Market Timing Strategies for Crypto currencies using Kalman Filters

Rakesh Sharma P. R

WorldQuant University, Master of Science Financial Engineering
rakeshsharma.pr@gmail.com

Abstract

Machine learning algorithms power the modern stock trading strategies, with complex modeling and noisy data prediction of stock prices remaining a challenge. Algo trading strategies rely on the accuracy of prediction and introduction of the Kalman filter in this paper is a search in this direction. In this paper I introduce Kalman Filters for forecasting stock prices and volume and use the signal generated for trading strategies. The Kalman Filter algorithm is used in dynamic systems for finding the next and is only dependent on the previous state. The previous state captures all the information of the history and removes the need of rolling features that make the model deterministic. Kalman Filters updates its current state based on the incoming data or observation and makes for an aggressive trading strategy on stock prices. I will be using crypto assets for forecasting and will be using other ML algorithms such as Linear Regression for comparison of the performance. Kalman filter forecasting will be used for back testing of trading strategies with various portfolio metrics. The code experiments are available in a ***Google Colab Jupyter*** *notebook and at the* ***Github*** *repo, links are shared below.*
https://colab.research.google.com/drive/1DHlMEGF2ZAOw8_1gXE1x8jITpTMfp4Hq

https://github.com/rakeshsharma14/WorldQuant-Capstone.
Kalman Filter methods have shown promise in this paper, both with the model validation metrics and portfolio strategy ratios as explained in various sections of this paper.

Keywords: *Kalman Filters, Machine Learning, Crypto Trading, Timing Strategies, ML Algorithms, Linear Regression, Kalman Forecast, Stock Price Forecast, Double Moving Average Crossover*

1. Introduction

Machine Learning algorithms are changing the way we are trading stocks for both buy and sell strategy. It allows financial managers to engage in newer ways to generate the alpha to beat the market. Financial data is inherently very noisy and this makes it very challenging for accurate forecasts. Moreover, there are numerous other factors that lead to information flow affecting stock prices, making data collection and identification of their correlations a tedious task. Traditional models such as ARIMA, ARMA and GARCH fail to capture these nuances [Diego Villarino, 1]. Deep learning techniques can be trained on large historical data sets and have found applications in financial industry use cases for a long time. They tend to outperform traditional Machine learning models [O S Alamu, 5]. Financial data is noisy and there are a lot of external factors that cause subsequent fluctuation of prices. Machine learning is a companion of a modern trader as stock price prediction can now follow data more closely than ever and need not be driven by a mean price or intuitions of the trader [O S Alamu, 5]. Linear Regression is a popular technique used to predict the trend of stock signals. In many cases they have given surprisingly good results. Linear Regression is easy to model and explain; it works on the minimization of difference between predictions and actual stock prices. Linear Regression models suffer from overfitting when used with rolling features [Dengxin, 2]. One problem with the

Machine Learning model is that machine learning techniques suffer from frequent retraining of the data due to the data drift with the new trends of stock. For aggressive pricing financial managers need to take into factor frequent retraining costs of the model. There is a general tradeoff between increase in data points, retraining cost and improvement of the accuracy. If the frequency of trading is daily or hourly then this becomes a bottleneck, as training a whole dataset with just new additional data points is not very efficient.

Other techniques that have gained attention recently are Kalman Filters, and have come from other fields such as Aerospace engineering and robotics [Claudio Urrea, 4]. Kalman Filter can overcome the bottleneck of frequent training as it makes use of the difference between priori prediction and current observation in the new state to be predicted [Claudio Urrea, 4]. Kalman Filter based trading strategies allow financial managers to participate in long and short selling strategies and can generate better alpha for their portfolio. The problem with time series data is that it is noisy, is hard to predict and fails to generalize. Kalman Filters is a filtering algorithm that can be used for noisy time series data. Time series data is unique in that it is a linear combination of signal, trend and seasonality. The challenge with time series for modeling is to separate out the underlying signal from the noise. The Kalman filter in finance is also known by the name alpha – beta – gamma system. It is based on the average update trick, which means as an estimate at any time *n* will be the average of all the previous estimates [Qiang Li, 3]. Calculating the average for the previous values will be computationally costly and Kalman Filter algorithm overcomes this drawback with the update method.

2. Scope and Objective

The objective of this research paper is to explore the potential use of Kalman Filter for crypto price prediction namely, Bitcoin and Ethereum for short term and long-term trading strategies and compare the results with traditional Machine Learning methods such as Linear Regression. Crypto data is highly volatile and it is highly non-linear in its underlying trend [Jingyang Wu, 12]. Kalman Filter on the other hand works well with dynamic systems which need real time data [Qiang Li, 3]. Using constant real time updates for crypto prices could work as an effective trading strategy for financial managers. Frequent training of data for short term prediction is not feasible in the financial world and in this paper, we will observe how Kalman Filter updates are more accurate and perform better with trading strategies.

3. Background - Kalman Filter

Origin of the Kalman Filter algorithm can be attributed to Rudolf E. Kalman, who introduced the term Kalman Filter in 1960 in his famous paper "*Describing A Recursive Solution to the Discrete Data Linear Filtering Problem*" for the estimation of non-observable state from events that may have error [Qiang Li, 3]. Kalman filters are used in systems that collect multiple data from various sensors, such as GPS systems, trying to estimate velocity and location. The problem with such systems is that the signals represent the hidden states and are used to predict the future system states. In such systems there is no direct way to measure state of the system and all the signals are indirect, leading to the noise in the signals [Qiang Li, 3]. It is also used in places for combining information in the presence of uncertainty like in the domain of Sensor Fusion where there is uncertainty and one has to make educated guesses [Camile J.J. Beckers, 10]. Our stock price prediction is such a system and Kalman Filter makes an ideal choice for this application. There are several strengths of using Kalman Filter in addition to the uncertain nature of the system. It is a linear model and is not computationally expensive. It is light on memory and as the previous state of the system captures the history there is no need to keep any historic states like a long-term memory [Camiel J.J. Beckers, 10]. Due to this nature, it is extremely fast and to be used in real time systems. Kalman systems assume that the state of the current system can be dependent on any independent variables, which are Gaussian distributed. Each variable has a mean μ and is the center of the distribution or the likely state [Camiel J.J. Beckers, 10]. The variance of the distribution or σ^2 is the uncertainty. At any point in time the future stock price can be any combination of these variables picked from the distribution but certain values are more likely than others, this is the very essence of the Kalman Filter Algorithm. With Kalman Filter updates stock price can be constantly updated with the incoming real time data and makes it easy to change or build trading strategies in search of superior α.

4. Literature Review

There are several papers that this research allowed me to explore and understand that there is a gap in the methods used in prediction of stock prices. ARCH, ARMA and GARCH models are popular for modeling and interpreting time series data. The ARCH model was proposed by Engel to explain clustering and persistence of the stock market. Bollerslev extended the ARCH model to GARCH, which could also model variance of error [Ningyi Li, 6]. On the similar lines ARMA is suitable for short term prediction and is suitable for the study of stationary stochastic processes. These models require time series to be random and stable [Ningyi Li, 6]. The paper from Ningyi Li et al clearly brings out the need for combination of ARMA, ARCH and GARCH for various lag levels (p, q) parameters to capture the time series data such as stock index yields, interest rate market risk, exchange rate volatility etc. [Ningyi Li, 6].

In the paper, by O S Alamu and Md Kamrul Siam, "Stock Price Prediction and Traditional Models: An Approach to Achieve Short-, Medium- and Long-Term Goals" the authors concluded that deep learning methods such as LSTM outperform traditional methods as they are able to capture complexities and nonlinear pattern of data [O S Alamu, 5]. These methods also make it less interpretable and require greater computational resources. Daily trading strategies using these models involve trade-offs on accuracy and computational complexity. Another paper from John Pahn and Hung-Fung Chang on "Leveraging Fundamental Analysis for Stock Trend Prediction for Profit" employs a CNN-LSTM model that uses technical analysis details to achieve greater accuracy [John Pahn, 9]. The paper indicates that feature engineering is an integral part of stock price prediction and there is value in incorporating external factors for improving accuracy.

In another paper by Dengxin Huang, he studied Apple stock for stock market forecasting and uses Fama French 3-factor, Linear Regression, Random Forest Regressor and Gradient Boosting Regression model for comparison and found 3-factors critical to performance improvement of stock price predictions. Linear Regression emerges as the best performing model in that case study [Dengxin, 2]. Simplicity of Linear Regression and its interpretability made a strong case for it to be used for stock price prediction with feature engineering techniques on stock data.

There is various research that has highlighted the use of the Kalman Filter algorithm and its usage for state estimation. Notable among those is one using Kalman Filter to estimate the state of storage battery capacity. The paper by Camiel J.J. Beckers et al uses this technique to estimate the aging of batteries and its economic lifetime on the uncertainty of real-world data [Camiel J.J Becker, 9]. The research uses a combination of joint Extended Kalman Filter, combined with Recursive Least Squares to estimate the impedance. As per the author the algorithms converge quickly to the trend of the capacity and resistance in this case. They also highlight that the cost of computation is very minimal in the process and the data gets updated in the real time driving scenario making it ideal in this scenario. Kalman Filter has been in use in industry for many years and is known for its robustness, this is explained in detail in a paper by Claudio Urrea et. al. in their paper "Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation" and inspires my research for its application in financial market use cases.

4.1 Competitor Analysis

The literature review above brings our focus on the need of a new method to improve stock price prediction and why Kalman Filter can be a robust method for the noisy and dynamic nature of the financial data. A competitor analysis of the current methods helped identify where current research fits in. Kalman Filter updates capture the history in the current state and represents the highest likelihood of the stock price value, which is the future state predicted. This offers an opportunity to predict more on real time data and allows frequent predictions like hourly data. Traditional Machine Learning methods would be computationally costly to train on real time data for higher frequencies. The longer the prediction we do, the more static the prediction will be, such models will also suffer from overfitting. The strength of this algorithm allows for high frequency trading strategies and can also be combined with stock price signals for more robust prediction and can be a scope for further research. It is clear from many of the research papers we have studied and cited above that Linear Regression with feature engineering works better than some of the Deep learning methods for equity stock prediction. CNN-LSTM based architecture increases the computational complexity and may not be ideal for prediction based on real time updates. Kalman Filter applications are more suitable for real time updates, where stock price prediction and forecast will have a shorter window for entry and exits. Traditional forecasting algorithms may be more suitable for forecasting for longer periods and executing long term strategies.

On the other hand, crypto currency trading has been gaining momentum in recent years. Even though adoption of crypto as a legitimate currency is debated by many, it has increased in valuation and is now accepted as an alternate asset to hedge equities or other forms of assets [Bingqia Luo, 13]. Crypto is characterized by high levels of volatility compared to other stocks. It is also considered as a high-risk investment. Predicting crypto prices using algorithms is a challenge due to these factors. The global value of Crypto currency was at $3 trillion in 2021, however next year the value dropped down to $1 trillion, such is the volatility and risk inherent in crypto trading [Duy Thien An Nguyen, 10]. Research has concluded that multivariate Convolutional LSTM gives better performance for Crypto Price prediction during Covid-19 period where the volatility was high. All these strengthen the need to find alternate methods such as Kalman Filter as a replacement for Crypto price prediction in this research.

