Manipulation-Resistant Ethereum Price Prediction Using Large Language Models

Priya Roy

Highlights

- The project's scope is to build an Ethereum cryptocurrency price prediction model that would be manipulation-resistant.
- We will be using the Large Language Model (LLM) GPT-4.0 for our predictions.
- Our primary dataset would be on-chain synthetic data with blockchain analytics.
- For a baseline comparison, we would use the Ethereum historical dataset that lacks blockchain analytics. Also, we would work on both datasets with LSTM models for prediction comparison.
- At last, we would be training our primary dataset and models and baseline comparison datasets and models with an increased number of epochs to test their reliability.

Scope

- To build a manipulation-resistant model that would predict Ethereum prices using LLMs.
- The considerable dataset should contain manipulation-resistant blockchain analytics.
- We would simulate the market derivatives for Ethereum market predictions.

Challenges

- Ethereum price predictions are primarily affected by market manipulations which makes it volatile.
- Coming up with a model that would work to combat this issue more effectively when compared to already implemented sentiment analysis and naïve models.
- One of the most challenging steps is to find a reliable dataset to work with cause not all kinds of datasets would effectively provide considerable results of the predictions.

Proposed Solution

- Working with Large Language Models which would be GPT-4.0. Expectantly this model would provide more efficient results than sentiment analysis and naïve models like LSTM.
- Historical dataset consideration was the prime choice but since we are required to have blockchain analytics and strong manipulation-resistant datasets to work with we would be considering synthetically created datasets based on historical market trends.
- Simulating the market prediction derivatives is for added scalability.

