Luyuan Fan USCID: 2520035085

The Comparison between ARIMA and LSTM
ISE537 Final Project

0 Abstract

This study evaluated the performance of ARIMA, LSTM, and GRU models for forecasting stock prices of JPMorgan Chase (JPM) and NVIDIA (NVDA), representing the financial services and technology sectors, respectively. The results revealed that ARIMA outperformed deep learning models in predictions, achieving the lowest RMSE values for both stocks, and GRU performed better than LSTM. While LSTM and GRU demonstrated potential in capturing temporal dependencies, their performance was hindered by the small dataset size, overfitting, and the relatively linear nature of the data. The relatively underwhelming performance of deep learning models in this analysis is not surprising, as similar results had been reported in prior research. For example, a study by Yamak et al. (2020) highlighted that ARIMA outperformed deep learning models for time series forecasting tasks involving smaller datasets or data with linear trends and seasonal patterns. This reinforces the findings of the current analysis.

1 Data Processing

1.1 Data acquisition

In the study, two stocks are selected, JPM and NVDA, and their daily close prices were from the yfinance library for the period from December 31, 2018 to December 31, 2023. Both datasets were split into two parts: the first four years (2018–2022) as the training data, and the last year (2023) as the testing data. This division ensured that the models were trained on past data and tested on unseen data, reflecting real-world scenarios.

1.2 Exploratory data analysis (EDA) for ARIMA

1.2.1 Transformations

To prepare for analysis, several key transformations of the close prices were computed:

- Price Differences (prices_diff): the first differences of the closing prices to capture daily changes.
- Logarithmic Prices (prices_log): the natural logarithm of the prices to stabilize variance.

- Log Price Differences (prices_log_diff): the first differences of the logarithmic prices.

1.2.2 Stationarity test

Using the Augmented Dickey-Fuller (ADF) test, the stationarity of each transformation was evaluated. Stationarity is crucial for ARIMA models. Results showed that close prices and logarithmic prices were non-stationary and price differences achieved enough stationarity, even better than log price differences for JPM, making them suitable for ARIMA modeling.

1.2.3 ACF and PACF analysis

The Autocorrelation (ACF) and Partial Autocorrelation (PACF) plots were generated for the differenced series to identify the lag structure and try to assess ARIMA model parameters.

1.3 Data standardization and regularization for LSTM and GRU

For the deep learning models (LSTM and GRU), the data was normalized to improve convergence during training. Prices were scaled to the range [0, 1] on the training data using Min-Max Scaler, and the test data was scaled using the fitted scaler, which retained the information from the training data to avoid data leakage or inconsistencies. L2 regularization was applied to the model layers in the GRU to prevent overfitting during training.

2 Model Introduction

2.1 ARIMA

The Autoregressive Integrated Moving Average (ARIMA) model is a classical statistical method commonly used for time series forecasting. It combines three components: Autoregression (AR), Integration (I) and Moving Average (MA), where AR captures the relationship between a time series and its past values, I accounts for differencing to make the series stationary and MA models the relationship between a time series and past forecast errors.

Through the ADF test, the integration parameter d was determined to be 1 for both JPM and NVDA, consistent with the results from auto_arima, which is a function in the pmdarima library that identifies optimal values for ARIMA parameters (p, d, q) by iterating through possible

combinations and evaluating them based on a chosen metric, such as the Akaike Information Criterion (AIC).

Since the p and q parameters could not be clearly identified from the ACF and PACF plots, the searchARMA function was implemented to systematically search through combinations of the AR (p) and MA (q) parameters to find the best model based on the AIC. The integration parameter (d) was set to 0, as differencing had already been applied.

For JPM, the optimal (p, q) was found to be (3, 2), which was consistent with the result from auto_arima. For NVDA, the optimal (p, q) was determined to be (5, 4), differing from the auto_arima result but yielding a lower AIC. This difference could be attributed to the stepwise algorithm employed by auto_arima, which, while computationally efficient, may skip certain parameter combinations that could lead to better performance. The parameters (5, 4) were chosen based on the results from searchARMA, as the AIC and BIC values for all parameter combinations were directly observable, making the model selection process more interpretable and reliable.