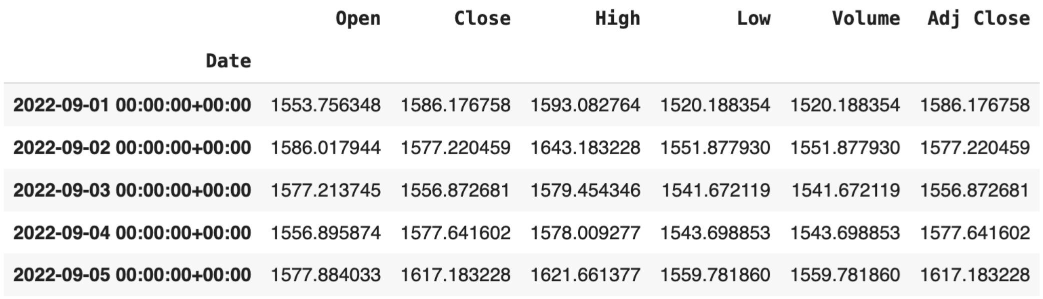
5. Methodology

In this section we would be describing the design and process of the research done to understand the performance of the Kalman Filter algorithm. The methodology follows guidelines and pitfalls highlighted in the research paper on Machine Learning Pitfalls [Michael A. Lone, 12]. We have used best practices for supervised learning followed in industry for our comparison. The below sections are various steps performed before reaching the conclusions given in this paper. All the coding is done in python language and we have relied on verified python libraries to implement the Kalman Filter and Linear Regression algorithms. Kalman Filter implementation is done using *pykalman*, an open-source package available at https://pykalman.github.io/. The filter defined in python is then used for forecasting of Ethereum and Bitcoin adjusted closing prices and is further used in back testing of portfolio (BTC and ETH) under Dual Moving Average Crossover Strategy (DMAC) using the python package available at https://vectorbt.dev/. *Vectorbt* is a highly popular open-source python package used for algorithmic trading, research and backtesting. In the backtesting process we will be comparing the prediction outputs from Linear Regression and Kalman Forecast under specific short term and long terms entry and exit strategies

5.1 Data Collection - For this research we have collected crypto currency data, namely Ethereum and Bitcoin for the last two years from yahoo finance website. Due to its incompleteness, and incorrectness we have decided to read data from ***finance.yahoo.com*** website. Other sources that we have considered are Kaggle and data available in Github. The observation window is from *2022-09-01* to *2024-08-31*. We have taken the data from the post Covid-19 event as that event has added a lot of volatility to the crypto data and wanted to remove watershed events from our study and observation. The data can be collected using the APIs from *yahoo finance package* which can be called in python and read into the data frame data structure for *pandas*. This makes it easy to analyze the data in a *jupyter notebook environment* without much hassle. The data points are at daily frequency, there are also lower frequencies available like hourly and its applicability of Kalman algorithm for real time updates could be a study for the future. The reason for choosing ETH and BTC are that they are the leaders in the market of more than 10,000 cryptocurrencies [Bigqiao Lu, 12]. They are also the oldest crypto currencies and represent information and nuances from this space such as NFTs and other emerging trends. Ethereum is traded with the ticker label ETH-USD and Bitcoin uses the ticker symbol BTC-USD. Both ETH-USD and BTC-USD are crypto linked to the US dollars for their value. Ethereum was developed by Vitalik Buterin and Gavin Wood in 2013 and has the largest market cap after the Bitcoin currency [Bigqiao Lu, 12]. Bitcoin was developed by Satoshi Nakamoto in 2009 and is the most popular crypto currency [Bigqiao Lu, 12]. Ethereum and Bitcoin can be traded with peers or can be received from apps and exchanges [Buterin, 14].

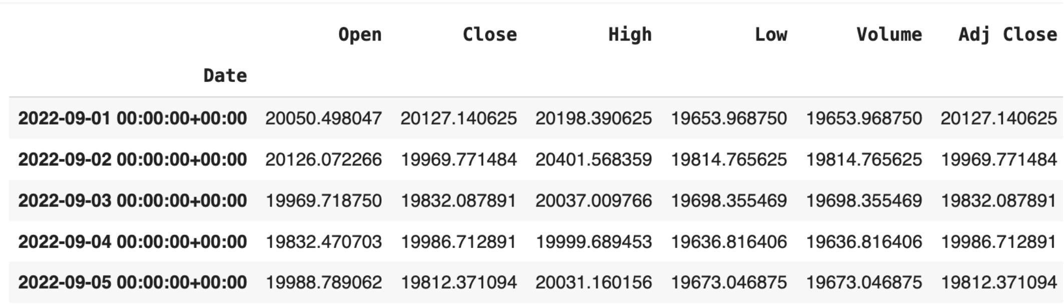
The below figure shows the top five rows of Ethereum (ETH-USD) and Bitcoin (BTC-USD) in their dataframe structure.

Fig 1. The first 5 rows of Data for ETH-USD



	Open	Close	High	Low	Volume	Adj Close
Date						
2022-09-01 00:00:00+00:00	1553.756348	1586.176758	1593.082764	1520.188354	1520.188354	1586.176758
2022-09-02 00:00:00+00:00	1586.017944	1577.220459	1643.183228	1551.877930	1551.877930	1577.220459
2022-09-03 00:00:00+00:00	1577.213745	1556.872681	1579.454346	1541.672119	1541.672119	1556.872681
2022-09-04 00:00:00+00:00	1556.895874	1577.641602	1578.009277	1543.698853	1543.698853	1577.641602
2022-09-05 00:00:00+00:00	1577.884033	1617.183228	1621.661377	1559.781860	1559.781860	1617.183228

Fig 2. The first 5 rows of Data for BTC-USD



	Open	Close	High	Low	Volume	Adj Close
Date						
2022-09-01 00:00:00+00:00	20050.498047	20127.140625	20198.390625	19653.968750	19653.968750	20127.140625
2022-09-02 00:00:00+00:00	20126.072266	19969.771484	20401.568359	19814.765625	19814.765625	19969.771484
2022-09-03 00:00:00+00:00	19969.718750	19832.087891	20037.009766	19698.355469	19698.355469	19832.087891
2022-09-04 00:00:00+00:00	19832.470703	19986.712891	19999.689453	19636.816406	19636.816406	19986.712891
2022-09-05 00:00:00+00:00	19988.789062	19812.371094	20031.160156	19673.046875	19673.046875	19812.371094

5.2 Data Preprocessing (Cleaning) - The dataset has been used for multivariate modeling and has these 4 columns namely $open, high, low, close, adj\ close\ and\ volume$ as values. The following checks are done as part of preprocessing. Both the data series are checked for $null\ and\ not\ a\ number\ (NA)$ values in the dataset. For rolling properties or features, where the column values are null are eliminated. There is data availability for all the date indices for BTC-USD and ETH-USD. We have applied and scaled for both data series. Standardization is the technique we have applied to scale the data. Post standardization the data series will have a mean of zero and standard deviation of one. It can be defined by the formula

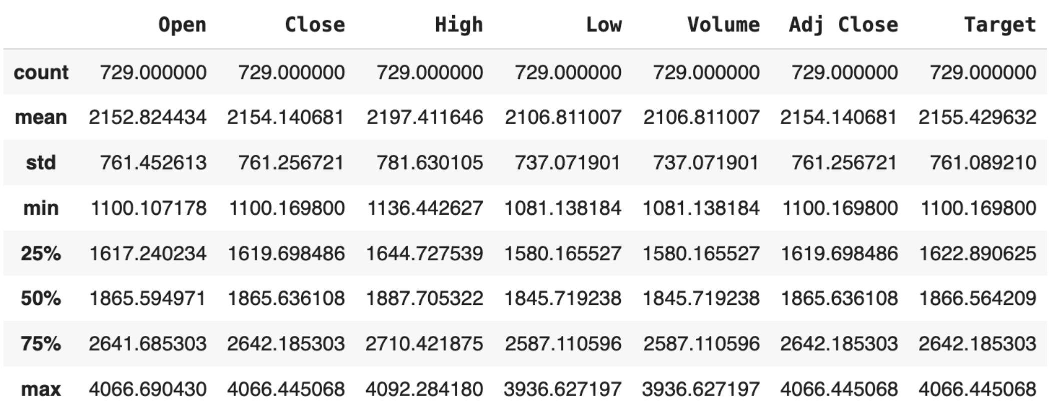
$$z = \frac{X - \mu}{\sigma} =$$

where X is the data point, μ is the mean of the data series and σ is the standard deviation. We have used the python library $scikit - learn$ to achieve the scaling. The output of the data preprocessing is the training dataset.

5.3 Exploratory Data Analysis - Summary statistics and visualization of time series will help understand the trend, seasonality and noise within crypto data. Exploratory analysis gave the following conclusion about the crypto data. Crypto data is noisy and will have to use techniques to separate out trend and seasonality. The use of Kalman Forecast separated out the signal from the noise. $Autocorrelation$ and $partial\ autocorrelation$ functions identified the lags significant for BTC-USD and ETH-USD data. Other columns will be used as independent variables. We have used the statsmodel package from python to plot the autocorrelation and have performed the Dickey fuller test to confirm the trend and seasonality.

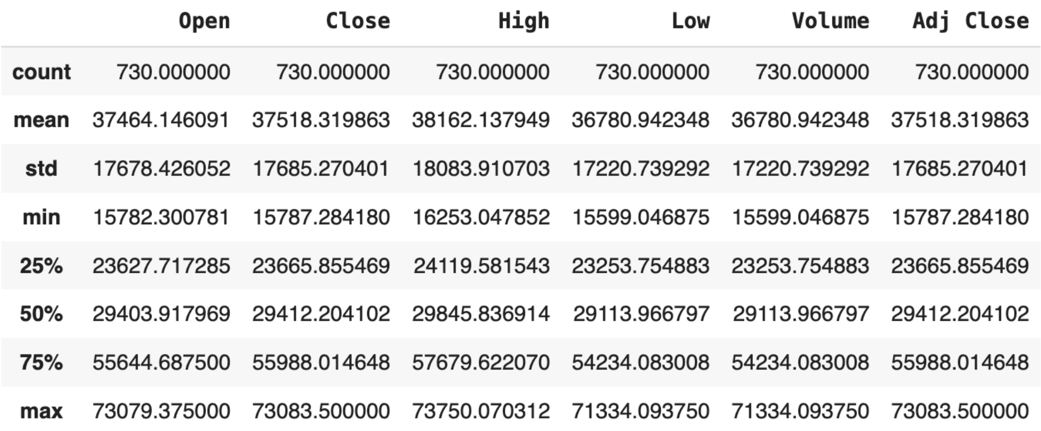
We can see the summary statistics of the ETH-USD and BTC-USD here.

Fig. 3. Summary statistics of ETH-USD



	Open	Close	High	Low	Volume	Adj Close	Target
count	729.000000	729.000000	729.000000	729.000000	729.000000	729.000000	729.000000
mean	2152.824434	2154.140681	2197.411646	2106.811007	2106.811007	2154.140681	2155.429632
std	761.452613	761.256721	781.630105	737.071901	737.071901	761.256721	761.089210
min	1100.107178	1100.169800	1136.442627	1081.138184	1081.138184	1100.169800	1100.169800
25%	1617.240234	1619.698486	1644.727539	1580.165527	1580.165527	1619.698486	1622.890625
50%	1865.594971	1865.636108	1887.705322	1845.719238	1845.719238	1865.636108	1866.564209
75%	2641.685303	2642.185303	2710.421875	2587.110596	2587.110596	2642.185303	2642.185303
max	4066.690430	4066.445068	4092.284180	3936.627197	3936.627197	4066.445068	4066.445068

Fig. 4. Summary Statistics of BTC-USD



	Open	Close	High	Low	Volume	Adj Close
count	730.000000	730.000000	730.000000	730.000000	730.000000	730.000000
mean	37464.146091	37518.319863	38162.137949	36780.942348	36780.942348	37518.319863
std	17678.426052	17685.270401	18083.910703	17220.739292	17220.739292	17685.270401
min	15782.300781	15787.284180	16253.047852	15599.046875	15599.046875	15787.284180
25%	23627.717285	23665.855469	24119.581543	23253.754883	23253.754883	23665.855469
50%	29403.917969	29412.204102	29845.836914	29113.966797	29113.966797	29412.204102
75%	55644.687500	55988.014648	57679.622070	54234.083008	54234.083008	55988.014648
max	73079.375000	73083.500000	73750.070312	71334.093750	71334.093750	73083.500000

5.4 Feature Engineering - I have used multivariate Linear Regression to train the model to predict 'Adj Close Price' and 'Volume'. Feature engineering is a process that involves transforming a variable x into a form $f(x)$ to be used in the modeling. We have defined the target variable for prediction in this research. The target variable is next day's $Adj\ Close\ Price$. We use the shift operator to generate this variable in the dataframe. In the case of Volume Prediction, we try to predict the next day's Volume as the target variable y. In the linear regression model, we have used standardized "*Open", "High", "Low"* and "*Close"* Price to predict "*Adj Close"* Price. The "*Adj Close"* Price is the dependent variable y and the rest of the variables are dependent variables or features $[x_1, x_2, x_3, \dots x_n]$. Rolling means with various lags are also part of the feature set as part of the exploratory analysis, they have not been included for training. For Volume prediction, $Volume$ is the dependent variable y and standardized "*Open", "High", "Low"* and "*Close"* Price are the features $[x_1, x_2, x_3, \dots x_n]$. Kalman Filter uses an update algorithm and is different from supervised learning and will not involve feature engineering. It predicts the current state of stock price, which can be expressed here as $(x_{t|t-1}$) and uncertainty $(P_{t|t-1})$ at time step t based on the stock price and uncertainty at time step $t-1$ [Mario Filho, 18].