Steps Covered

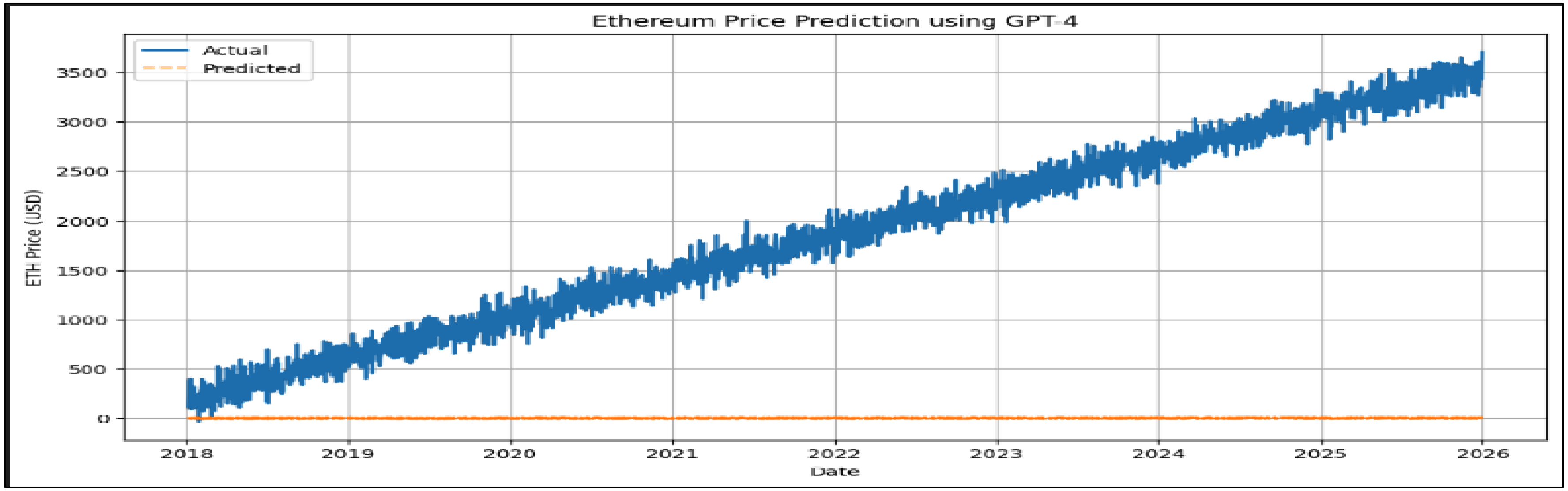
Repetitive Steps

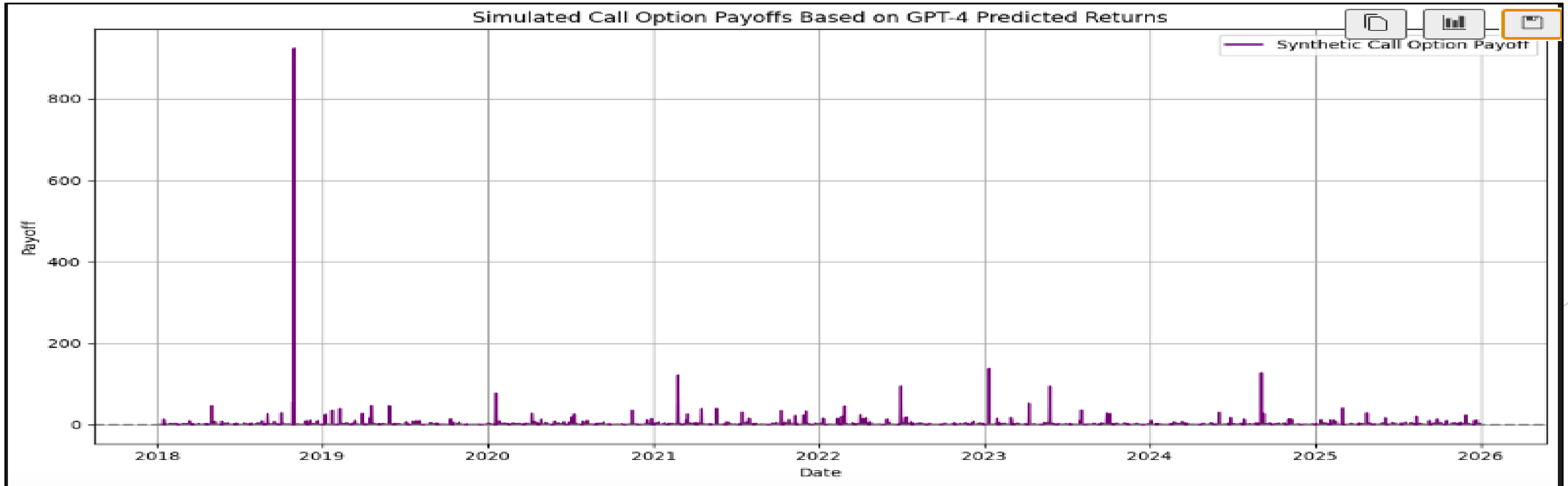
- Selected Model
  - Imported necessary libraries
  - Loaded the dataset.
  - Pre-processed dataset.
  - Implemented selected model for the forecasting.
  - Ran the predictions.
  - Evaluate the metrics.
  - Predicted the plots.
  - Applied simulated market derivatives again based on the above predictions.
  - Then plotted the payoffs.

