SYDE 660A Systems Design Graduate Workshop 1 - AI and Machine Learning

Team 3 - Final Project Report

Authors:
Shinong Mao (ID: 20570392)
Zhiliu He (ID: 21090731)
Beilin Ye (ID: 21053851)

Date: September 24, 2024

Contents

1 Introduction

In the financial trading market, the stock market stands out as a crucial asset class. Developing prediction models for stock markets using artificial intelligence (AI) is a promising field of research [2]. However, predicting stock prices is challenging due to the highly nonlinear and volatile nature of stock data, which is sensitive to factors such as political changes, global financial crises, investor psychology, and macroeconomic variables [3].

Small and Medium-sized Enterprises (SMEs), defined by their limited number of employees and revenue, play a critical role in fostering new businesses and driving innovation, significantly contributing to national economic growth [4]. According to McKinsey & Company [5], SMEs help prevent monopolies and promote a healthier economy through competition. In the stock market, SMEs exhibit high uncertainty in stock price trends but also present significant investment potential due to their agility and growth opportunities. Despite their importance, there is a notable scarcity of research specifically focused on stock prediction for SMEs.

Current solutions include human-conducted investment analyst reports, which tend to be expensive and slow [6]. While generalized AI models for stock predictions exist, they are often validated by and applied to stock prediction for large-cap companies, such as Apple Inc., Amazon Inc., and Google LLC [7], limiting their effectiveness for small and medium-sized businesses' stock prediction. Our project aims to address this gap by developing AI models tailored to the unique attributes of SME stocks.

2 Project Scope & Objectives

2.1 Problem Scope

The primary issue is the lack of research and tools specifically for predicting the stock prices of small-cap stocks. Most existing studies focus on large-cap stocks or use generalized models. For instance, Wang et al. [8] used S&P 500 and Dow Jones Industrial Average datasets, Fischer and Krauss [9] utilized S&P 500 index data, and Thakur and Kumar researched NASDAQ, Dow Jones, and S&P 500 indices [10], all dominated by large-cap stocks.

Small-cap stocks (SMEs) have unique characteristics, such as high volatility, low liquidity, limited analyst coverage, and scant AI model research. These factors necessitate a specialized model to provide targeted, efficient, and accurate predictions.

Investing in small-cap stocks can be highly profitable, as SMEs are often undervalued [11] and tend to outperform large caps, especially after recessions [12]. Small caps usually peak and trough before large caps in market cycles and can outperform during young bull markets and economic recoveries [13]. Despite their growth potential, SMEs present higher risks due to narrow economic moats, leading to increased compe-

tition and volatility [14]. Therefore, tools are needed to support personal investment decisions in small caps, helping investors navigate risks and capitalize on opportunities. Small-cap stocks are more sensitive to interest rates, consumer spending, and market sentiment [13]. Incorporating these insights into our AI model can enhance its predictive accuracy and utility. However, current AI-based stock prediction models for SMEs are limited and focused on individual companies.

Our primary users are individual investors who favor shorter-term trading and are willing to take moderate risks for higher returns. Secondary users include those seeking signal indicators for mergers and acquisitions, conducting thorough research based on initial tool indications. Our solution is not intended for institutional investors who require maximum accuracy and prefer fundamental over technical analysis.

User feedback indicates a high demand for cost-effective and efficient predictive tools with moderate accuracy and transparency for non-technical users. Users prefer limited customization options but demand frequent, real-time updates and alerts for informed decision-making.

Based on our analysis, the identified problem is significant among our users. The shortage of tailored solutions for predicting small-cap stock performance, considering all SME characteristics, is a widespread concern. Existing research predominantly focuses on large-cap stocks, leaving a gap for tools that cater specifically to small-cap investors. Therefore, our project addresses a prevalent need within the investment community, offering solutions to enhance decision-making and improve profit outcomes for our users.

2.2 Project Objectives

Based on the identified needs, we finalized the design requirements and constraints, which are categorized into those achieved by the AI stock prediction model and those implemented through the user interface (UI).

Firstly, the AI stock prediction model is designed to deliver highly accurate predictions, taking into account the unique characteristics and market behavior of small-cap stocks. It should accurately predict trends and provide predicted prices within a specified tolerance. In addition to accuracy, the model must generate results quickly, ideally within seconds, to ensure timely decision-making and meet user expectations for acceptable waiting times [15]. It should also consume minimal computational resources, making it suitable for deployment on standard personal computing devices and ensuring cost-effectiveness. The AI model should be affordable for personal investors, ensuring that the cost of using the tool does not outweigh the potential revenue earned from its use.

For UI design, the tool should be user-friendly and effective for personal investors. Firstly, the tool should be easy to use, ensuring accessibility for users without any technical knowledge. This includes a straightforward and intuitive user interface (UI) and clear usage documentation. Additionally, the tool is expected to support personal investors in making decisions by providing clear and actionable insights. Thus, the UI

should also offer frequent, real-time updates and alerts, enabling informed decision-making.
These requirements are detailed in the Quality Function Deployment (QFD) chart (see Figure1) in the Appendix. It highlights that the most critical factor for our users is the accuracy of the stock price predictions. Personal investors expect to use our tool to generate revenue from the stock market, making reliable predictions essential. Inaccurate predictions can lead to poor investment decisions and financial losses. However, cost-effectiveness is also crucial. The QFD chart indicates that low computational resource consumption and relatively high efficiency are the second most important factors. Personal investors, who typically have smaller investment amounts compared to large institutions, are sensitive to costs. If the tool's costs exceed the potential revenue, they are unlikely to use it.
In summary, our project aims to provide a robust, user-friendly, and cost-effective tool that enhances decision-making and improves profit outcomes for personal investors by accurately predicting small-cap stock prices.

3 Designed Solution

In our designed solution, we prioritized achieving high fidelity for AI code, ensuring robust and accurate implementation of predictive models. Conversely, we adopted a low-fidelity approach for the user interface2, with an initial focus on model performance over UI design. Therefore, the following content will concentrate on the development and evaluation of the AI models.

3.1 Model Training Process

The model training process involves several key steps to ensure accurate and efficient prediction capabilities. Initially, data collection is performed by downloading data from open public resources followed by data pre-processing to clean and prepare the dataset. One important pre-experiment step is feature selection. Based on previous studies, multi-feature models are considered superior to single-feature models due to several key advantages [16]. Firstly, the richness of information provided by multiple features offers a more comprehensive dataset, enhancing the model's ability to capture the underlying patterns. Additionally, using multiple features contributes to model stability by making the model less sensitive to noise in any individual feature, thus improving overall robustness. Moreover, incorporating multiple features enables the model to capture complex nonlinear relationships and interactions that a single feature might miss. This is particularly important in financial datasets where such relationships are common.
To identify the most relevant features, we employed general feature selection methods and initially evaluated each feature by combining the results. The first method is correlation analysis [17], by which we select features with an absolute correlation greater

than 0.3 with the target variable, i.e. close price. Another method is Recursive Feature Elimination (RFE) [18], which iteratively removes the least important features based on a specified model until the optimal set is reached. Features marked as True by RFE are retained for the final model. The last method we applied is L1 regularization (Lasso regression) [19], which inherently performs feature selection by driving the coefficients of less important features to zero, thus choosing only the most impactful ones for the model.

In the next steps, specific experiments and analyses will be conducted for different models to determine the most suitable features. This ensures that the feature selection process is tailored to the characteristics and requirements of each model, maximizing its performance. Once the input data is ready, it is fed into the model for training. During the training phase, the model parameters and the features will be further adjusted to learn from the processed data. Subsequently, fine-tuning of hyperparameters is conducted to minimize loss and error, optimizing the AI model's performance. Finally, model validation is carried out using the training set to evaluate and confirm the model's effectiveness and accuracy before deployment. This systematic approach ensures that the model is well-trained, optimized, and validated for robust predictive performance.

3.2 Candidate Models

We have five candidates for our AI stock prediction model, comprising three tree-based models and two neural network-based models. The tree-based models include Random Forest (RF), Gradient Boosting Decision Tree (GBDT), and Adaptive Boosting (Adaboost). The neural network-based models are Long-Short Term Memory (LSTM) and Echo State Network (ESN).

We selected these models due to their proven effectiveness in time series prediction. LSTM [7] and ESN are well-known for their capabilities in handling sequential data, making them ideal for stock price forecasting. Additionally, our choice of tree-based models is supported by recent studies. For instance, Liu et al. [20] utilized Random Forest, GBDT, and Adaboost to predict stock prices for innovative SMEs listed on China's SSE STAR market. Their model incorporated 34 determinants from historical trading data, stock price-related indices, and exchange rates, demonstrating impressive accuracy and robustness.

By leveraging these models, we aim to explore their performance and suitability for predicting SME stock prices in our study. In this section, we will introduce the Random Forest (RF), Gradient Boosting Decision Tree (GBDT), Adaptive Boosting (Adaboost), and Long-Short Term Memory (LSTM) models. The Echo State Network (ESN) will be discussed separately in the next section, as it stands out among the five candidates as our best solution.

3.2.1 Random Forest (RF)

Random Forest (RF) [21] is an ensemble learning method that constructs multiple decision trees during training and merges their results to improve accuracy and control overfitting. Each tree is built using a different subset of the training data, and the final prediction is made by averaging the predictions of all trees. In our one-day stock prediction and ten-day stock prediction comparative experiments (see Figures 3, 6, 7), the Random Forest (RF) model achieved good prediction accuracy with an RMSE of 13.27 for 1-day predictions and 27.10 for 10-day predictions. However, its prediction efficiency was relatively low and unacceptable, requiring 2210.79 seconds for 1-day and 220.25 seconds for 10-day predictions, with a relatively high level of CPU usage.

3.2.2 Gradient Boosting Decision Trees (GBDT)

Gradient Boosting Decision Trees (GBDT) [22] is an ensemble technique that builds trees sequentially, where each tree tries to correct the errors of the previous one. This method focuses on minimizing the loss function by adding weak learners in a stage-wise manner, leading to a highly accurate model. In our one-day stock prediction and ten-day stock prediction comparative experiments (see Figures 4, 6, 7), Gradient Boosting Decision Trees (GBDT) showed high accuracy with an RMSE of 13.18 for 1-day and 25.15 for 10-day predictions. It was more efficient than RF, with an unacceptable time consumption of 1948.43 seconds for 1-day and 177.42 seconds for 10-day predictions.

3.2.3 Adaptive Boosting Adaptive Boosting (Adaboost)(Adaboost) [1]

combines multiple weak learners to create a strong learner. Each weak learner is trained sequentially, with more focus on the incorrectly predicted instances from the previous learners. The final model is a weighted sum of these weak learners, providing a robust predictive performance. In our one-day stock prediction and ten-day stock prediction comparative experiments (see Figures 5, 6, 7), Adaptive Boosting (Adaboost) had moderate accuracy with an RMSE of 14.06 for 1-day and 27.71 for 10-day predictions. It excelled in efficiency, taking 668.25 seconds for 1-day and 67.72 seconds for 10-day predictions, with a relatively lower level of CPU usage.

3.2.4 Long Short-Term Memory Networks (LSTM)

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data. LSTM networks use memory cells to store information over time and gates to regulate the flow of information, making them effective for time series prediction tasks. In our one-day stock prediction and ten-day stock prediction comparative experiments (see Figures 6, 7), Long Short-Term Memory Networks (LSTM) had the lowest accuracy among the models, with an RMSE of 59.10 for 1-day and 301.93 for 10-day predictions. Despite this,

LSTM was highly efficient, requiring only 40.03 seconds for 1-day and 160.12 seconds for 10-day predictions, with low CPU usage at 2.2% and 2.9%, respectively.

3.3 Final Solution - Echo State Networks (ESN)

3.3.1 Overview of ESN

An Echo State Network (ESN) is mainly used for analyzing sequences of data, such as stock prices. It is a type of recurrent neural network (RNN) with a loosely connected hidden layer, also known as a reservoir. And only the connections from the reservoir to the output are trained, which makes it simpler and faster compared with other techniques. The model also fades the influence of the past input gradually for higher stability. Those help the model to capture various sequential patterns of the time sequence data.

In this project, a standard ESN algorithm implemented in Python was chosen as our method. We trained the model with the previous closing price of SCMI and predicted the future closing with the trained model. The model was also validated by splitting previous data into a train set and a test set.

3.3.2 Training and optimization

ESN model training process consists of 3 steps, which are the initial training with default parameters for validating the training process, the hyperparameter optimization for achieving the best performance, and prediction with various length for testing its performance on long/short term prediction.

1. Initial training with default parameter
   The closing price data was divided into the train set and test set at the ratio of 9:1. The train set (about 2500 points of data) was fed to the algorithm as the output layer, and an array of ones was fed as input. They are used to train the model with the default parameters provided by the source code to verify the training process. Then the trained model was used to predict the future closing price by feeding the previous price values. At this stage, we only care about whether the code will run and produce reasonable results. We will optimize and validate the model later.