5.5 Training and Modeling - The dataset was divided into training, validation sets. The dataset obtained from preprocessing is divided into ratios of 8:2 to training, validation.
Training set is further split into training and testing sets. Training set is the dataset used to train the model and learn the underlying pattern and coefficients (weights). Validation set is used to backtest the data. Trading strategy DMAC will be tested using backtesting techniques to understand the overall portfolio return. For the Kalman Filter forecast, the dataset will be updated from the last point of the training set. The value will be updated for each observation in the validation set sequentially. Models trained are Linear Regression and Kalman Filter. The comparison of the models using the performance metric is done on the validation set.

5.5.1 Model Analysis

We are employing 2 models here
Linear Regression - It is an algorithm that predicts the dependent variable based on independent variables. It identifies a best fit line or surface that minimizes the error difference between the predicted values and actual observations. It is used for forecasting values and understanding the relationship between the dependent variable y and independent variables [O.S. Alamu, 5].

Kalman Filters - The crux of this paper is around Kalman Filter which is used for estimating and predicting states of dynamic systems. Kalman Filters uses the Kalman gain update process to update the value of a point estimate for the future value. The future value is the most likely estimate of the point based on the history of the value [Camiel J.J. Backer, 9].

5.6 Prediction Metrics - We have used **R-squared**, **RMSE**, **MSE**, **MAE** for comparison.

R-Squared measures the proportion of variance of the dependent variable or y that can be explained by the independent variables. It is also known as goodness of fit. The value varies from 0 to 1 and in some cases, it can be negative, in such cases the mean value of the data is found to be a better fit than the model. It is given by the formula

$$R^2 \quad = \ 1 - SS_{res} \,/\, SS_{tot}$$

Where $SS_{res}$is the total residual sum of squares and SS_{tot} is the total sum of squares

RMSE stands for root means square error - It is the average difference between values predicted and actual values in a regression model. It is given by the formula

$$RMSE \ = \ \sqrt{1/N \sum_{i\,=\,1}^{n} \quad (\hat{y_i} \ - \ y_i)^2}$$

It represents standard deviation of residuals and value ranges from 0 to ∞. Lower the value better the model. The choice of RMSE is due to the fact that it is an absolute measure of error and it can assess prediction precision.

MSE - Mean square error, it is the squared difference between predicted and actual values. It represents the residuals, the values which the regression model is not able to account for. It can be represented mathematically as below

$$MSE \ = \ 1/N \sum_{i=1}^{n} \quad (y_i \ - \ y)^2$$

There are several advantages of using this metric, it eliminates the negative values and can be compared to variance. Irrespective of the direction this is always a positive value. For a perfect model the MSE value will be 0. Squaring magnifies the larger error than smaller ones and so penalizes such differences in the model and learns to keep differences small in training.

MAE - Mean absolute error, it is the sum of absolute error divided by the sample size. The sum of absolute difference is also called Manhattan distance. It is given as below formula

$$MAE \ = \ 1/N \sum_{i=1}^{n} \quad |y_i \ - \ y|$$

Unlike squared errors, it gives equal weightage to each of the observation and prediction pairs and is useful for understanding the magnitude of errors. The reason for choosing this metric is its robustness to outliers, interpretability irrespective of direction of the prediction and its intuitive nature as it is Manhattan distance. Smaller the MAE better the prediction and ideally, we would like to get a MAE of 0 which means observation and prediction are the same.

5.5 Model Comparison

Comparison of Kalman Filter has been done with Linear Regression. We have done comparisons of price forecast of 2 cryptocurrencies for different periods, 10, 50, 100, 200 days. We have also done a comparison with volume forecasts of the 2 cryptocurrencies for 10, 50, 100, 200 days. The model metrics defined above are used for comparison.

5.6 Backtesting Strategy

Backtesting is a method for assessing the potential of trading strategy along with the prediction model on historical data. If our backtesting shows promising results on the historical data, then we can use it as a strategy for the future. The backtesting strategy used for this portfolio is DMAC. It stands for Dual Moving Average Crossover and is a cross-signal trading strategy [Jevtic, 16]. It is simple to interpret and easy to implement. In this strategy we use moving averages for 2 different time periods, short term and long term. Moving averages are called smoothers, the larger the MA window the smoother the time series will become. When we compare it with a smaller MA window the change in pattern becomes more visible [Liechty, 17]. One can define Buy and Sell signals in this strategy. I have defined 1 day as a short term for Ethereum and 2 days for Bitcoin Price Prediction. 2 and 3 days are defined as long term for Ethereum and Bitcoin respectively. In this strategy we buy Ethereum and Bitcoin when the short-term moving average crosses the long-term moving averages. We sell them when the long-term MA exceeds short-term moving average [Liechty, 17].

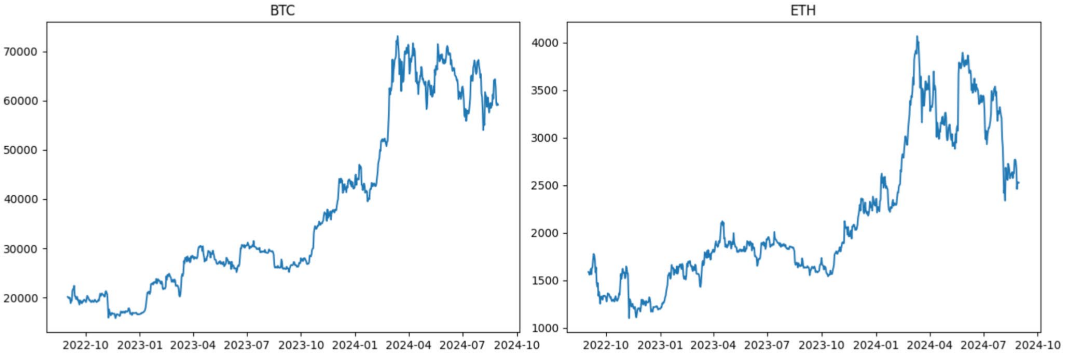
6. Results and Discussion

Before building any models, we should explore cryptocurrency data thoroughly to understand the underlying complexities, noise, and trends. In the below section we will look into details of the nature of the ETH-USD and BTC-USD and understand this time series.

6.1 Understanding Crypto Data

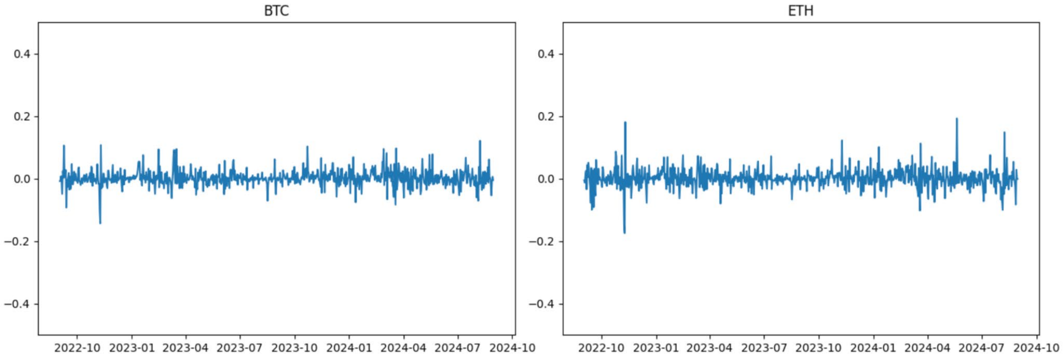
We will be modeling crypto data for the **Adjusted closing price** and **Volume.** These two factors contribute to our trading strategies.

Fig 5. BTC and ETH Price Trend during our research period



From the figure above it is clear that crypto data has an upward trend. BTC-USD has a higher value compared to ETH-USD for the same time period. Both the time series have a similar trend during the observation period.

Fig 6. BTC and ETH percentage change trend during our observation



The plot for percent change in the above figure for the time period shows downward and upward jumps in the data. Both time series have indicated high volatility for the time period. We can apply some level of smoothing to separate the noise and understand the underlying trend as seen below.

Fig.7. Rolling means of Adj Close Price for various time intervals for BTC

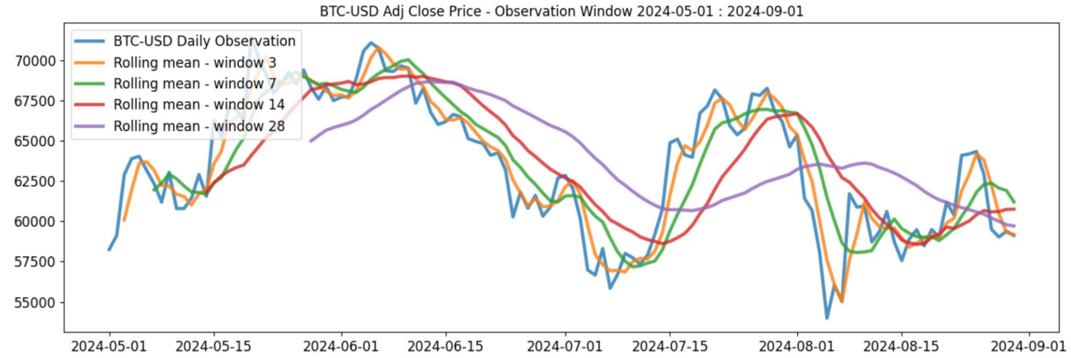
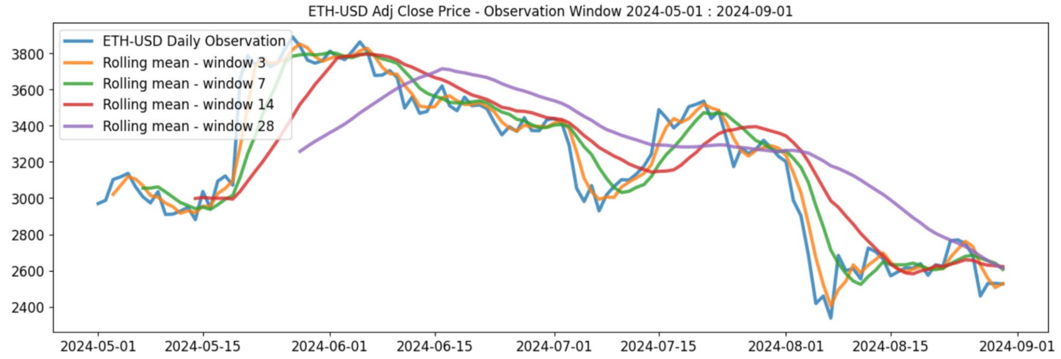


Fig. 8. Rolling means of Adj Close Price for various time intervals for ETH



Overall Ethereum has a smoother trend for the rolling period than BTC-USD, for the observation period. The rolling prices cross the original stock signal at various points and these are significant for our trading strategies.

Let us look at the Volume shocks for the data. Smoothed Volume data shows less volatility and remains stationary. Smoothed volume signals could be used for alternate trading strategies and can be explored further in the future.