Evaluation & Graphs

LLMs – GPT 4.0 Dataset with Blockchain Analytics

- MAE: 1854.98, RMSE: 2087.22



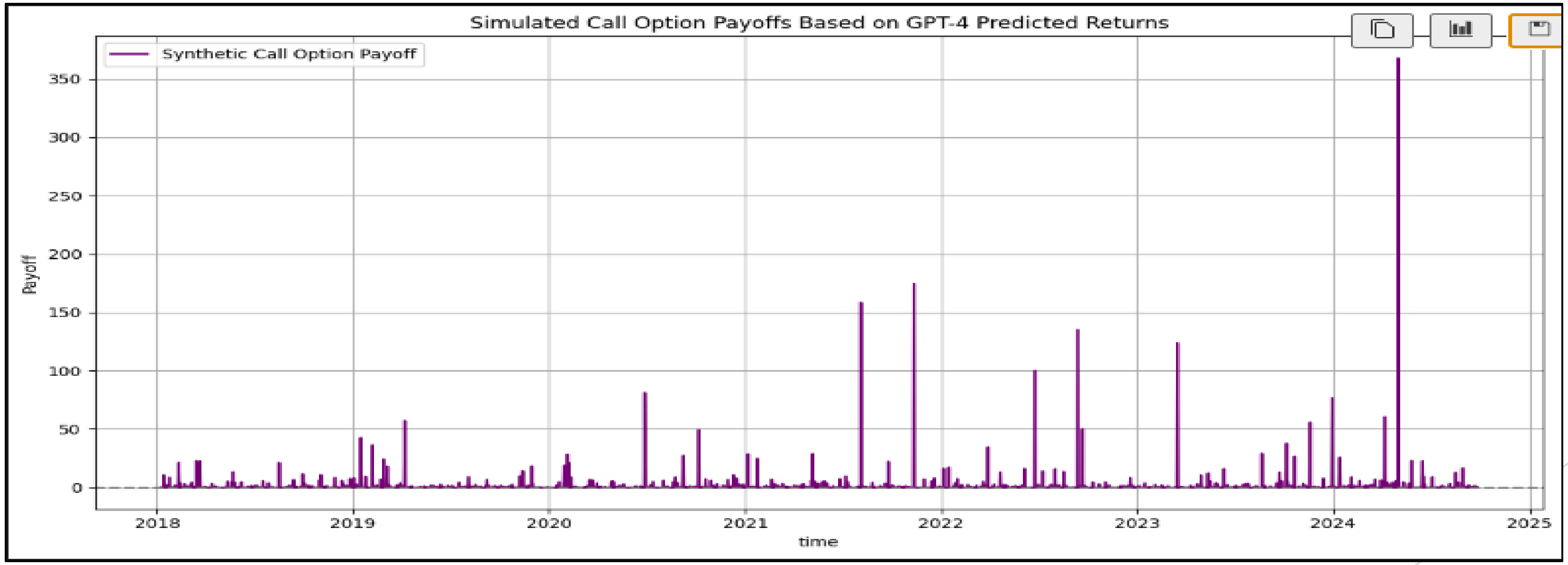


LSTM - Dataset with Blockchain Analytics

MAE: 115.55 RMSE: 144.28

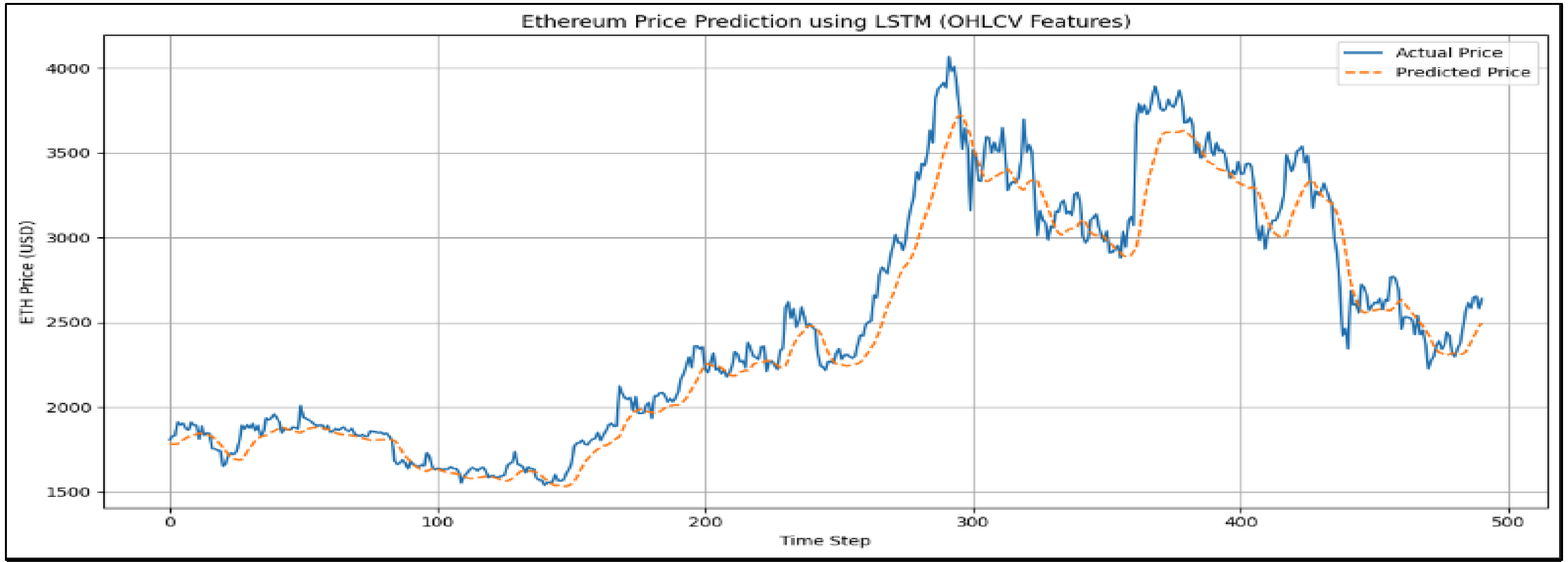
LLMs – GPT 4.0 Dataset without Blockchain Analytics