2. Optimizing hyperparameters
   After confirming the training process working well, we started to optimize the model by adjusting the hyperparameters. Spectrum radius and noise were selected to tune the model. Other hyperparameters affects the performance of the model. Due to the limitation of time and availability, they are not studied through the optimization process.

Spectrum radium is one of the critical parameters for the ESN model to ensure the stability and dynamics of the network. By definition, the spectrum radius of a matrix is the largest absolute value of its eigenvalue. In an ESN, the spectral radius of its reservoir weight matrix ensures that the effect of input fades over time. In other words, the influence of past inputs on the current state diminishes as time goes on. If the spectral radium is less than 1, the past state will not have a strong influence on the current state. And that influence increases as we increase the spectral radius. In this project, we tested the spectral radius range from 0.5 to 1.5 as the step of 0.1.
Noise is another critical parameter in the ESN model. It introduces stochasticity to the network, which brings three benefits. They are increasing the robustness of the model, preventing overfitting, and improving the generalization. Details are not in the scope of this project. We just tested with the noise value from 0.0001 to 0.01 with about 10 intervals.
In the optimization process, various combinations of spectral radius and noises were tested using the same method in the initial training process. The RSME value of each combination was calculated shown in the below figure. It can be concluded that the best combination of spectral radius and noise is 1 and 0.0001.

3. Test predictability versus prediction length
The ESN model is able to predict multiple lengths of data points. There is a trade-off between the prediction length versus accuracy. From the users' perspective, a longer prediction length is always more useful, because it helps them better make longer-term decisions. However, as we increase the prediction data length, the error propagates and accumulates over time, which makes the prediction result unrealizable. Thus, in this project, we also studied the accuracy of prediction length. We set the length from 1 to 10, repeated the initial training process, and then calculated the RSME values of each length. They are plotted in the below figure. It is obvious that as the prediction length increases, we get higher RSME results. In other words, the model produces less accurate results. For comparing performance with other models, we choose length as 1.

3.3.3 Model validation

After we obtained the best spectral radius and noise pair and confirm the prediction length. We can validate the model by comparing the predicted values with the real numbers. Since the model only produces 1 data point, we move the train set 1 step forward (which will borrow values from the test set) to generate the next day's result and keep doing this till the end of the test set.
In our one-day stock prediction and ten-day stock prediction comparative experiments, Echo State Networks (ESN) demonstrated the best overall performance for 1-day predictions with an RMSE of 12.79 and excellent efficiency (see Figures 6, 7), requiring only 53.6 seconds. For 10-day predictions, its accuracy decreased with an RMSE of

101.29, but it remained the most efficient, taking only 5.4 seconds, and had the lowest CPU usage at 2.4% for 1-day and 0.5% for 10-day predictions.

4 Design Analysis & Testing

4.1 Engineering Analysis

4.1.1 Analyzing the Problem/Problem Space

Market Analysis: Problem Overview and Challenges
The primary issue at hand is the lack of specialized research and predictive tools tailored to small-cap stocks, predominantly represented by Small and Medium-sized Enterprises (SMEs). Small-cap stocks differ significantly from large-cap stocks due to their high volatility, low liquidity, and limited analyst coverage. These unique characteristics render traditional generalized models ineffective for predicting the stock prices of SMEs. This deficiency restricts investors' ability to make well-informed investment decisions in SMEs, which, despite their potential for high profitability, carry inherent risks due to their distinctive market behaviors. Therefore, there is a critical need for targeted, efficient, and accurate prediction models for small-cap stocks to support better investment decisions.
Primary Users: Our primary users are individual investors who frequently use our tool for short-term trading and are willing to take moderate risks to maximize returns.
Secondary Users: Secondary users include individuals who occasionally utilize the solution, such as for signal indicators related to merger and acquisition opportunities. These users perform detailed research on investment decisions after receiving initial indications from our tool.
Anti-Users: Institutional investors, who demand maximum accuracy and prefer long-term investments based on fundamental analysis, are not our target audience and are classified as anti-users.
Requirements Analysis
Based on our analysis and user feedback, the following user needs have been identified:

1. **Cost-Effectiveness:** Users prioritize affordable solutions for stock price prediction.

2. **Operational Efficiency:** The tool must be easy to use, even for non-technical users, and provide transparent predictions.

3. **Moderate Accuracy:** Users require reliable predictions that are sufficient to inform their trading decisions, even if not maximally accurate.

4. **Real-Time Updates:** Frequent or real-time updates and alerts are crucial for informed decision-making.

5. **Transparency and Accessibility:** The predictions and workings of the tool should be easily understandable by users without a technical background.

These needs guide the design and functionality of our predictive tool, ensuring it effectively addresses the requirements of our target user base. In the following section, we will revisit these requirements to ensure we have met them.

4.2 Modeling

4.2.1 Data Analysis & Exploration

We have conducted extensive research on existing methodologies for predicting stock prices using AI and Machine Learning techniques. Our findings indicate two major approaches: (1) using time series data such as open/close/high/low prices and trading volume to predict future closing prices, and (2) employing NLP techniques to analyze market sentiment. Due to time constraints and our expertise, we have chosen to focus on the first method.

Data Sources: To support this approach, we sourced historical stock price data (open, close, high, low, trading volume) from Nasdaq Data Link and Yahoo Finance. The data from these two sources are equivalent, allowing us to use them interchangeably. Research indicates that SME stocks are particularly sensitive to interest rates, consumer spending, and market sentiment [23]. Therefore, we plan to incorporate these economic indicators to potentially improve prediction accuracy.

- **Interest Rates:** Daily prime rate data from the Federal Reserve Economic Data.
- **Market Sentiment:** Market Volatility (VIX index) data from Yahoo Finance.

Stocks Selection: The definition of SMEs varies across different contexts and can be determined by criteria such as staff headcount or balance sheet total(Reference 2). The Russell 2000 Index [24], which includes 2000 SMEs, is the most recognized index for SMEs. However, only the largest 10 by capitalization are publicly available, and no reliable sources provide a complete list of all 2000 corporations. For our experiment, we are focusing on the top 10 stocks in the Russell 2000 index, with a primary emphasis on SMCI, the top stock in this index. SMCI belongs to the technology sector, similar to prominent companies like Apple and Microsoft. Due to the popularity and significant gains in tech companies, many studies use giants such as Apple, Microsoft, and Nvidia for stock analysis.

Data Characteristics: Our study utilizes time-series data, focusing on features such as open/close/high/low prices, trading volume, interest rates, consumer spending, and the VIX index. This data is essential for predicting stock prices, particularly for SMEs.
However, there are limitations and biases in our study. While the Russell 2000 index is a well-known representation of SMEs, the companies within it grow over time, and the

top 10 may be better managed compared to traditional SMEs. Unfortunately, we do not have access to the complete list of smaller companies in the Russell 2000 index, limiting our analysis to only the largest, publicly available stocks. This restriction means our model may not capture certain features or patterns present in smaller publicly traded businesses.

Regarding interest rates, we use the bank prime rate to represent this variable. However, this rate does not fluctuate with interest-rate-dependent products in the market, such as government bonds, potentially omitting some aspects of the true interest rate. Additionally, its lack of fluctuation may make it less suitable for integration with time-series data.

Data Exploration: Given that many models in existing research primarily use large-cap stocks like Apple (AAPL) or Microsoft, we aim to investigate the differences between large-cap stocks and the SME stocks we are interested in. To do this, we conducted data exploration on both AAPL and SMCI (representing large-cap and SME stocks, respectively), focusing on their characteristics within the technology sector.

Correlation Matrix & Lagged Correlation Matrix: We analyzed the correlation matrices for AAPL and SMCI (10)(11)(12)(13). The results show that SMCI's close price has a stronger linear relationship with trading volume compared to AAPL. This gap widens when trading volume is lagged by 60 days. Additionally, SMCI appears to be more sensitive to interest rate changes, although this is shown only in terms of linear relationships.

Scatter Plot with LOWESS Fit: To examine non-linear relationships, particularly with market volatility (VIX), we used scatter plots (14)(15). The analysis indicates that SMCI's close price tends to be low when market volatility is high, whereas with low market volatility, the close price varies significantly.

ACF & PACF: The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots (16)(17)(18)(19) reveal that AAPL's time series exhibits a strong trend or seasonal component, with significant autocorrelation at longer lags and notable negative correlation, suggesting long-term trends or cycles. In contrast, SMCI's time series shows a rapid decay of autocorrelation to near zero at shorter lags, indicating a lack of long-term trends or seasonality and maintaining stationarity at longer lags. These ACF plots provide an initial understanding of the structure of AAPL and SMCI time series, guiding subsequent modeling steps.

Moving Averages: Analyzing the moving averages (20)(21), we observe that SMCI is less stable and exhibits steeper patterns, which may complicate the learning process for models.

Decomposition of Close/Last Price: Trend, Seasonality, and Residuals We decomposed the close prices into trend, seasonality, and residual components (22)(23). Each component was tested on an LSTM model, revealing that the model performs well on seasonal and residual components, with errors mainly arising from the trend component.

This detailed data exploration provides critical insights into the differences between

large-cap and SME stocks, informing the development and adjustment of our predictive models.

4.2.2 Data Preprocessing

In our data preprocessing phase, we undertook several steps to ensure the data's quality and suitability for model training. This included handling missing values, normalization, and feature selection.

Handling Missing Values For interest rates, while stock prices are available for each trading day, some trading days lack corresponding prime rate data. To address this, we interpolated these missing values by averaging the available data from the preceding day (t-1) and the following day (t+1).

Normalization We employed Min-Max normalization, which is particularly suitable for time-series data where the range of values is important for interpreting trends and patterns. This method scales the data to a fixed range, typically [0, 1], which is beneficial for neural networks as it ensures input data within a consistent range, thereby improving convergence rates during training.

Although Min-Max normalization is sensitive to outliers, we chose it over other methods after testing. We also evaluated Z-Score normalization and Robust Scaler to mitigate sensitivity to outliers. However, these alternatives reduced the accuracy of our models, leading us to favor Min-Max normalization despite its susceptibility to outliers.

4.2.3 Model Selection

Algorithm Choice We selected three types of models for our study: Long Short-Term Memory (LSTM) networks, Echo State Networks (ESN), and tree-based models. Each model was chosen based on its unique strengths: ESN is adept at handling volatile data, LSTM networks are widely used in stock prediction, and tree-based models offer interpretability and ease of understanding.

Model Comparison To evaluate the performance of these models, we conducted comparisons over two investment horizons: 1-day and 10-day periods. We initially focused on the 1-day horizon because the models were originally designed to predict stock prices on a daily basis. This aligns with the primary purpose of these models in stock prediction.

In the financial context, short-term investment horizons are typically considered to be 30 to 90 days. While no model we tested was able to accurately predict stock prices over such extended periods, we included a 10-day horizon in our evaluation. This extended period covers approximately two weeks of business days and allowed us to assess whether the models could perform adequately over a slightly longer term. However, the results for the 10-day horizon were less satisfactory (Table), highlighting the increased challenge of forecasting for SMEs, which generally exhibit greater volatility.

Tree-based Models For the three tree-based models, to address the need for predicting future stock prices, we created lag features using time series forecasting techniques. Then, we experimented with different window sizes to optimize model performance (see Figure27). By testing various sliding and prediction windows, we identified configurations that balanced the amount of historical data with the need for timely predictions, ensuring that the model had sufficient context to make accurate forecasts.

Additionally, to address the challenge that training times were excessive due to the large number of features, we employed advanced Python data processing techniques to reshape and optimize the data processing, which significantly reduced training times from approximately one hour to 30 minutes per experiment. This allowed for more rapid iteration and refinement of the model parameters, ensuring a balance between accuracy and efficiency.

LSTM To adapt our LSTM model for SME stocks, we expanded the feature set based on extensive research and data exploration. Research indicates that SME stock prices are highly sensitive to interest rates and market sentiment. Consequently, we included the U.S. daily prime rate and the VIX closing price in our dataset to separately measure these factors. Additionally, we incorporated standard stock price information, including opening price, closing price, daily high and low prices, and trading volume.

Through data exploration, we found that the opening price, closing price, daily high, and low prices exhibited very strong linear relationships with each other, as demonstrated by our heat maps (cite heat map). Including these highly correlated features introduced extra noise into the model without providing additional useful information. As a result, eliminating features with strong pairwise relationships improved model performance.

In addition to focusing on efficiency and accuracy, we emphasized transparency to ensure that users can trust and understand the model's predictions.

ESN ESN model was also evaluated for SME stocks. Due to the limitation of the ESN training code we used, we only applied closing price as the input feature. We fed the stock price data of SCMI to the training code to generate model. And we further optimized the model by tuning the critical parameters of ESN with SCMI to achieve the best prediction accuracy. The optimized model reveals high prediction accuracy and high efficiency on SME stock price prediction.