Fig. 9. Rolling means of Volume for various time intervals for BTC

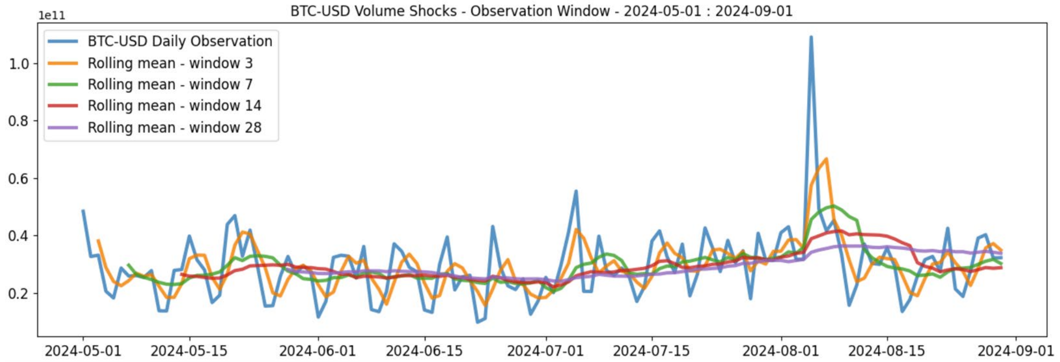
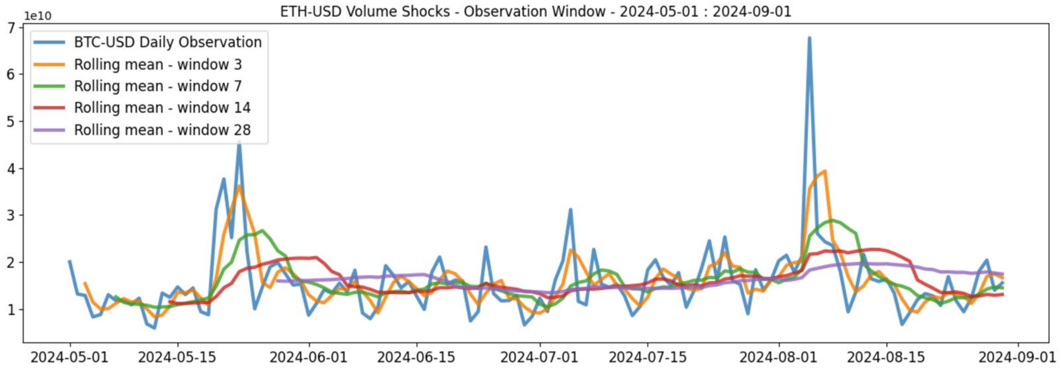


Fig. 10. Rolling means of Volume for various time intervals for ETH



We also observe that there is significant autocorrelation, the previous value of the stock price determines day's closing price. From the PACF plot it is clear that the price of certain periods of time is more correlated than others, say for example price on day 4, 9, 12, 15 etc.

Fig. 11. Autocorrelation Function Plot for BTC and ETH

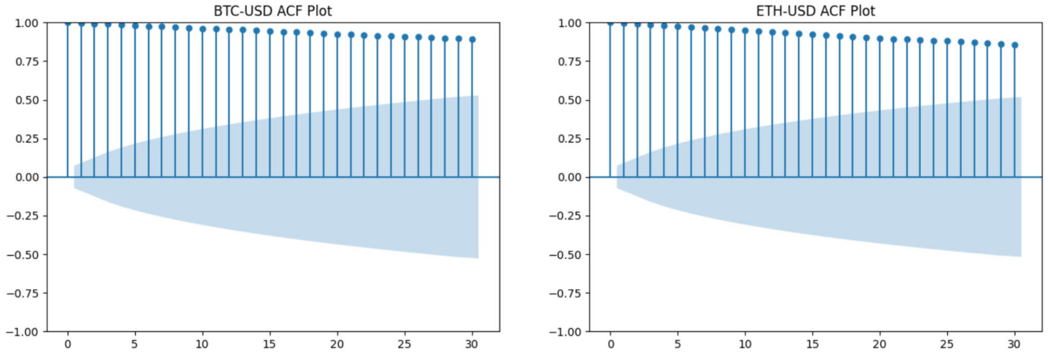


Fig. 12. Partial Autocorrelation Function Plot for BTC and ETH

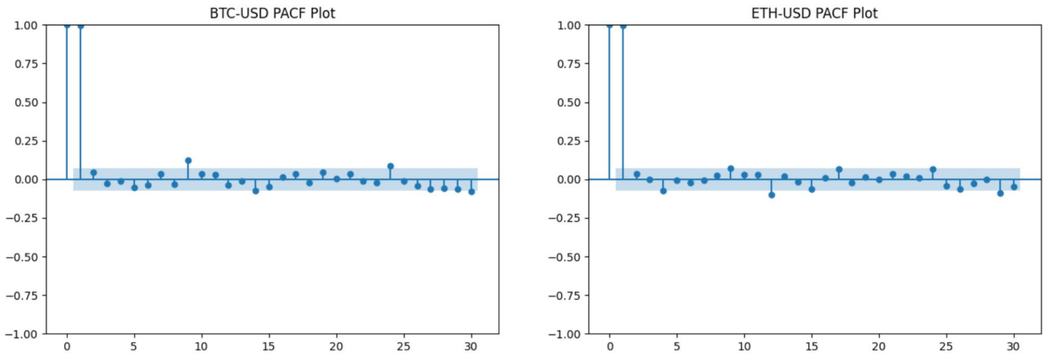


Fig. 13. Autocorrelation Function Plot for Volume Traded in BTC and ETH

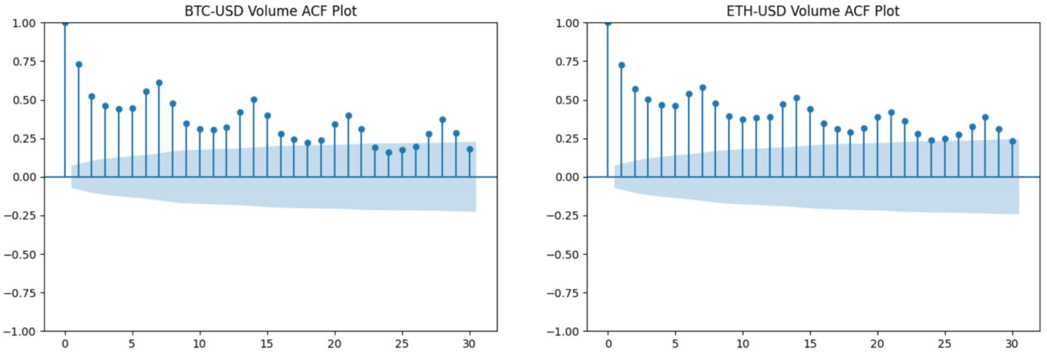
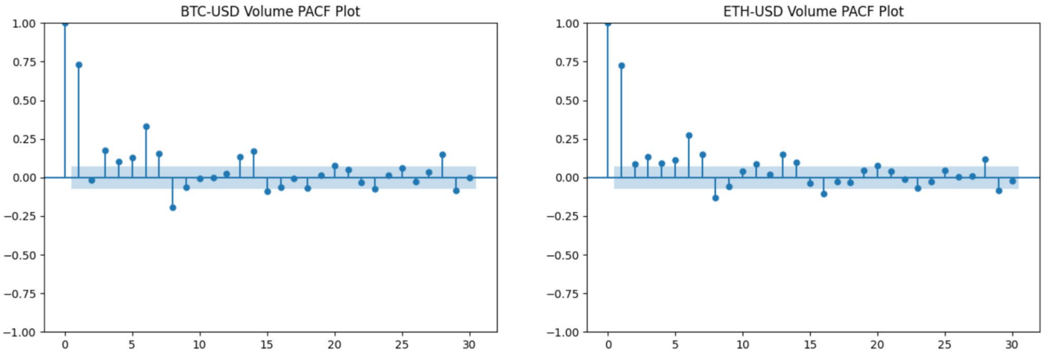


Fig. 14. Partial Autocorrelation Function Plot for Volume Traded in BTC and ETH



The ACF and PACF plots for Volume show significant correlation with observations at the previous time slots. This validates that regression on observation on previous time slots with a constant will be good to model this time series. This explains our choice of Linear Regression and Kalman Filter model for prediction as it makes use of auto-regression.

6.2 Kalman Filter

From the plots above it is inferred that crypto data is noisy, and highly volatile. There is an uptrend with strong autocorrelation with certain values at time spot $t - k$. In this section we will bring the observations from the Kalman Filter forecast.

Fig. 15. Kalman Filter prediction for Adjusted Closing Price for BTC

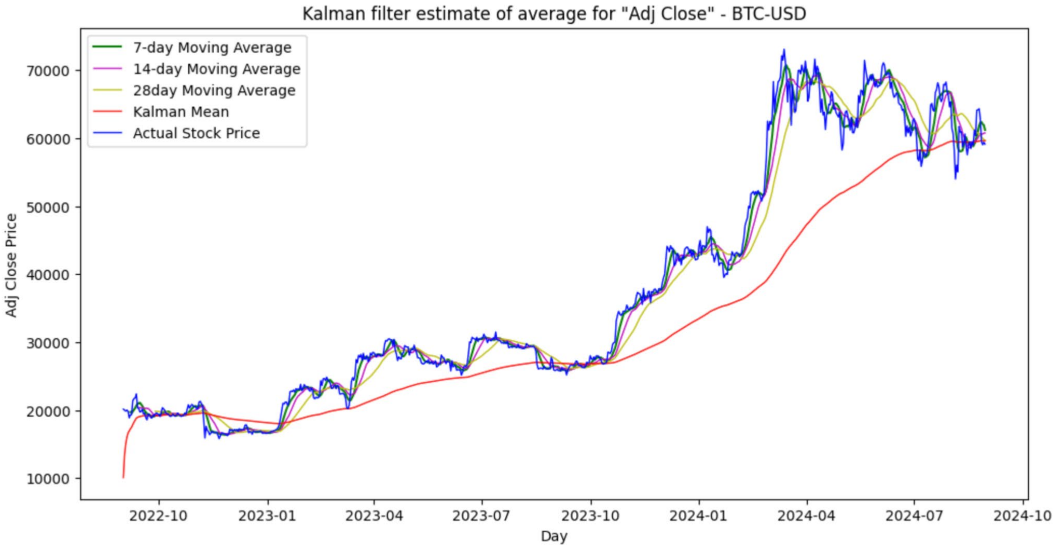
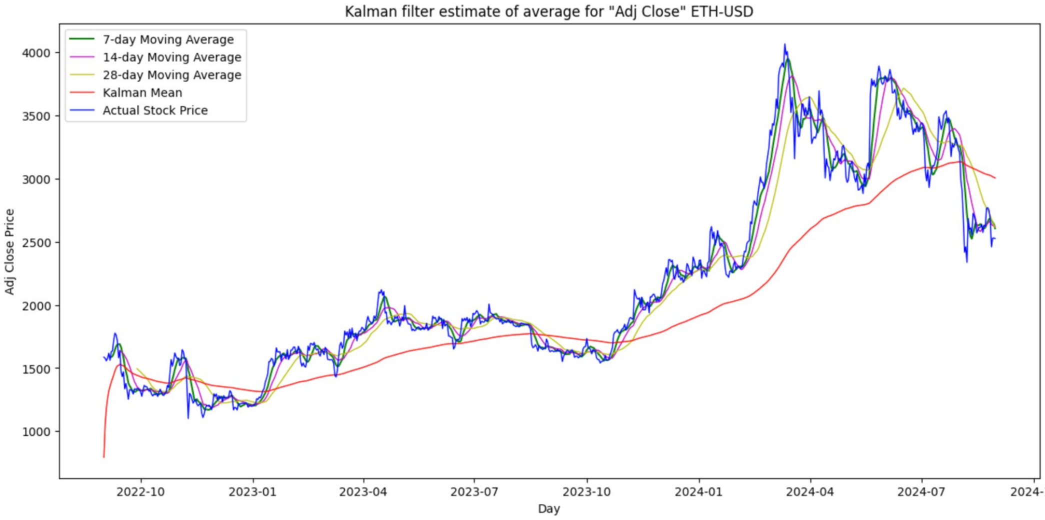


Fig. 16. Kalman Filter prediction for Adjusted Closing Price for ETH



Kalman forecast is able to separate the signal from noise and is smoother than any of the rolling smoothing techniques to understand the trend. The Kalman Filter algorithm returns the Kalman mean and covariance. The Kalman mean represents the most likely state of the Ethereum and Bitcoin signal if they have a Gaussian Distribution. Bitcoin and Ethereum have an upward trend and are devoid of any seasonality. Kalman Filter forecasts do not suffer from overfitting compared to the rolling means. This will be clearer in the plots Fig 6. and Fig. 7 in the appendix section, where we plot the difference between Kalman Mean and stock price vis-a-vis the rolling means.

