CISC 5352: Machine Learning in Finance
Project Report

Tipu Sultan

Abstract: This project applies machine learning in MATLAB to create a pair trading strategy, specifically focusing on high-frequency trading. It identifies JPMorgan Chase & Co. (JPM) and BlackRock, Inc. (BLK) as highly correlated stock pairs from the chosen dataset. The strategy employs a sliding window technique and examines different regression models, like linear and polynomial, to understand the dynamics between JPM and BLK. A key innovation of this research is using Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, to forecast the trading strategy's cumulative profit and loss (P&L). LSTMs are particularly adept at recognizing temporal patterns in stock prices, an essential factor in financial forecasting. The project includes extensive hyperparameter tuning to refine the LSTM model, aiming for precise stock movement predictions. The model's effectiveness is evaluated using the Root Mean Square Error (RMSE) metric. This study showcases MATLAB's capabilities in deploying sophisticated machine learning methods in finance. It offers a detailed framework for a solid pair trading strategy, emphasizing LSTM's role in financial market predictions. The results are significant for traders and analysts, illustrating how integrating machine learning can enhance decision-making and profitability in high-frequency trading. Drawing from the insights of Elliott, Van Der Hoek, and Malcolm [1], the methodology has potential applications for generating wealth from any quantities in financial markets observed to be out of equilibrium."

Introduction:

I. Objective:

The core objective of this project is to develop a data-driven pair trading strategy, focusing on the statistical analysis and prediction of stock returns for JPMorgan Chase & Co. (JPM) and BlackRock, Inc. (BLK). Leveraging MATLAB's computational capabilities, the project analyzes the relationship between these two highly correlated stocks within defined time windows, each encompassing 60 data points. The primary approach involves using linear and quadratic polynomial regression to predict JPM's returns based on BLK's, a method well-suited for capturing the nuances in pair trading dynamics. The strategy calculates the residuals or the difference between actual and predicted returns of JPM for each window, culminating in a cumulative profit and loss (P&L) representation over time. This project aims to demonstrate the potential of statistical models in predicting stock performance in a pair trading context and explore how these insights can be translated into actionable trading strategies. While the focus remains on the statistical interplay between JPM and BLK stock returns, the project sets the groundwork for the future integration of comprehensive trading rules and risk management strategies, enhancing the practical application of pair trading in financial markets**.** We link the profitability to the presence of a common factor in the returns, different from conventional risk measures [2]."

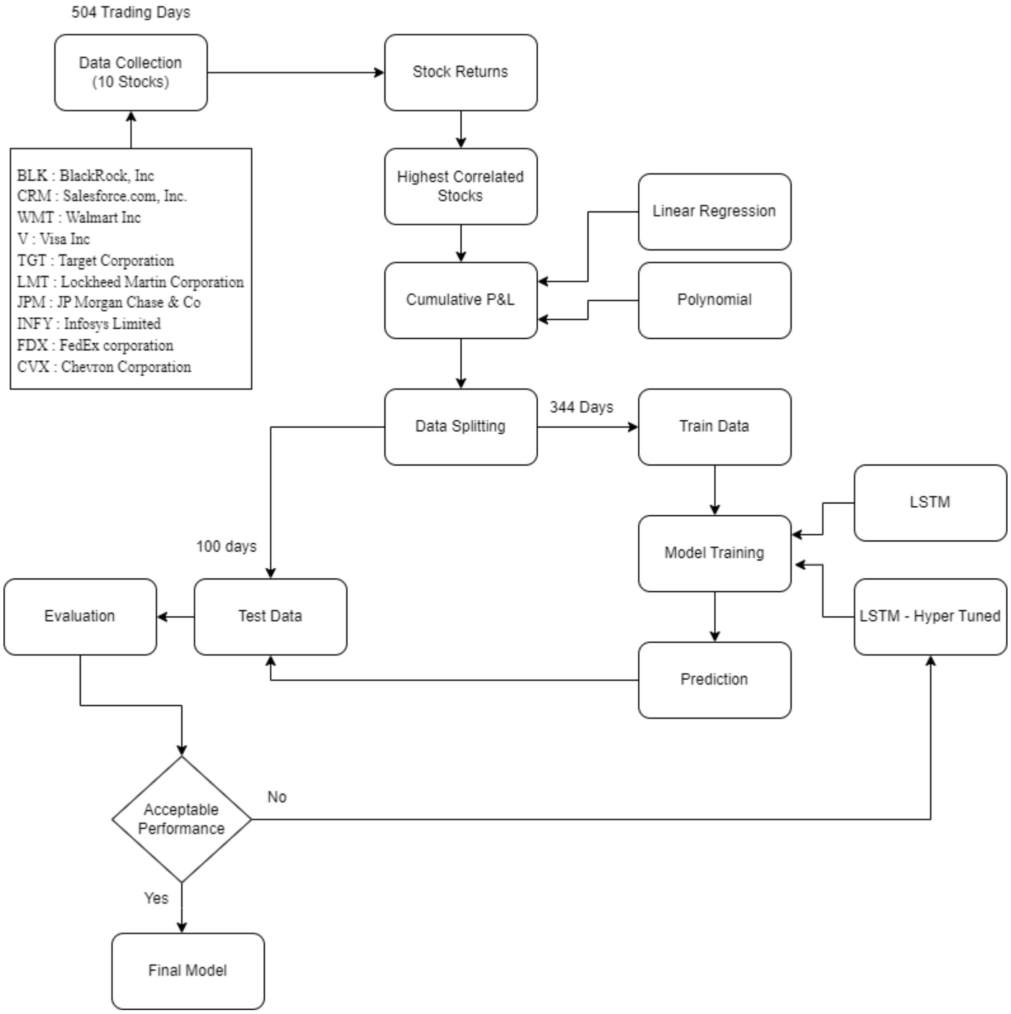


Fig1: Framework for this Project

II. Methodology:

A. Data Preparation:

Initially, ten distinct stocks were identified for analysis. Utilizing Python and the 'yfinance' library, which facilitates the acquisition of historical stock data from Yahoo Finance, data for these selected stocks was retrieved for two years. Subsequently, the closing prices for each stock were pivoted, transforming the dataset such that the columns represented the stocks and the rows indicated the closing prices. Following this transformation, the data was exported to an Excel/CSV format compatible with MATLAB. Finally, the stock data file was imported into MATLAB as a numerical matrix.

Stocks List:

Stock Details and Tickers		
S.No	Ticker	Stock Name
1	CVX	Chevron Corporation
2	FDX	FedEx Corporation
3	INFY	Infosys Limited
4	JPM	JP Morgan Chase & Co
5	LMT	Lockheed Martin Corporation
6	TGT	Target Corporation
7	V	Visa Inc
8	WMT	Walmart Inc
9	CRM	Salesforce.com, Inc.
10	BLK	BlackRock, Inc

Table1:Stocks List

B. Correlation:

Upon exporting the data to MATLAB, a script was executed to compute the stock returns, effectively converting the closing price series into corresponding returns. Subsequently, a correlation matrix was constructed to delineate the correlation of returns among the ten selected stocks. This correlation matrix was then utilized to generate a heatmap, enabling a visual interpretation of the stocks demonstrating the highest correlation. Selecting stocks with the highest correlation is a pivotal step in the pair trading strategy, as it underpins the approach's effectiveness.

