Cryptocurrency Pattern Recognition

Adam Yanai - May ‘25

Introduction

The aim of this project is to build upon findings from a previous study of cryptocurrency activity (Fintech Pattern Analysis) and analyze certain trends and patterns across the top eleven cryptocurrencies by market capitalization (BNB - Binance Coin, BTC - Bitcoin, XRP - Ripple, SOL - Solana, ADA - Cardano, USDT - Tether, DOGE - Dogecoin, ETH - Ethereum, TRX - TRON, USDC - USD Coin, and LDO - Lido). Moreover, the analysis will be used to predict the future movements of those cryptocurrencies through the utilization of machine learning models. Cryptocurrencies are integral components of the evolving financial ecosystem, and understanding their behavior and influence can prove crucial for investors, analysts, and institutions. The provided datasets are organized by date (one data point per day), price, volume, and market capitalization. Therefore, the central objective is to predict future values of cryptocurrency prices and market capitalization, as well as identify the underlying patterns and trends that influence their movements.

Although the data set includes price, volume, and market capitalization, this report will focus on the correlation between price and market capitalization and the effectiveness of these points in demonstrating the behaviour of cryptocurrencies. Furthermore, the study will explore predictions for a day, week, month, and year, and ultimately showcase that the month-based predictions are the most effective. The analysis primarily relies on linear regression and ridge regression models, but finds that the ridge regression models are far more effective, as they are better suited for time-series forecasting in financial markets.

The cryptocurrency data used in this project was provided by the Faculty of Engineering at the University of Sydney. The methodology for this project consisted of a series of major components: (1) data transformation & cleaning, (2) data visualization, (3) predictive modeling, (4) trading strategy, and (5) risk management & performance evaluation. Throughout the first component, the data was cleaned, normalized, and transformed to be high quality and optimal for machine learning modeling. The second component had the treated data visualized, which allowed for initial calculations and pattern identifications to be made. The third component used the information gathered throughout the second stage to create models that make predictions on the price and market capitalization of the cryptocurrencies, using ARIMA and LSTM. The fourth stage placed an emphasis on long term predictions per cryptocurrency, and the fifth and final stage used risk management and performance evaluation metrics to evaluate the final investment decisions. Tools used include Python (Pandas, NumPy, Sklearn, Matplotlib) and Google Colab.

(I) Data Cleaning and Transformation

Initial Data Visualization

To gain a foundational understanding of the project, the dataset was plotted in its entirety. The data of all eleven cryptocurrencies was formatted in an identical CSV file structure, in which a single data point existed per day (taken at 00:00:00 UTC+0), with each cryptocurrency having varying timelines based on its creation and growth. Each data point presented the Price (cost of a unit), Volume (total value of all coins traded in a day), and Market Capitalization (total value of all coins in circulation). Plotting was done in three parts according to the three fields, and in four graphs for each part to facilitate viewing and understanding (the cryptocurrencies with the larger values were removed in sequential order). The three figures below (Figure 1, Figure 2, and Figure 3) are the plots for the three fields.

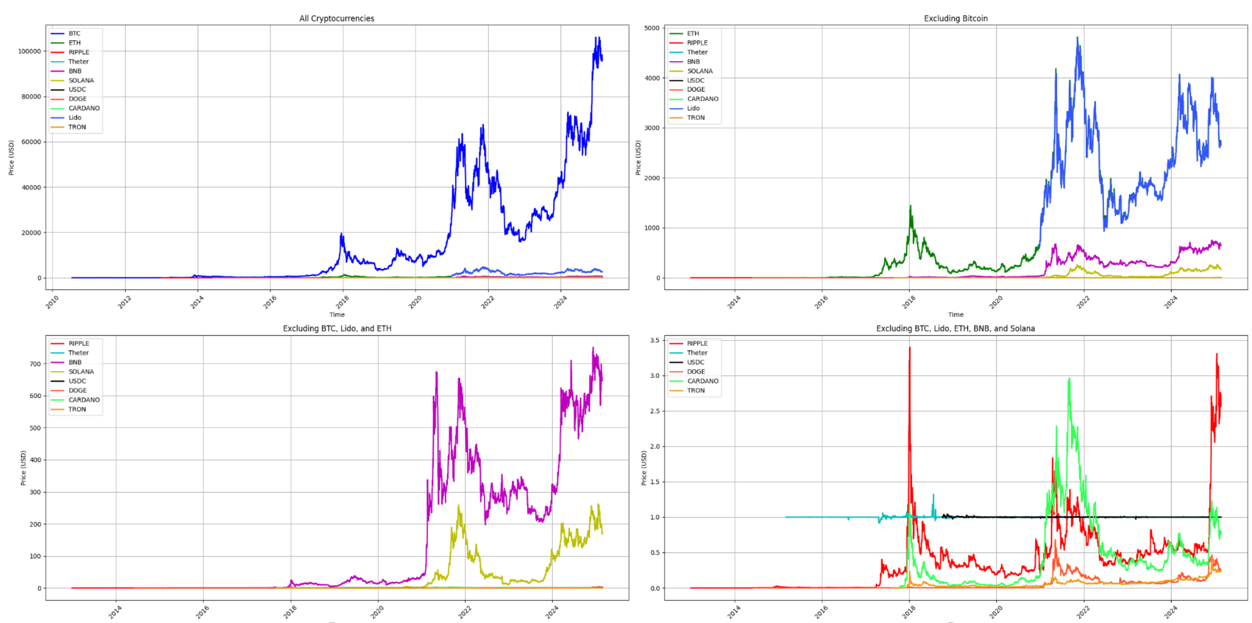


Figure 1: Initial Data Visualization (Price)

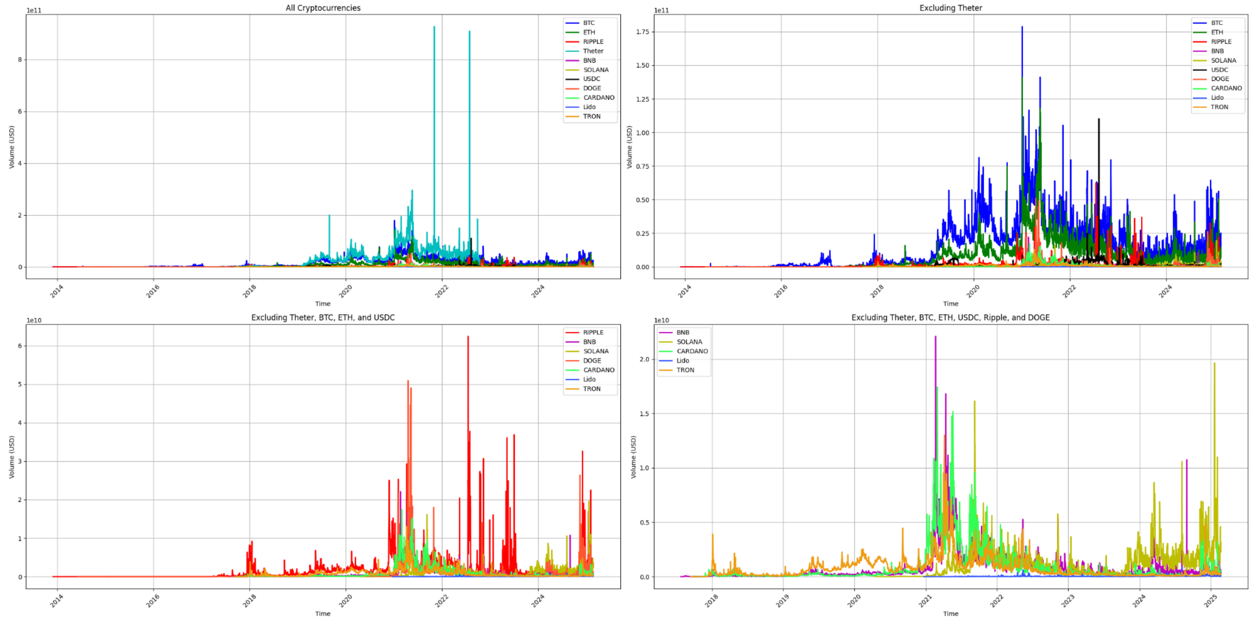


Figure 2: Initial Data Visualization (Volume)

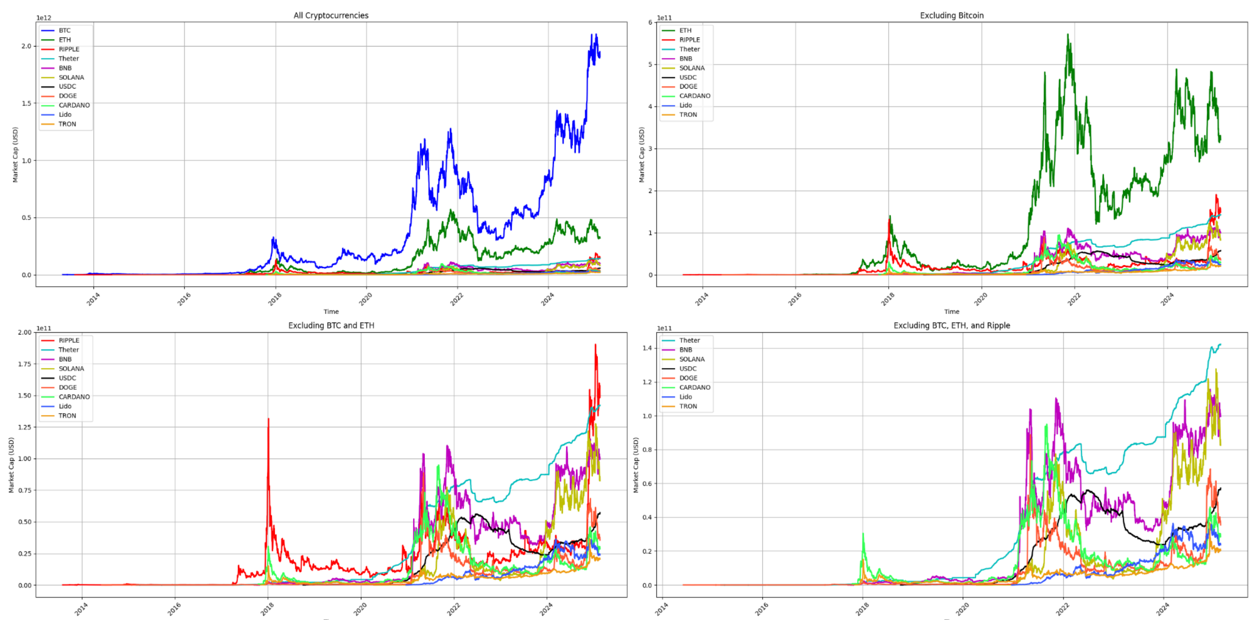


Figure 3: Initial Data Visualization (Market Capitalization)

Through the visualization of the data, I observed that the cryptocurrencies have different starting dates and that they have widely different value ranges, which would create difficulty in analyzing them together. Therefore, solutions must be found for the starting dates (cleaning and transformation) and the variance in the value ranges (normalization).

Data Cleaning & Transformation

To begin the data cleaning process, I first checked for missing values and duplicate rows in the dataset by reading through the files with pd.read_csv(file). I did so by searching for all null cells and cells that share exact values with other cells, and I organized the outputs by category as shown in Figure 4.

```
Missing values in BTC.csv: Date (0), Price (1), Volume (1232), Market_cap (1018)
Duplicate rows in BTC.csv: 0
Missing values in ETH.csv: Date (0), Price (1), Volume (1), Market_cap (2)
Duplicate rows in ETH.csv: 0
Missing values in RIPPLE.csv: Date (0), Price (0), Volume (333), Market_cap (220)
Duplicate rows in RIPPLE.csv: 0
Missing values in Theter.csv: Date (0), Price (23), Volume (23), Market_cap (23)
Duplicate rows in Theter.csv: 0
Missing values in BNB.csv: Date (0), Price (2), Volume (22), Market_cap (75)
Duplicate rows in BNB.csv: 0
Missing values in SOLANA.csv: Date (0), Price (0), Volume (0), Market_cap (0)
Duplicate rows in SOLANA.csv: 0
Missing values in USDC.csv: Date (0), Price (0), Volume (0), Market_cap (0)
Duplicate rows in USDC.csv: 0
Missing values in DOGE.csv: Date (0), Price (1), Volume (1), Market_cap (2)
Duplicate rows in DOGE.csv: 0
Missing values in CARDANO.csv: Date (0), Price (0), Volume (8), Market_cap (25)
Duplicate rows in CARDANO.csv: 0
Missing values in Lido.csv: Date (0), Price (0), Volume (0), Market_cap (0)
Duplicate rows in Lido.csv: 0
Missing values in TRON.csv: Date (0), Price (0), Volume (16), Market_cap (73)
Duplicate rows in TRON.csv: 0
```

Figure 4: Output of Missing & Duplicate Values Check

The output demonstrated that many cryptocurrencies had high levels of missing values, with Bitcoin and Ripple having thousands and hundreds of missing values, respectively. Fortunately, it also demonstrated that there were no duplicate rows, so I could focus entirely on the missing values. I wanted to ensure a high-quality dataset with minimal noise and consistency, so I decided to duplicate the dataset files and in the new files drop every row with missing values through df.dropna(inplace=True).