We can further observe this phenomenon with the Kalman forecast both for Ethereum and Bitcoin. Kalman forecasts for **"Adj Close**" price of **ETH-USD** price have shown varying performance for different time periods and can be seen in the table below.

Table 1. Kalman Forecast - ETH-USD Model Performance Metrics

Metric	***200 Days***	***100 Days***	***50 Days***	***10 Days***
R-Square	**0.975**	**0.9925**	**0.988**	**0.912**
MAPE	**0.013**	**0.008**	**0.0097**	**0.0074**

RMSE	60.615	37.68	40.752	31.72
MAE	42.222	26.072	27.716	19.30

The below table captures the results for similar time periods for Linear Regression.

Table 2. Linear Regression - ETH-USD Model Performance Metrics

Metric	*200 Days*	*100 Days*	*50 Days*	*10 Days*
R-Square	0.911	0.832	0.873	0.48
MAPE	0.025	0.028	0.029	0.020
RMSE	114.61	132.088	129.925	78.785
MAE	82.39	94.76	100.33	59.43

From the above 2 tables it can be inferred that Kalman Mean is an ideal signal for devising strategies for buy or sell events as the model performance is superior to Linear Regression for similar time windows

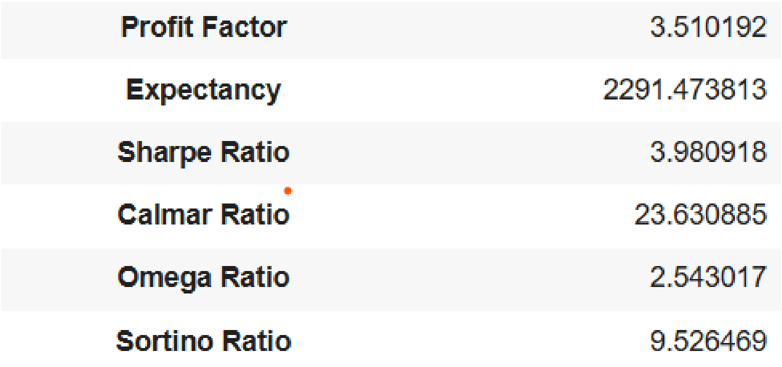
7. Conclusion and Future Work

The Kalman Filter algorithm is able to separate signals from the noise for stock prices and volume predictions. The algorithm can be controlled by the hyperparameter Transition Covariance. A high value of the hyperparameter - Transition Covariance parameter makes returns values closer to real time prices. Kalman mean value with a low Transition Covariance is also an ideal signal for buying and entry strategies for trading based on the intersection of the stock price with this signal. In order to compare the algorithms, we generate predictions using Linear Regression and Kalman Filter and define smoothers of these signals for 2 different time periods to generate entry and exit trading strategies.

We observe that Kalman Filter forecasts are better than Linear Regression predictions with respect to R-square, MAPE, MAE and RMSE for all different time buckets. The results for ETH-USD with Kalman Filter and Linear Regression can be seen from Table 1. and Table 2. figures above. We see similar results with Volume predictions, the results for which are given in the Table. 1 and 2. in the appendix section.

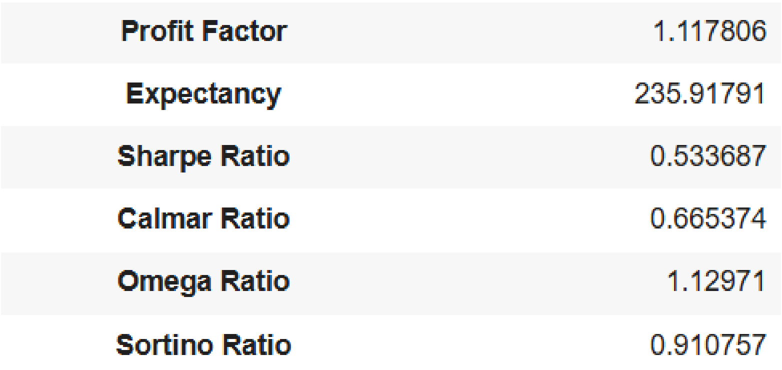
Backtesting Kalman Filter forecast and regression predictions using the DMAC strategy show that Kalman Filter outputs give superior performance with respect to portfolio testing parameters such as Sharpe ratio, Calmar ratio, Omega ratio and Sortino Ratio. The profit factor is 3.510, whereas it is only 1.117 for the prediction from the Linear Regression model. The total profit earned by Kalman Filter is much higher for a test scenario of $100000 as initial investment under the Double Moving Average Crossover Strategy. The results of the comparison can be seen in Table 3 and 4. below with respect to our portfolio. The cumulative portfolio performance and various buy and sell stages for ETC-USD and BTC-USD are shown in the figures 17. and 18. below.

Table 3. Portfolio performance with DMAC for Kalman Filter model



Profit Factor	3.510192
Expectancy	2291.473813
Sharpe Ratio	3.980918
Calmar Ratio	23.630885
Omega Ratio	2.543017
Sortino Ratio	9.526469

Table 4. Portfolio performance with DMAC for Linear Regression model



Profit Factor	1.117806
Expectancy	235.91791
Sharpe Ratio	0.533687
Calmar Ratio	0.665374
Omega Ratio	1.12971
Sortino Ratio	0.910757

Fig. 17. Ethereum Portfolio Performance with Kalman Forecast

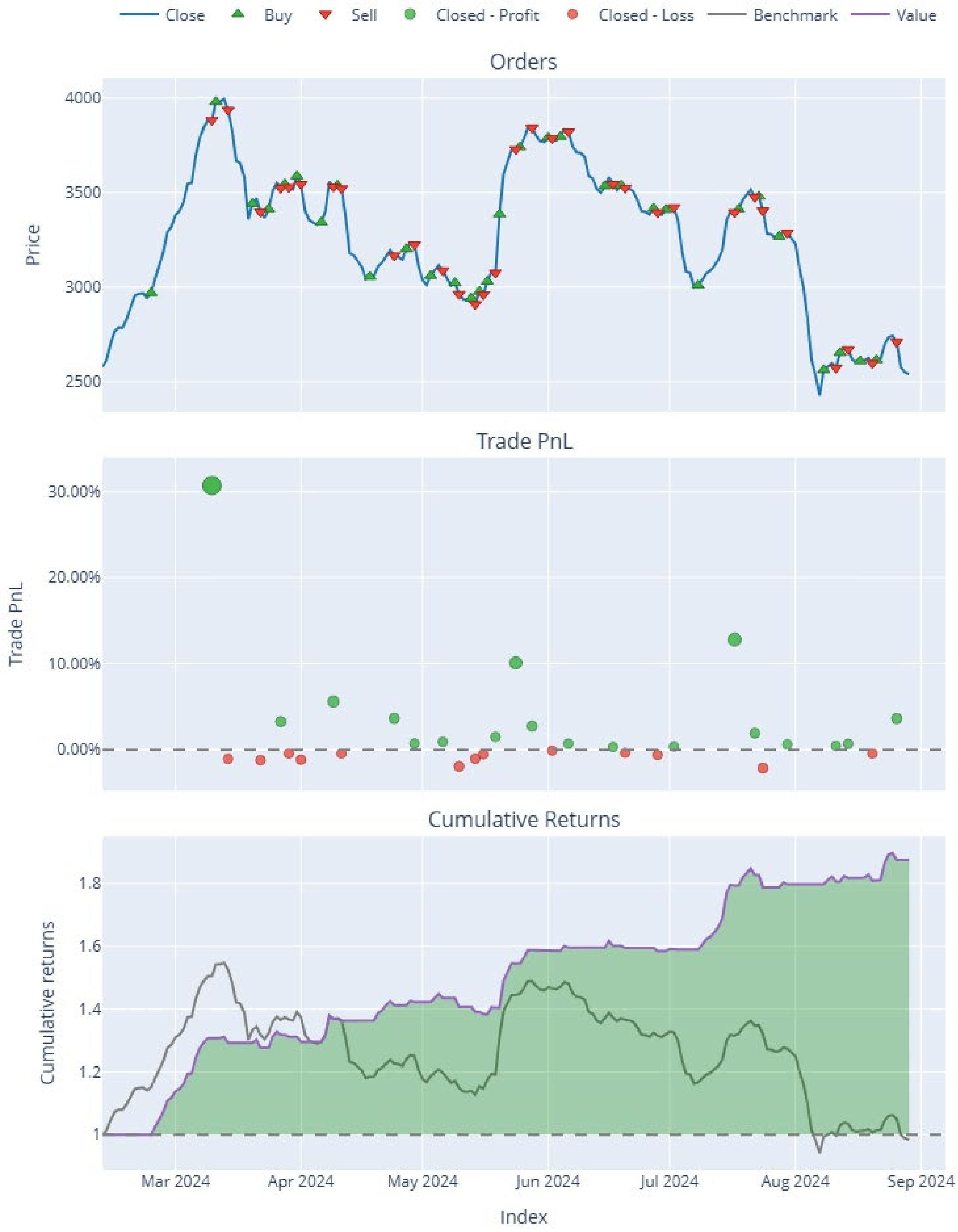
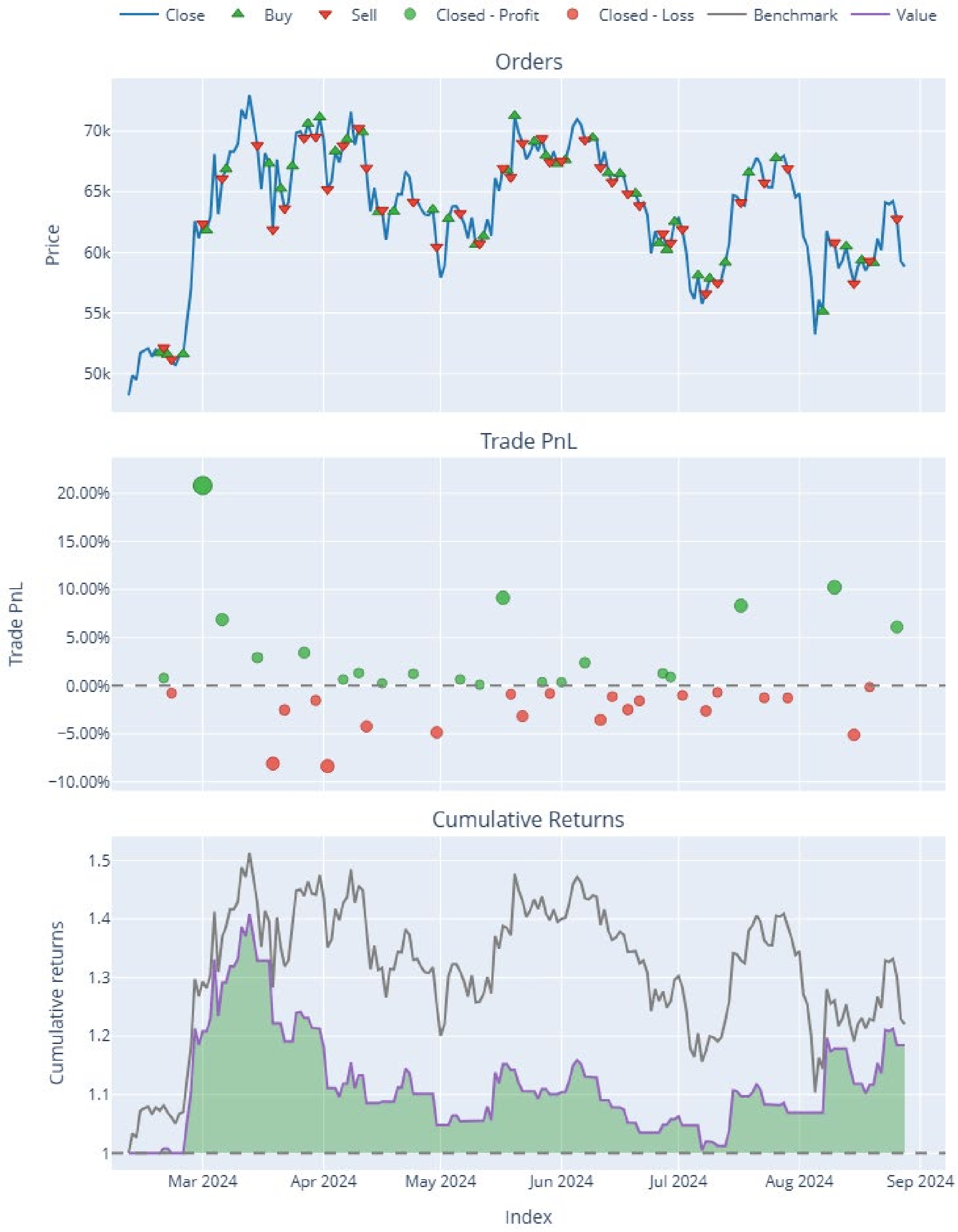