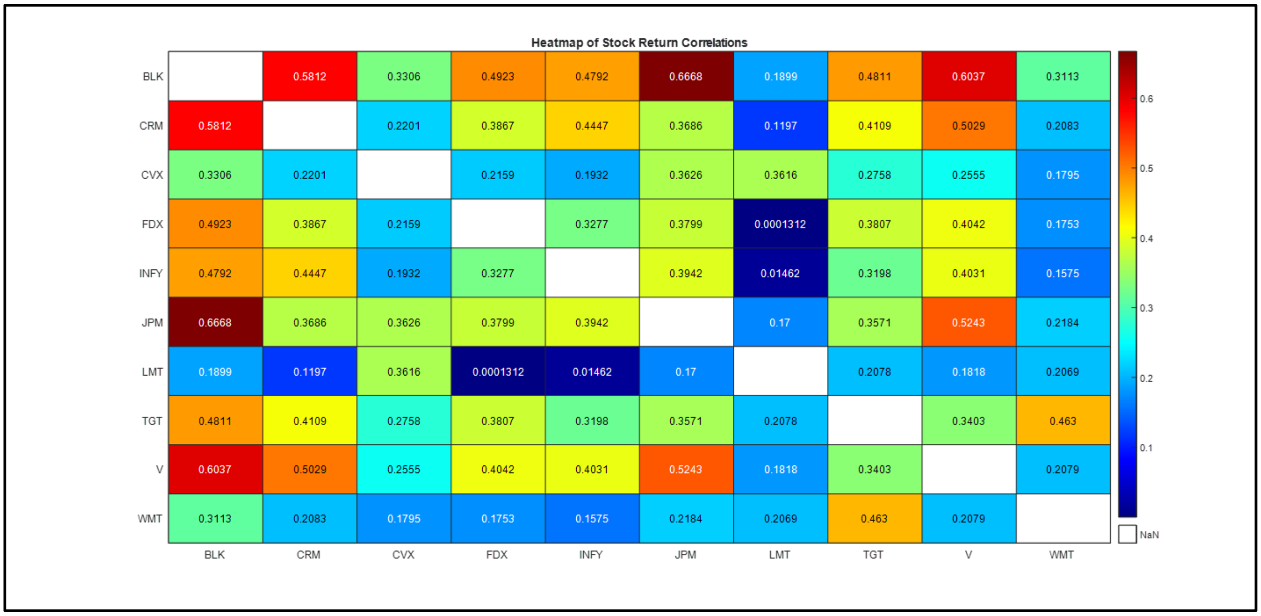


Fig2: Heatmap for Stock Returns

Based on the heatmap analysis, it was determined that JP Morgan (JPM) and BlackRock (BLK) exhibited the highest correlation in terms of returns. Consequently, these two stocks were selected for more in-depth analysis in the subsequent phases of the study. This decision aligns with the core objectives of pair trading strategies, where identifying highly correlated stocks is crucial for predictive modeling and strategy formulation.

The highest correlation in returns is 0.666815 between JPM and BLK

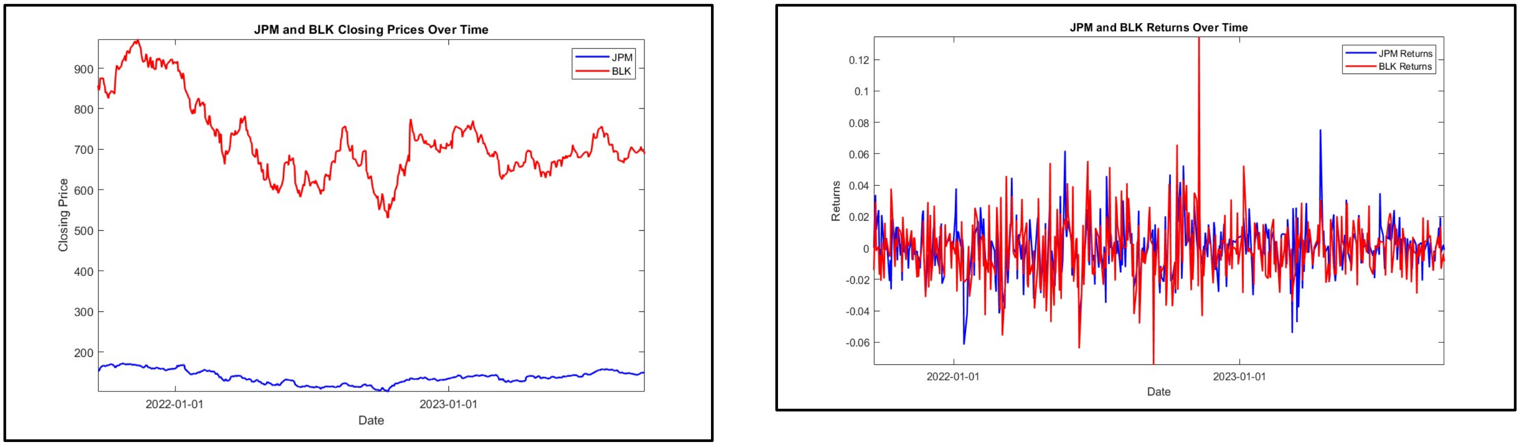


Fig3:JPM and BLK Closing Prices and Return Over Time

C. Cumulative P&L:

Sliding Window:

Next, using MATLAB script, we create a sliding window to calculate the residuals or cumulative P&L. The sliding window approach is a pivotal technique employed in this project to analyze the stock prices of JPMorgan Chase & Co. (JPM) and BlackRock, Inc. (BLK) over time. A window of 60 data points is used, which slides across the entire dataset, segment by segment. This method allows for examining the stock prices and their returns in smaller, more manageable subsets of data, enabling a dynamic analysis that reflects the evolving nature of financial markets. Focusing on these specific windows, the strategy captures the short-term relationships and trends between the stock pairs, which is crucial for making timely and effective trading decisions. Using a sliding window also aids in understanding how the correlation between the stocks changes over time, offering insights into their interdependencies and potential predictive patterns.

Regression Analysis:

a) Linear Regression: Linear regression is utilized to establish a fundamental relationship between the stock returns of JPM and BLK. It models the returns of JPM as a linear function of BLK's returns, providing a baseline understanding of how these stocks move in relation to each other. This method is crucial for its simplicity and interpretability, often serving as a starting point in financial modeling.