After cleaning the dataset, I wanted to restructure it for optimal use. First, I cleaned the date column by converting the datapoints into actual datetime objects through the function df['Date'] = pd.to_datetime(df['Date'], utc=True, errors='coerce') for better manipulation during analysis. I also resorted the date column in ascending order so that the data is shown from the earliest dates to the most recent ones, as well as made Date the index of the dataset with df.set_index('Date', inplace=True) so that the other data columns are all organized directly by the date. I also implemented another check to ensure that all the values in the Price, Volume, and Market Capitalization columns are real numbers.

Data Normalization

To tackle the problem of the wide variance in the value ranges, I decided to use a normalization, which standardizes the values of all cryptocurrency data sets to be between 0 and 1 (0 is the minimum of the cryptocurrency and 1 is the maximum) so that the focus is not on the values themselves, but rather on the patterns and trends. To do this, I used the equation shown in Figure 5, in which *X* is the original numeric value in a column, *Xmin* is the minimum value in that column, *Xmax* is the maximum value in that column, and *X′* is the new normalized value in the column.

$$X' = \frac{X - Xmin}{Xmax - Xmin}$$

Figure 5: Normalization Equation

In the code, this was implemented through importing and using MinMaxScaler, a tool that scales all values in a column between 0 and 1 based on the relationship between the values. The normalized values were attained through MinMaxScaler().fit_transform(df[numeric_cols]), and were input into new columns in the duplicate dataset files (which already had the cleaned versions of the Price, Volume, and Market Capitalization columns) as Price_normalized, Volume_normalized, and Market_cap_normalized (shown in Figure 6 below). Once the new cleaned, transformed, and normalized files were complete, they were saved under “Cleaned_[cryptocurrency].csv” in an entirely new data folder.

Cleaned_TRON

Date	Price	Volume	Market_cap	Price_normalized	Volume_normalized	Market_cap_normalized
2017-11-09 00:00:00+00:00	0.00238682	1224287.170176	156404161.61769100	0.0013732423927244600	9.27736850222454E-05	0.0010417584357038600
2017-11-10 00:00:00+00:00	0.00204444	990422.8195	133968506.2492820	0.0005660546258981280	7.48099797722263E-05	0.0004294137357534310
2017-11-11 00:00:00+00:00	0.00191476	707642.964341	125470649.38402900	0.0002603238308691020	5.30889523730697E-05	0.00019747856261372500
2017-11-12 00:00:00+00:00	0.00180434	814789.232403	118235245.9032630	0.0	6.13191250032363E-05	0.0
2017-11-13 00:00:00+00:00	0.00201768	894985.567811	132386427.48493500	0.0005029658221120640	6.74792059528841E-05	0.00038623347066449900
2017-11-14 00:00:00+00:00	0.00241536	1073924.144644	158479487.413139	0.0014405276864484600	8.12239252433381E-05	0.0010984010757625200
2017-11-15 00:00:00+00:00	0.0023203	1179822.352577	152242301.026122	0.0012164162631336900	8.935823125584E-05	0.0009281672241554670
2017-11-16 00:00:00+00:00	0.00224758	1324833.2011	147470996.18371	0.0010449731461186400	0.00010049687694250700	0.0007979422236318490
2017-11-17 00:00:00+00:00	0.00202291	1905625.229589	132729588.66321	0.0005152959582780250	0.0001451089641681970	0.0003955994965413840
2017-11-18 00:00:00+00:00	0.00202237	2184964.124249	132693856.773722	0.0005140228658249430	0.0001665656827917650	0.000394624252770346
2017-11-19 00:00:00+00:00	0.00198527	1832646.818395	130259776.10292	0.0004265566991409760	0.00013950331001648700	0.00032818998225128200
2017-11-20 00:00:00+00:00	0.00213324	1679383.57715	139968958.501911	0.0007754076070716130	0.00012773077742043500	0.0005931863144398040
2017-11-21 00:00:00+00:00	0.00212957	1420954.846383	139727776.589602	0.0007667552935478890	0.00010788022074424300	0.0005866036466593790
2017-11-22 00:00:00+00:00	0.00233308	1345920.56312	153081053.186036	0.0012465461178566300	0.0001021166499208710	0.0009510595993173040
2017-11-23 00:00:00+00:00	0.00213762	3417832.230171	140256267.319897	0.000785733801413278	0.00026126536319587600	0.0006010279410405780

Figure 6: Example of Cleaned Cryptocurrency File (TRX - TRON)

Confirmation of High Quality Data

Once the entire data cleaning, transforming, and normalizing process was over, I ran the cleaned cryptocurrency files through the missing values and duplicate rows checking program and confirmed that the data had been properly cleaned. The output demonstrated that no missing values were remaining in any of the cryptocurrency files, as shown in Figure 7 below.

```
Missing values in Cleaned_BTC.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_BTC.csv: 0
Missing values in Cleaned_ETH.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_ETH.csv: 0
Missing values in Cleaned_RIPPLE.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_RIPPLE.csv: 0
Missing values in Cleaned_Theter.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_Theter.csv: 0
Missing values in Cleaned_BNB.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_BNB.csv: 0
Missing values in Cleaned_SOLANA.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_SOLANA.csv: 0
Missing values in Cleaned_USDC.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_USDC.csv: 0
Missing values in Cleaned_DOGE.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_DOGE.csv: 0
Missing values in Cleaned_CARDANO.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_CARDANO.csv: 0
Missing values in Cleaned_Lido.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_Lido.csv: 0
Missing values in Cleaned_TRON.csv: Date (0), Price (0), Volume (0), Market_cap (0), Price_normalized (0), Volume_normalized (0), Market_cap_normalized (0)
Duplicate rows in Cleaned_TRON.csv: 0
```

Figure 7: Output of Missing & Duplicate Values Check (Using Cleaned Files)

(II) Data Visualization

High Quality Data Visualization

With the data entirely cleaned and converted into a high-quality set, I could begin the analysis. First, I re-visualized the data to demonstrate the impact of the cleaning, transforming, and normalizing process and to better analyze the data before beginning to evaluate ways of detecting trends and patterns. I first re-plotted the cleaned Price, Volume, and Market Capitalization data using the same 'sequential removal by size' strategy that I had used for the initial data visualization, as demonstrated in Figure 8, Figure 9, and Figure 10 below.

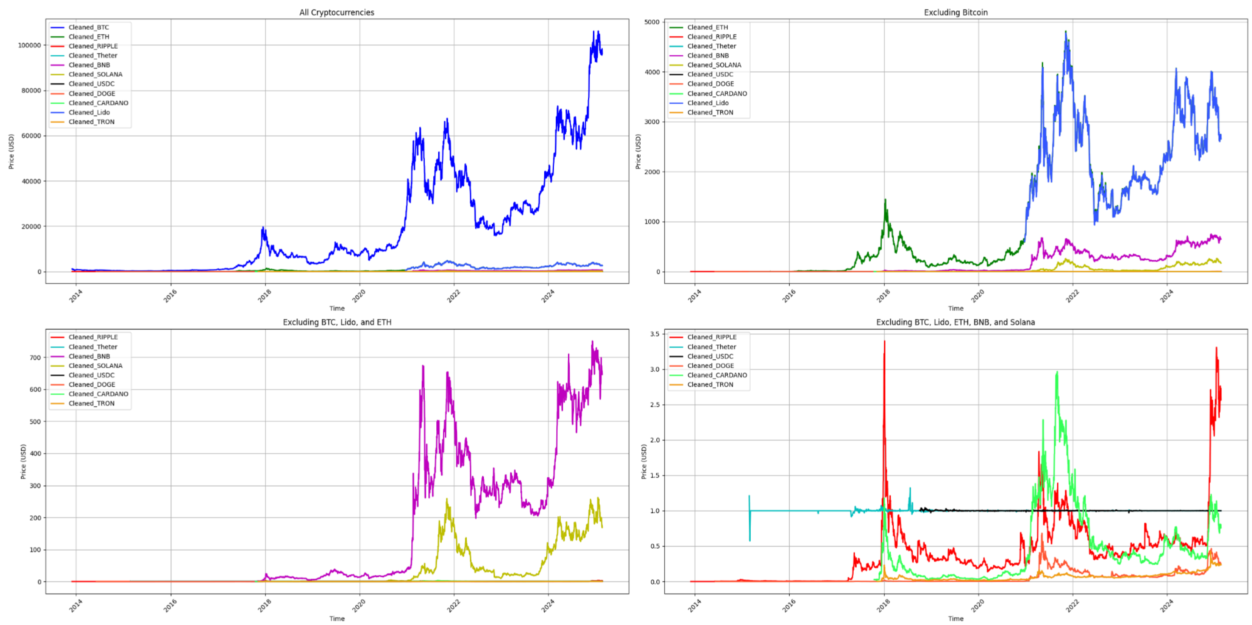


Figure 8: Cleaned Data Visualization (Price)

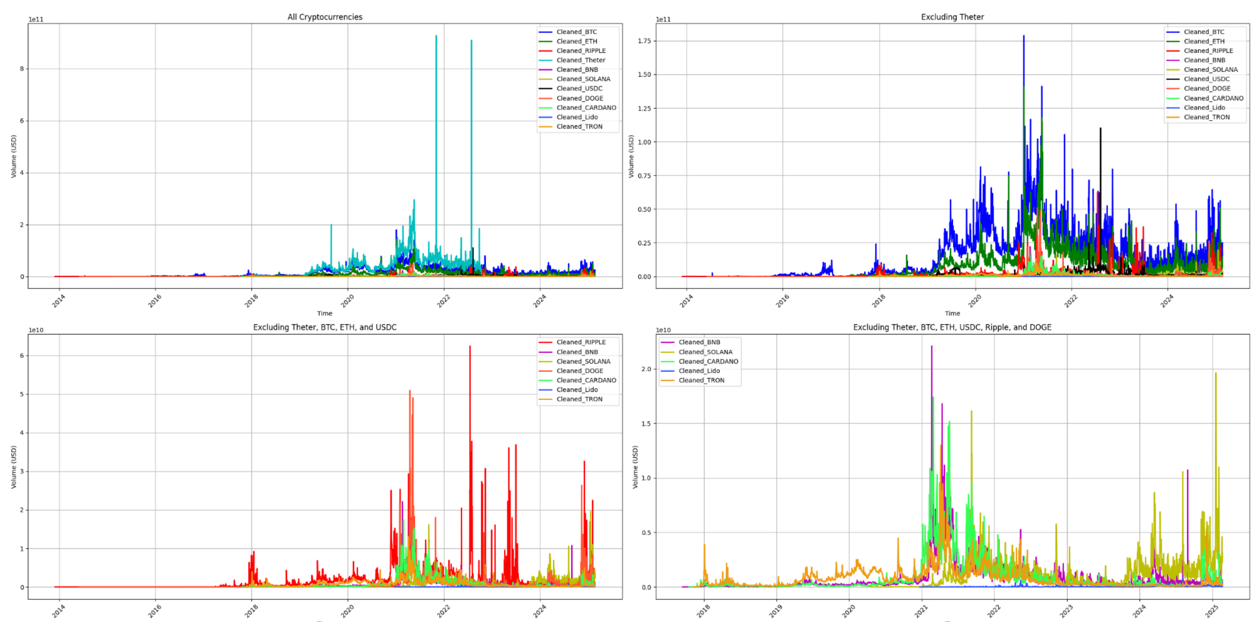


Figure 9: Cleaned Data Visualization (Volume)