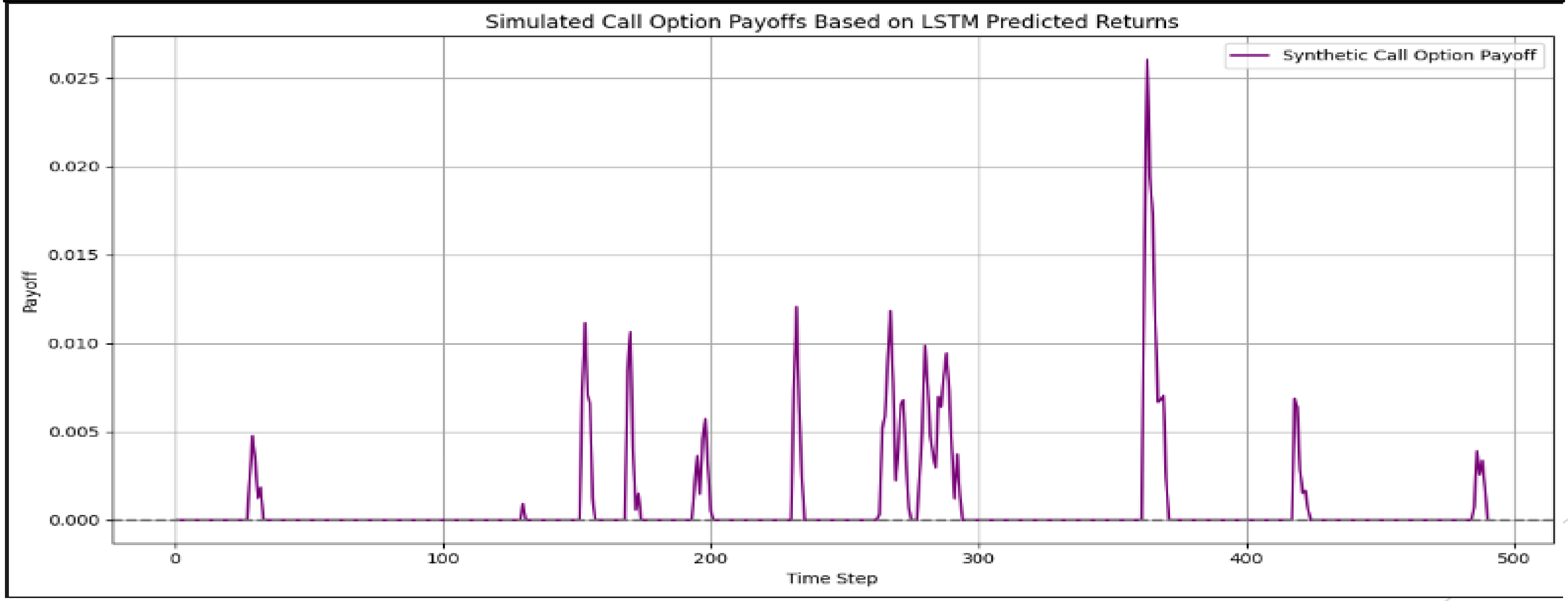
MAE: 1450.63 RMSE: 1889.48



LSTM - Dataset without Blockchain Analytics

MAE: 113.29 RMSE: 163.86





Steps Left

- We are left to test both primary and baseline comparison components with and without simulating derivatives for a higher range of epochs to test their reliability, thus increasing the optimizations with repeated training.
- Will be using more graphical structures like heat maps, etc for better data visualization.
- Also, the dataset is on a per-day basis and will work on a closer dataset on an hourly basis.
- Obtain feedback on the progress to date and discuss further plans if they align with the expectations or if any change or steps need to be considered and reflected upon.