Fig. 18. Bitcoin Portfolio Performance with Kalman Forecast



As part of future work, I will further generate further strategies using *vectorbt* python implementation around Kalman mean signal for stock price interfering with original stock price for various entry and exit points. Volume predictions of the two crypto currencies can also be explored for various entry and exit points of trading strategies. Similarly, we can introduce comparison between LSTM and Kalman Filter for model performance and explore if LSTM does a better job at short term prediction strategies. We can also bring in external factors as features engineered variables to improve the prediction of the target variable. All these possibilities make research potential in this space ripe for financial engineers and can further the development of this research.

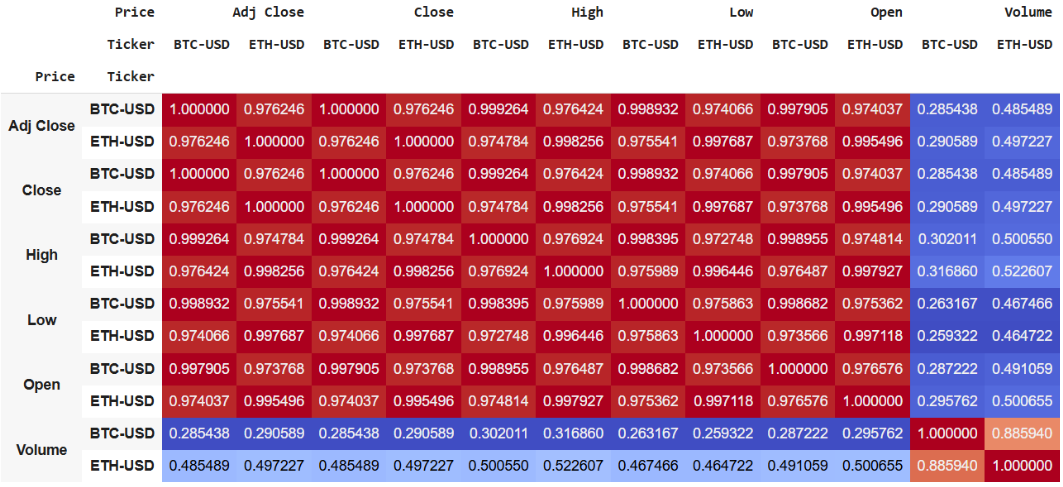
8. References:

1. Diego Vallarino. Oct. 2024. arxiv.org. A Dynamic approach to Stock Price Prediction: Comparing RNN and Mixture of Experts Models Across Different Volatility Profiles. https://arxiv.org/pdf/2410.07234
2. Dengxin Huang. Apr. 2023. arxiv.org. Application of Machine Learning in Stock Market Forecasting: A case study of Disney Stock https://arxiv.org/pdf/2401.10903
3. Qiang Li, Ranyang Li. Kaifan Li. 2015. Kalman Filter and Its Applications. https://www.researchgate.net/publication/305871722_Kalman_Filter_and_Its_Application
4. Claudio Urrea, Ramon Agromonte. Sep. 2021. doi.org. Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation https://onlinelibrary.wiley.com/doi/10.1155/2021/9674015
5. Opeyemi Sheu Alamu, Md Kamrul Siam. Sep. 2024. arxiv.org. Stock Price Prediction and Traditional Models.: An Approach to Achieve Short-, medium- and long-term goals. https://arxiv.org/pdf/2410.07220
6. Ningyi Li, Chennan Ju et al. Nov. 2023. Arxiv.org. Forecasting and Analysis of CSI 300 Daily Index and S&P 500 Index Based on ARMA and GARCH Models. https://arxiv.org/pdf/2312.14162
7. Vikram Krishnamurthy, Christian R. Rojas. Oct. 2024. arxiv.org. Slow Convergence of Interacting Kalman Filters in Word-of-Mouth Social Learning. https://arxiv.org/pdf/2410.08447
8. John Phan, Hung-Fung Chang. Oct. 2024. arxiv.org. Leveraging Fundamental analysis for Stock Price Prediction. https://arxiv.org/pdf/2410.03913
9. Camiel J.J. Beckers, Feye S.J Hoekstra. et al. Oct. 2024. arxiv.org. HiL Demonstration of Online Battery Capacity and Impedance Estimation with Minimal a Priori Parametrization Effort. https://arxiv.org/pdf/2410.03528
10. Duy Thien An Nguyen, Ka Ching Chan, Nov. 2024. International Journal of Information Management Data Insights. Cryptocurrency trading: A systematic mapping study. https://www.sciencedirect.com/science/article/pii/S2667096824000296
11. Jingyang Wu, Xinyi Zhang et. al. Jun. 2024. arxiv.org. Review of Deep Learning Models for Crypto Price Prediction: implementation and evaluation. https://arxiv.org/pdf/2405.11431
12. Micheal A. Lones. Jan. 2024. arxiv.org. How to avoid Machine Learning Pitfalls: A guide for academic researchers. https://arxiv.org/pdf/2108.02497v4
13. Bingqiao luo. Mar. 2024. arxiv.org. When Crypto Economics Meet Graph Analytics and Learning. https://arxiv.org/pdf/2403.06454v1
14. Vitalik Buterin. 2014. ethereum.org. A Next-Generation Smart Contract and Decentralized Application Platform. https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf
15. Mario Filho. Apr. 2023. forecastegy.com. Kalman Filter for Time Series Forecasting in Python. https://forecastegy.com/posts/kalman-filter-for-time-series-forecasting-in-python/
16. Danijel Jevtic, Romain Deleze et.al. Jun. 2022. Artificial Intelligence for Trading Strategies. https://arxiv.org/pdf/2208.07168
17. Merrill Liechty, Lance Stover. n.d. Dual Moving Average Crossover Strategy. Finance 453, Global Asset Allocation. https://people.duke.edu/~charvey/Teaching/BA453_2002/CCAM/CCAM.htm#:~:

text=In%20the%20dual%20moving%20average,of%20thought%3A%20Technical%20and%20Value.

9. Appendix

Fig. 1. Correlation plot between ETH-USD and BTC-USD



	Price	Adj Close		Close		High		Low		Open		Volume	
	Ticker	BTC-USD	ETH-USD	BTC-USD	ETH-USD	BTC-USD	ETH-USD	BTC-USD	ETH-USD	BTC-USD	ETH-USD	BTC-USD	ETH-USD
Price	Ticker												
Adj Close	BTC-USD	1.000000	0.976246	1.000000	0.976246	0.999264	0.976424	0.998932	0.974066	0.997905	0.974037	0.285438	0.485489
	ETH-USD	0.976246	1.000000	0.976246	1.000000	0.974784	0.998256	0.975541	0.997687	0.973768	0.995496	0.290589	0.497227
Close	BTC-USD	1.000000	0.976246	1.000000	0.976246	0.999264	0.976424	0.998932	0.974066	0.997905	0.974037	0.285438	0.485489
	ETH-USD	0.976246	1.000000	0.976246	1.000000	0.974784	0.998256	0.975541	0.997687	0.973768	0.995496	0.290589	0.497227
High	BTC-USD	0.999264	0.974784	0.999264	0.974784	1.000000	0.976924	0.998395	0.972748	0.998955	0.974814	0.302011	0.500550
	ETH-USD	0.976424	0.998256	0.976424	0.998256	0.976924	1.000000	0.975989	0.996446	0.976487	0.997927	0.316860	0.522607
Low	BTC-USD	0.998932	0.975541	0.998932	0.975541	0.998395	0.975989	1.000000	0.975863	0.998682	0.975362	0.263167	0.467466
	ETH-USD	0.974066	0.997687	0.974066	0.997687	0.972748	0.996446	0.975863	1.000000	0.973566	0.997118	0.259322	0.464722
Open	BTC-USD	0.997905	0.973768	0.997905	0.973768	0.998955	0.976487	0.998682	0.973566	1.000000	0.976576	0.287222	0.491059
	ETH-USD	0.974037	0.995496	0.974037	0.995496	0.974814	0.997927	0.975362	0.997118	0.976576	1.000000	0.295762	0.500655
Volume	BTC-USD	0.285438	0.290589	0.285438	0.290589	0.302011	0.316860	0.263167	0.259322	0.287222	0.295762	1.000000	0.885940
	ETH-USD	0.485489	0.497227	0.485489	0.497227	0.500550	0.522607	0.467466	0.464722	0.491059	0.500655	0.885940	1.000000