Model Architecture We performed predictions over 1-day and 10-day horizons. For training and validation, we used a dataset spanning 10 years, dividing it into 90% for training and 10% for validation. The model architecture includes RNN layers with a time window of 60 days. This setup enables the model to learn relationships between each 60-day period and the subsequent 1 day for 1-day predictions, and between each 60-day period and the subsequent 10 days for 10-day predictions.

The performance of the model, as discussed in the testing section, was quantified and iteratively refined to improve results.

4.3 Testing

Our testing approach involved a comprehensive evaluation using Root Mean Squared Error (RMSE) to measure accuracy, runtime to assess efficiency, and GPU consumption to evaluate usability on personal computers. We backtested the models on historical data and iteratively refined them based on performance feedback.

How the results related to Requirements

- Moderate Accuracy: Accuracy was quantified using RMSE. The model's performance was benchmarked against acceptable accuracy levels for stock price prediction.
- Cost-Effectiveness: We aimed to provide an affordable solution for stock price prediction. The current model is resource-efficient, running on a personal computer with a runtime of less than 1 minute.
- Real-Time Updates: For the demo, real-time stock prices were downloaded as CSV files. While using APIs for real-time data is feasible, limitations and costs associated with free APIs restricted this option.
- Transparency and Accessibility: We ensured that users without a technical background could understand the model's functionality. An anonymous questionnaire will be used to rate user trust in the model, on a scale from 1 to 5, with 5 indicating high trust.
- Operational Efficiency: The tool's ease of use for non-technical users was measured by the time taken from initiating the application to viewing the results. This time was further divided into model runtime and operational time, with the latter being the time taken excluding model execution.

Summary

We successfully met the key project objectives of moderate accuracy, cost-effectiveness, and real-time updates, and ensured model transparency and operational efficiency. However, due to time constraints, user opinions on transparency and operational efficiency were not tested, highlighting the need for a stronger user interface.

Model Performance

- ESN (Echo State Network): ESN exhibited the best accuracy and efficiency for volatile data but struggled with the incorporation of additional features.
- LSTM (Long Short-Term Memory): While LSTM was less accurate and efficient compared to ESN, it showed improvement with feature expansion tailored to SME data.
- Tree-Based Models: These models delivered good accuracy but had longer runtimes and lower efficiency compared to LSTM and ESN.

Overall, the results underscore the strengths and limitations of each model in the context of SME stock prediction.

5 Limitations & Challenges

The Recurrent Neural Network (RNN) models used in this study were initially designed for 1-step predictions. Although we attempted to extend these models to multiple-step forecasts, the results were not satisfactory. This limitation highlights the challenges associated with adapting RNN models for longer prediction horizons.
Furthermore, stock market dynamics are influenced by a multitude of factors. In this study, we focused on a limited set of economic indicators. There is potential for improvement by incorporating additional factors, as discussed in the Future Work section.
The metrics used to measure efficiency, such as runtime and CPU usage, may not be highly accurate due to variations across different computers and operating conditions. More advanced tools would be required for precise measurement. Our current metrics provide a general indication that the model performs efficiently on personal computers, which aligns with the needs of our target users.

6 Conclusion & Future Work

Several potential enhancements were considered but not yet implemented:
Firstly, we plan to conduct stress testing under various market conditions and across different sectors of SME stocks. This additional testing will help validate our conclusions and assess the model's robustness in diverse scenarios.
Secondly, our research revealed an intriguing fact: stock movements can be statistically decomposed into three components—Trend, Seasonality, and Residual. We applied LSTM models to each of these decomposed datasets and found that the model performed well on Seasonal (25) and Residual (26) components, with errors primarily arising from the Trend component (24). The Trend is influenced by various factors such as company revenue, financial performance, economic conditions, policies, and market sentiment.
This insight suggests a promising approach for future work: integrating AI models to handle Seasonal and Residual components while incorporating human input or interaction to address the Trend component. Combining these approaches could lead to more optimized and accurate predictions.

7 Declaration

Generative AI tools, including ChatGPT and Grammarly, are only used for the purpose of correcting grammatical errors.

References

[1] K. W. Walker and Z. Jiang, "Application of adaptive boosting (adaboost) in demand-driven acquisition (dda) prediction: A machine-learning approach," **The Journal of Academic Librarianship**, vol. 45, no. 3, pp. 203–212, 2019. pages 2, 7

[2] F. Dakalbab, M. A. Talib, Q. Nassir, and T. Ishak, "Artificial intelligence techniques in financial trading: A systematic literature review," **Journal of King Saud University-Computer and Information Sciences**, p. 102015, 2024. pages 3

[3] S. Naseem, M. Mohsin, W. Hui, G. Liyan, and K. Penglai, "The investor psychology and stock market behavior during the initial era of covid-19: a study of china, japan, and the united states," **Frontiers in Psychology**, vol. 12, p. 626934, 2021. pages 3

[4] A. M. Abdulsaleh and A. C. Worthington, "Small and medium-sized enterprises financing: A review of literature," **International Journal of Business and Management**, vol. 8, no. 14, p. 36, 2013. pages 3

[5] D.-Y. Lin, S. N. Rayavarapu, K. Tadjeddine, and R. Yeoh, "Beyond financials: Helping small and medium-sized enterprises thrive," https://www.mckinsey.com/industries/public-sector/our-insights/beyond-financials-helping-small-and-medium-size-enterprises-thrive, January 26 2022. pages 3

[6] C. D. Johnson, B. C. Bauer, and F. Niederman, "The automation of management and business science," **Academy of Management Perspectives**, vol. 35, no. 2, pp. 292–309, 2021. pages 3

[7] M. M. Billah, A. Sultana, F. Bhuiyan, and M. G. Kaosar, "Stock price prediction: comparison of different moving average techniques using deep learning model," **Neural Computing and Applications**, pp. 1–11, 2024. pages 3, 6

[8] J. Wang, J. He, C. Feng, L. Feng, and Y. Li, "Stock index prediction and uncertainty analysis using multi-scale nonlinear ensemble paradigm of optimal feature extraction, two-stage deep learning and gaussian process regression," **Applied Soft Computing**, vol. 113, p. 107898, 2021. pages 3

[9] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," **European journal of operational research**, vol. 270, no. 2, pp. 654–669, 2018. pages 3

[10] J. Jiang, E. Chen, X. Yan, and Y. Feng, "Applications of artificial intelligence in stock market prediction: A survey," **Journal of Big Data**, vol. 7, no. 1, pp. 1–30,

2020. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6 pages 3

[11] MorningStar, "Where we see opportunities as june stocks recover losses," 2023, accessed: 2024-06-07. [Online]. Available: https://www.morningstar.com/stocks/where-we-see-opportunities-june-stocks-recover-losses pages 3

[12] MSCI, "Small caps have been a big opportunity," 2021, accessed: 2024-06-07. [Online]. Available: https://www.msci.com/www/blog-posts/small-caps-have-been-a-big/03951176075 pages 3

[13] T. M. Fool, "When to buy small-cap stocks," 2024, accessed: 2024-06-07. [Online]. Available: https://www.fool.com/investing/stock-market/types-of-stocks/small-cap-stocks/when-to-buy/ pages 3, 4

[14] MorningStar, "2024 outlook: Stock market economy," 2024, accessed: 2024-06-07. [Online]. Available: https://www.morningstar.com/markets/2024-outlook-stock-market-economy pages 4

[15] S. Schlagkamp and J. Renker, "Acceptance of waiting times in high performance computing," in **International Conference on Human-Computer Interaction**. Springer, 2015, pp. 709–714. pages 4

[16] N. Cao, S. Ji, D. K. Chiu, and M. Gong, "A deceptive reviews detection model: Separated training of multi-feature learning and classification," **Expert Systems with Applications**, vol. 187, p. 115977, 2022. pages 5

[17] N. J. Gogtay and U. M. Thatte, "Principles of correlation analysis," **Journal of the Association of Physicians of India**, vol. 65, no. 3, pp. 78–81, 2017. pages 5

[18] B. F. Darst, K. C. Malecki, and C. D. Engelman, "Using recursive feature elimination in random forest to account for correlated variables in high dimensional data," **BMC genetics**, vol. 19, pp. 1–6, 2018. pages 6

[19] D. Vidaurre, C. Bielza, and P. Larranaga, "A survey of l1 regression," **International Statistical Review**, vol. 81, no. 3, pp. 361–387, 2013. pages 6

[20] W. Liu, Y. Suzuki, and S. Du, "Forecasting the stock price of listed innovative smes using machine learning methods based on bayesian optimization: Evidence from china," **Computational Economics**, pp. 1–34, 2023. pages 6

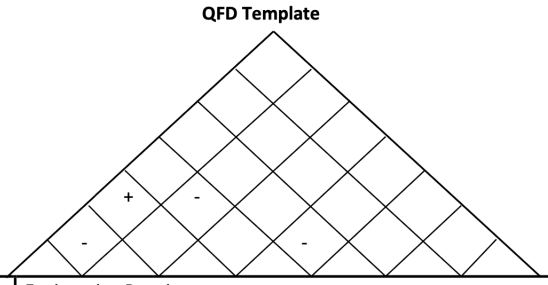
[21] S. J. Rigatti, "Random forest," **Journal of Insurance Medicine**, vol. 47, no. 1, pp. 31–39, 2017. pages 7

[22] J. Feng, Y. Yu, and Z.-H. Zhou, "Multi-layered gradient boosting decision trees," **Advances in neural information processing systems**, vol. 31, 2018. pages 7

[23] J. Bowman, “When is the best time to invest in small-cap stocks?” 2024, accessed: 2024-07-22. [Online]. Available: https://www.fool.com/investing/stock-market/types-of-stocks/small-cap-stocks/when-to-buy/ pages 11

[24] London Stock Exchange Group, “Ftse russell indices - russell us,” 2024, accessed: 2024-07-22. [Online]. Available: https://www.lseg.com/en/ftse-russell/indices/russell-us pages 11

Appendix



		Engineering Requirements							Competitive Products		
User Needs/Functions	Importance (What)	Should be easy to use	Should not be customizable	Should give prediction result rapidly	Should consume low computation resources	Should output accurate price prediction			Existing AI for stock price prediction	Financial investment analyst	
Know the future trend of the stock price	15			1		9			4	3	
Making revenues from stock markets	35			3	1	9			2	3	
Support personal investors to make decision	20	1		3		9			3	5	
Required few technical knowledge	10	9	6						4	5	
Affordable	20				9				2	1	
Unit		likert	likert	second	CDN	%					
Importance (HOW)		110	60	180	215	630	1195				
Importance (%)		9	5	15	18	53					
Minimum Threshold		1	1	1	1	60					
Desired Target		8	6	10	10	90					
Maximum Threshold		10	10	600	50	99					

