "Responding to the COVID-19 and pandemic protection gap in insurance." 2021.