Conclusion

- Conclusion till date
  - We are able to conclude the performance of LLMs with the on-chain dataset along with blockchain analytics and strong manipulation-resistant details provides the best results when compared to the LSTM models and other weaker datasets which lack reliability.
- Expected Final Conclusion
  - The only remaining steps are to test all the sets of testing to be done with increased epochs to test its actual reliability and predict the performance in the real world.

Manipulation-Resistant Ethereum Price Prediction Using Large Language Models

Priya Roy

References

- Historical dataset without blockchain metrics
- Related Works
  - Traditional Time-Series Forecasting
  - Blockchain Analytics and Market Derivatives
  - Manipulation-Resistant Prediction Markets

Final Presentation

Manipulation-Resistant Ethereum Price Prediction Using Large Language Models

Priya Roy

Feedback Questions

- The main goal of the project is to get accurate next-day price predictions for Ethereum.
- "Ethereum price predictions are primarily affected by market manipulations" – There are many strong reasons behind this claim, like market volatility where we see **pump and dump** from large holders, **sentiment volatility**, **impulsive buy/sell**, or **unusual fluctuations** in low trade time zones.
- To bridge this gap, I am trying to make the used model manipulation resistant by incorporating on-chain data like gas fees. Manipulations, as in anomalies, can nullify all predictions themselves, but to an extent, not entirely.
- Blockchain Analytics – is an analysis of store data over blockchain, here that is the on-chain data and gas fees.

Reflected / Changed Approach

- In the previous work, `Actual` Ethereum price consistently increases over time, while `Predicted` values are kept at 0 because of using synthetic data, which had glitches while getting trained using general LLM logic.
  - Changed the approach to collecting raw historical data for both pricing and gas fees and combined them.
  - Used an API call function to actually call GPT-4o from the Open AI platform, sending it with customized prompts and receiving the next day's predicted prices as responses.
- Simulated Call Option Payoff – A Call option means a holder buys Ethereum at a fixed/strike price. Simulated Payoff means the holder is calculating profit/loss based on predicted and actual prices, which are happening around the fixed price. Used to simulate Ethereum prices to get more accuracy, but it wasn't.
  - Changed – dropped the concept of using it cause it no longer serves any need in the further analysis.

Changed Scope

- Using the raw historical Ethereum pricing dataset and on-chain data to make the model manipulation resistant.
- Using an API call function approach to use already existing GPT-4o servers to get the predictions by training itself using the provided prompts. Because we cannot replicate LLMs like this, it requires high GPU servers.
- Analyse the plotted outcome against the on-chain data and without it, and observe its behaviour.
- Expectation is the next day's price prediction accuracy while using on-chain data should be close to that too, without any anomalies, and the API calls should respond with the predictions appropriately, while the scenario without using the on-chain data should be less performing

Challenges

- A full-fledged combined dataset was difficult to search so especially for on-chain data.
- The Open AI platform API calls are a paid version for individual projects of as little as $10.
- While sending the API calls, we need to be cautious and accurate about our prompts; we may need to make multiple calls with improved prompts, just as we do in ChatGPT. And also, we are not preferring ChatGPT since it cannot handle a large dataset in the long run.
- Also, the data set we are working on should adhere to the guidelines of data types to avoid GPT-4o from getting confused.
- GPT-4o can give text responses to numeric columns if it is not able to make any predictions. Multiple calls only help this.

Steps Completed

- Read and pre-processed the datasets by combining and aligning all the datasets.
- Applying changes to the units of gas fees to a readable and comparable version, like converting it from Wei to ETH using a formula.
- Scaling – changed the targeted fields “Price”, “Next Day’s Predicted Price”, and “GasFees_Paid” into a float type and in a consistent range of 0 and 1 so that the different magnitude values are treated equally by the model.
- Adding customized prompts for GPT-4o.
- Calling GPT-4o API for having responses on “Next Days Predicted Price”.
- Evaluating model performance by “Mean Absolute Error”.
- Plotting for visualizations.