Figure 1: QFD chart

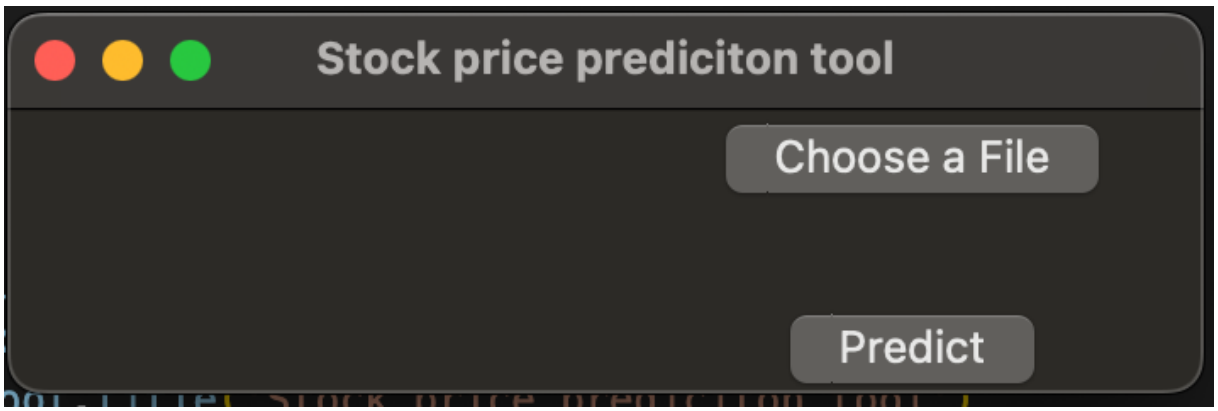


Figure 2: User Interface Protocol

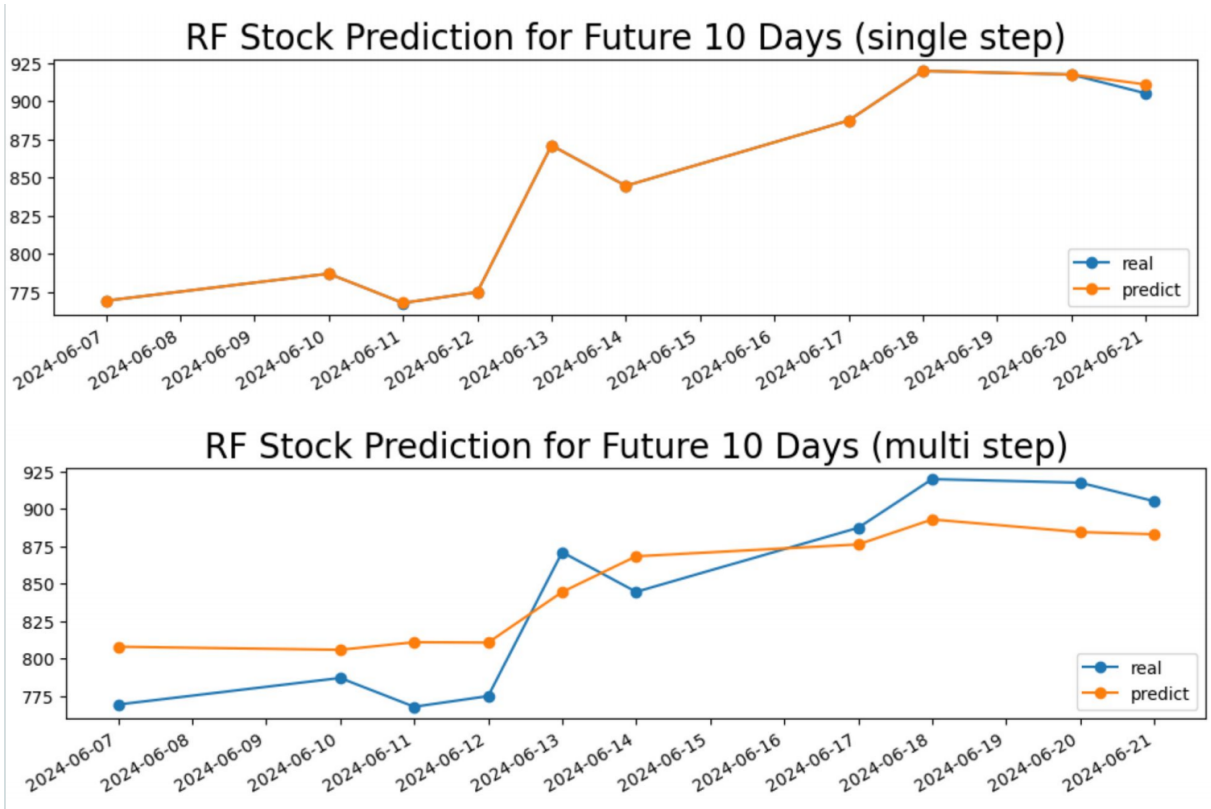


Figure 3: RF Stock Prediction Results

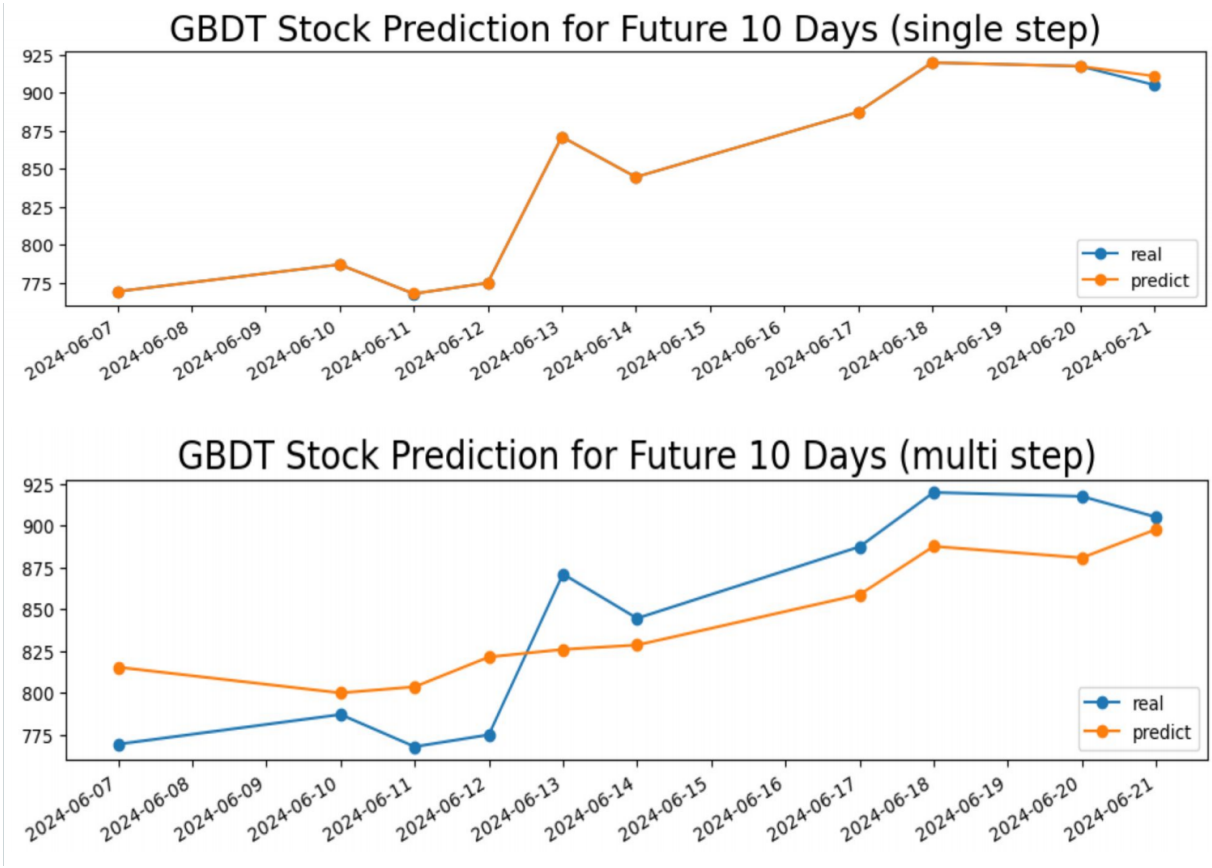


Figure 4: GBDT Stock Prediction Results

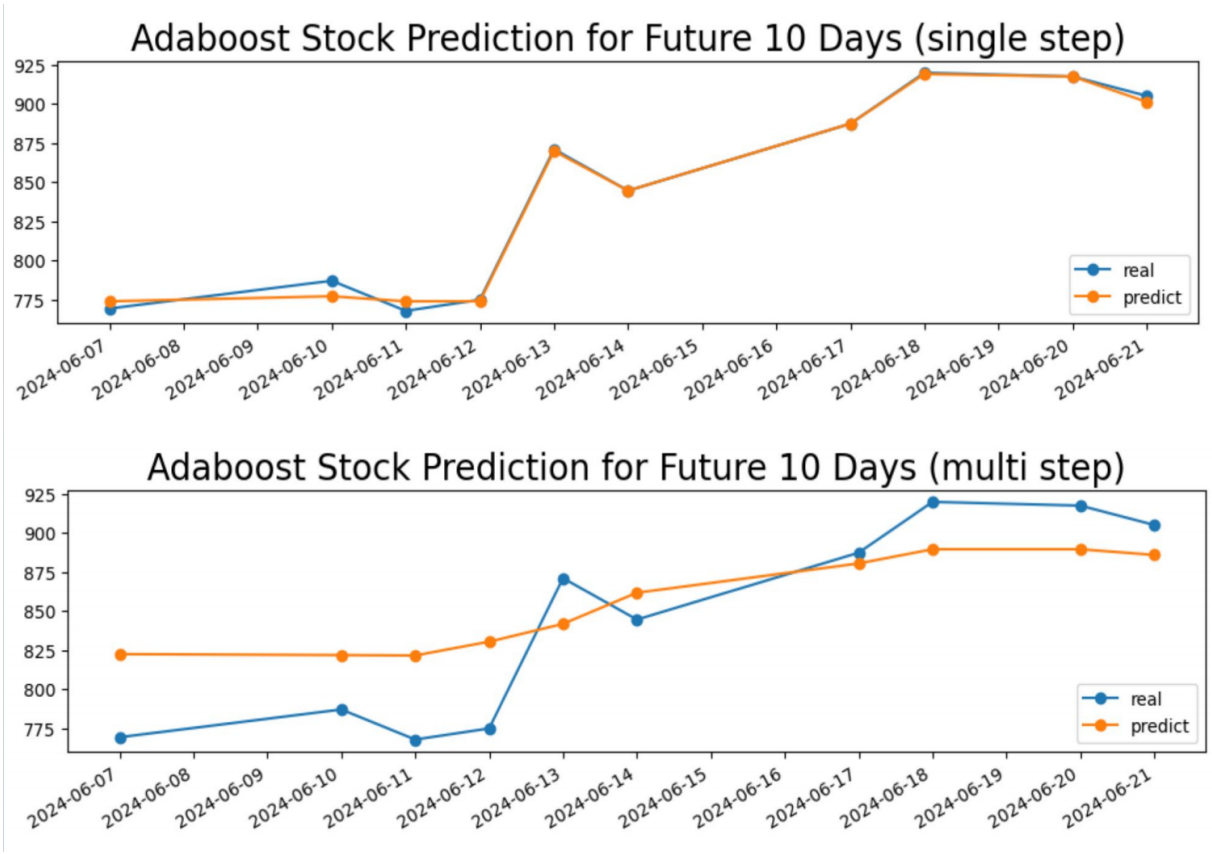


Figure 5: Adaboost Stock Prediction Results

Designed Solution - Conclusion for 1-Day Prediction

Model	Prediction Accuracy (RMSE)	Prediction Efficiency (Time Consumption)	Computation Resources (CPU Usage)
Random Forest (RF)	13.27	2210.79 seconds	6.4%
Gradient Boosting Decision Trees (GBDT)	13.18	1948.43 seconds	5.7%
Adaptive Boosting (Adaboost)	14.06	668.25 seconds	4.6%
Long Short-Term Memory Networks (LSTM)	59.10	40.03 seconds	2.2%
Echo State Networks (ESN)	12.79	53.6 seconds	2.4%

Figure 6: Conclusion for 1 Day Prediction

Designed Solution - Conclusion for 10-Day Prediction

Model	Prediction Accuracy (RMSE)	Prediction Efficiency (Time Consumption)	Computation Resources (CPU Usage)
Random Forest (RF)	27.10	220.25 seconds	3.1%
Gradient Boosting Decision Trees (GBDT)	25.15	177.42 seconds	2.5%
Adaptive Boosting (Adaboost)	27.71	67.72 seconds	2.3%
Long Short-Term Memory Networks (LSTM)	301.93	160.12 seconds	2.9%
Echo State Networks (ESN)	101.29	5.4 seconds	0.5%