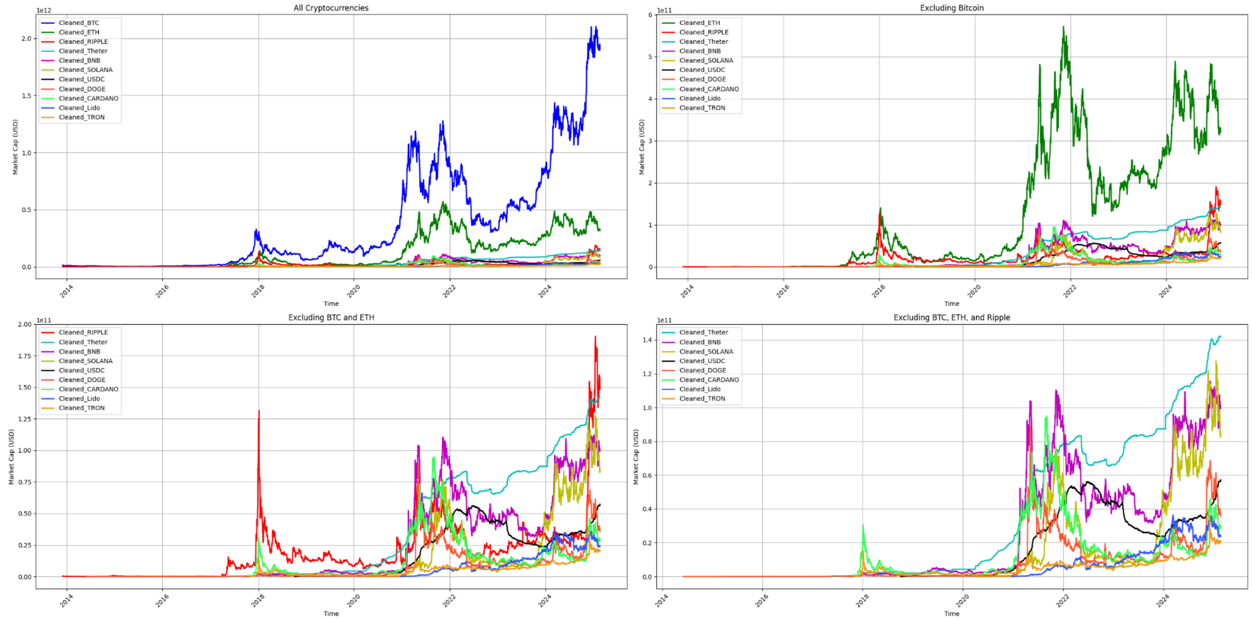


Figure 10: Cleaned Data Visualization (Market Capitalization)

Particularly for the Price plots, the data was significantly better formatted for machine learning modeling on the entire dataset. The dates across the cryptocurrencies were now more aligned and consistent, which facilitated making observations on the market of the eleven cryptocurrencies as a whole. The cleaned but not normalized plots are also effective in showing the similarities in some basic trends, even though they have incredibly different value ranges. For example, in the Market Capitalization plots, there is a clear pattern of peaks around 2022 for many of the currencies, followed by a dip into 2023 and a later increase to another peak around 2025. These very basic trends were much easier to identify with the cleaned data that limited some of the noise and inconsistency from earlier. However, it was still difficult to identify patterns due to the vast difference in value range that persisted. Fortunately, I hadn't yet plotted the normalized data, which is more focused on demonstrating trends and patterns do to its elimination of the role of the value range. The normalization plots are below in Figure 11, Figure 12, and Figure 13.

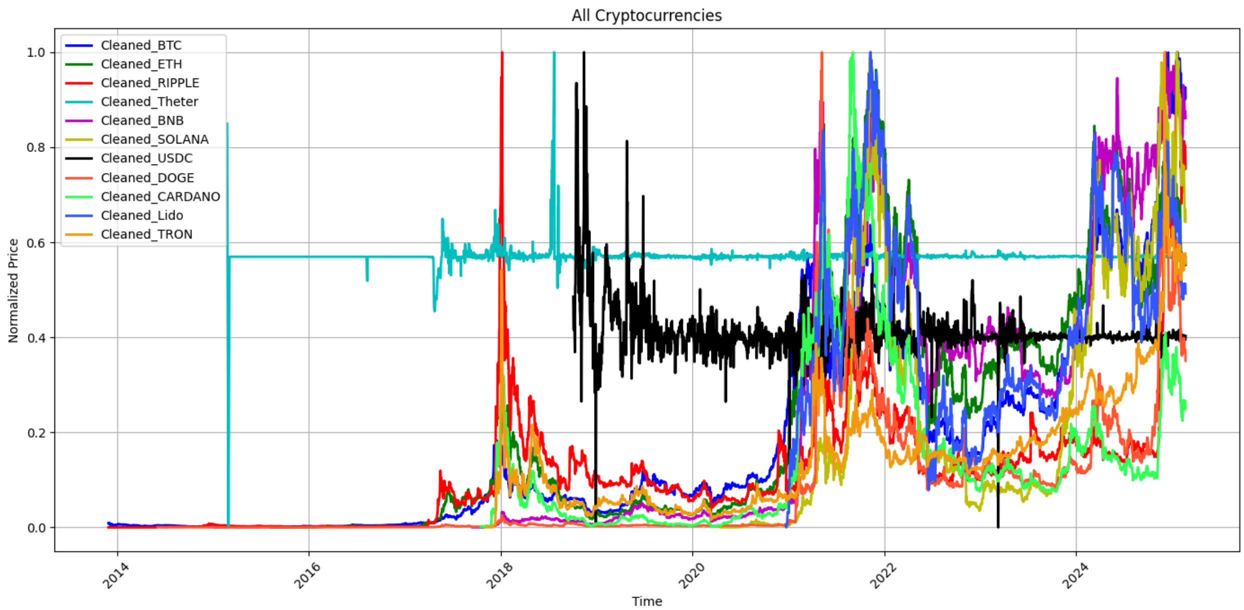


Figure 11: Normalized Data Visualization (Price)

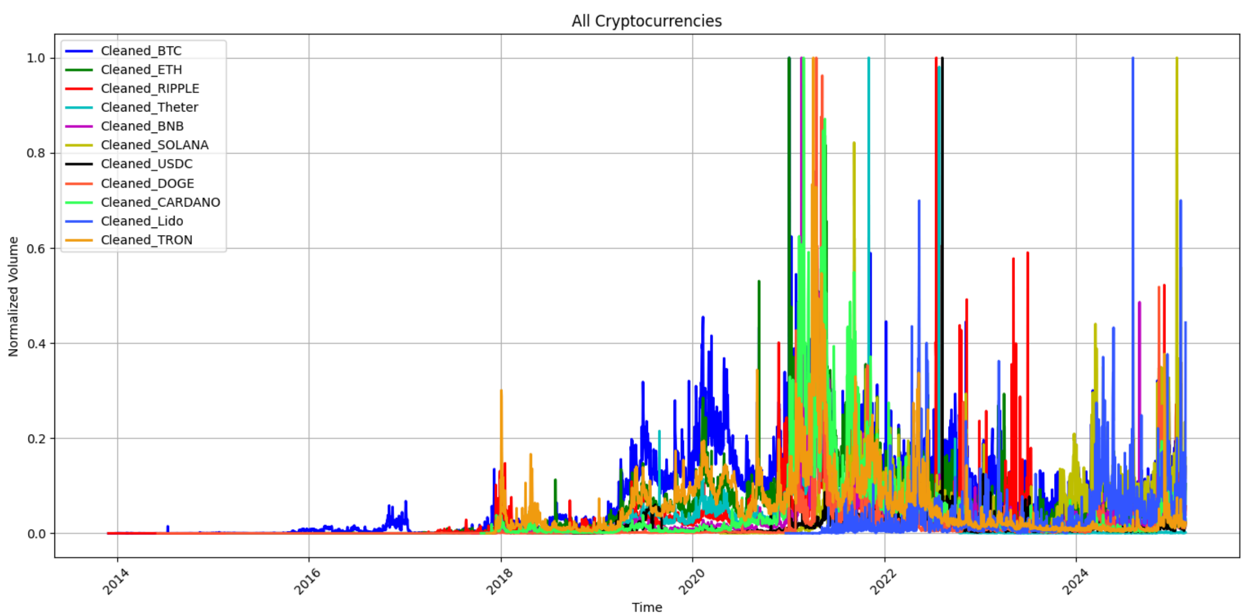


Figure 12: Normalized Data Visualization (Volume)

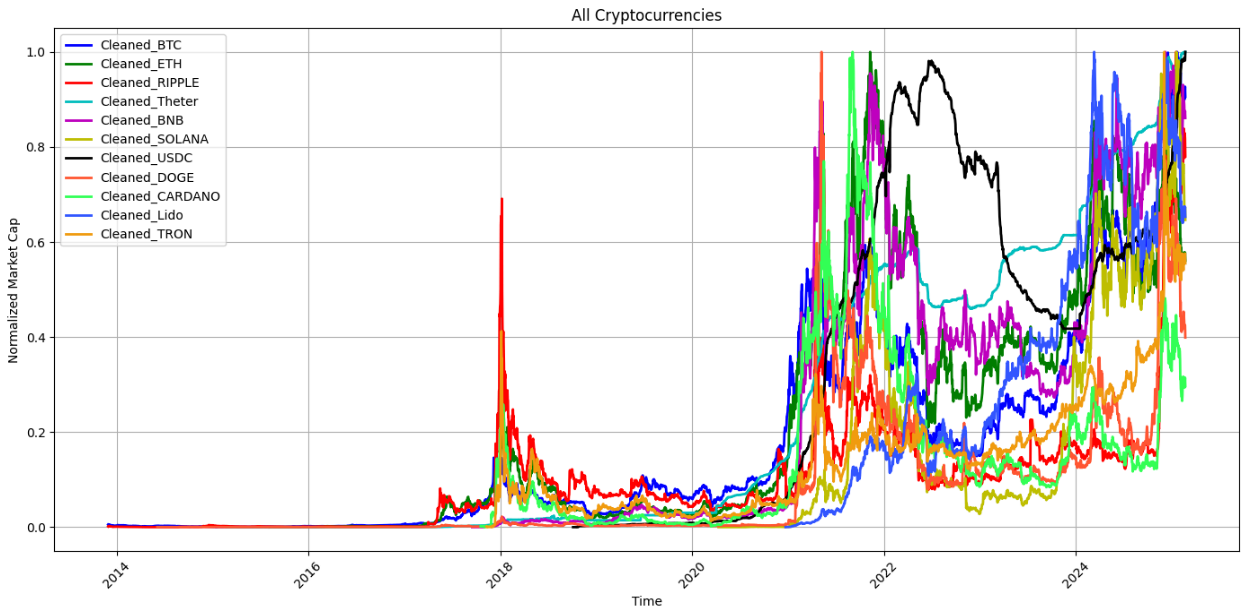


Figure 13: Normalized Data Visualization (Market Capitalization)

Since the problem of value range was now eliminated from plotting, I decided to remove the sequence-based plotting I did before and instead focus solely on all cryptocurrencies plotted together. However, another plotting problem popped up: the Cleaned_Thether and Cleaned_USDC datasets are based on Tether and USDC, which are cryptocurrency stablecoins that are pegged to the United States dollar, meaning that they are designed to remain stable and consistently be worth around one dollar ($1). Therefore, the activities of these currencies in the Price and Market Capitalization plots were completely different from the rest, and due to their distinction as stablecoins, I decided to recreate the Price and Market Capitalization plots with the stablecoins removed, as shown below in Figure 14 and Figure 15.