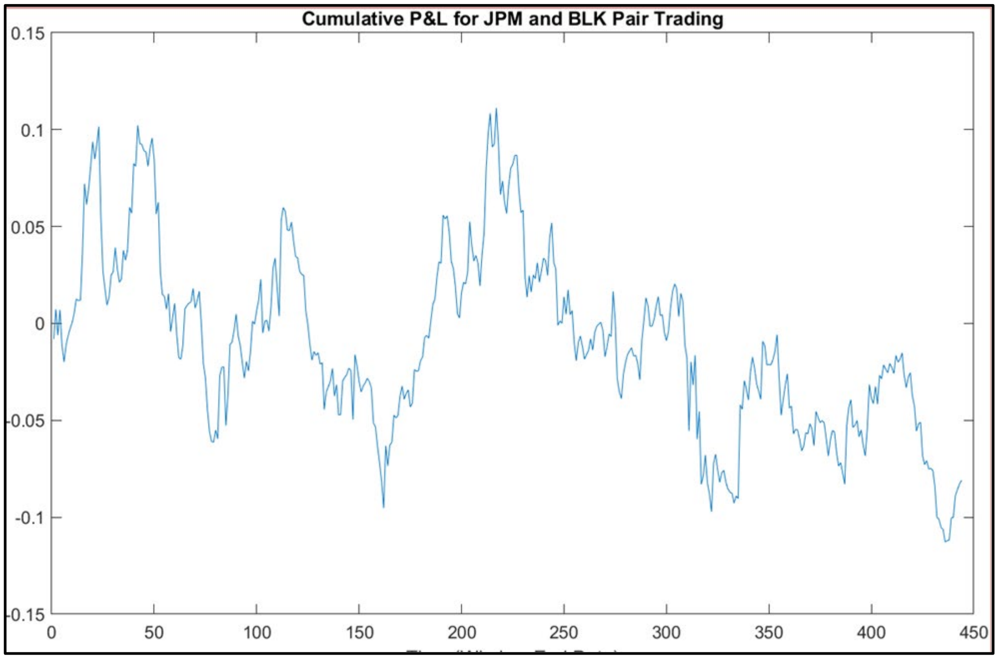


Fig4: Cumulative P&L for JPM and BLK Pair Trading

b) Polynomial Regression: The project advances to quadratic polynomial regression to capture more complex relationships between the stock returns. Unlike linear regression, polynomial regression can model non-linear patterns, making it more suitable for financial data that often exhibit such characteristics. Specifically, a quadratic model is chosen better to fit the potentially curved trends in the data, offering a more nuanced view of the stock dynamics.

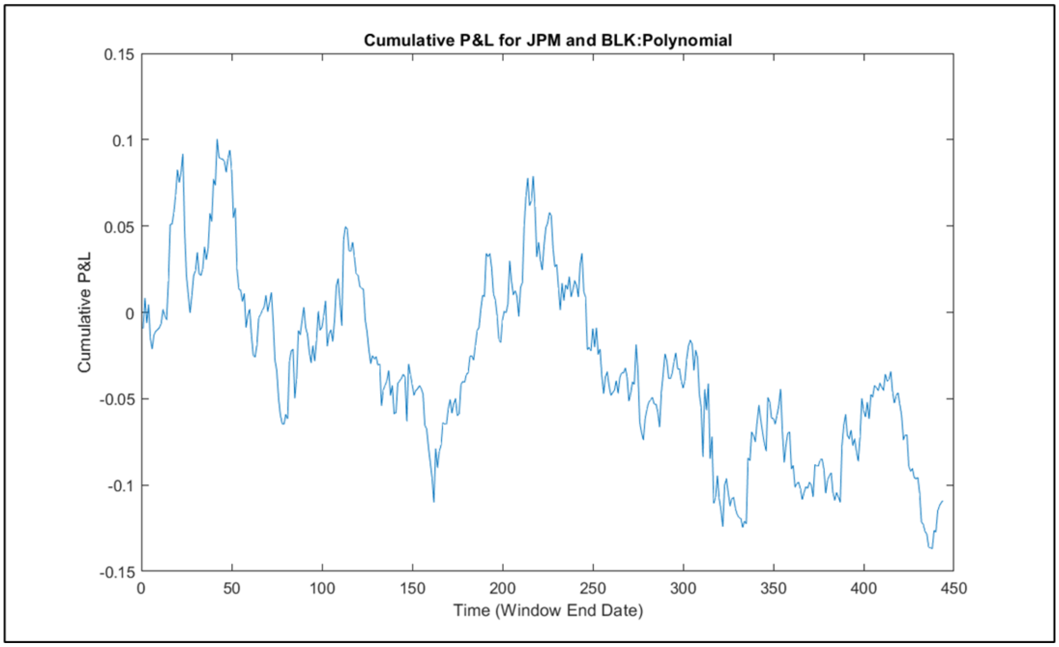


Fig5: Cumulative P&L for JPM and BLK(Polynomial)

The selection of these specific linear and polynomial regression methods is driven by the need to comprehensively understand and model the relationship between JPM and BLK stock returns. Linear regression offers a straightforward model, and polynomial regression provides a fit for non-linear patterns. This combination of methods ensures a well-rounded analysis, catering to the various facets of financial data behavior. Each technique contributes uniquely to the overall strategy, enabling the development of a more effective and reliable pair trading model.

D. Data Splitting & Transformation:

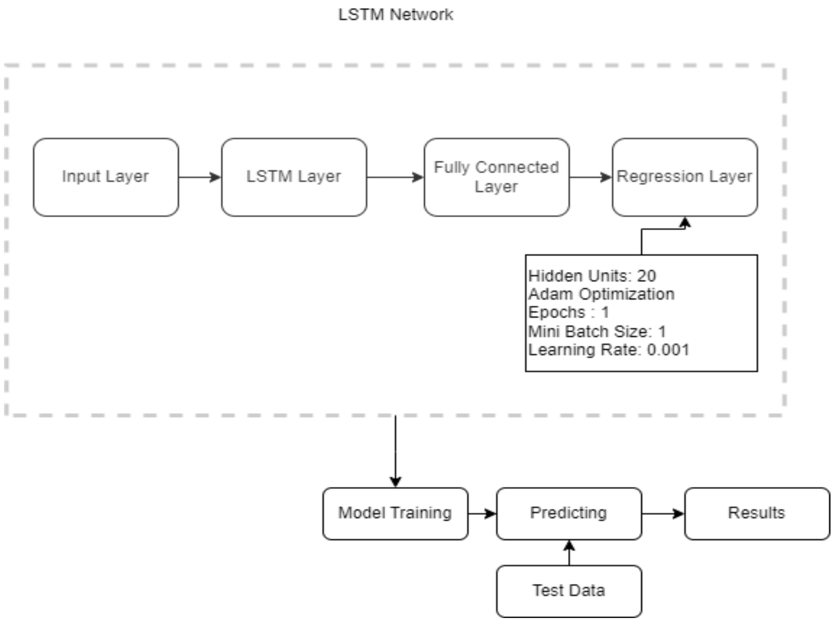
Having calculated the Cumulative Profit and Loss (P&L) for both linear and polynomial regression models over 444 days (accounting for a reduction in dataset size due to the implementation of a 60-day sliding window), the study then transitions to the preparation phase for the Long Short-Term Memory (LSTM) architecture. For this purpose, the dataset is divided into two segments: a training set encompassing 344 days and a testing set covering the remaining 100 days. This partitioning is essential for evaluating the LSTM model's performance and generalizability. Following the data split, a crucial transformation is undertaken, converting the dataset into a row vector format. This format is specifically tailored for compatibility with the LSTM model, ensuring that the data structure aligns with the requirements of the LSTM's input layer.