Figure 7: Conclusion for 10 Day Prediction

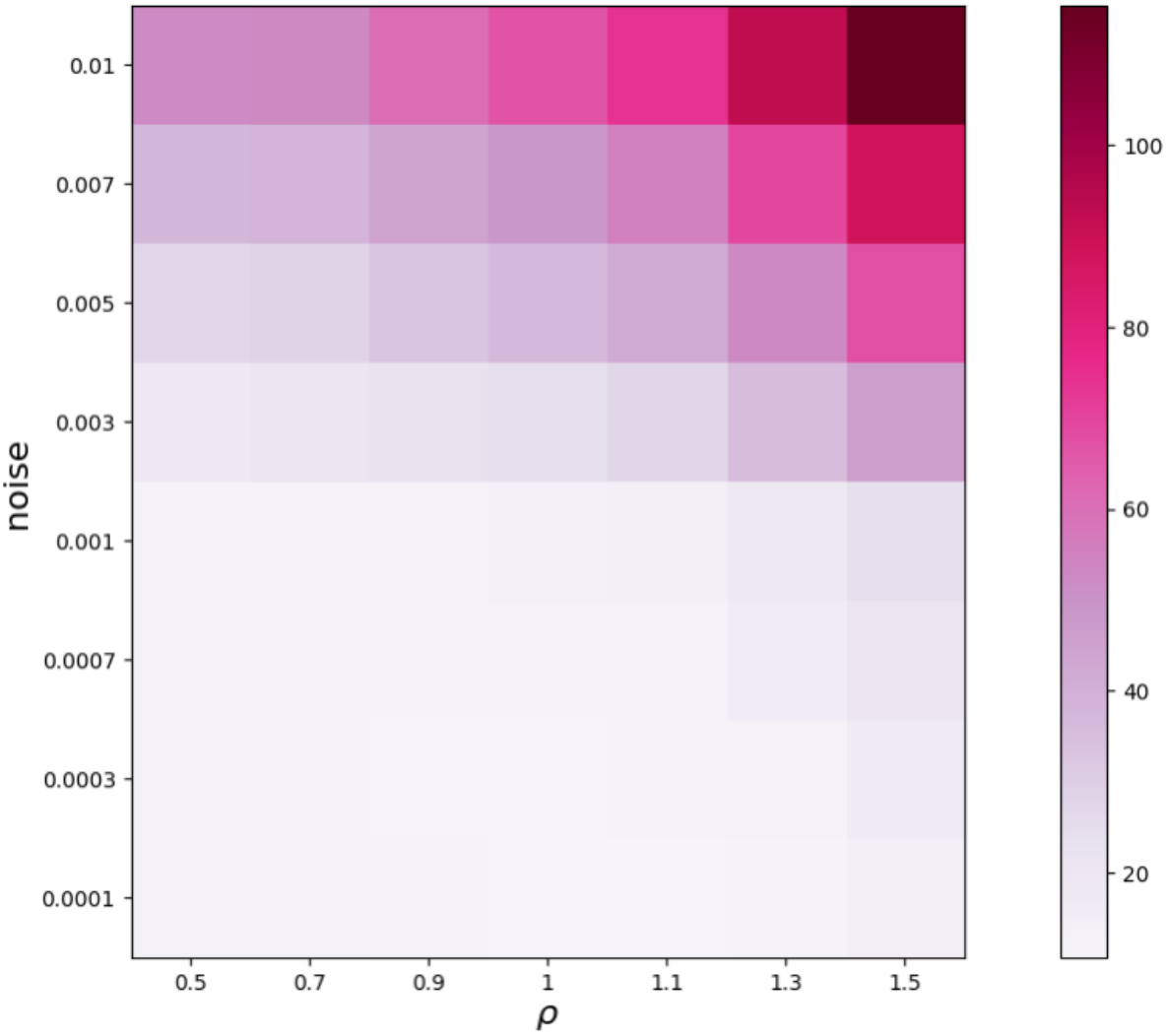


Figure 8: ESN Parameters Optimization - Noise

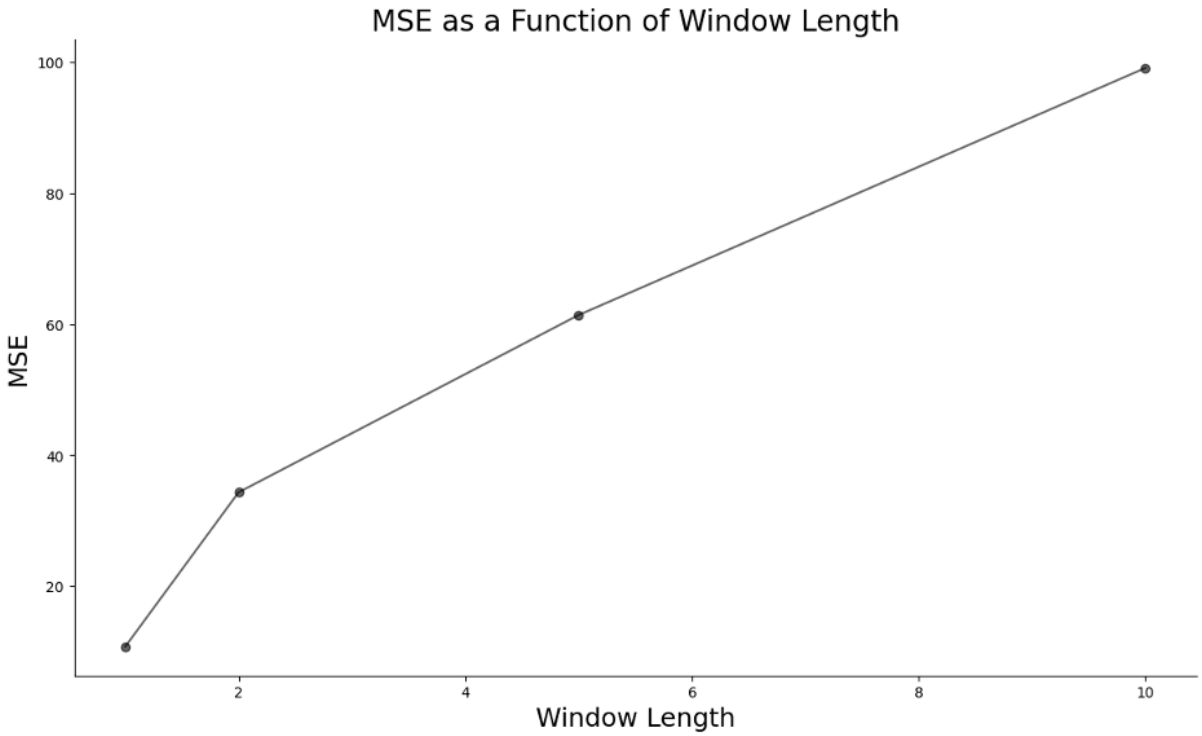


Figure 9: ESN Parameters Optimization - Window Size

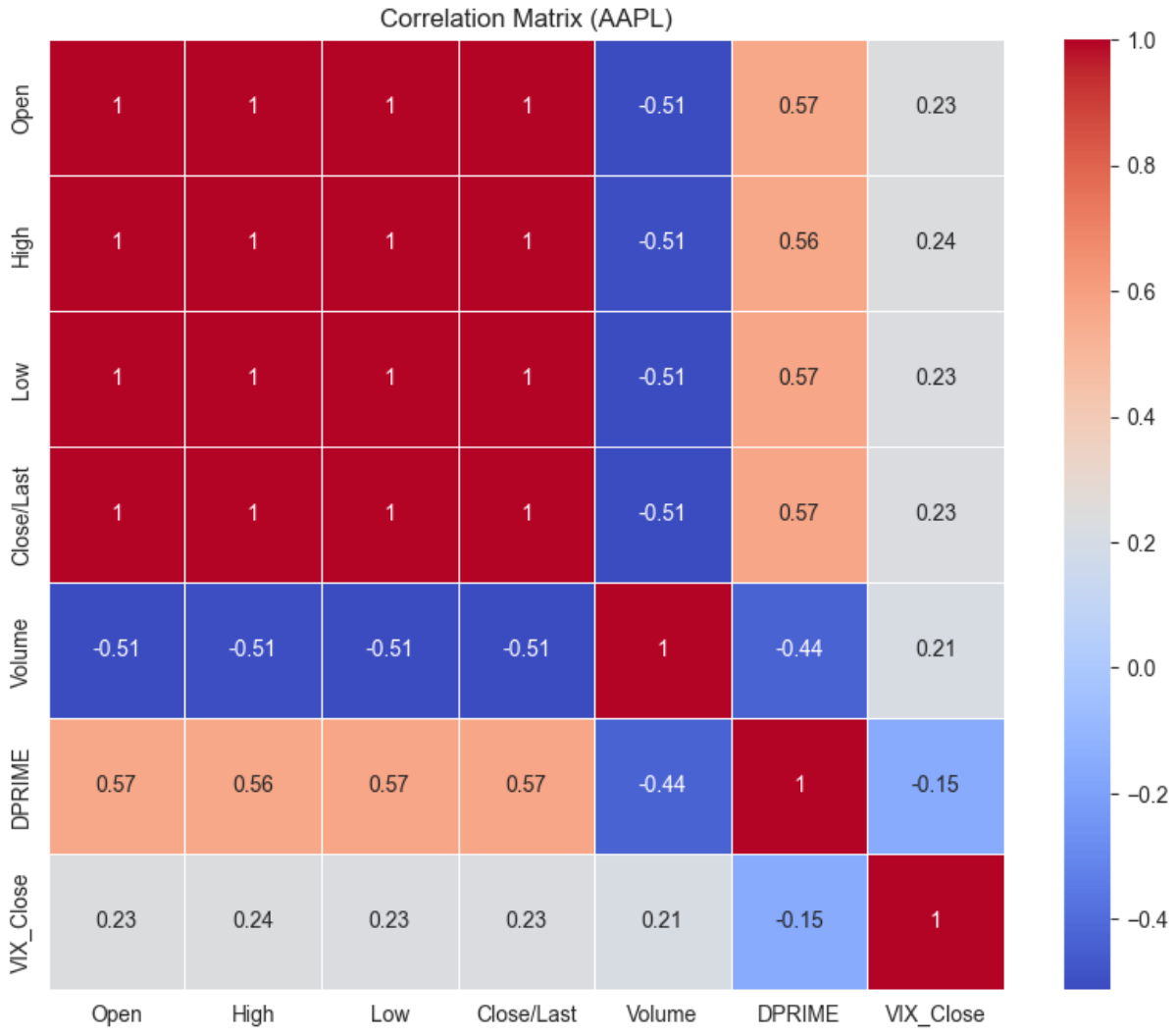


Figure 10: Correlation Matrix for Apple Stock (AAPL)

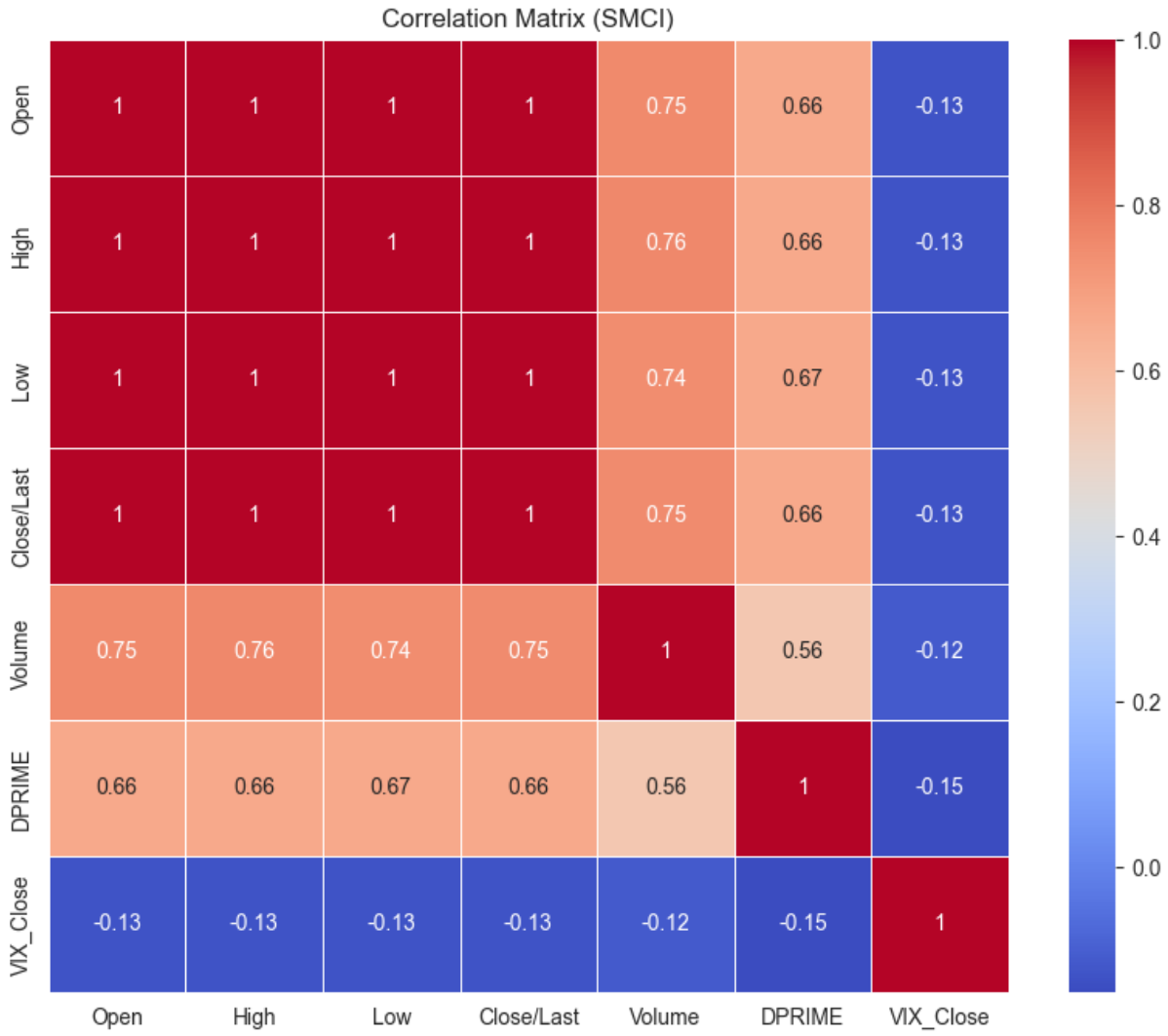


Figure 11: Correlation Matrix for Super Micro Computer Inc Stock (SMCI)