2.2 LSTM

The Long Short-Term Memory (LSTM) network is a type of recurrent neural network (RNN) designed to handle sequential data. It is particularly effective in learning long-term dependencies by using memory cells and gating mechanisms. The key features of LSTM include Forget Gate, which determines what information to discard from the memory, Input Gate, which decides what information to update in the memory and Output Gate, which outputs relevant information for the current timestep.

In this study, the LSTM model was implemented with two LSTM layers. The first LSTM layer contains 128 units, while the second LSTM layer has 64 units. This structure was followed by two fully connected dense layers, with 25 units in the first dense layer and 1 unit in the output layer. For JPM, the input consisted of sequences of 20 past observations, while for NVDA, sequences of 10 past observations were used as input. Other parameters, including the optimizer and training configuration, were the same for both models. The Adam optimizer was used with Mean Squared Error (MSE) as the loss function, and the models were trained for 100 epochs with a batch size of 16 and a validation split of 20%.

2.3 GRU

The Gated Recurrent Unit (GRU) is another variant of RNN that simplifies the architecture of LSTM while retaining its ability to learn long-term dependencies. GRU uses only Reset Gate, which controls how much past information to forget and Update Gate, which balances between retaining past information and incorporating new information.

GRU was chosen as an alternative to LSTM due to its simpler architecture, which is better suited for the small dataset used in this study. The simpler design often results in faster training while maintaining comparable accuracy. Additionally, GRU is more efficient in handling the vanishing gradient problem. Its gating mechanism captures long-term dependencies more directly, with the reset gate controlling the integration of new and old information and the update gate determining the extent of state updates. This makes GRU better at capturing key long-term dependencies in certain time series tasks. Moreover, GRU's design allows it to flexibly learn patterns in time series data, automatically deciding what to remember and what to forget. To further enhance its robustness and prevent overfitting, L2 regularization was applied to the first layer, and the model underwent extensive hyperparameter tuning to optimize its performance.

3 Model Evaluation

3.1 ARIMA

To evaluate the performance of ARIMA, both in-sample and out-of-sample results were analyzed, alongside residual diagnostics, for JPM and NVDA.

3.1.1 In-Sample Performance

The in-sample evaluation demonstrated that the ARIMA (3, 1, 2) model for JPM and the ARIMA (5, 1, 4) model for NVDA successfully captured the primary trends and fluctuations in the respective time series. The fitted values closely aligned with the actual prices as shown in Figure 1, reflecting the models' ability to explain the variations within the training dataset.