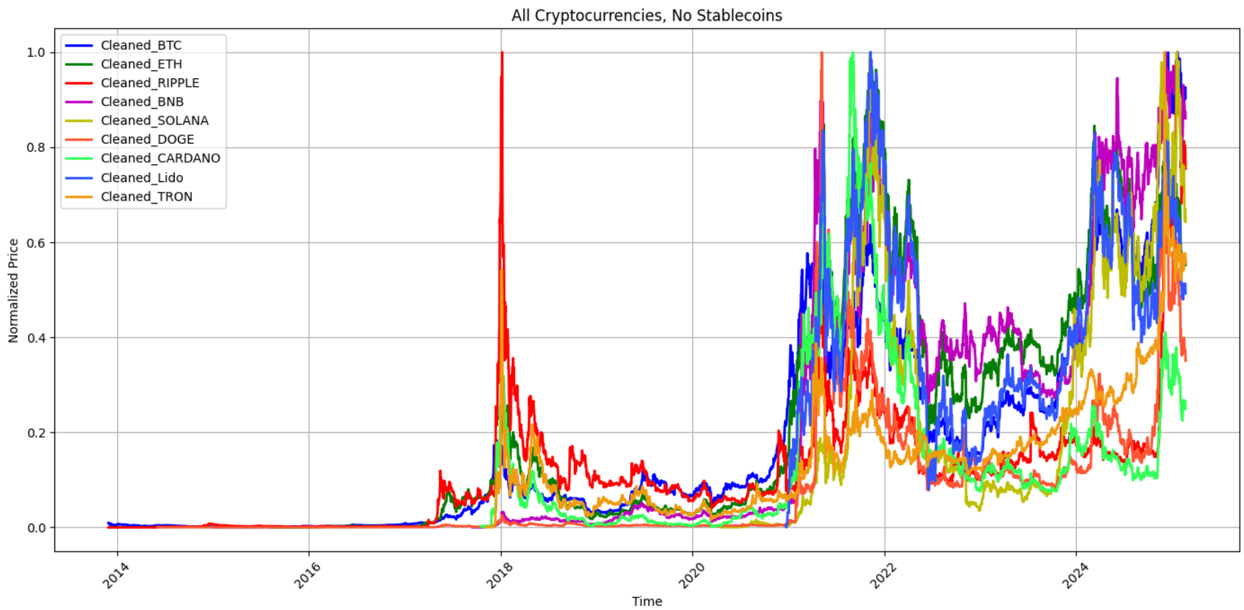


Figure 14: Normalized Data Visualization, No Stablecoins (Price)

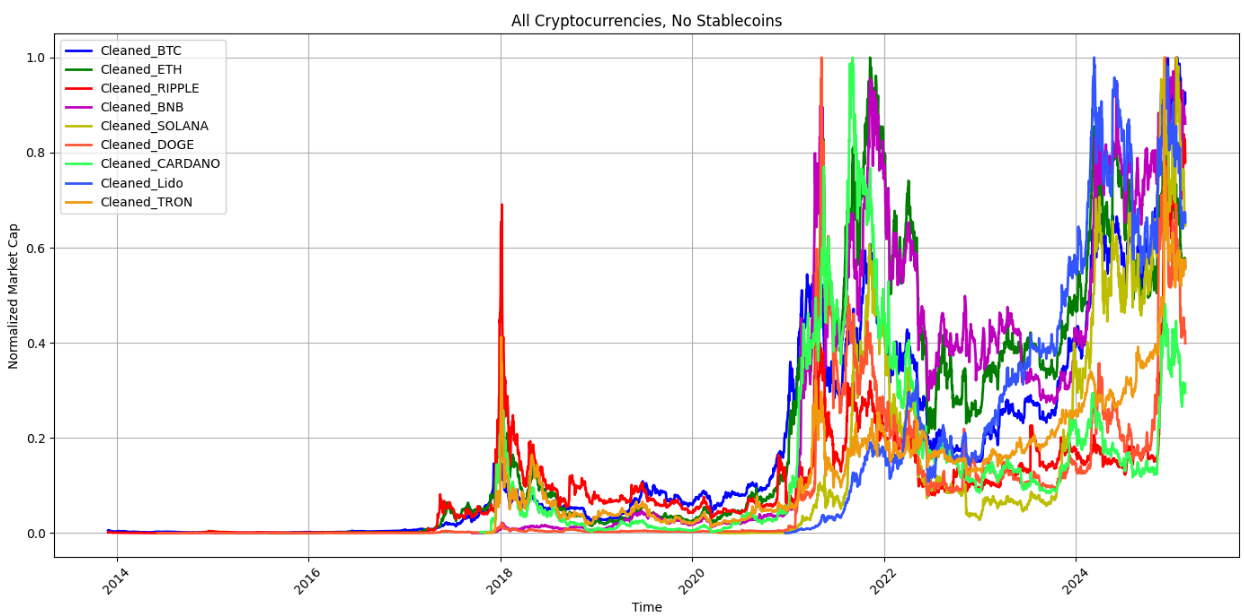


Figure 15: Normalized Data Visualization, No Stablecoins (Market Capitalization)

After removing the stablecoins, the patterns in the non-stablecoin cryptocurrencies are far easier to identify, with the pattern described earlier taking full effect. Many cryptocurrencies experienced a small peak in 2018, followed by a stabilization into 2020, a major peak in 2022, a re-stabilization in 2023 (with much higher values than in the 2020 stabilization), and another peak past 2024 and into 2025, for both the Price and Market Capitalization plots. Figure 14 and Figure 15 demonstrate the incredibly strong correlation between Price and Market Capitalization, which are evidently the most impactful data points for accurate prediction modeling.

(III) Predictive Modeling

Early Testing: Strengths of ARIMA & LSTM

The first part of the predictive modeling involved in this project required a thorough exploration and evaluation of two major time-series forecasting methods: Autoregressive Integrated Moving Average (ARIMA) models and Long Short-Term Memory (LSTM) neural networks. ARIMA models are a foundational method in classical statistical time-series analysis, and are known for their high ability to model linear dependencies within data. They have three key components, which are the autoregressive (AR) part (which uses the relationship between an observation and a number of lagged observations), the integrated (I) part (which involves differencing observations to make the time series stationary), and the moving average (MA) part (which models the dependency between an observation and a residual error from a moving average model applied to lagged observations). The primary strengths of the ARIMA models are their interpretability, relative simplicity, and effectiveness for time series exhibiting clear stationarity and linear patterns. The project's initial tests with ARIMA involved defining optimal p, d, and q parameters, training the model on historical data, and later evaluating its performance using metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE). These metrics provided foundational quantitative assessments of the model's ability to forecast future price movements. ARIMA served as the essential baseline, which demonstrated the potential for linear relationships within the cryptocurrency data, though its limitations in capturing complex, non-linear dynamics were also noted, as demonstrated in Figure 16.

Figure 16: Price Forecasting Using ARIMA Model

On the other hand, Long Short-Term Memory (LSTM) networks, a specialized architecture within the recurrent neural networks (RNNs), offers a much more sophisticated approach to the predictive modeling problem. Notably, the LSTM model is particularly suited for the inherent complexities of financial time series like cryptocurrency prices. Moreover, unlike ARIMA, LSTMs are designed to learn and retain information over long sequences, which mitigates the vanishing gradient problem that is common in simpler RNNs. This capability is extremely important in cryptocurrency markets, where the price fluctuations are often influenced by interconnected factors, market sentiment shifts, and macroeconomic events that unfold over extended periods and can prove to be highly complex to predict. The unique gating mechanism within LSTM cells (input, forget, and output gates) allows the network to selectively remember or forget information, which enables it to model intricate, non-linear dependencies and long-term patterns that traditional linear models cannot (demonstrated in Figure 17 below). Early testing indicated that while the ARIMA model could identify some very basic trends for a fraction of the cryptocurrencies, the LSTM model was far more effective and captured more nuanced and volatile behaviors, offering a more robust framework for predictive modeling in the dynamic environments of cryptocurrency trading.

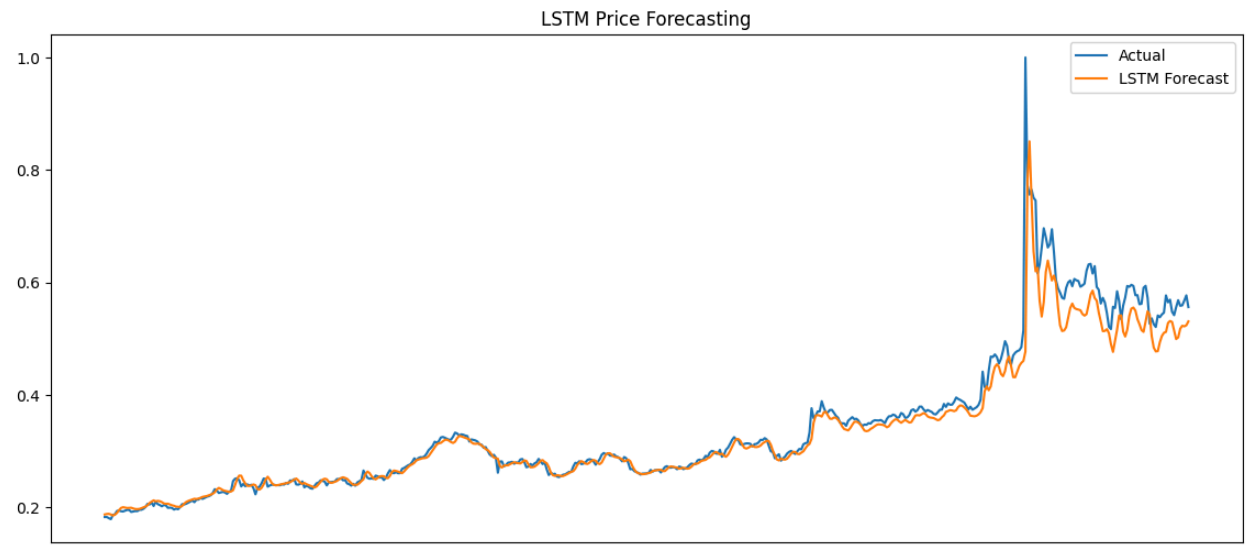


Figure 17: Price Forecasting Using LSTM Model

(IV) Trading Strategy

Model Selection: ARIMA vs. LSTM

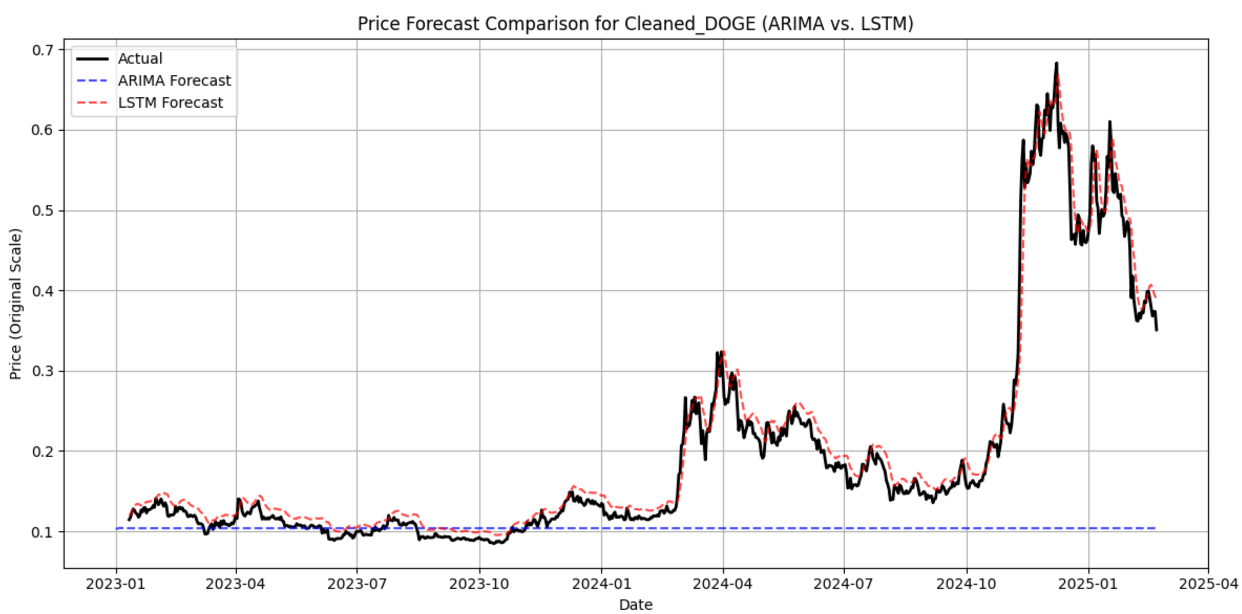
The selection between ARIMA and LSTM for the central predictive modeling component of the trading strategy was a critical decision, and it was heavily influenced by the distinct characteristics of cryptocurrency markets that require specific modeling. While linear regression and ridge regression were explored as initial statistical models (after their use in the previous project), their inherent linearity posed significant limitations in capturing the complex, non-linear, and volatile patterns that define many of the cryptocurrency price movements. Similarly, ARIMA (and the other simpler models) is best suited for data with clear linear relationships and stationarity. However, the financial time series in the crypto field rarely reflect those ideal conditions, and instead exhibit erratic and long-term dependencies influenced by external factors and sentiment. Therefore, the decision to pivot towards more advanced deep learning models like LSTM was motivated by the necessity to address these complexities and properly model the inherently complex nature of cryptocurrency. LSTM's architectural design, specifically its memory cells and gating mechanisms, allows it to effectively learn and remember intricate, non-linear relationships over extended sequences of data, which is demonstrated in Figure 18.