E. Long Short-Term Memory:

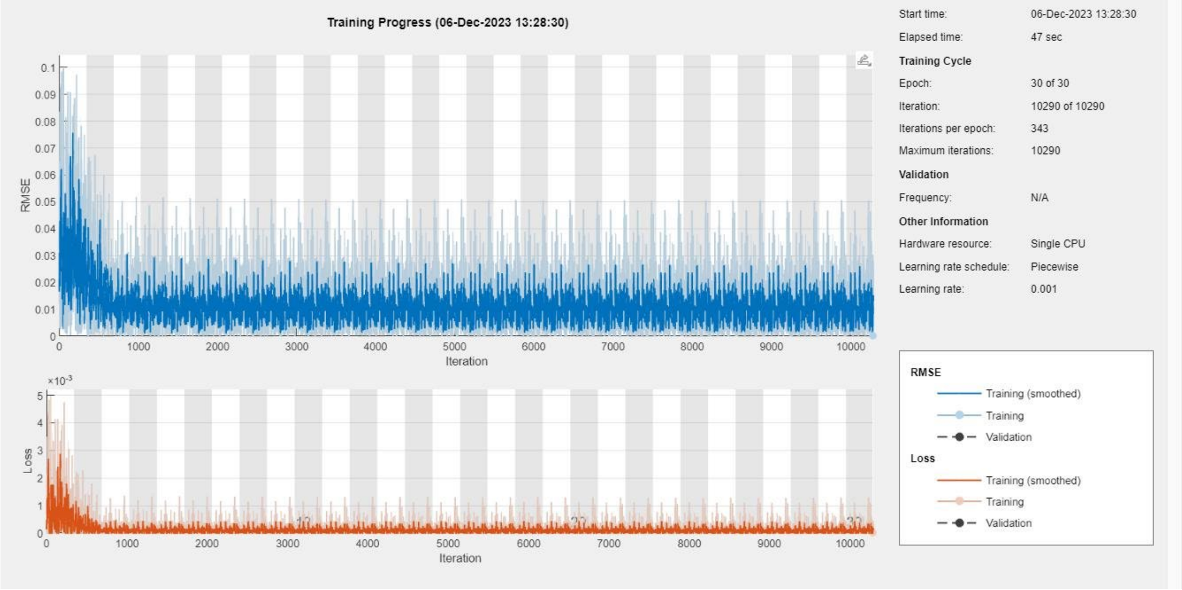
In this project, a Long Short-Term Memory (LSTM) network, a specific type of recurrent neural network (RNN), is employed to forecast the cumulative profit and loss (P&L) in pair trading. The LSTM architecture is designed to process data sequences (in this case, financial time series), capturing long-term dependencies and patterns that are pivotal for accurate predictions. "In this project, we draw inspiration from the Pairs Trading strategy as outlined in [3], where the authors emphasize the challenges of finding robust pairs in the context of growing data availability."

a) Basic LSTM Network:

Initially, we implemented a basic Long Short-Term Memory (LSTM) network to forecast the cumulative profit and loss (P&L) from pair trading of JPM and BLK stocks. The LSTM network, characterized by its ability to capture temporal dependencies in time series data, consists of an input layer, an LSTM layer with 20 hidden units, a fully connected layer, and a regression output layer. This architecture is optimized for sequence prediction tasks, making it ideal for financial time series analysis. The network is trained using the Adam optimization algorithm over 30 epochs, with specific training options like a mini-batch size of 1 and a learning rate schedule to enhance convergence. Post-training, the model is employed to predict future stock returns, with its performance evaluated using the Root Mean Square Error (RMSE) between the predicted and actual returns. This LSTM model serves as a sophisticated tool in our analysis, offering insights into future performance based on historical data, a crucial aspect of our pair trading strategy.



Linear Regression:



Start time: 06-Dec-2023 13:28:30

Elapsed time: 47 sec

Training Cycle

Epoch: 30 of 30

Iteration: 10290 of 10290

Iterations per epoch: 343

Maximum iterations: 10290

Validation

Frequency: N/A

Other Information

Hardware resource: Single CPU

Learning rate schedule: Piecewise

Learning rate: 0.001

Fig6: Training Process(Linear Regression)

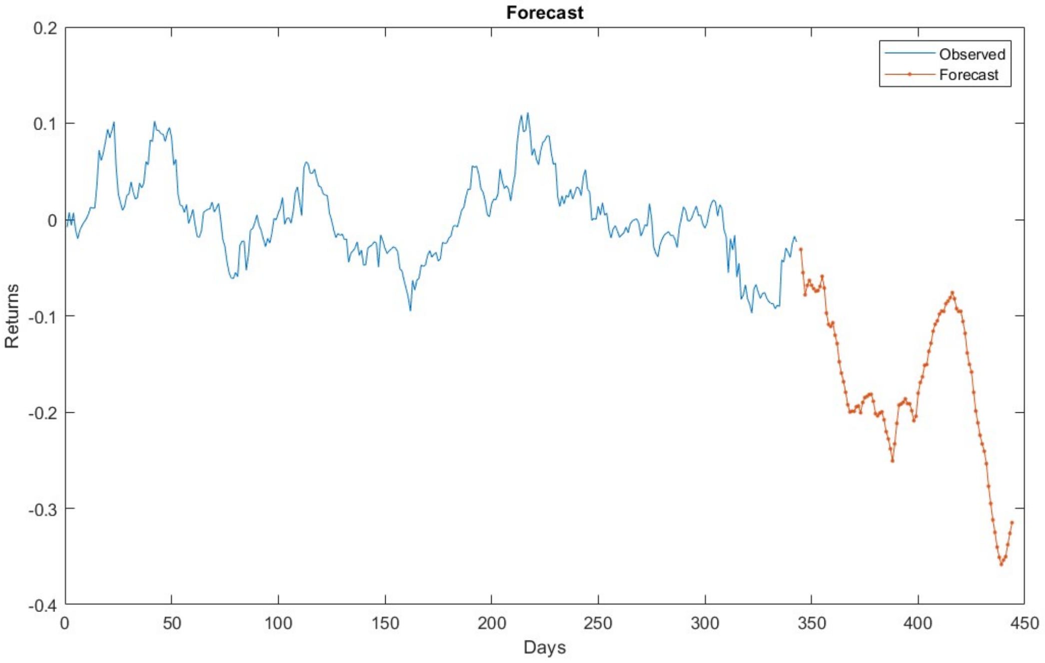


Fig7: Forecast(Linear Regression)

Utilizing the trained linear regression model, we forecasted the unknown cumulative profit loss for a 100-day period. The plot visually represents the forecasted values alongside the trained data, offering a comprehensive view of the model's predictive performance. This graphical representation is a valuable tool for assessing the model's ability to project cumulative profit loss trends over the specified timeframe, providing insights for informed decision-making in financial analysis.