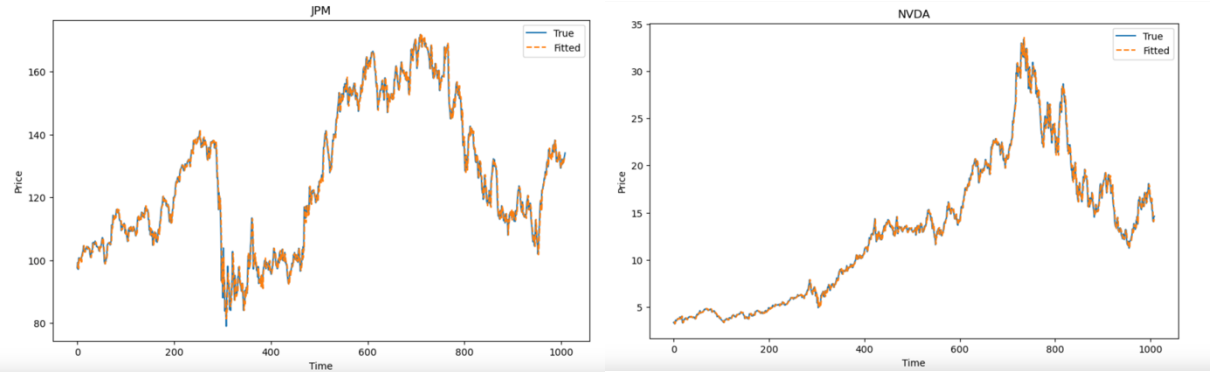


Figure 1: ARIMA in-sample performances of JPM (left) and NVDA

3.1.2 Residual Diagnostics

Residual diagnostics were conducted to assess the adequacy of the models. The residual plots for both stocks indicated randomness, with residuals distributed around zero and no significant autocorrelation detected, as confirmed by the Ljung-Box test (p-values > 0.05). The histograms of residuals showed approximate normality, while the Q-Q plots indicated reasonable adherence to a normal distribution, with only minor deviations at the tails. Furthermore, the residual variance appeared stable over time, suggesting the models effectively captured the underlying patterns without substantial heteroscedasticity.

3.1.3 Out-of-Sample Performance

The out-of-sample performance was evaluated using the Root Mean Square Error (RMSE) metric. For JPM, the ARIMA (3, 1, 2) model achieved an RMSE of approximately 1.8070, while for NVDA, the ARIMA (5, 1, 4) model achieved a RMSE of 1.0446. These results indicated that both models demonstrated strong predictive capabilities, as illustrated in Figure 2.

The strengths of the ARIMA models lie in their interpretability and their ability to model stationary data effectively. The differencing transformations applied to the price series ensured stationarity, enabling the ARIMA models to deliver robust results.

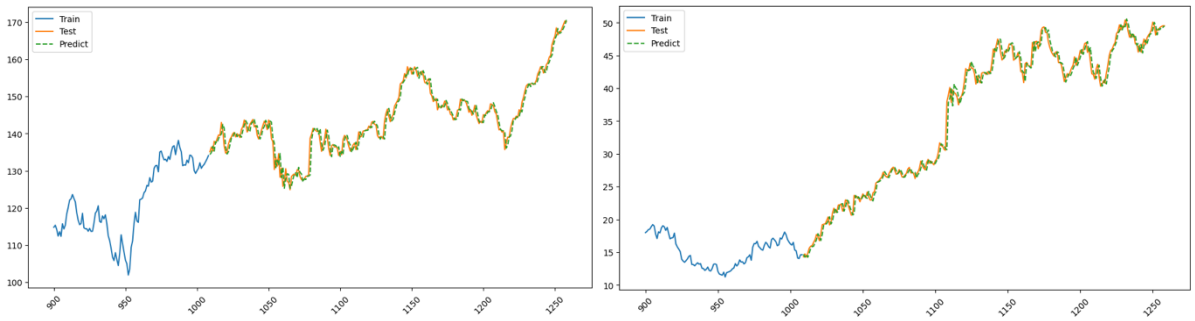


Figure 2: ARIMA out-of-sample performances of JPM (left) and NVDA

3.2 LSTM

The LSTM model was evaluated on both JPM and NVDA datasets. For JPM, the model achieved an RMSE of 1.8816. As shown in Figure 3, the predictions closely followed the trend of the actual data, with slight deviations in highly volatile regions. For NVDA, the model achieved an RMSE of 1.7146, but the predicted values showed noticeable deviations from the actual data in certain regions, particularly during periods of high volatility, indicating limitations in the model's ability to fully capture temporal dependencies.

Despite its ability to model complex patterns, the LSTM underperformed compared to the ARIMA model in terms of forecasting accuracy. This was likely attributed to the limited training data and the overfitting tendency of neural networks when applied to small datasets.