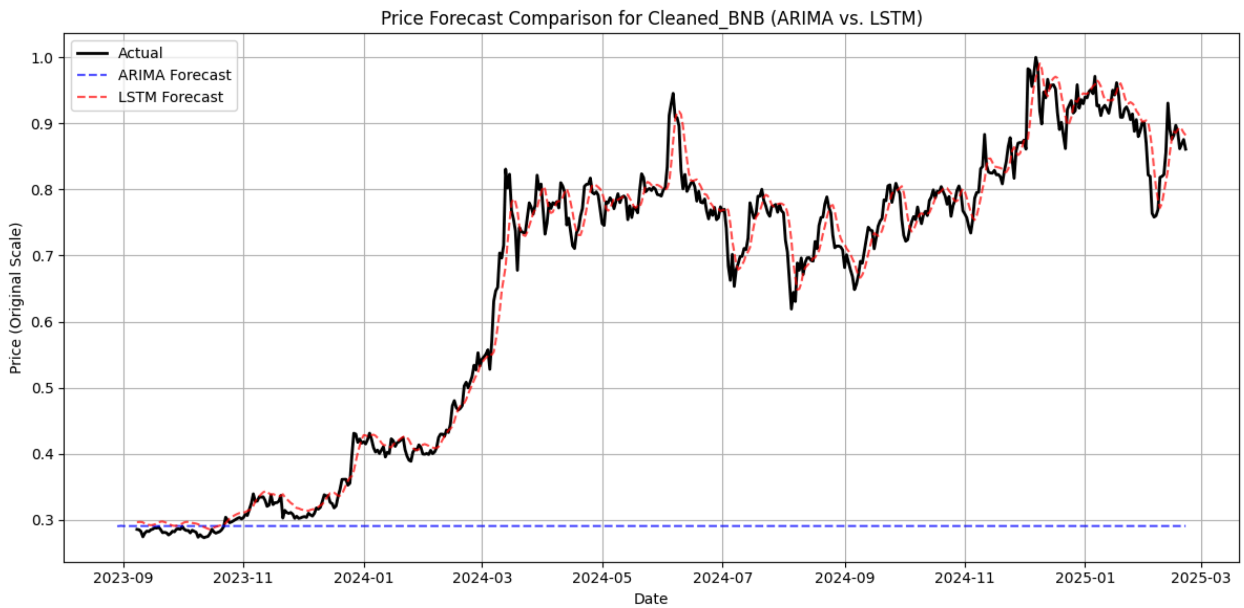
--- Model Performance Summary ---

	Crypto	ARIMA_MSE	ARIMA_MAE	ARIMA_R2	LSTM_MSE	LSTM_MAE	LSTM_R2
0	Cleaned_DOGE	0.026739	0.093106	-0.431435	0.000459	0.015219	0.975604
1	Cleaned_BNB	0.175890	0.354638	-2.434827	0.000841	0.020265	0.983096
2	Cleaned_Lido	0.013224	0.097141	-0.069740	0.001817	0.030616	0.857486
3	Cleaned_ETH	0.053352	0.182146	-1.483879	0.001146	0.027260	0.946515
4	Cleaned_Theter	0.000003	0.000952	-0.000634	0.000004	0.001645	-0.460884
5	Cleaned_CARDANO	0.014914	0.092911	-1.349650	0.000238	0.010813	0.962250
6	Cleaned_SOLANA	0.044051	0.165996	-1.651958	0.001697	0.031649	0.897759
7	Cleaned_BTC	0.150767	0.308406	-1.707973	0.000316	0.012269	0.994271
8	Cleaned_RIPPLE	0.046539	0.107560	-0.324093	0.000939	0.013137	0.973525
9	Cleaned_USDC	0.000068	0.005056	-0.009041	0.000093	0.006664	-0.340045
10	Cleaned_TRON	0.041202	0.157051	-1.490566	0.002319	0.027215	0.858676

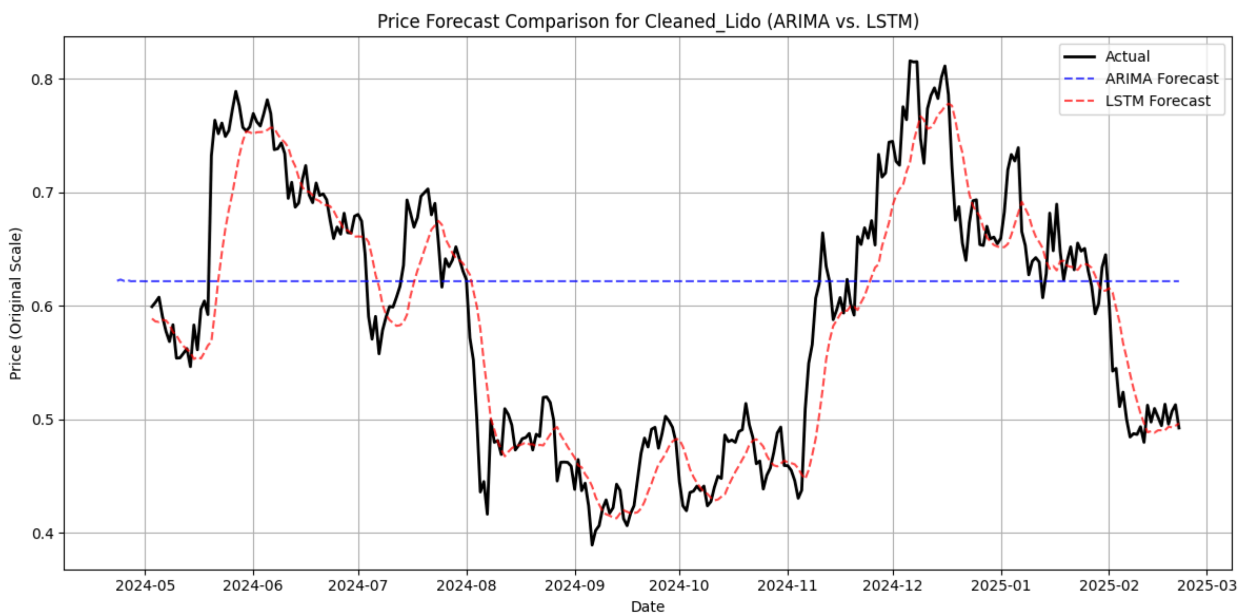
Figure 18: Model Performance Comparison (ARIMA and LSTM)

This ability to model long-term dependencies and adapt to sudden shifts in market dynamics is of utmost importance for accurate forecasting in a highly unpredictable market like cryptocurrency. While ARIMA offered a foundational understanding of temporal relationships, LSTM's capability in handling non-linearity and complex temporal patterns effectively made it the preferred model for developing a robust and adaptable trading strategy. Below in Figure 19, the success of the LSTM models and the shortcomings of the ARIMA models is demonstrated.