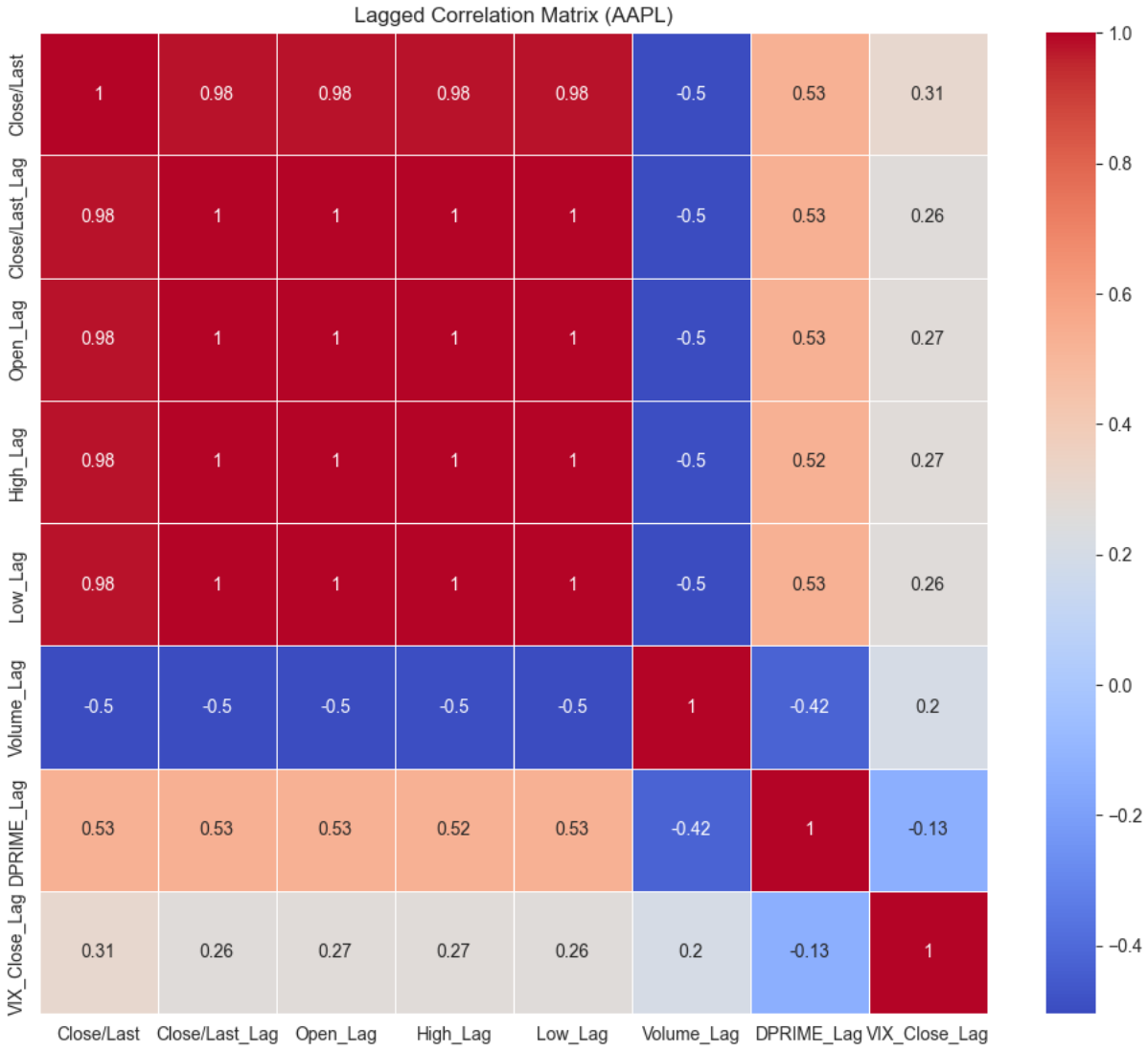


Figure 12: Lagged Correlation Matrix for Apple Stock

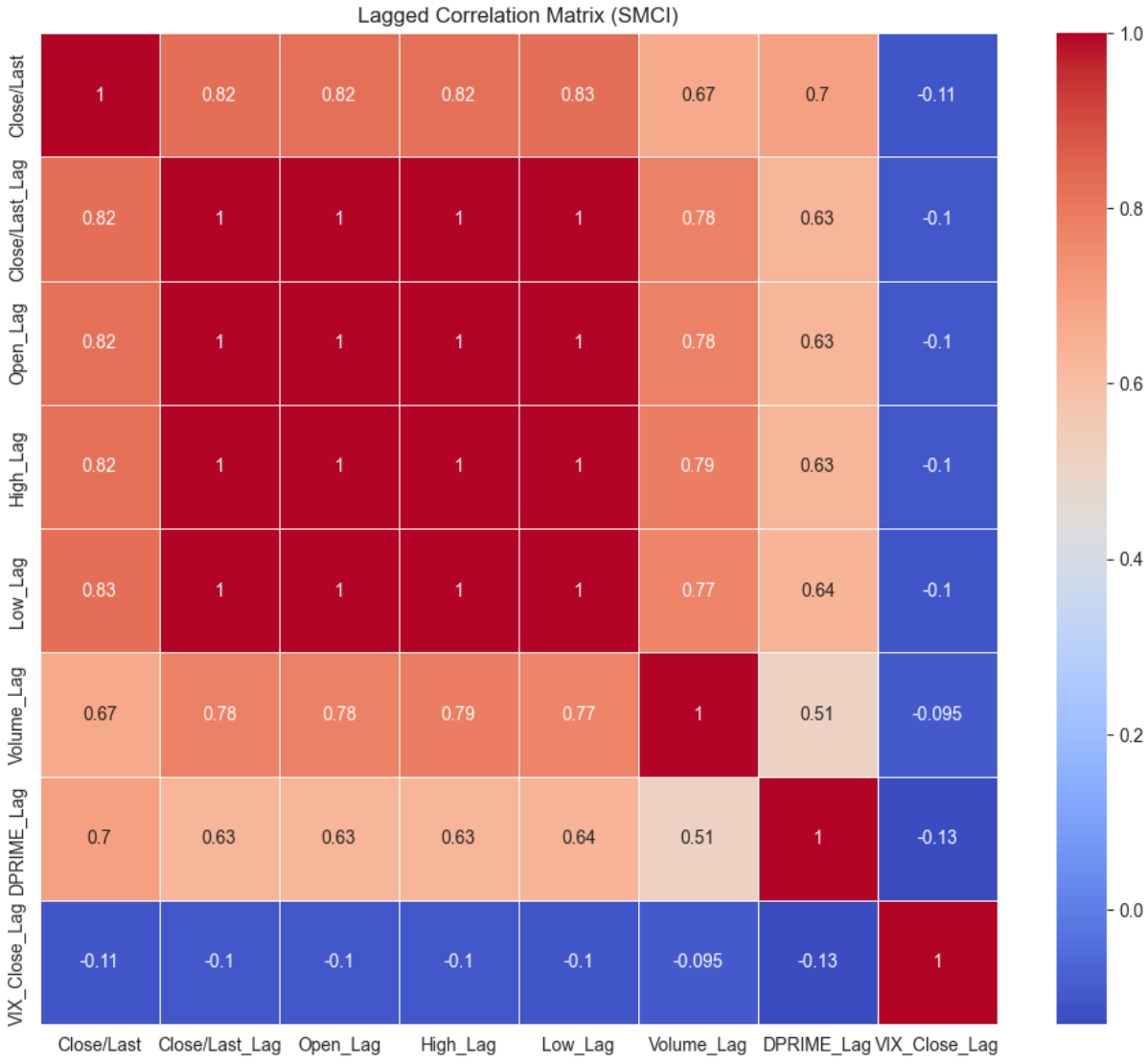


Figure 13: Lagged Correlation Matrix for Super Micro Computer Inc Stock

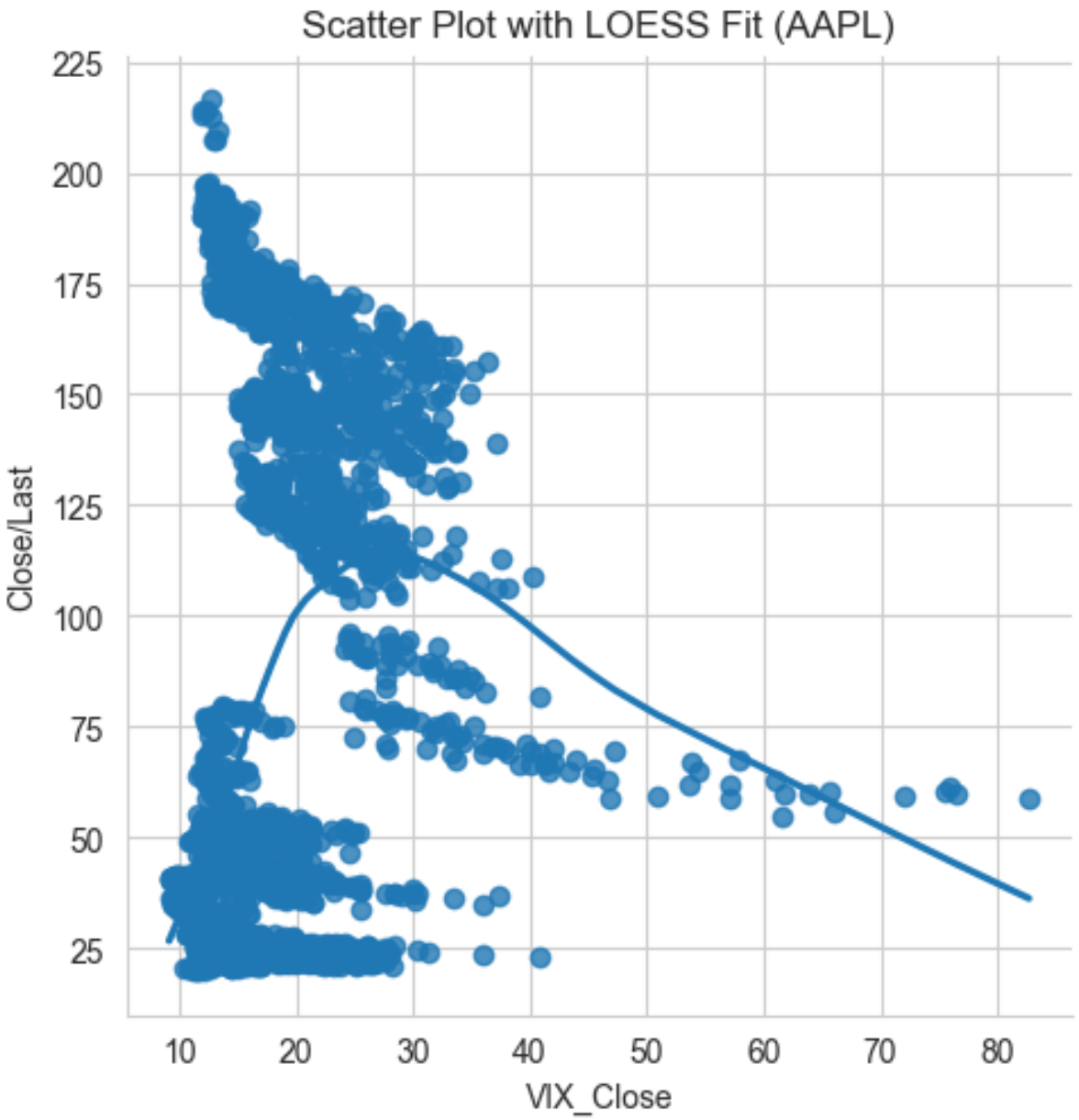


Figure 14: Scatter Plot with LOESS Fit (AAPL)

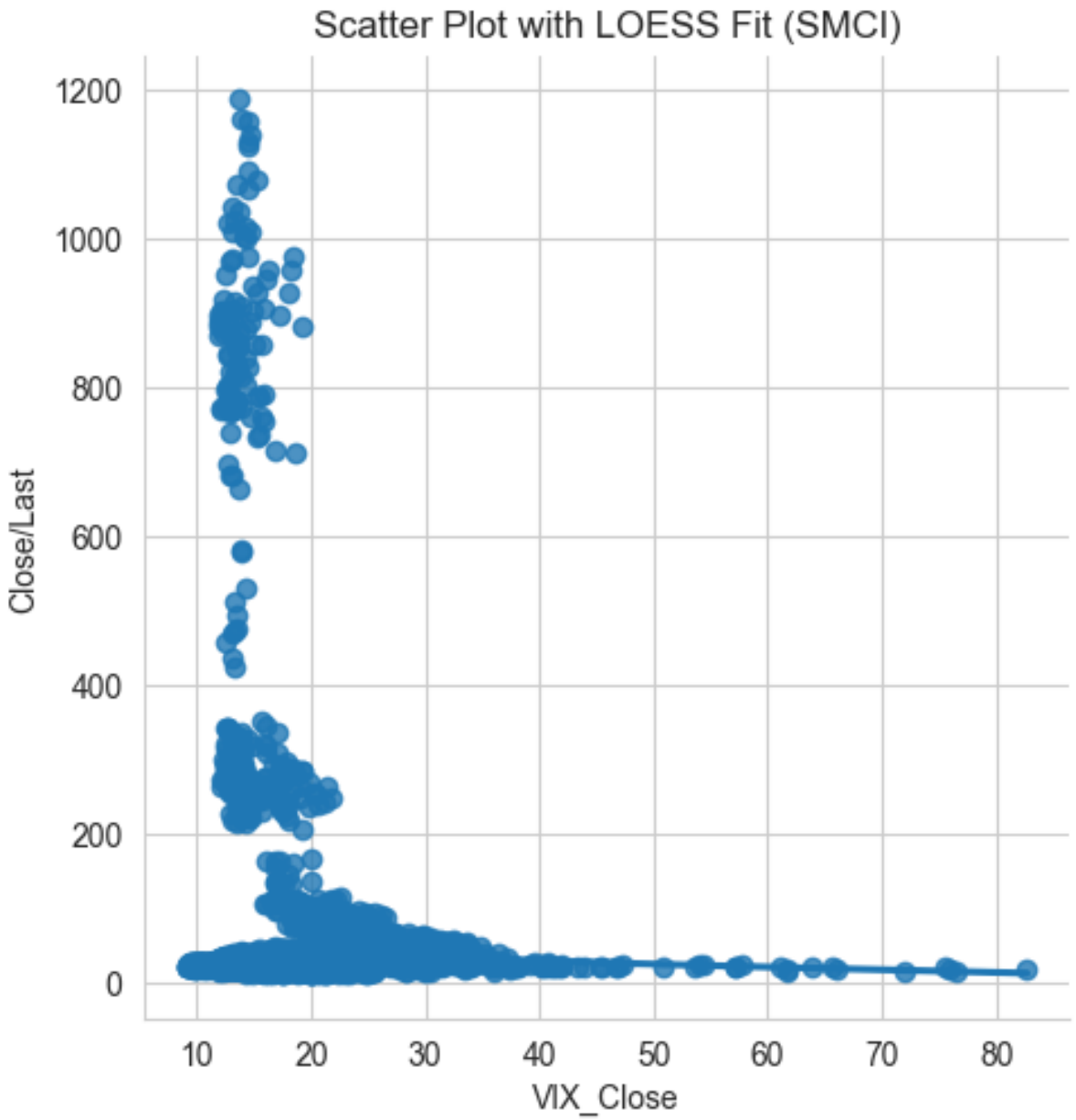


Figure 15: Scatter Plot with LOESS Fit (SMCI)