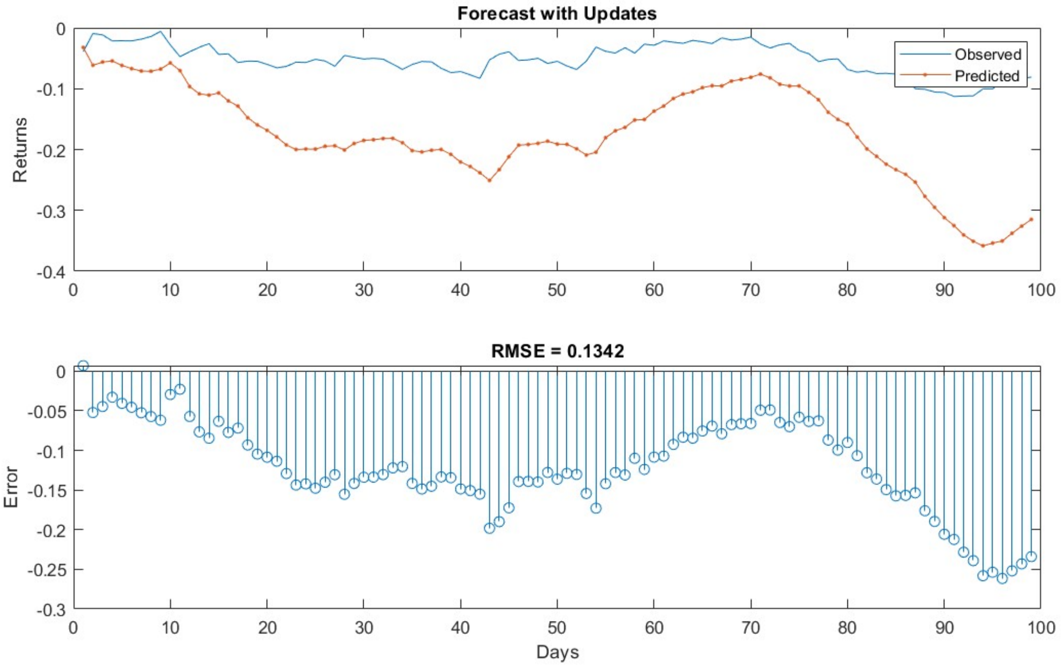


Fig8: Forecast with Updates

The plot illustrates the performance of our basic LSTM model in forecasting the test dataset, revealing a suboptimal outcome with a **Test RMSE of 0.1342**. Despite the model demonstrating some predictive capability, there is room for improvement. Acknowledging this, we plan to enhance performance through hyperparameter tuning, optimizing the model's parameters for better accuracy. The intention is to refine the model's predictive capabilities, achieving an improved alignment between forecasted and actual values for more robust and reliable predictions in future analyses.

Polynomial Regression:

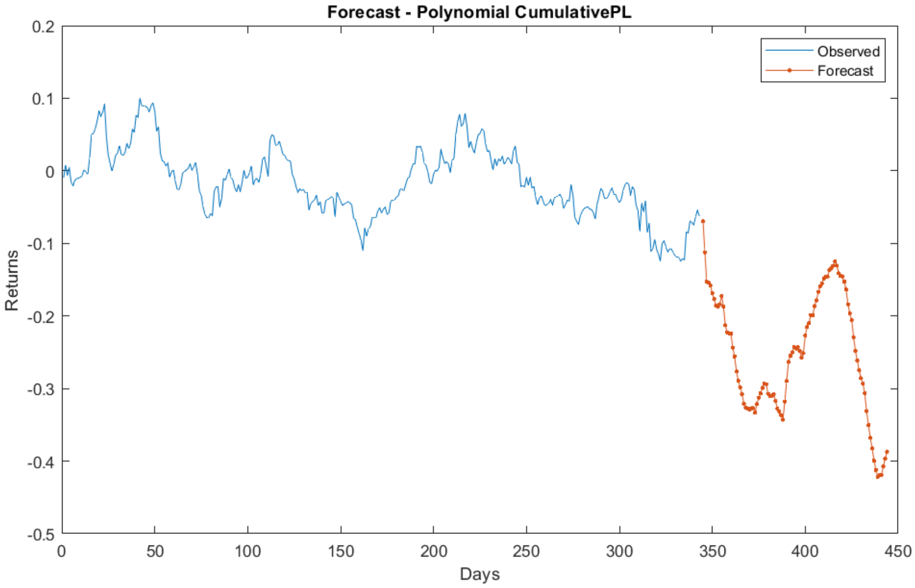


Fig9: Forecast Polynomial Regression

Leveraging the trained polynomial regression model, we conducted a forecast for the unknown cumulative profit loss over 100 days. The plot visually represents the forecasted values in conjunction with the trained data, comprehensively depicting the model's predictive capacity. This graphical representation is a valuable tool for evaluating the model's effectiveness in projecting cumulative profit loss trends over the specified timeframe, offering insights for informed decision-making in financial analysis.

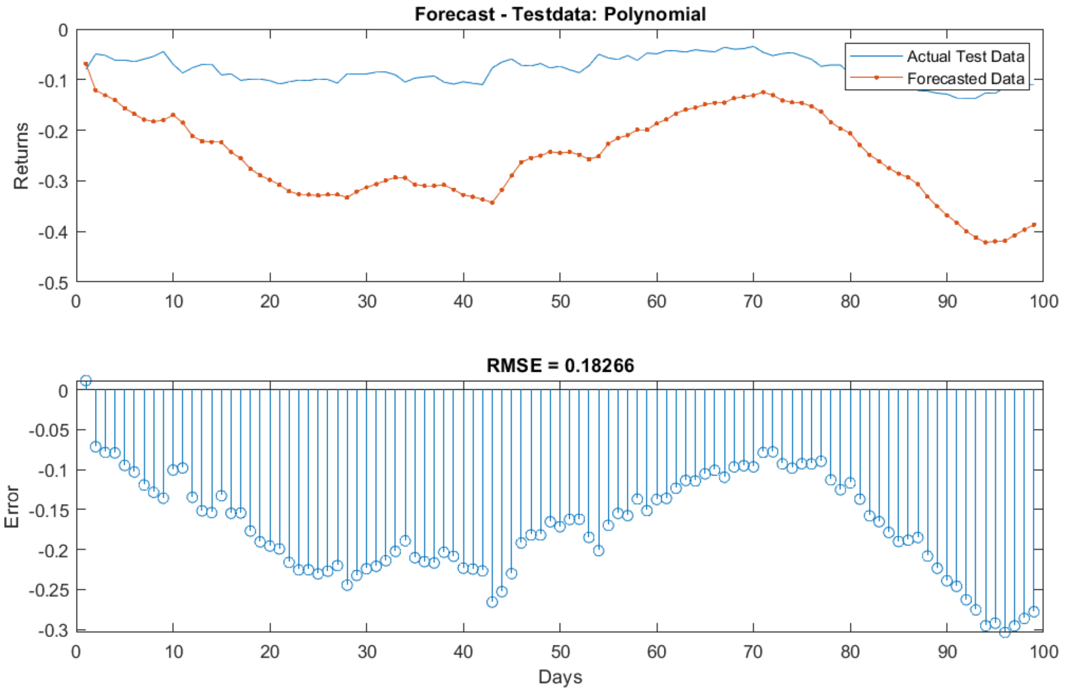


Fig10: Forecast in Test Data(Polynomial Regression)

The plot depicts the performance of our basic polynomial regression model in forecasting the test dataset, revealing a suboptimal outcome with a **Test RMSE of 0.1826**. While the model exhibits some predictive capability, there is room for improvement. Acknowledging this, we plan to enhance performance through hyperparameter tuning, optimizing the model's parameters for better accuracy. The intention is to refine the model's predictive capabilities, achieving an improved alignment between forecasted and actual values for more robust and reliable predictions in future analyses.