Fig. 2. Volume Trends for BTC-USD and ETH-USD

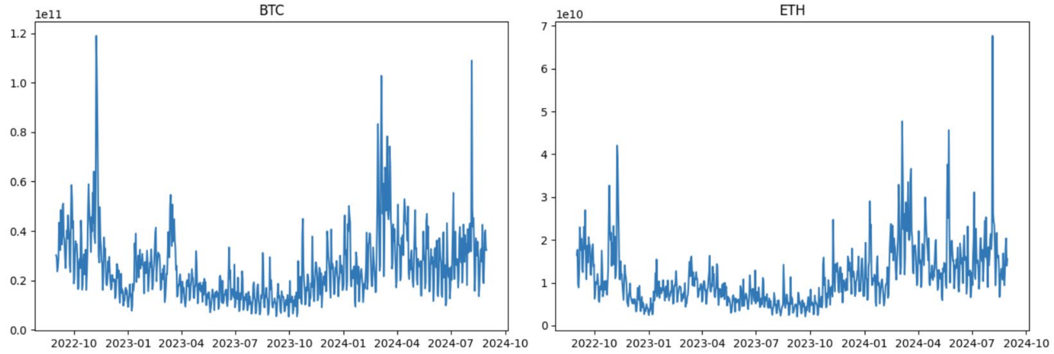


Fig. 3. First difference Volume Trend for BTC and ETH

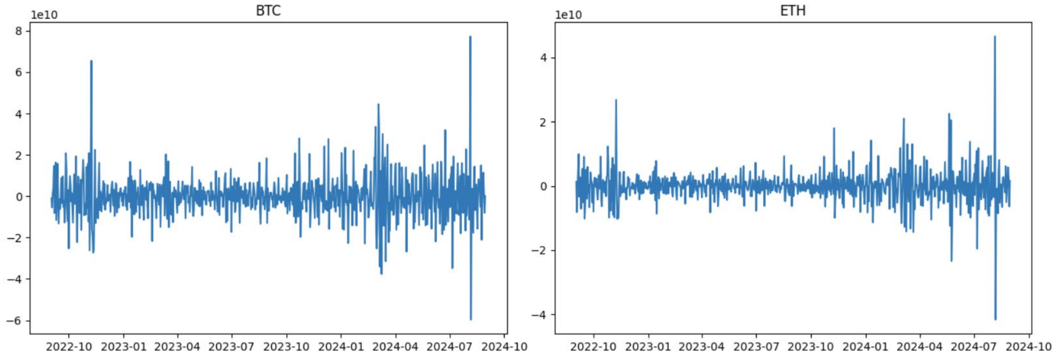


Fig. 4. Second difference Volume Trend for BTC and ETH

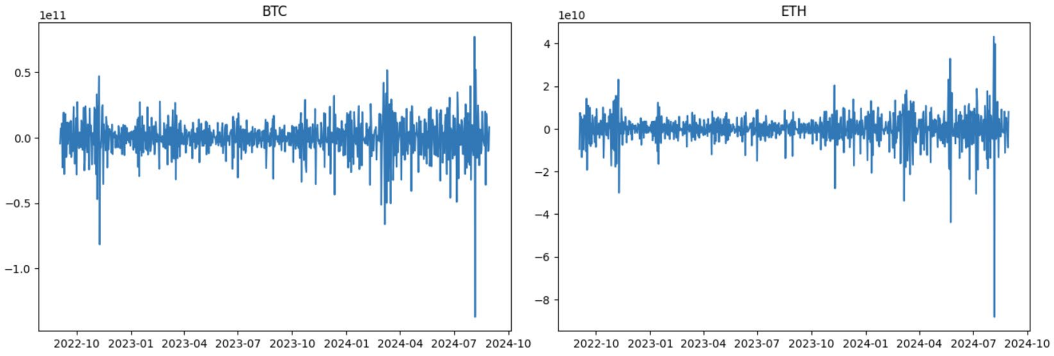


Fig. 5. Percentage change trend for Adj Close Price for BTC and ETH

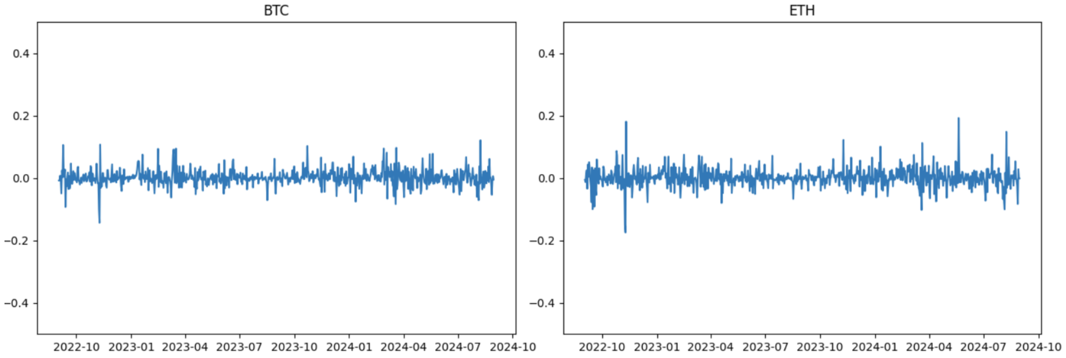


Fig. 6. Difference between Kalman Mean and rolling mean and stock price for BTC

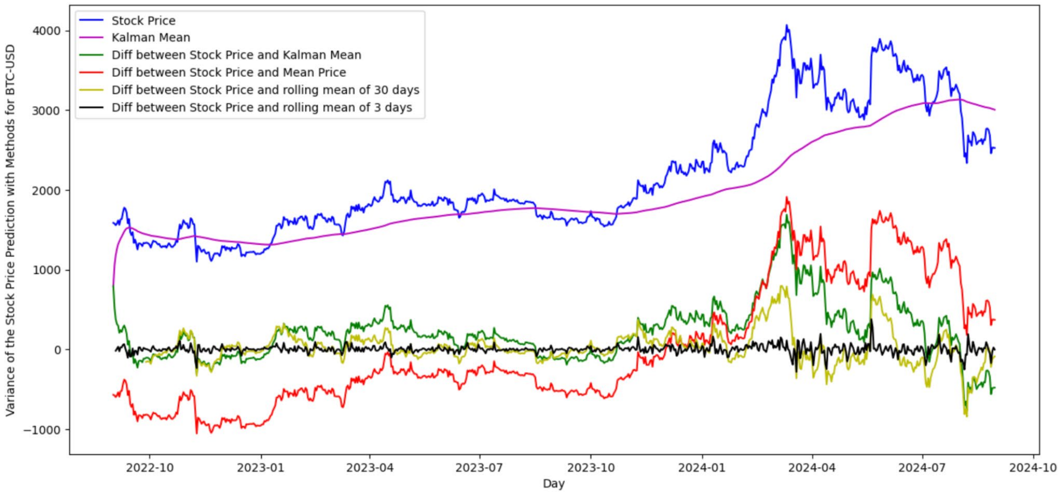


Fig. 7. Difference between Kalman Mean and rolling mean and stock price for ETH

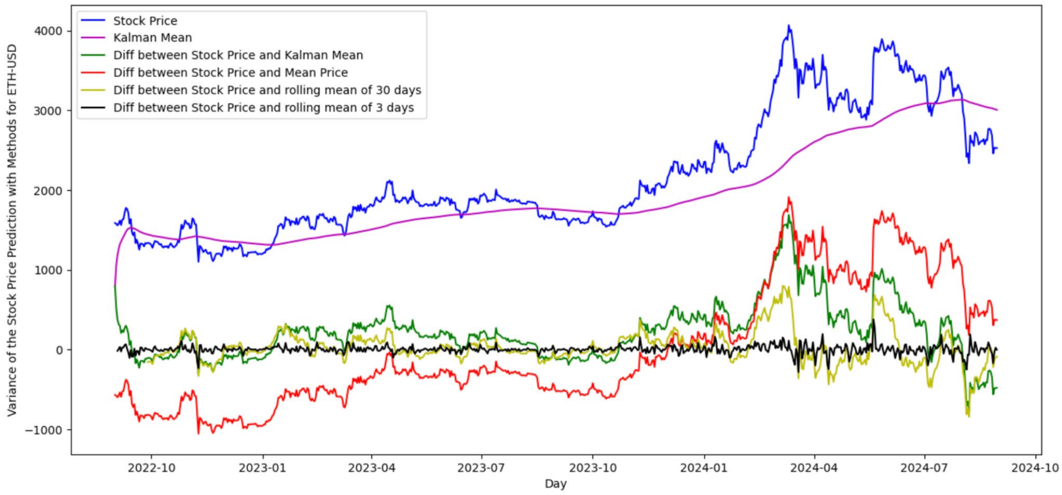


Fig. 8. Kalman Forecast for Adjusted Closing price for ETH for 100 Days

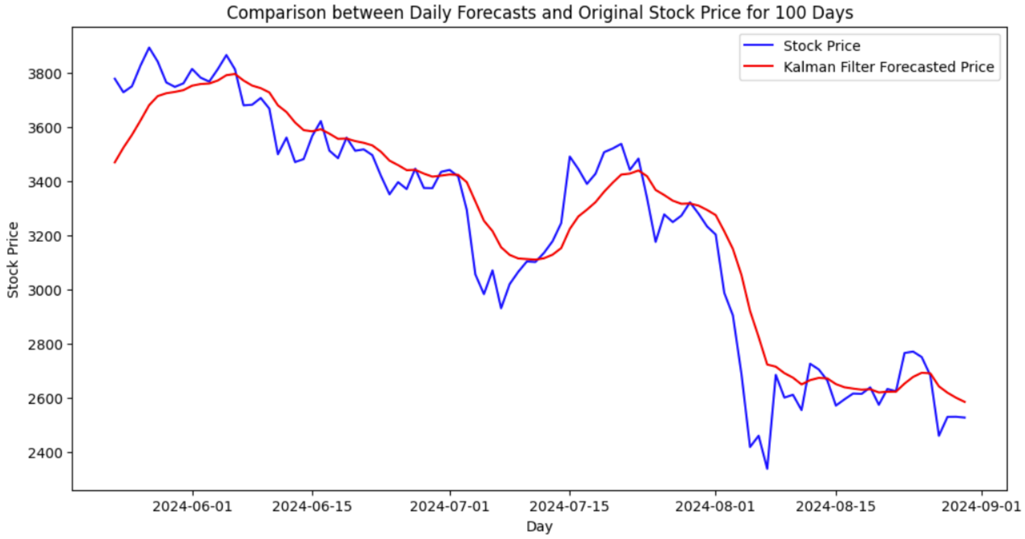


Fig. 9. Kalman Forecast for Adjusted Closing price for ETH for 50 Days

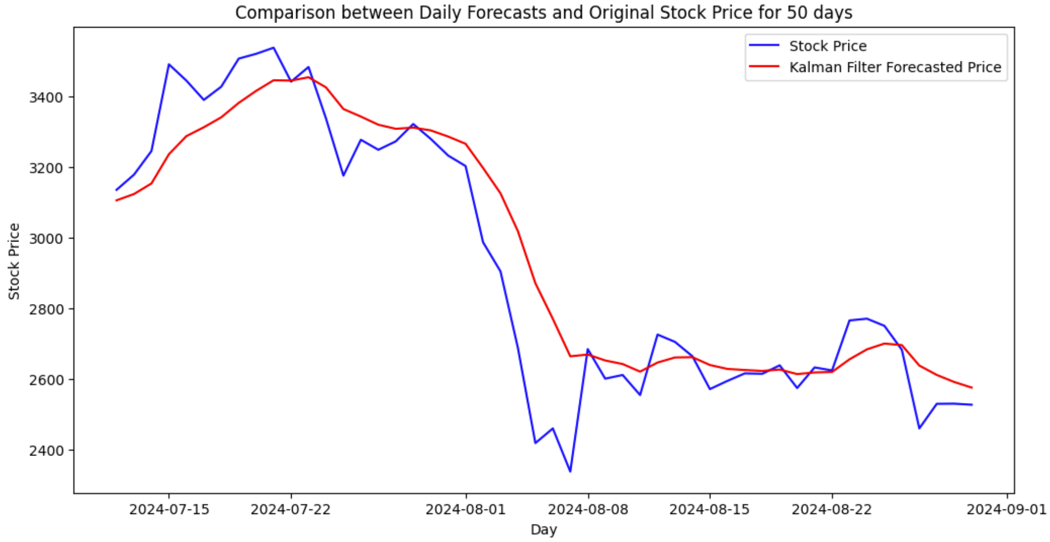


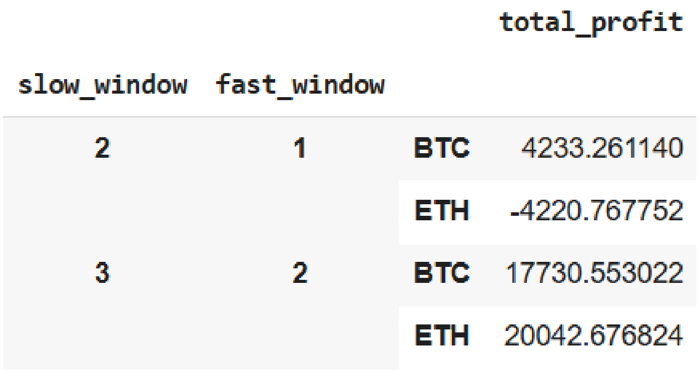
Table. 1. Kalman Filter – BTC-USD 'Adj Close Price' Prediction Model Performance Metrics

Metric	*200 Days*	*100 Days*	*50 Days*	*10 Days*
R-Square	**0.867**	**0.869**	**0.903**	**-0.37**
MAPE	**0.021**	**0.024**	**0.026**	**0.0103**
RMSE	**1874.731**	**2127.71**	**2336.574**	**814.062**
MAE	**1376.424**	**1589.83**	**1706.557**	**535.661**