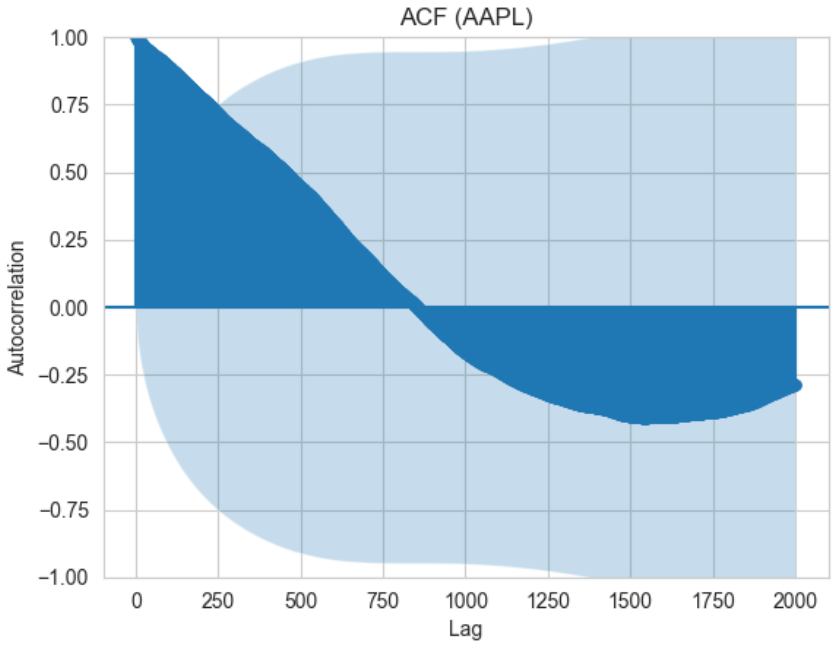


Figure 16: ACF (AAPL)

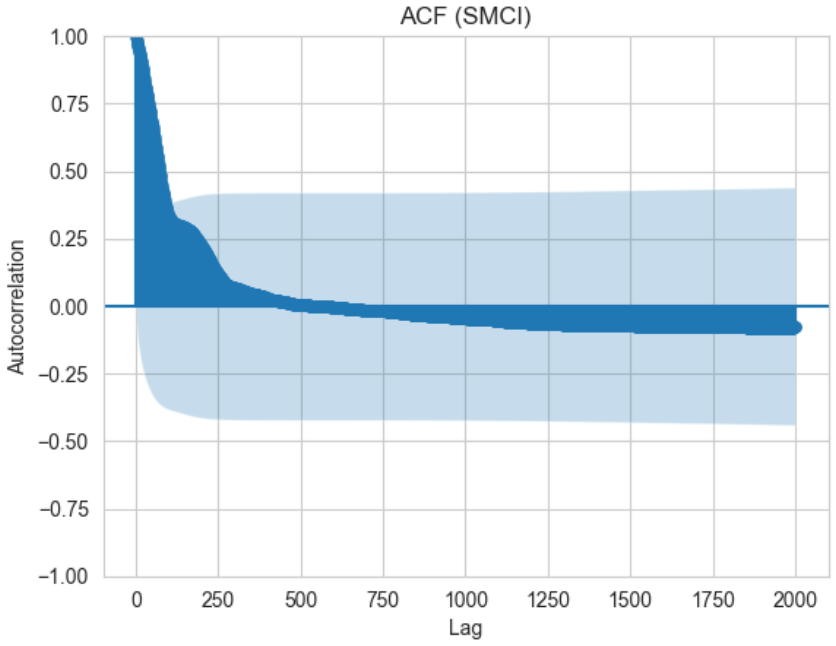


Figure 17: ACF (SMCI)

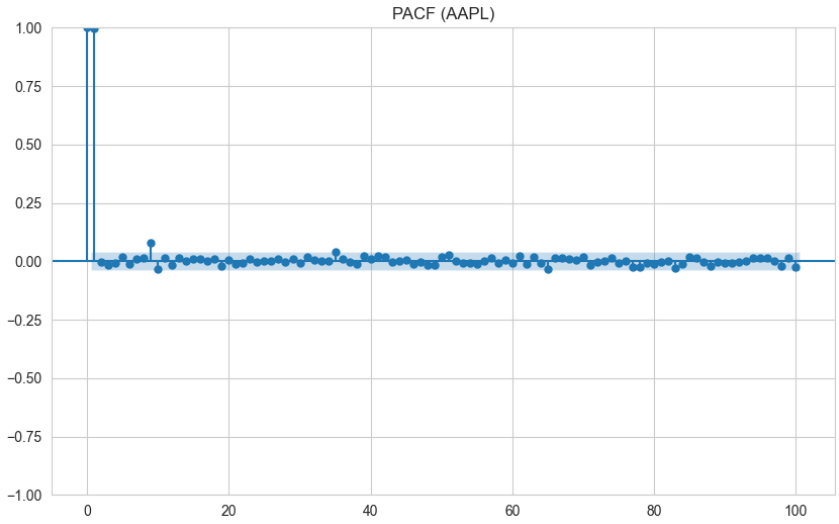


Figure 18: PACF (AAPL)

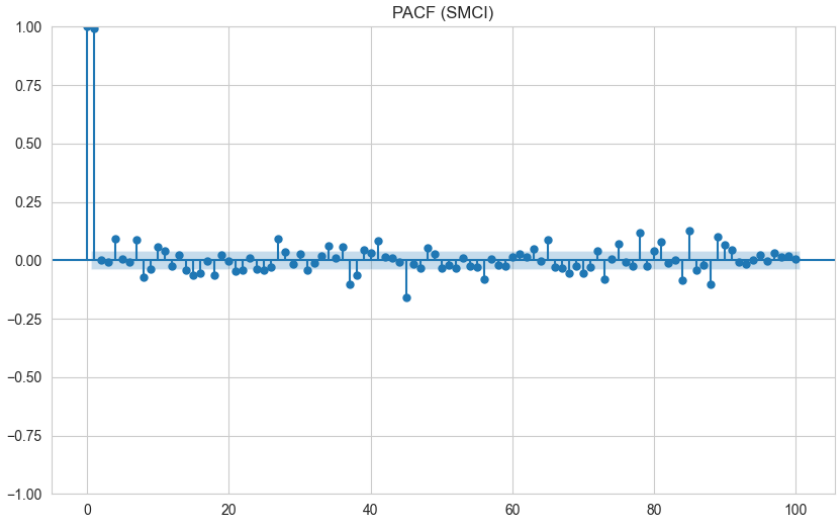


Figure 19: PACF (SMCI)

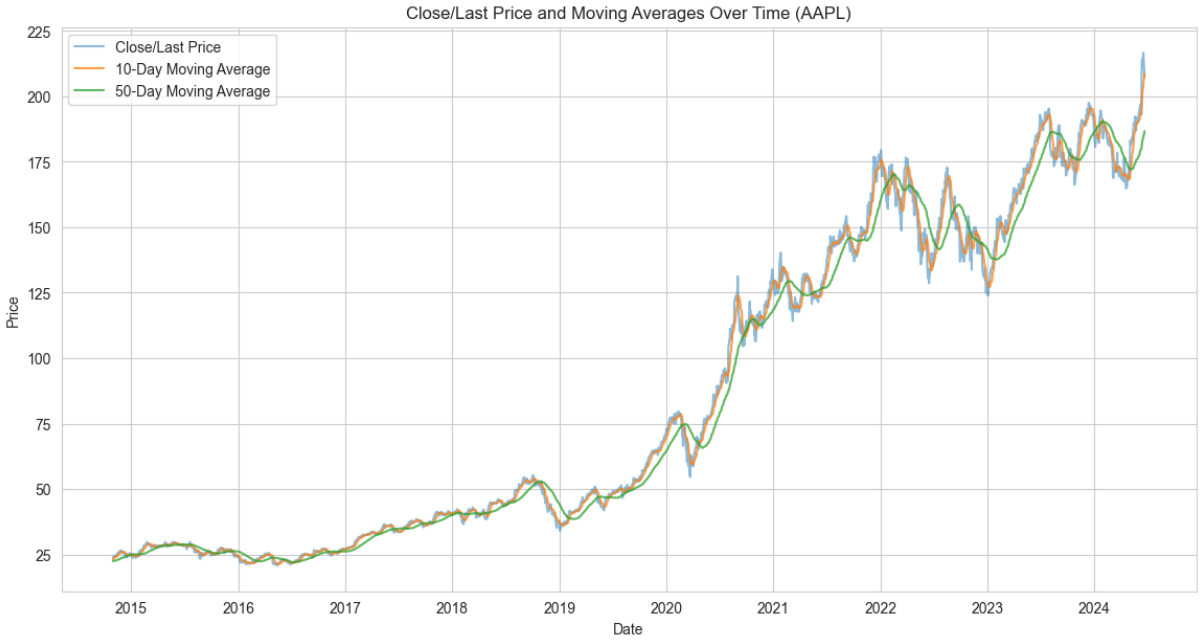


Figure 20: Close/Last Price and Moving Averages Over Time (AAPL)

Figure 21: Close/Last Price and Moving Averages Over Time (SMCI)