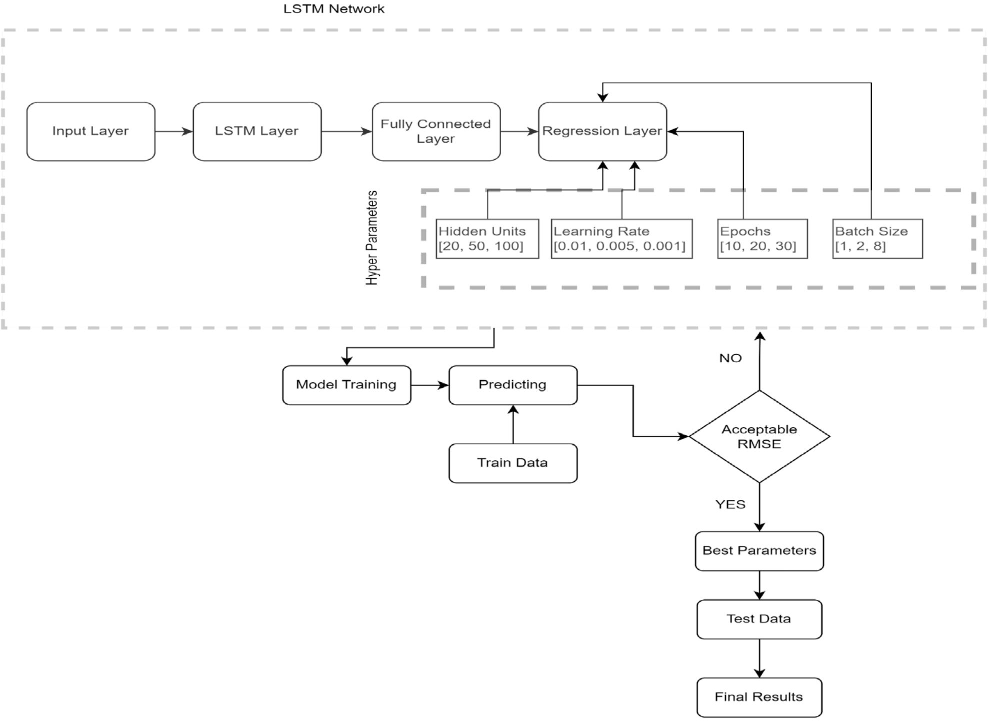
The outcome for LSTM Basic Model:

The analysis of the plots reveals that the basic Long Short-Term Memory (LSTM) model did not perform as effectively in predicting or forecasting values as initially anticipated. In comparison, the linear regression model demonstrated superior performance. This observation is further corroborated by the Root Mean Square Error (RMSE) values, where the RMSE for the polynomial regression model is notably higher than that of the linear regression. This discrepancy in RMSE values indicates that the linear regression model was more accurate in

capturing the underlying trends and patterns in the dataset for this specific pair trading strategy. Overall, there are better models than this. We need to plug and play with the parameters to get a better model output.

b) Hyper-Tuned LSTM Network:

To get a better result, a systematic approach to hyperparameter tuning was undertaken to optimize the LSTM network for forecasting stock returns. The tuning process involved experimenting with different combinations of hyperparameters, including the number of hidden units, learning rates, epochs, and batch sizes. Specifically, we tested hidden units of 20, 50, and 100, learning rates of 0.01, 0.005, and 0.001, epoch options of 10, 20, and 30, and batch sizes of 1, 2, and 8. For each combination, the LSTM network, comprising a sequence input layer, an LSTM layer set to 'sequence' output mode, a fully connected layer, and a regression layer, was trained using the Adam optimizer. The performance of each model configuration was evaluated based on the Root Mean Square Error (RMSE) calculated from the training set predictions. To minimize this metric, the optimal set of hyperparameters was identified by comparing the RMSE values. This rigorous tuning process ensured that the LSTM model was finely adjusted to capture the complex financial time series data patterns, leading to more accurate and reliable predictions. The best-performing model was then used to forecast on both the training and test datasets, with its effectiveness evidenced by the respective RMSE values, providing a robust framework for our pair trading strategy.



Linear Regression:

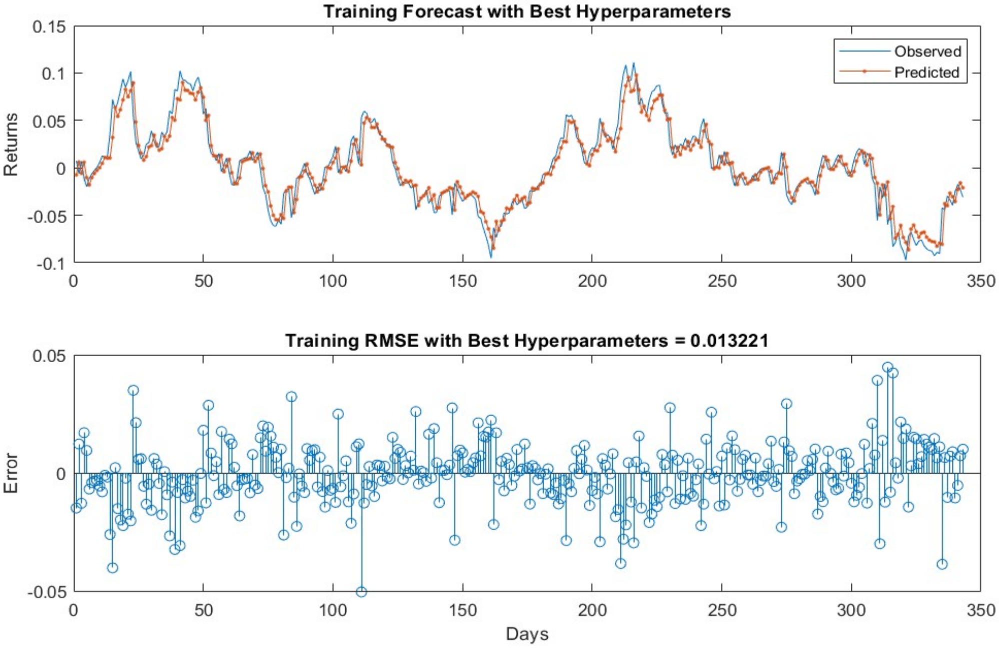


Fig11: Training Forecast with Best Hyperparameters(Linear Regression)

The observed versus forecasted comparison for the training dataset, utilizing the hyper-tuned LSTM model with optimized parameters, demonstrates refined predictive performance. The plot visually illustrates how the model predicted the training dataset with a Train RMSE of 0.0132, indicating a close alignment between the forecasted and actual values. This low RMSE affirms the heightened accuracy achieved through meticulous parameter tuning, highlighting the model's proficiency in capturing temporal patterns within the training data.