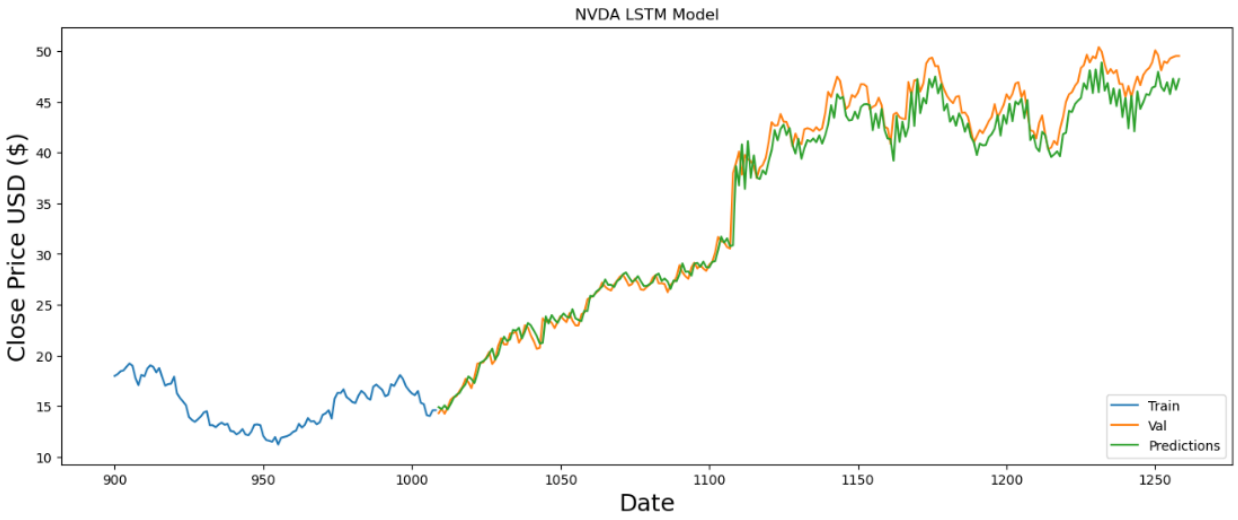
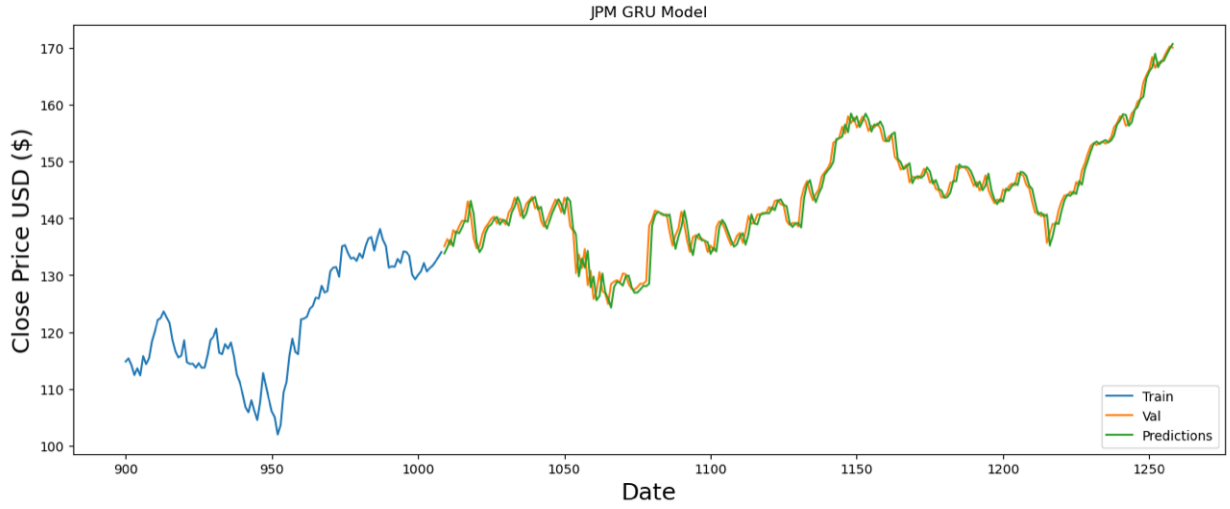


Figure 3: LSTM out-of-sample performances of JPM and NVDA

3.3 GRU

The GRU model demonstrated significantly better performance than LSTM for both datasets and achieved results close to those of the ARIMA model. For JPM, the GRU achieved an RMSE of 1.8299, outperforming the LSTM. For NVDA, the GRU's RMSE was 1.0930, showing a substantial improvement over the LSTM. As illustrated in Figure 4, the GRU model provided a much closer fit to the true values for NVDA compared to the LSTM. The GRU's simpler architecture compared to the LSTM allowed it to achieve superior results while being computationally more efficient.