Table. 2. Kalman Filter – ETH-USD Volume Prediction Model Performance Metrics

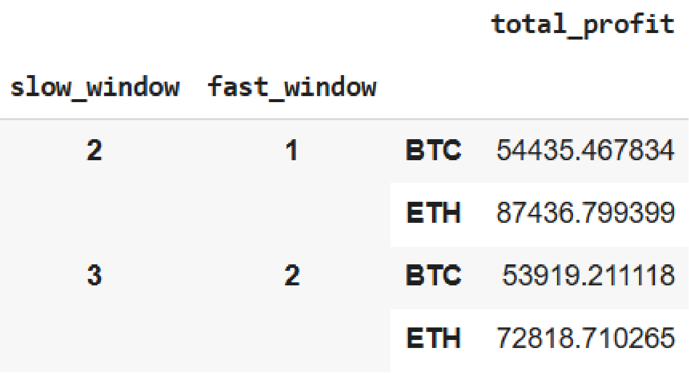
Metric	*200 Days*	*100 Days*	*50 Days*	*10 Days*
R-Square	**0.928**	**0.943**	**0.929**	**0.880**
MAPE	**0.081**	**0.0716**	**0.076**	**0.067**
RMSE	**2049202655**	**1813420368.387**	**2278519588.39**	**1137850706**
MAE	**1374597299**	**1144993804.71**	**1348608161.76**	**957798580**

Table. 3. Linear Regression DMAC Portfolio Strategy Total Profit



			total_profit
slow_window	fast_window		
2	1	BTC	4233.261140
		ETH	-4220.767752
3	2	BTC	17730.553022
		ETH	20042.676824

Table. 4. Kalman Filter DMAC Portfolio Strategy Total Profit



			total_profit
slow_window	fast_window		
2	1	BTC	54435.467834
		ETH	87436.799399
3	2	BTC	53919.211118
		ETH	72818.710265

Fig. 10. Total Return with different windows of Exit and Entry for Portfolio

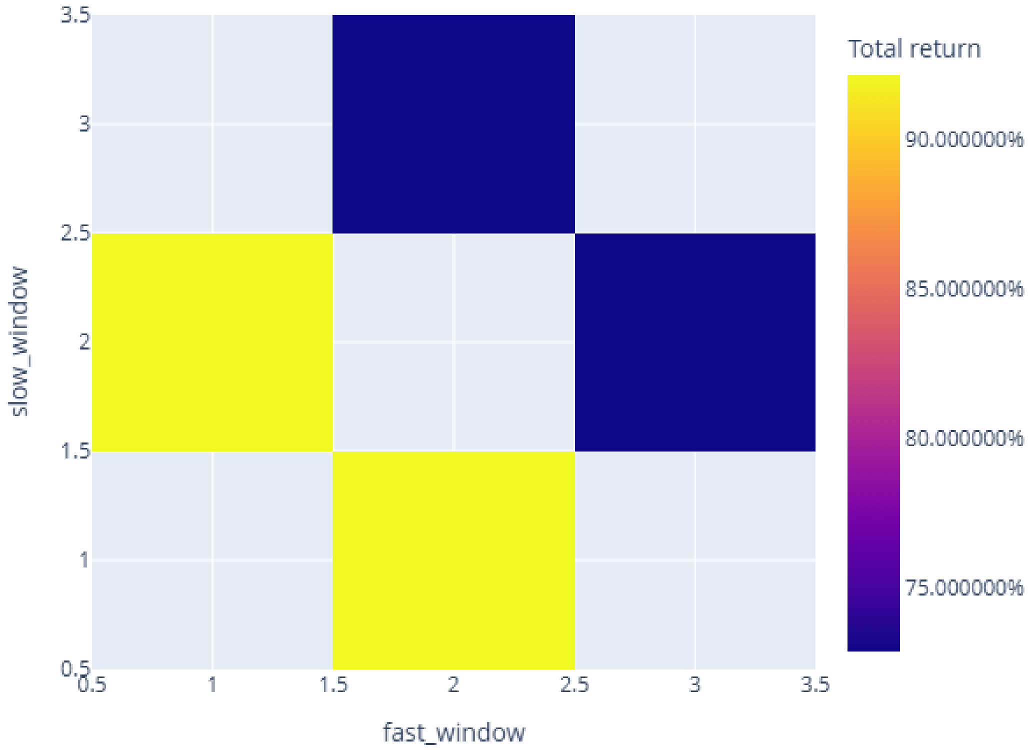


Fig. 11. Ethereum Portfolio Performance with Linear Regression Forecast

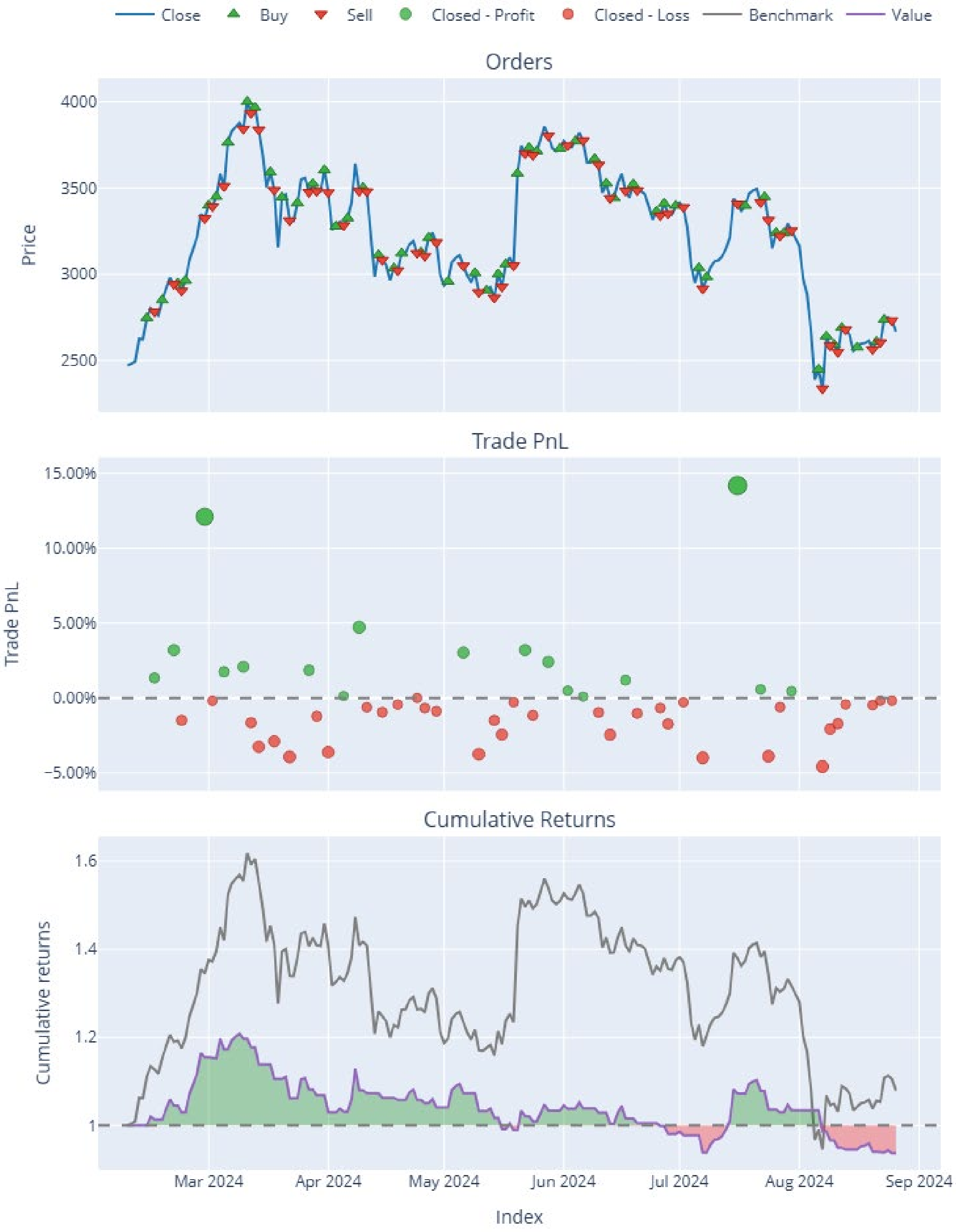
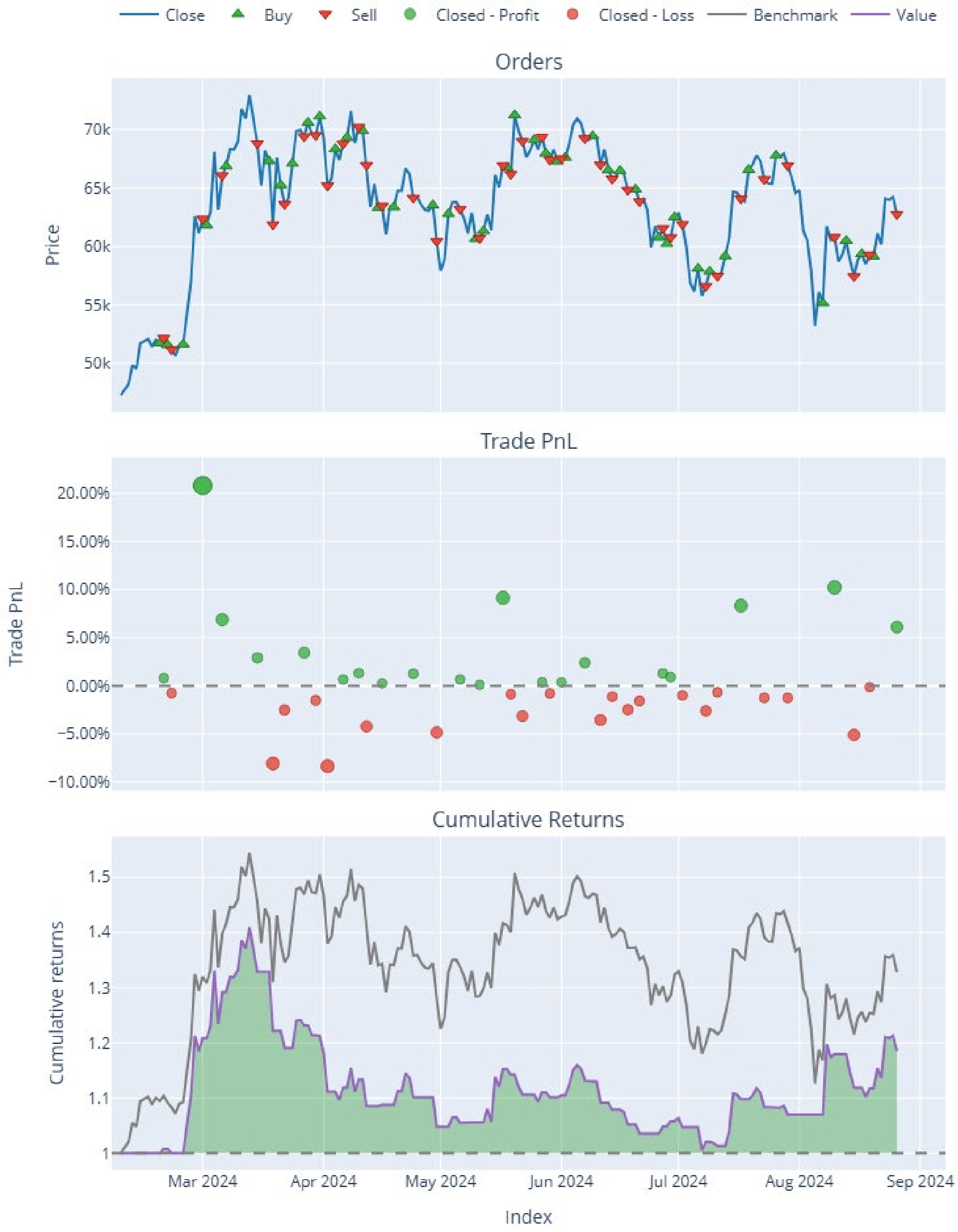


Fig. 12. Bitcoin Portfolio Performance with Linear Regression Forecast


Close
Buy
Sell
Closed - Profit
Closed - Loss
Benchmark
Value
Orders
Price
70k
65k
60k
55k
50k
Trade PnL
Trade PnL
20.00%
15.00%
10.00%
5.00%
0.00%
−5.00%
−10.00%
Cumulative Returns
Cumulative returns
1.5
1.4
1.3
1.2
1.1
1
Mar 2024
Apr 2024
May 2024
Jun 2024
Jul 2024
Aug 2024
Sep 2024
Index