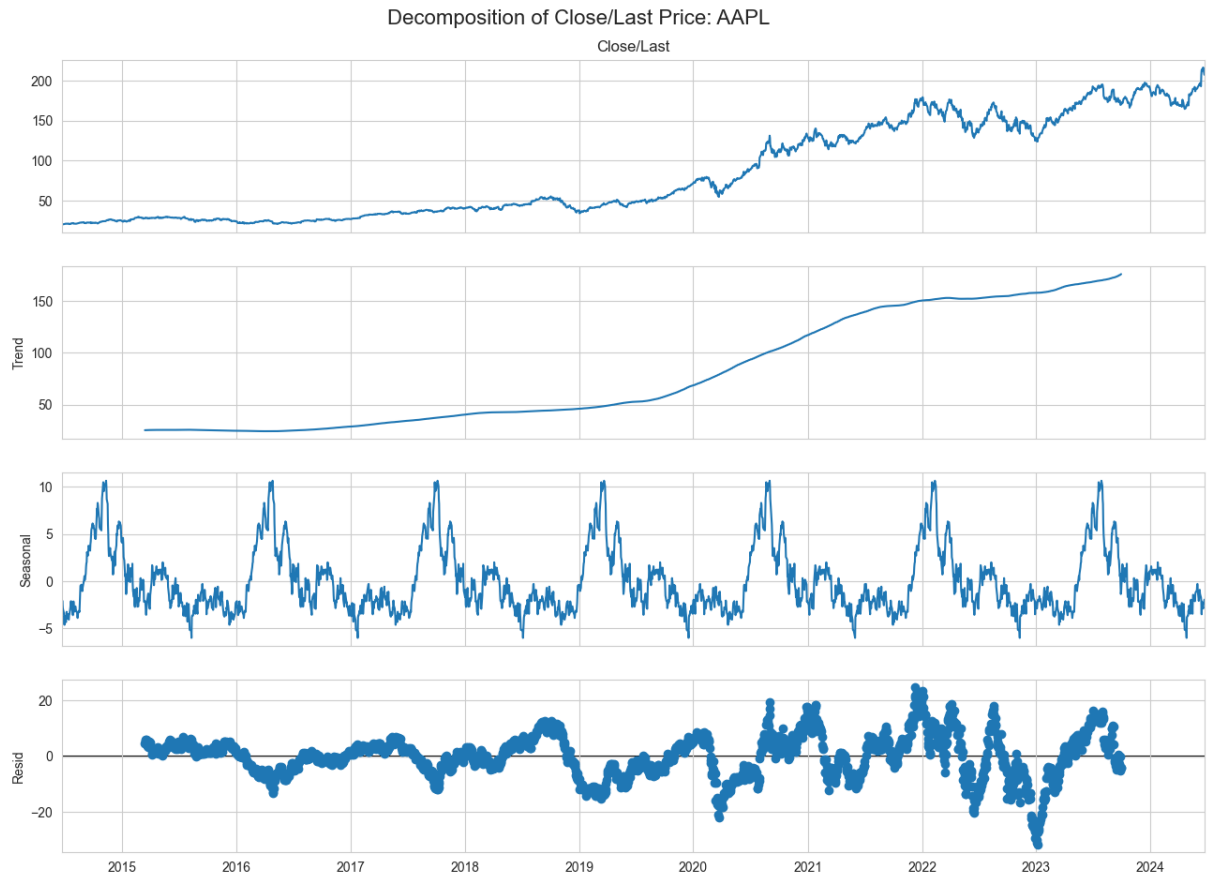


Figure 22: Decomposition of Close/Last Price: AAPL

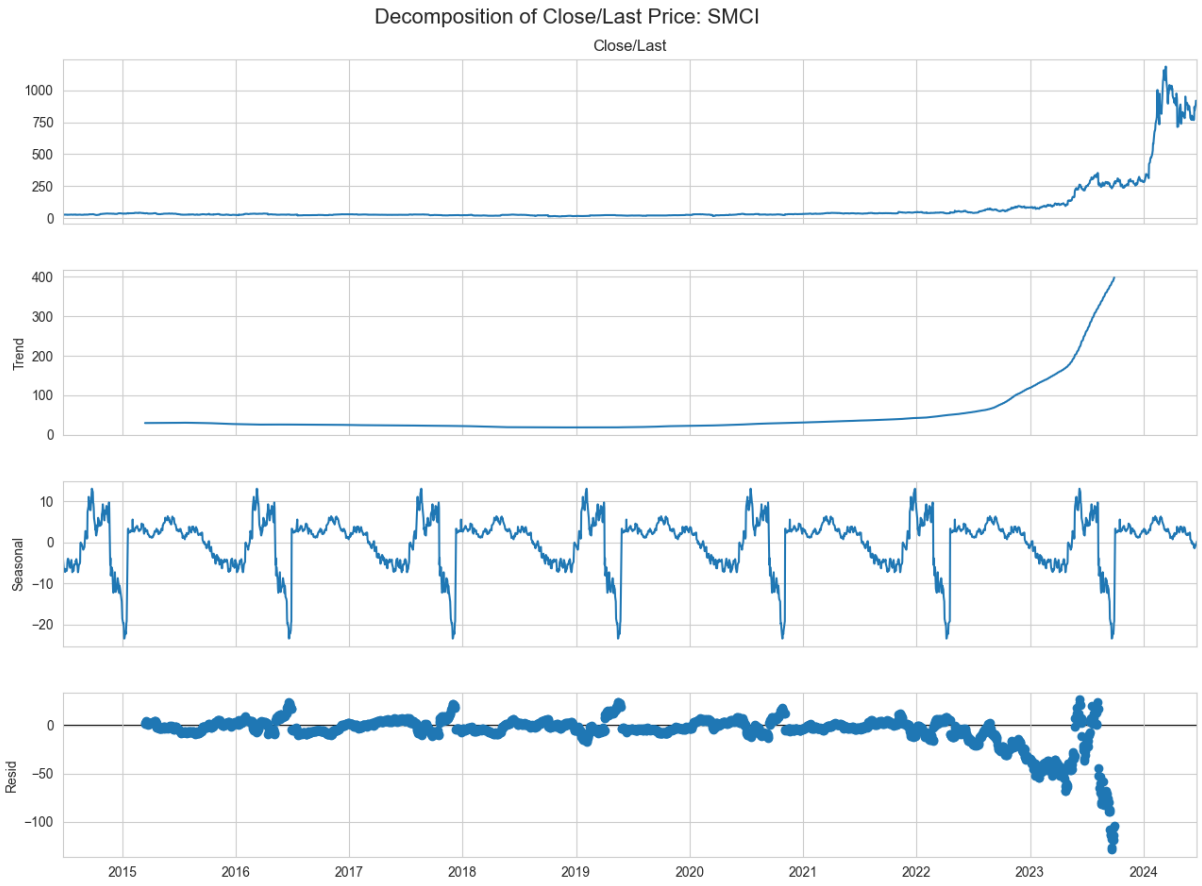


Figure 23: Decomposition of Close/Last Price: SMCI

```
LSTM RMSE on validation set for Trend: 108.84583075801046
```

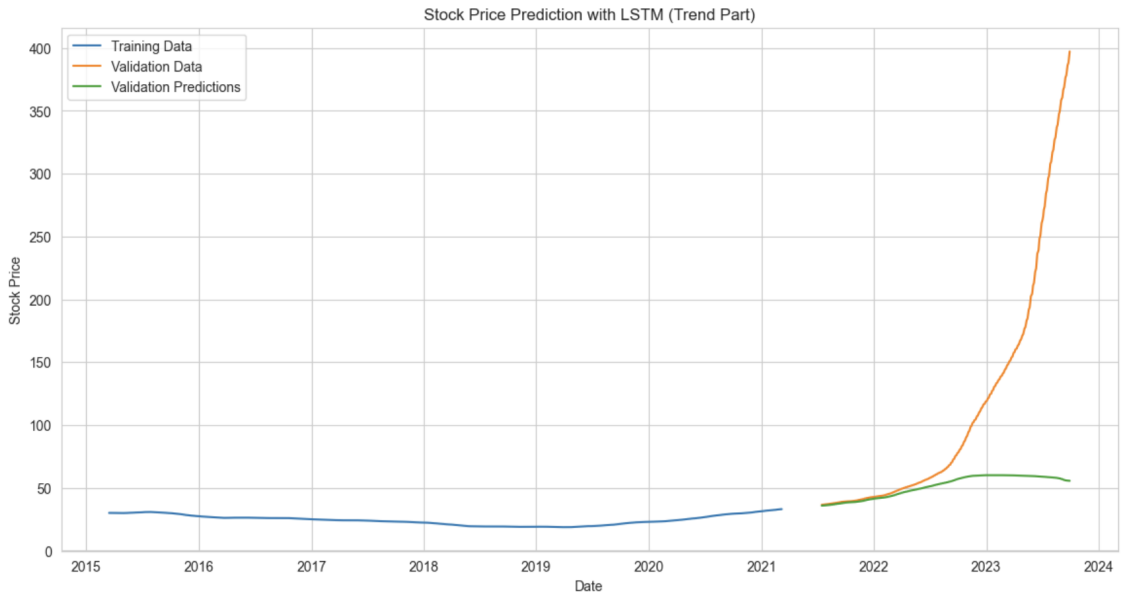


Figure 24: Stock Price Prediction with LSTM (Trend Part) for SMCI

```
LSTM RMSE on validation set for Seasonal: 2.352114413619583
```

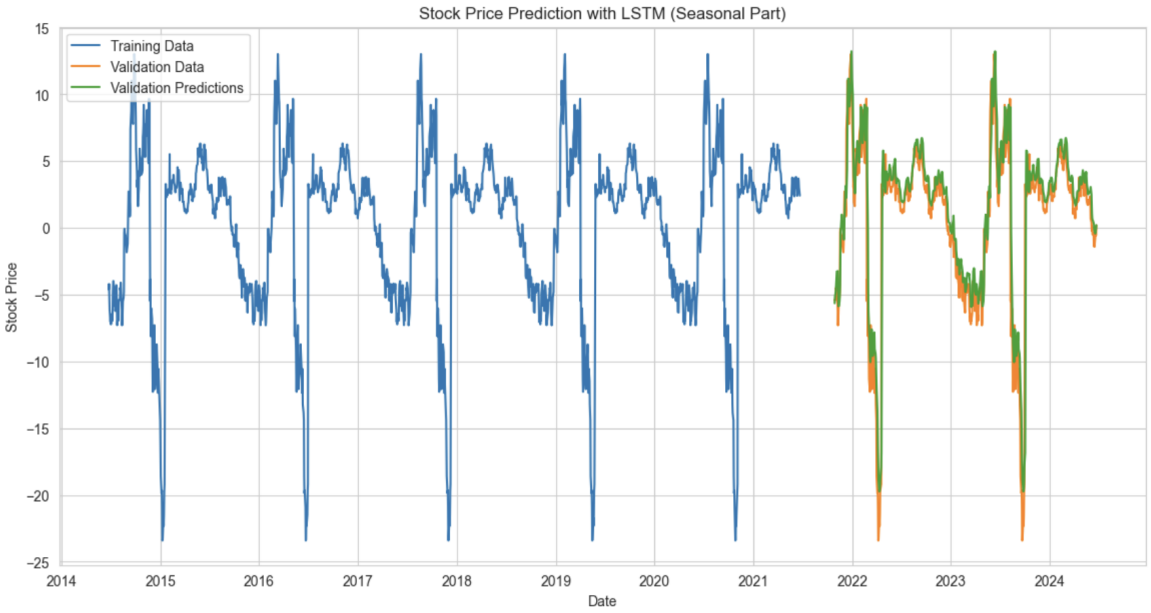


Figure 25: Stock Price Prediction with LSTM (Seasonal Part) for SMCI

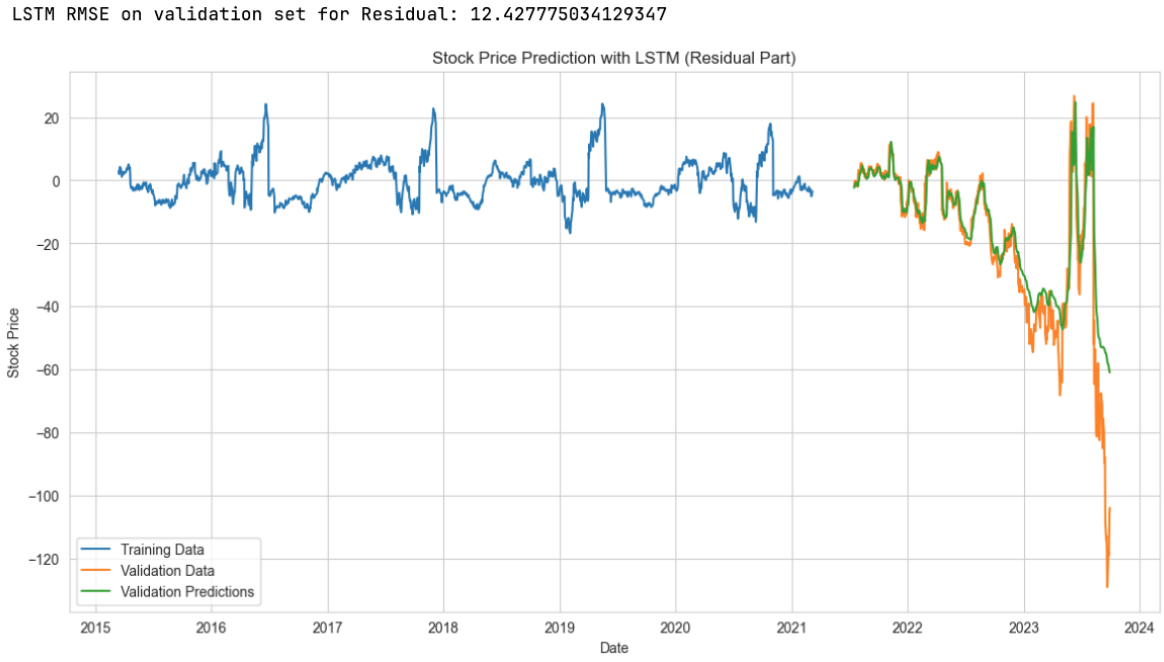


Figure 26: Stock Price Prediction with LSTM (Residual Part) for SMCI

Window Size	RMSE	RMSE Improved by(%)	Time Consumption	Time Consumption Improved by(%)	CPU Usage	CPU Usage Improved by(%)
15	49.72		40.16 seconds		1.6%	
30 (optimal option)	**27.54**	**44.61%**	**59.01 seconds**	**-46.94%**	**2.3%**	**-43.75%**
60	30.92	-12.27%	164.45 seconds	-178.68%	6.2%	-169.57%
90	26.65	13.81%	342.45 seconds	-108.24%	4.1%	33.87%

Figure 27: Feature Selection Experiments Examples