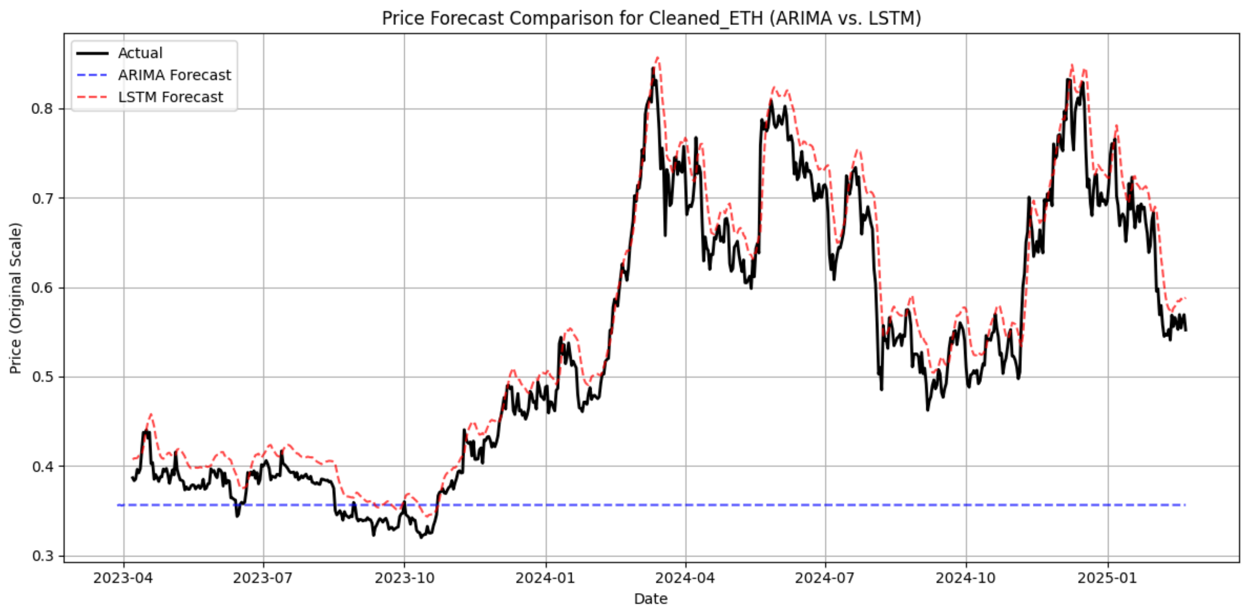


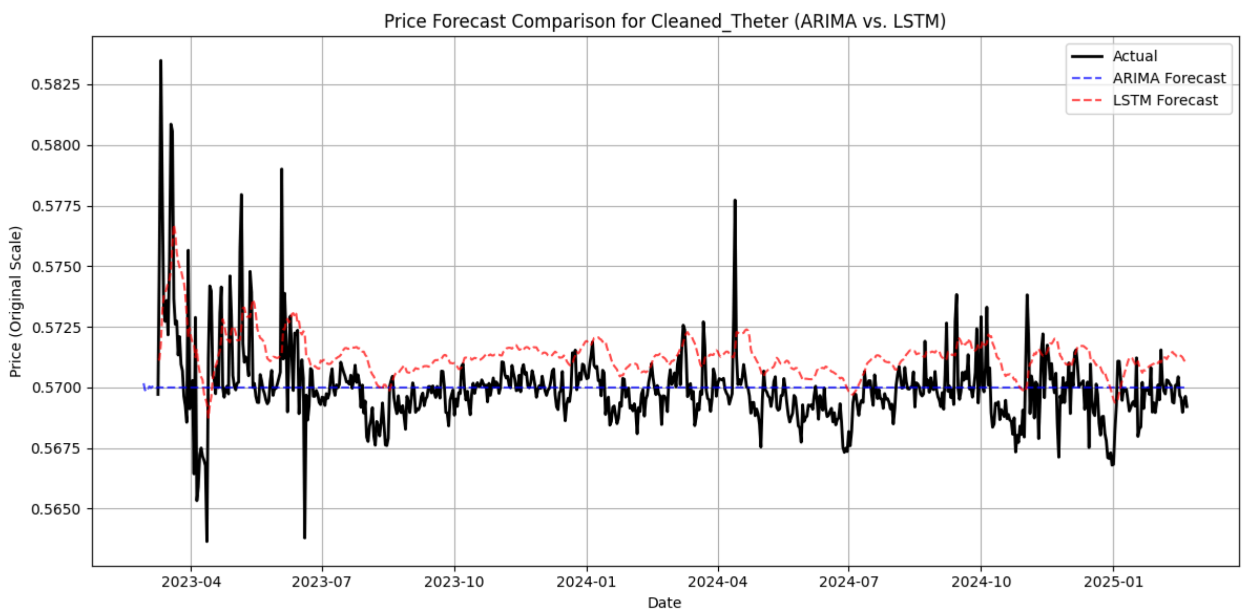


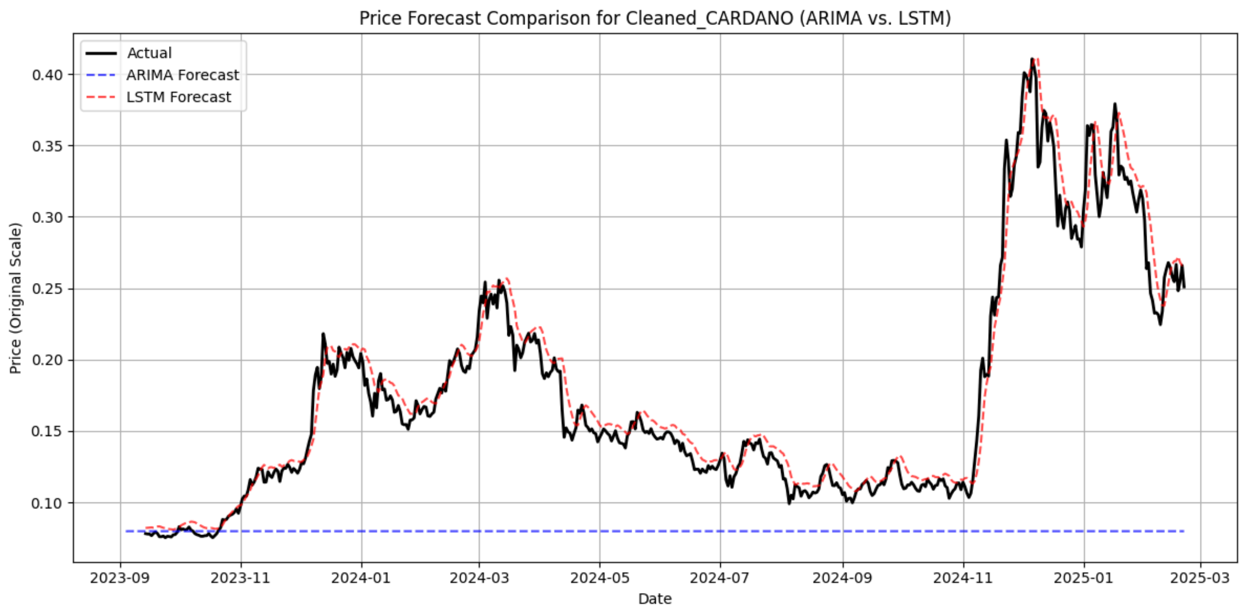
Price Forecast Comparison for Cleaned_BNB (ARIMA vs. LSTM)
Actual
ARIMA Forecast
LSTM Forecast
Price (Original Scale)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
2023-09
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01
2025-03
Date

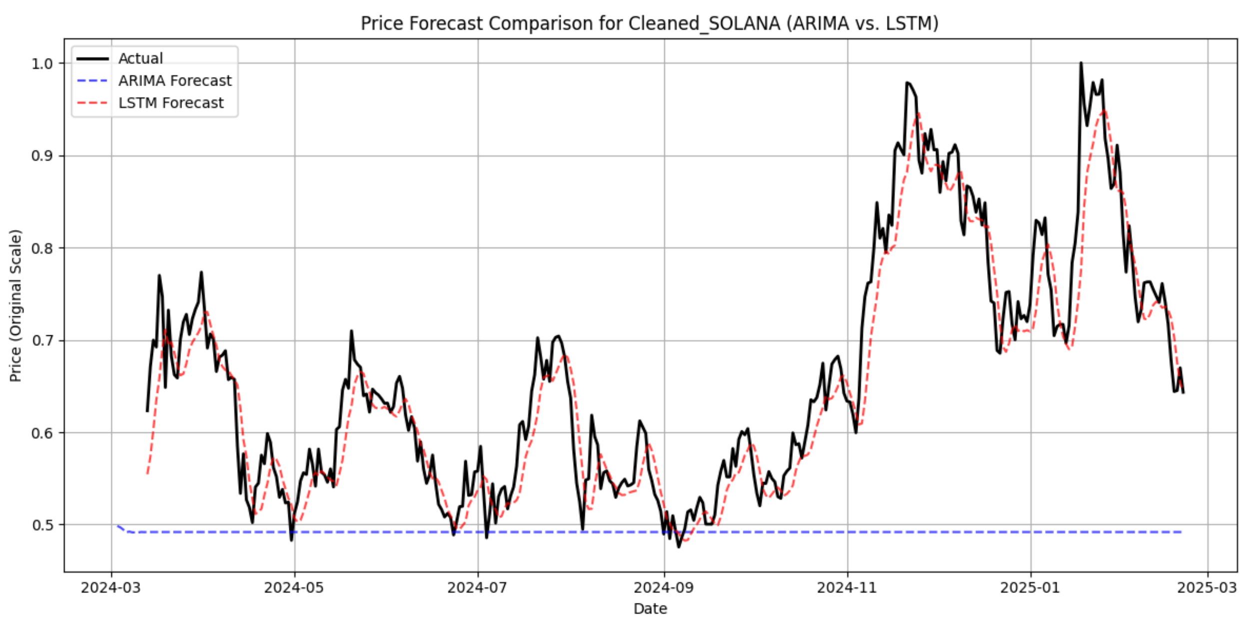


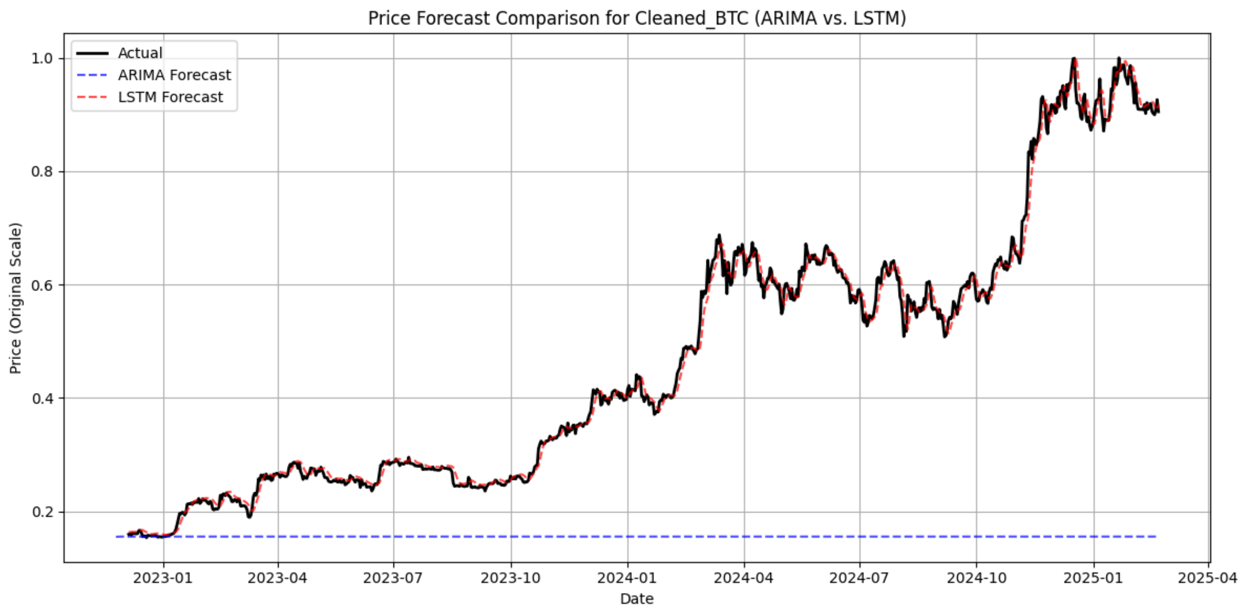
Price Forecast Comparison for Cleaned_Lido (ARIMA vs. LSTM)
Actual
ARIMA Forecast
LSTM Forecast
Price (Original Scale)
0.8
0.7
0.6
0.5
0.4
2024-05
2024-06
2024-07
2024-08
2024-09
2024-10
2024-11
2024-12
2025-01
2025-02
2025-03
Date

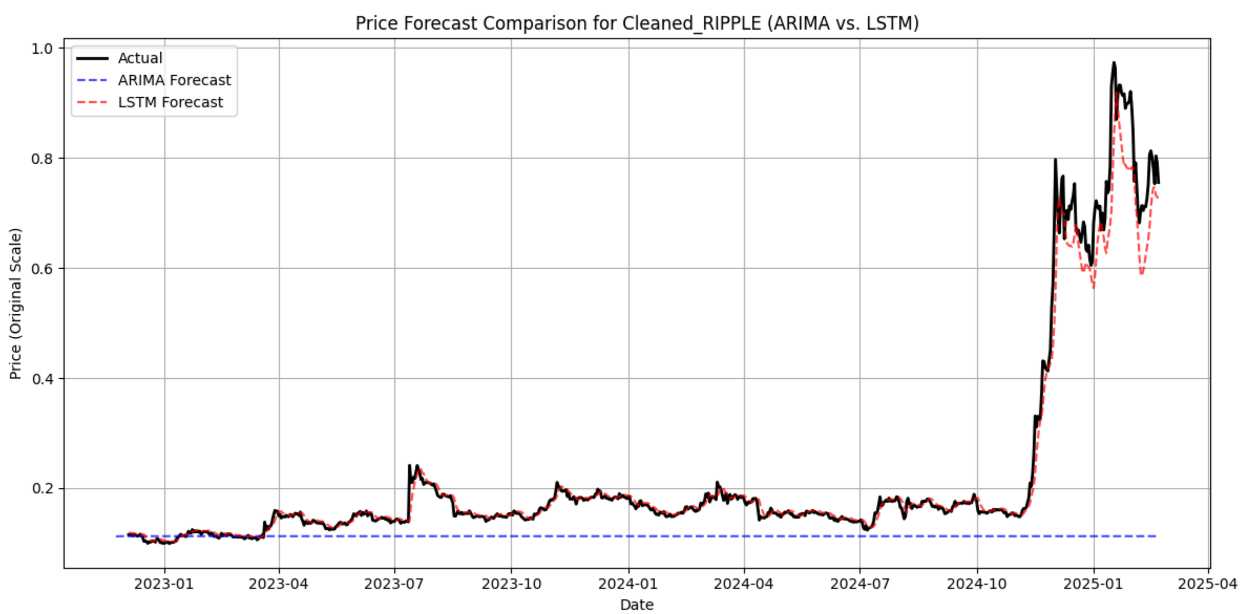

Price Forecast Comparison for Cleaned_ETH (ARIMA vs. LSTM)
Actual
ARIMA Forecast
LSTM Forecast
Price (Original Scale)
0.8
0.7
0.6
0.5
0.4
0.3
2023-04
2023-07
2023-10
2024-01
2024-04
2024-07
2024-10
2025-01
Date