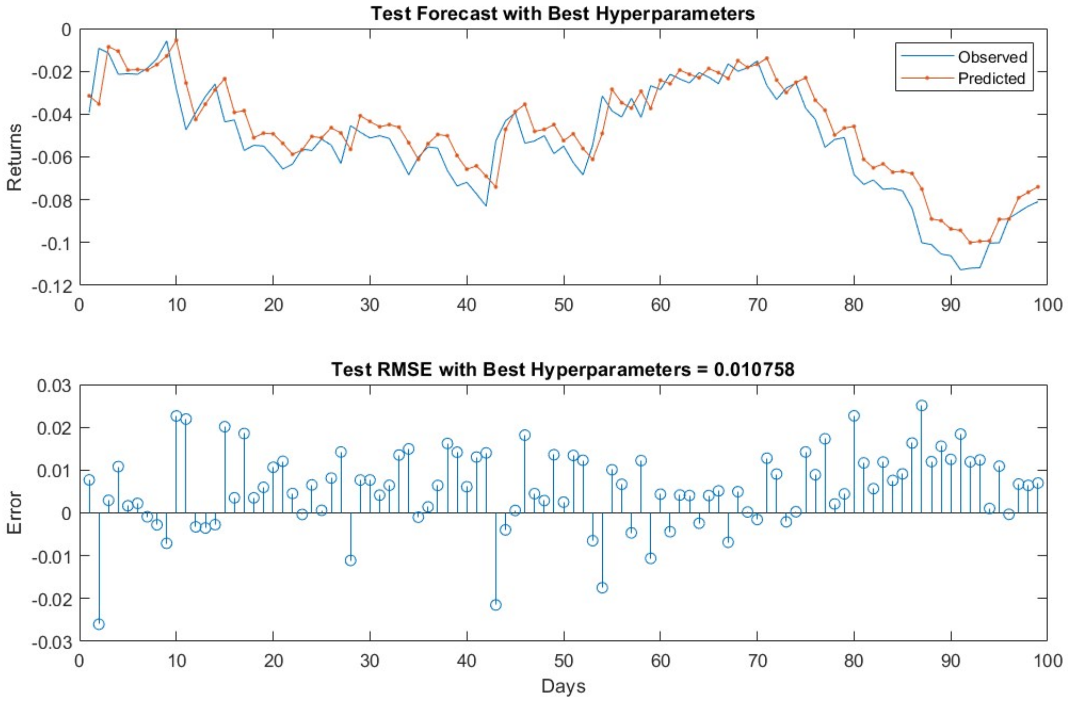


Fig12: Test Forecast and RMSE with best Hyperparameters

The observed versus forecasted comparison for the test dataset, utilizing the hyper-tuned LSTM model with optimized parameters, reveals superior performance. The plot visually indicates that the model outperformed the basic LSTM, with a Test RMSE of 0.010758, showcasing enhanced accuracy in predicting the test data. This lower RMSE underscores the success of the hyper-tuning process, emphasizing the model's proficiency in capturing underlying patterns and nuances within the test dataset.

Polynomial Regression:

Train Data: In the observed versus forecasted comparison for the training dataset, the hyper-tuned LSTM model with the best parameters, including 50 hidden units, a learning rate of 0.010000, 20 epochs, and a mini-batch size of 1, demonstrates refined predictive performance. The plot visually depicts how the model predicted the training dataset, showcasing a Train RMSE of 0.012598. This low RMSE underscores the success of the hyperparameter tuning process, emphasizing the model's proficiency in capturing intricate patterns within the training data, as reflected in the plot.