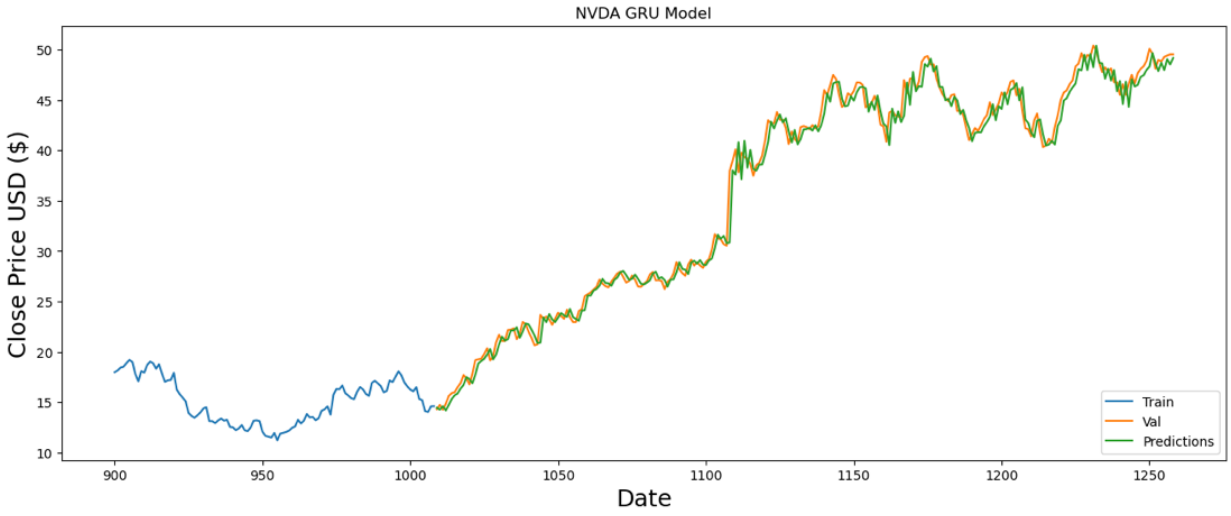


Figure 4: GRU out-of-sample performances of JPM and NVDA

Deep learning models like LSTM and GRU are powerful tools for handling complex and non-linear time series. However, their performance depends heavily on the availability of large datasets, effective hyperparameter tuning, and the complexity of the underlying patterns. In this analysis, the simpler, interpretable ARIMA model outperformed deep learning methods due to the relatively small dataset size and the linear nature of the data. Additionally, ARIMA's ability to treat noise as random disturbances enhanced its robustness, whereas deep learning models struggled with overfitting and required larger datasets to learn non-linear patterns effectively.

4 Financial Interpretation

The differences in model performance for JPM and NVDA can be explained by the distinct characteristics of their industries. JPM, operating in the financial services sector, is influenced by relatively stable and linear macroeconomic factors, such as interest rates and regulatory cycles, which are well-suited for ARIMA's statistical approach. In contrast, NVDA, with a smaller RMSE but also a significantly smaller data scale, a leader in the highly volatile technology sector, experiences non-linear growth driven by innovation, market sentiment, and global supply chain dynamics. These complex patterns challenge both ARIMA and deep learning models, with deep learning further limited by the small dataset and risk of overfitting. Industry-specific dynamics play a critical role in determining the suitability and effectiveness of forecasting models.

5 Conclusion

This study evaluated the performance of ARIMA, LSTM, and GRU models for forecasting stock prices of JPM and NVDA, representing the financial services and technology sectors, respectively. The results showed that ARIMA outperformed deep learning models in both accuracy and robustness. GRU demonstrated significantly better performance than LSTM and came close to ARIMA, highlighting its adaptability and efficiency. However, the deep learning models were constrained by the small dataset size, risk of overfitting, and the relatively linear nature of the data. While NVDA's RMSE values were smaller, the lower overall scale of its dataset values must be considered when comparing model performance. These findings underscore the importance of aligning model selection with the characteristics of the dataset and the specific dynamics of the industry. Future work could explore hybrid models or larger datasets to leverage the strengths of both statistical and deep learning approaches.