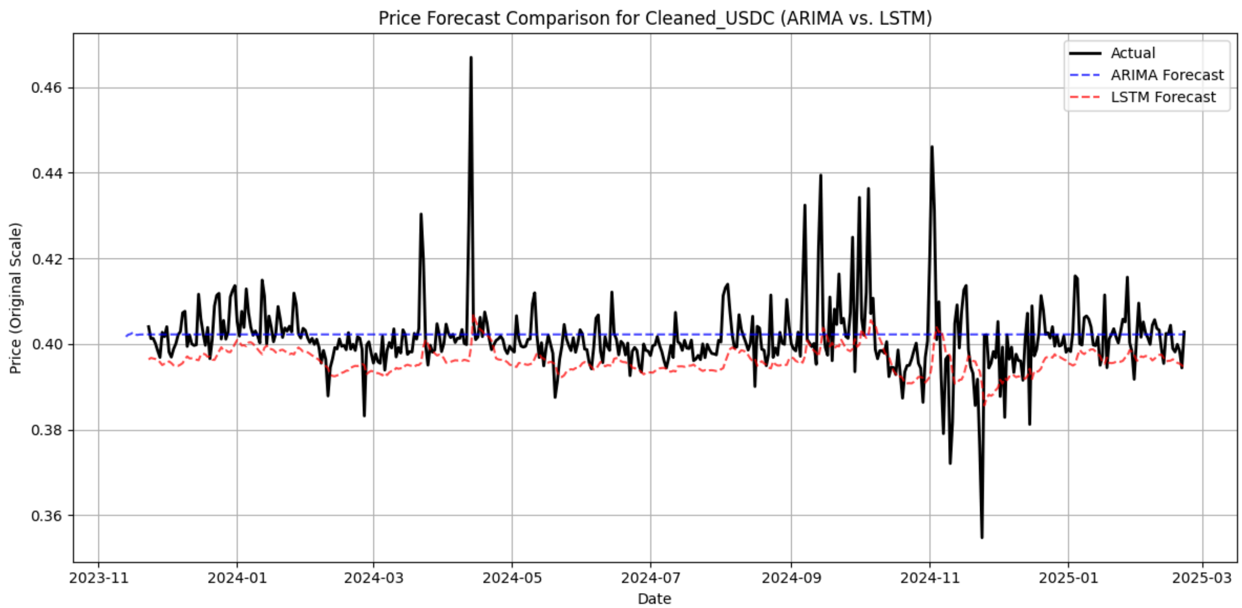

Price Forecast Comparison for Cleaned_Theter (ARIMA vs. LSTM)
Actual
ARIMA Forecast
LSTM Forecast
Price (Original Scale)
0.5825
0.5800
0.5775
0.5750
0.5725
0.5700
0.5675
0.5650
2023-04
2023-07
2023-10
2024-01
2024-04
2024-07
2024-10
2025-01
Date


Price Forecast Comparison for Cleaned_CARDANO (ARIMA vs. LSTM)
Actual
ARIMA Forecast
LSTM Forecast
Price (Original Scale)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
2023-09
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01
2025-03
Date


Price Forecast Comparison for Cleaned_SOLANA (ARIMA vs. LSTM)
Actual
ARIMA Forecast
LSTM Forecast
Price (Original Scale)
1.0
0.9
0.8
0.7
0.6
0.5
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01
2025-03
Date


Price Forecast Comparison for Cleaned_BTC (ARIMA vs. LSTM)
Actual
ARIMA Forecast
LSTM Forecast
Price (Original Scale)
1.0
0.8
0.6
0.4
0.2
2023-01
2023-04
2023-07
2023-10
2024-01
2024-04
2024-07
2024-10
2025-01
2025-04
Date


Price Forecast Comparison for Cleaned_RIPPLE (ARIMA vs. LSTM)
Actual
ARIMA Forecast
LSTM Forecast
Price (Original Scale)
1.0
0.8
0.6
0.4
0.2
2023-01
2023-04
2023-07
2023-10
2024-01
2024-04
2024-07
2024-10
2025-01
2025-04
Date



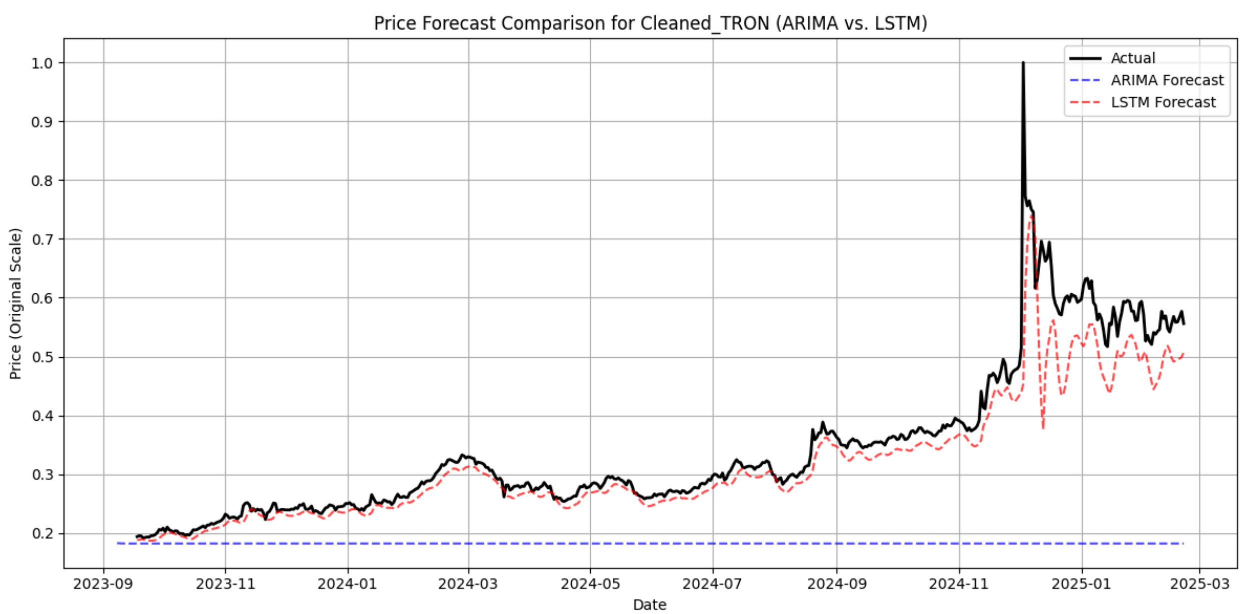


Figure 19: ARIMA vs. LSTM Graphs of All 11 Cryptocurrencies

Predictive Modeling Using LSTM

The implementation of LSTM for predictive modeling began with a thorough data preparation phase, tailored to optimize the network's learning capabilities. The cleaned and normalized cryptocurrency datasets were transformed into sequential formats required by LSTM architectures. This involved creating input sequences (X) and corresponding target values (y), where X typically comprised a look-back window of historical data (ex. the past 30 days of price and market capitalization), and y represented the future value to be predicted (ex. the price or market capitalization for the next day, week, month, or year). The data was then reshaped into a 3D array (samples, timesteps, features), which is the standard input format for Keras/TensorFlow LSTM layers. This transformation enabled the model to learn from the temporal context and sequential dependencies that were inherent in the data.

The LSTM model architecture was then carefully designed to effectively capture these complex patterns. It typically incorporated multiple LSTM layers, often followed by Dense (fully connected) layers for output. Dropout layers were also included between LSTM layers to prevent overfitting by randomly dropping units during training, enhancing the model's generalization capabilities. The model was compiled using the Adam optimizer, which is known for its efficiency in handling large datasets and non-stationary objectives, and Mean Squared Error (MSE) as the loss function, which is a standard choice for regression tasks. The training process involved fitting the model to the training data over a defined number of epochs, or iterations, and with a specified batch_size, allowing the network to adjust its internal weights to minimize prediction errors as it iterated through the model.

Upon completion of training, the LSTM model was used to generate predictions on some unseen test data that was set aside. Inverting the min-max normalization that was applied during the data cleaning process was a crucial step here. Since the model was trained on scaled values (between 0 and 1), the predictions were also in this normalized range. The MinMaxScaler was used to reverse this scaling, transforming the predictions back into their original price and market capitalization units, which made them directly comparable and interpretable against actual market values. The performance of the LSTM predictions was rigorously evaluated using quantitative metrics such as MSE and MAE, providing precise measures of prediction accuracy. Additionally, extensive visualizations, plotting the actual price and market capitalization against the LSTM-predicted values, were critical for qualitative assessment. These plots consistently demonstrated the LSTM model's superior ability to track and predict the nuanced, volatile movements characteristic of cryptocurrency markets, often outperforming simpler linear models and validating its selection as the primary predictive tool.

Trading Strategy

The trading strategy developed in this project is built upon the well-established Moving Average Crossover Strategy, a common technical analysis tool used to generate buy and sell signals. This strategy uses the smoothed data provided by moving averages to identify potential shifts in market momentum. The core concept involves two moving averages: a short-term moving average (SMA) and a long-term moving average (LMA). The notebook illustrates this by defining a class named MovingAverageCrossoverStrategy which calculates these averages for a selected cryptocurrency. A buy signal is generated when the short-term moving average crosses above the long-term moving average since it indicates that the asset's recent price action is stronger than its long-term trend and suggests an upward momentum. Conversely, a sell signal is triggered when the short-term moving average crosses below the long-term moving average, which implies that the recent price action is weaker and suggests a potential downward trend or reversal. The strategy's implementation involved iterating through the cryptocurrency data and calculating the daily returns based on these buy and sell signals. If a buy signal was active, the daily percentage change in price was considered a gain; if a sell signal was active, the position was exited or shorted (though primarily focused on long positions in this context). The accumulated effect of these daily returns then formed the basis for evaluating the simulated portfolio's performance. This simulation allowed for a direct assessment of how the strategy would have performed in comparison to the actual historical data.

	Crypto	Sharpe	Max_Drawdown	Cumulative_Return	Recommendation	Confidence_%	Risk_Level
0	Cleaned_DOGE	0.056093	0.368118	1.932690	SELL	63	High
1	Cleaned_BNB	0.098312	0.208496	1.700575	SELL	21	High
2	Cleaned_Lido	-0.012926	0.389980	-0.209570	HOLD	7	Low
3	Cleaned_ETH	0.024886	0.321902	0.234750	HOLD	5	High
4	Cleaned_Theter	-0.117156	0.176722	-0.156823	HOLD	1	Low
5	Cleaned_CARDANO	0.090496	0.216595	2.825371	SELL	22	High
6	Cleaned_SOLANA	0.046899	0.320800	0.413244	SELL	100	High
7	Cleaned_BTC	0.083240	0.187225	2.150915	HOLD	9	High
8	Cleaned_RIPPLE	0.046826	0.328623	1.192574	SELL	62	High
9	Cleaned_USDC	-0.161620	0.715961	-0.715961	HOLD	7	Low
10	Cleaned_TRON	0.062664	0.458315	2.053538	BUY	32	High

Figure 20: Quantitative Evaluations and Forecasting for All Cryptocurrencies

To properly and quantitatively evaluate the effectiveness of this trading strategy, several key portfolio metrics were calculated, as illustrated above in Figure 20. These included cumulative returns, which illustrate the total growth of the investment over the simulation period and provide a clear visual representation of profitability. The mean daily return (not shown above but present in the code) provided an average daily profit or loss generated by the strategy. Volatility (also present in the code), was measured as the standard deviation of daily returns, and served as a crucial indicator of the strategy's risk level; higher volatility implies greater price fluctuations and thus higher risk. The Sharpe Ratio, a very important component of risk-adjusted performance, was calculated to assess the return earned per unit of risk taken, allowing for a standardized comparison of the strategy's efficiency. Finally, the Maximum Drawdown, representing the largest percentage drop from a peak in equity to a subsequent trough, was calculated to highlight the worst-case scenario and the potential capital at risk during adverse market conditions. These metrics, alongside the visualization of cumulative portfolio returns, provided a comprehensive framework for understanding the strategy's profitability, risk profile, and overall viability for investment decisions in the cryptocurrency market.

(V) Risk Management & Performance Evaluation

Identification of Risks and Performance Evaluations of Predictions

In the world of financial modeling and particularly within the highly speculative cryptocurrency market, the identification of risks and comprehensive performance evaluation of predictions are extremely important. This final stage of the project included the transition from predictive accuracy to assessing the practical viability and financial implications of deploying the generated forecasts and predictions within a trading strategy. The evaluation process focused on quantifying the risk-reward profile of the strategy derived from the predictive models, providing a holistic understanding beyond simple forecasting errors.

Key portfolio metrics were calculated to achieve this comprehensive assessment. The Mean Daily Return offered a straightforward measure of the strategy's average daily profitability. A higher positive mean daily return indicates a consistently profitable approach, but this alone is insufficient to gauge true performance. To contextualize returns, Volatility, quantified as the standard deviation of daily returns, served as a primary indicator of risk. A higher volatility signifies greater fluctuations in portfolio value, implying a more unpredictable and potentially riskier investment. Managing volatility is a significant part of the process because it ensures the stability and sustainability of any trading strategy.