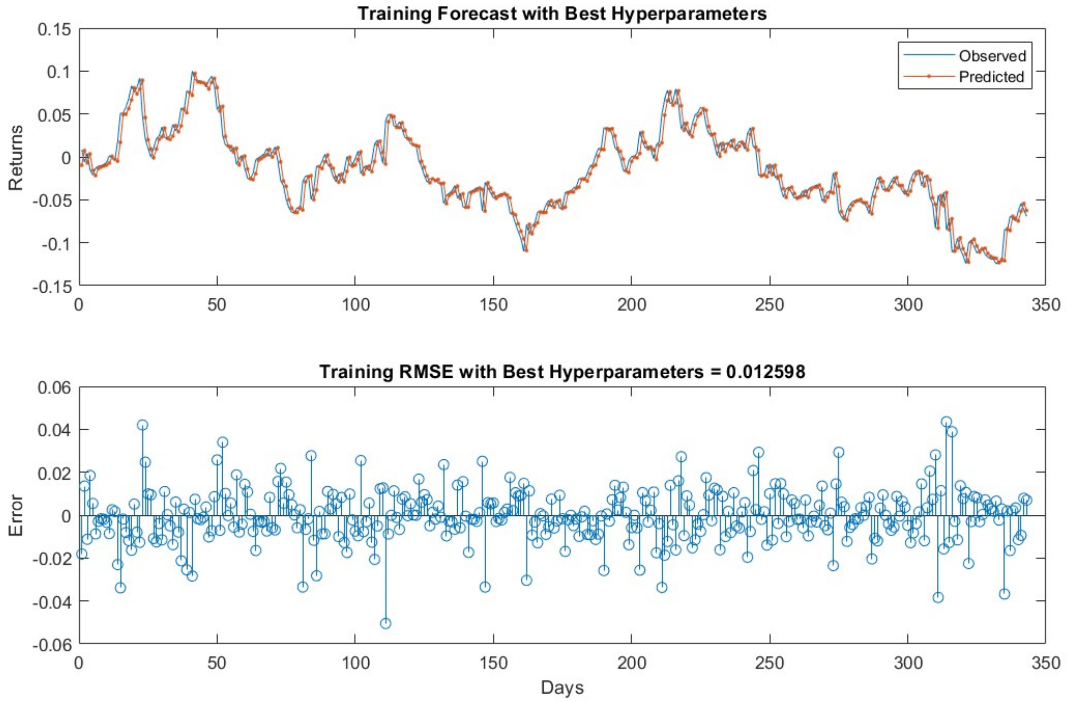


Fig13: Training Forecast and Training RMSE with Best Hyperparameters

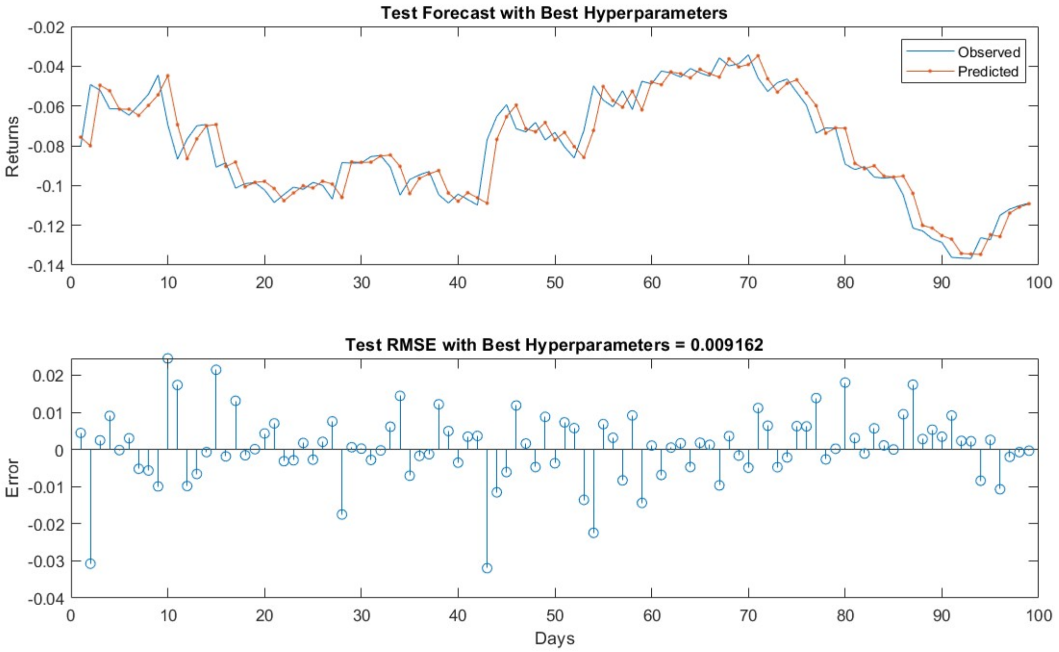
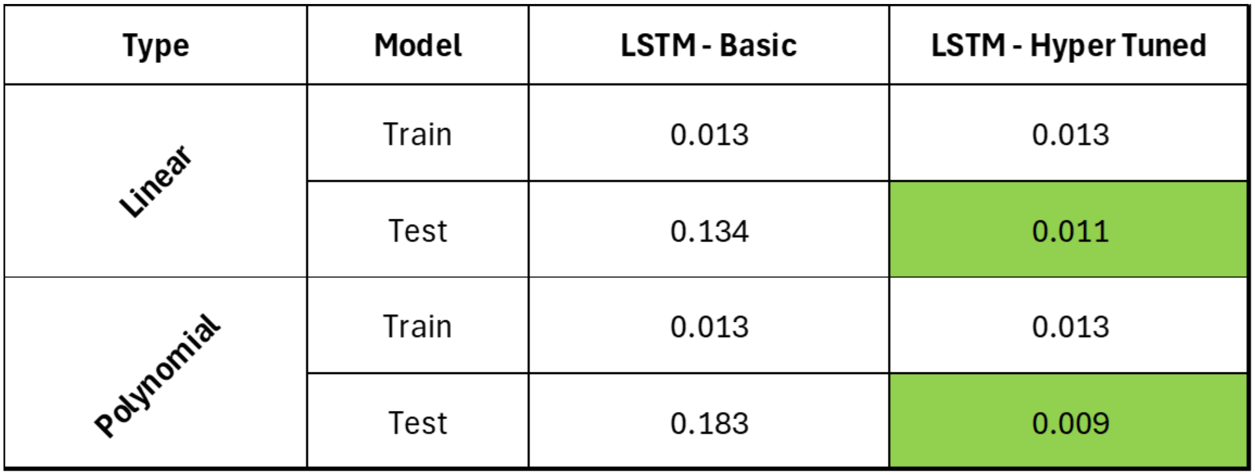


Fig14: Test Forecast and Test RMSE with Best Hyperparameters

Test Data: In the observed versus forecasted comparison for the test dataset, the hyper-tuned LSTM model with polynomial regression and optimized parameters demonstrated superior performance. The plot visually indicates that the model outperformed the basic LSTM, with a Test RMSE of 0.009162, highlighting enhanced accuracy in predicting the test data. This lower RMSE underscores the success of the hyper-tuning process, emphasizing the model's proficiency in capturing underlying patterns and nuances within the test dataset, as evidenced in the plot.

III. Results:



Type	Model	LSTM - Basic	LSTM - Hyper Tuned
Linear	Train	0.013	0.013
	Test	0.134	0.011
Polynomial	Train	0.013	0.013
	Test	0.183	0.009

Table 2: Result

Our project's quantitative analysis yielded insightful results, particularly when comparing the performance of the basic LSTM model with the hyperparameter-tuned LSTM model. Both models were assessed using linear and polynomial regression approaches to forecast stock returns and calculate the cumulative P&L for the pair trading strategy.

For the **linear regression-based forecasts**, the basic LSTM model achieved an RMSE of 0.013 on the training dataset, indicating a decent fit to the historical data. However, its performance on the test dataset showed an RMSE of 0.134, suggesting a disparity in its predictive accuracy when confronted with unseen data. Post hyperparameter tuning, while the training RMSE remained consistent at 0.013, a remarkable improvement was observed in the test RMSE, which reduced significantly to 0.011. This enhancement in the test results illustrates the effectiveness of hyperparameter optimization in bolstering the model's generalization capability.

Similarly, in **the polynomial regression-based forecasts**, the basic and hyper-tuned LSTM models both recorded an RMSE of 0.013 on the training data. Nevertheless, the test dataset starkly contrasted the two models' proficiency. The basic LSTM model's test RMSE stood at 0.183, which was substantially reduced to 0.009 after hyperparameter tuning. This substantial reduction in RMSE underscores the hyper-tuned model's superior predictive performance and the value of meticulous hyperparameter optimization.

Overall, the hyper-tuned LSTM model outperformed the basic LSTM model in the testing phase for both linear and polynomial regressions, maintaining robustness and reducing prediction errors. These results highlight the potential of LSTM networks in financial time series forecasting, incredibly when fine-tuned to the dataset's nuances. The hyper-tuned model's superior performance on the test data suggests it may offer a more reliable foundation for real-world trading strategy deployment.

IV. Conclusion:

The project's exploration into machine learning applications for pair trading strategies in financial markets has culminated in significant findings. The implementation of a Long Short-Term Memory (LSTM) network to predict the cumulative profit and loss (P&L) of JPMorgan Chase & Co. (JPM) and BlackRock, Inc. (BLK) provided a deep understanding of the intricate dynamics of stock price movements. The results demonstrated the basic LSTM model's capability to capture patterns in financial time series data, with further enhancements achieved through rigorous hyperparameter tuning. The hyper-tuned LSTM model's outstanding performance, particularly in the test phase, reflects its robust predictive power and potential for adaptation to complex market conditions.

These findings underline the crucial role of hyperparameter optimization in developing high-performing predictive models. The improved test RMSE values in both linear and polynomial regression-based forecasts post-tuning indicate that careful calibration of machine learning models is essential for achieving generalizability beyond the training dataset.

Furthermore, this project emphasizes the importance of machine learning in augmenting traditional financial analysis methods. Integrating LSTM networks into pair trading strategies represents a significant advancement in the field, offering a sophisticated tool for traders and analysts. In conclusion, while acknowledging the challenges inherent in financial market predictions, this project reaffirms the transformative impact of machine learning techniques in trading strategies. Future work can extend this research by incorporating additional variables, exploring different machine learning models, and applying the strategy across various financial instruments to validate further and enhance the model's applicability and effectiveness in real-world scenarios.

V. References:

[1] Elliott, R. J., Van Der Hoek, J., & Malcolm, W. P. (2007). Pairs Trading. Pages 271-276. Received 27 Dec 2004, Accepted 11 Apr 2005. Published online: 18 Feb 2007. DOI: 10.1080/14697680500149370.

[2]Gatev, E., Goetzmann, W. N., & Rouwenhorst, K. G. (2006). Pairs Trading: Performance of a Relative-Value Arbitrage Rule. The Review of Financial Studies, Volume 19, Issue 3, Pages 797–827. DOI: 10.1093/rfs/hhj020.

[3] Sarmento, S. M., Horta, N. (Year). Enhancing a Pairs Trading strategy with the application of Machine Learning. Journal/Conference Name. Available at: [https://doi.org/10.1016/j.eswa.2020.113490]