The Sharpe Ratio stood out throughout this project as a particularly critical metric for evaluating risk-adjusted performance. This ratio calculates the return generated per unit of risk taken, specifically the excess return over a risk-free rate divided by the standard deviation of the strategy's returns. A higher Sharpe Ratio implies that the strategy is delivering superior returns for the level of risk assumed, making it an invaluable tool for comparing the efficiency and attractiveness of different investment approaches (see Figure 21). This metric transforms raw returns into a measure of investment quality. Furthermore, the Maximum Drawdown (Max Drawdown) provided essential insight into potential capital at risk. It represents the largest percentage decline from a peak in portfolio value to a subsequent trough before a new peak is reached. This metric is vital for understanding the worst-case scenario an investor might face, reflecting the resilience of the strategy during adverse market conditions. A lower maximum drawdown suggests better capital preservation.

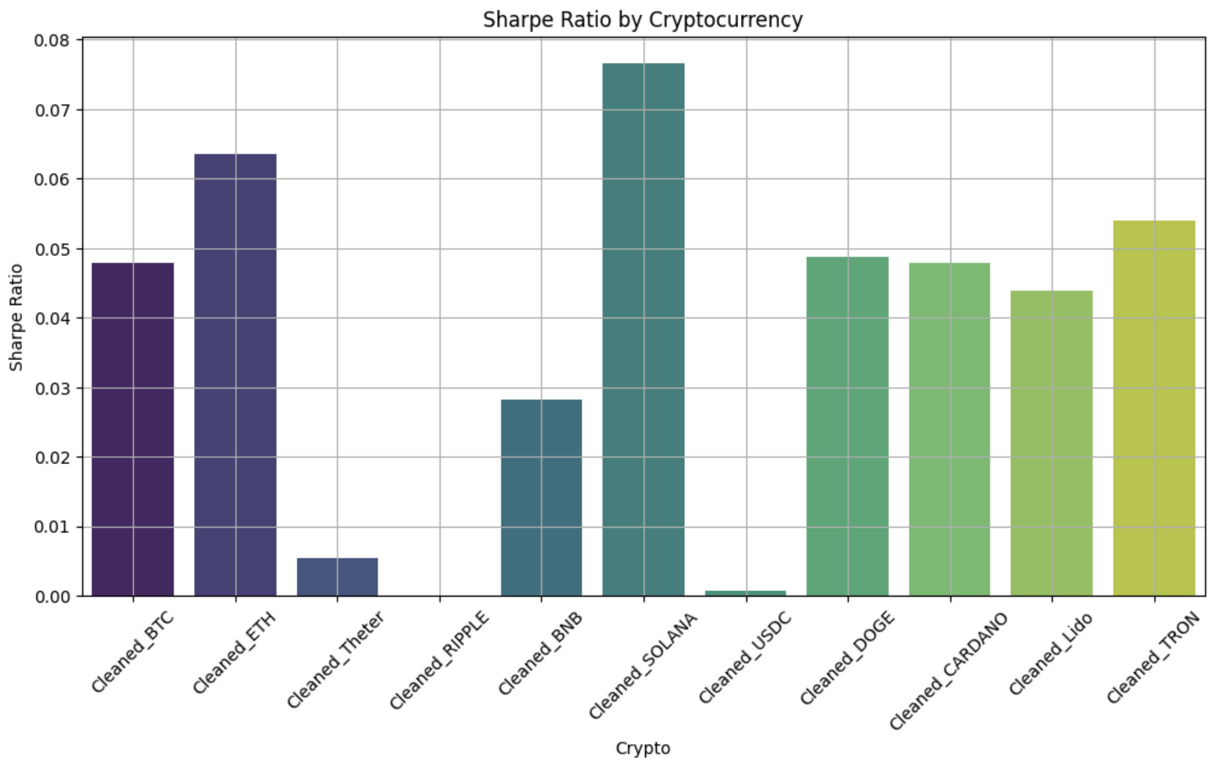


Figure 21: Comparison of Sharpe Ratios Across All Cryptocurrencies

Beyond these quantitative measures, the visualization of Cumulative Returns offered a powerful qualitative assessment of the strategy's long-term performance. Plotting the growth of the portfolio over time clearly illustrated periods of gains and losses, stability and volatility, providing an intuitive understanding of the strategy's trajectory, as demonstrated in an example below in Figure 22. Collectively, these risk management and performance evaluation metrics provided the necessary framework to critically assess the effectiveness and viability of the predictive models and their derived trading strategies, guiding future refinements and informing practical investment decisions in the volatile cryptocurrency landscape.

Figure 22: Return Predictions for a Cumulative Portfolio Consisting of TRON & ETH

Bonus & Conclusion

App Design: TrendWind

The utilization of these data analysis/evaluation and pattern recognition tools could be highly beneficial in aiding investors make informed investment decisions. Thus, this presents an opportunity for an impactful application aimed at facilitating the trade actions of data-based investors by providing extremely valuable insights using LSTM statistical modeling. With a platform that uses normalized data to make easy-to-understand visualizations, LSTM-powered software to make prediction modeling and quantitative risk mitigation and performance evaluations, investors could see far greater returns on their cryptocurrency investments.

The main concern for the success of this platform would be ensuring that the data provided by the platform demonstrates high levels of accuracy. Since the cryptocurrency world is highly unpredictable – or at least far more difficult to predict than the stock market – due to the fact that cryptocurrencies are highly impacted by random societal trends rather than physical events. Therefore, the development of a successful application of this sort would require intensive testing and training of the models (such as the LSTM model) using a far greater amount of data.

To demonstrate how a user might interact with this kind of application, I built a prototype frontend design for an app called TrendWind, as shown below in Figure 23. The user interface includes market leaders, accuracy, strategy, and risk control for all eleven of the cryptocurrencies used for this project, though ideally, the app would provide these services for all major global cryptocurrencies.

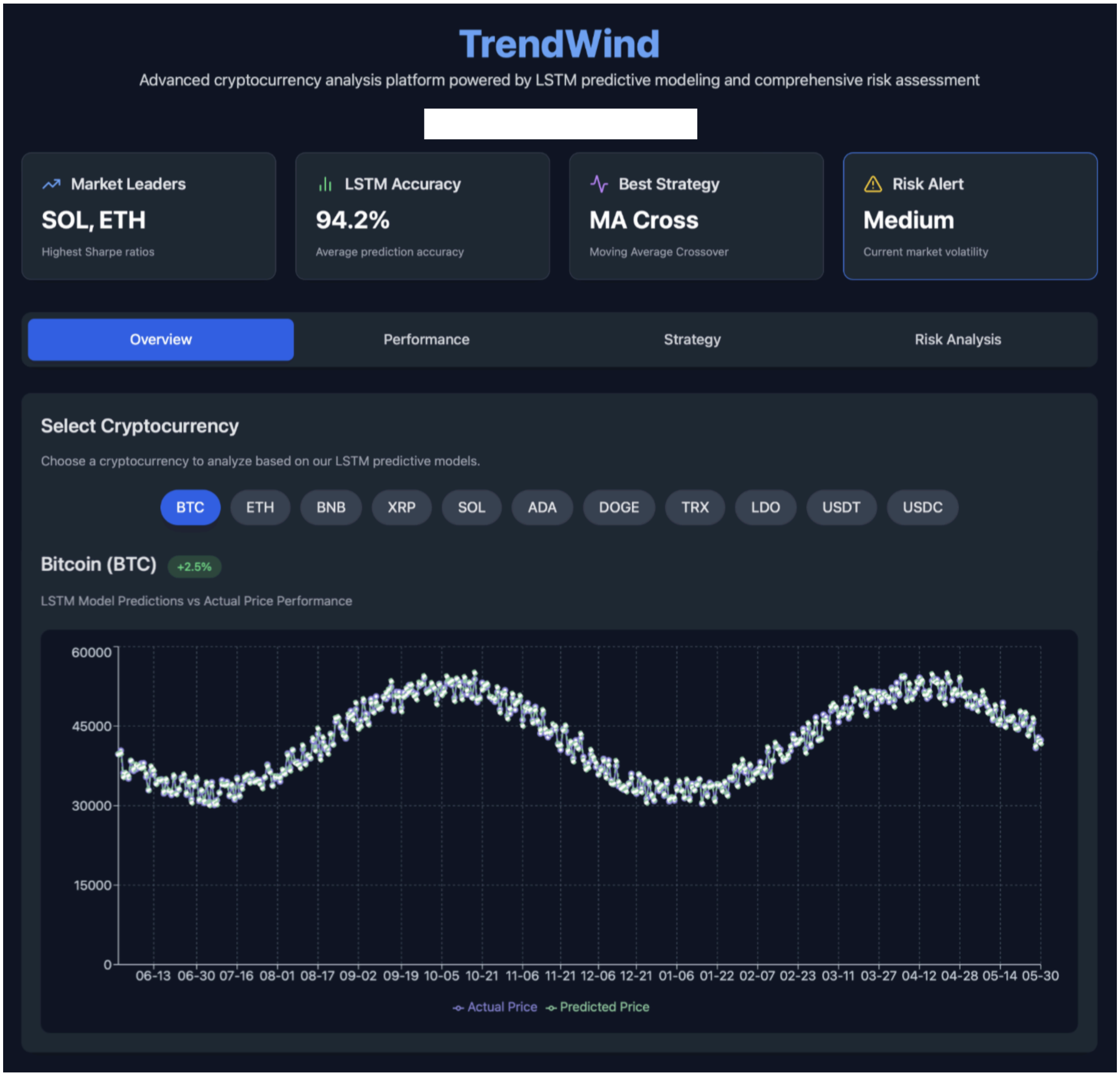


Figure 23: TrendWind User Interface

Conclusion and Final Interpretation of Results

The project consisted of several critical parts. The initial data cleaning and transformation allowed me to optimize the model-building process and create impactful data analysis. Particularly, the normalization of the data allowed for more productive pattern and trend identification because it removed the impact of varying value ranges on the overall analysis of the data. The normalization was likely the most crucial aspect of the first section of the project.

Following the data cleaning, transformation, and normalization was the visualization and statistical pattern recognition, in which I used the high-quality data set I created to present the data in effective formatting that allowed for visual observations of trends and patterns that would later be the foundation of the more precise and program-based data analysis.

Subsequently, the project moved into predictive modeling using ARIMA and LSTM algorithms. The effectiveness of these forecasting models was quantitatively assessed using metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE), which measured the average magnitude of prediction errors, with lower values indicating higher accuracy. Additionally, the R-squared (R2) metric was employed to determine the proportion of variance in the actual prices that could be explained by the models' predictions, providing insight into their explanatory power.

Building upon the forecasting results, a simulated trading strategy was implemented to evaluate its potential financial performance. Key financial metrics were calculated, including the Sharpe Ratio, which quantified the risk-adjusted return of the strategy, indicating how much return was generated per unit of volatility. The Maximum Drawdown was used to measure the largest historical percentage loss from a peak, serving as a critical indicator of downside risk. Finally, the Cumulative Return provided a direct measure of the overall percentage gain or loss of the portfolio over the simulated period. These metrics collectively offered a comprehensive view of the strategy's profitability and inherent risks.

The project also included a custom Risk Level assessment. While intended to provide a qualitative categorization of risk, its interpretation, particularly in conjunction with low or negative Sharpe Ratios, highlighted the nuanced challenges of synthesizing multiple risk metrics into a single, intuitive score. This aspect underscores the importance of carefully designing composite indicators to accurately reflect financial risk, especially in volatile markets where strategies may experience overall losses.

Cryptocurrencies are heavily influenced by external factors such as news, regulations, and current events. These factors make it incredibly difficult to predict cryptocurrency with only data on historical behavior, as often, there are many more factors at play. Volatility and market sentiment can always change quickly and unpredictably, and making very short or very long-term predictions is always uncertain. However, by leveraging historical data as well as unstructured data such as Twitter/X posts, news articles, and other forms of relevant and current information, powerful cryptocurrency prediction models can be built and implemented to effectively predict future movements.

References & Appendices

Reference

Kessel, Andrew. “Bitcoin, Crypto Stocks Fall as Tariff Fears Grip Markets.” *Investopedia*, 7 Apr. 2025, https://www.investopedia.com/bitcoin-crypto-stocks-fall-as-tariff-fears-grip-markets-11710116.

Code

The Jupyter Notebook file can be accessed through my GitHub at:
https://github.com/adamyanai/Cryptocurrency-Pattern-Recognition-Two