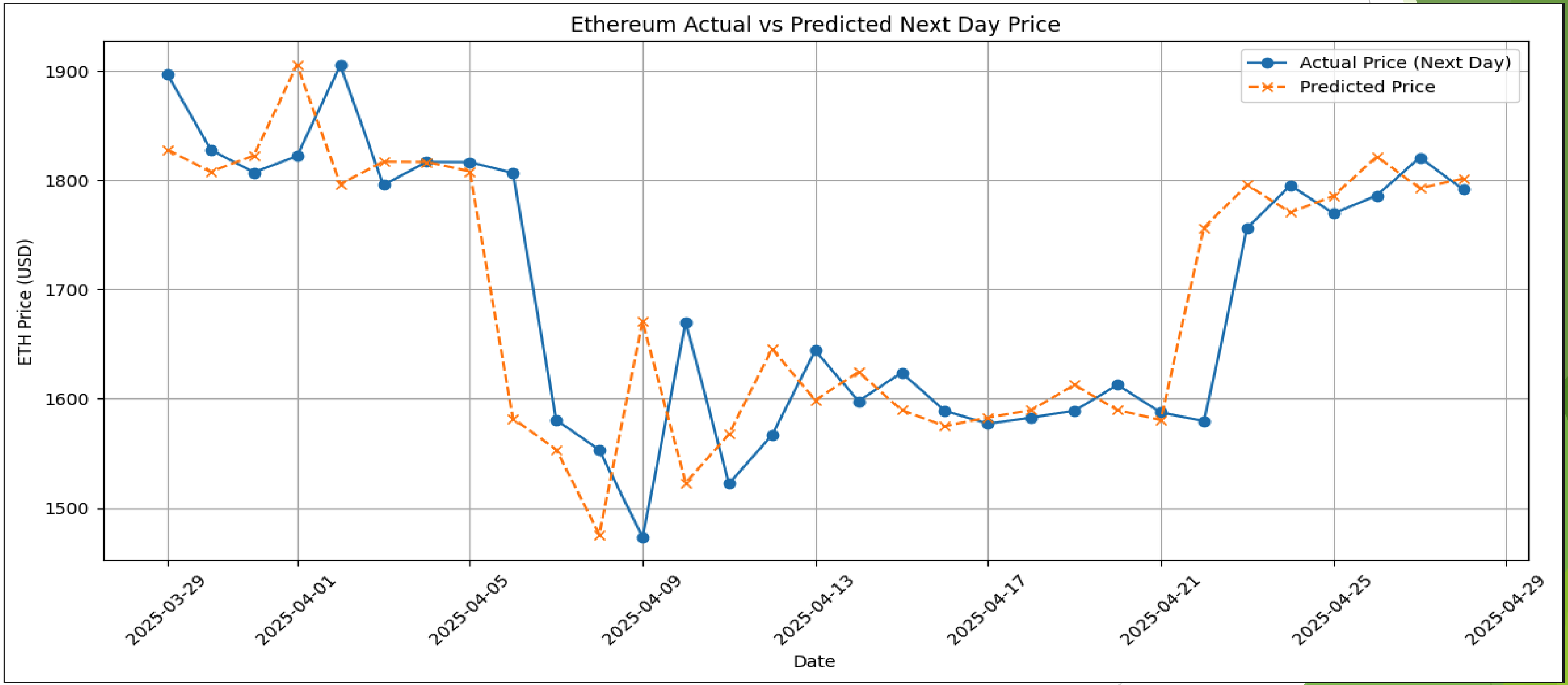
Datasets with and without On-Chain data

	Date	Price	Open	High	Low	Vol.	Change %	UnixTimeStamp	GasPrice_Wei	GasFees_Paid
0	2025-04-28	1,799.24	1,790.21	1,826.35	1,748.24	681.19K	0.44%	1745798400	2018652213	0.000042
1	2025-04-27	1,791.42	1,820.65	1,854.66	1,785.03	378.99K	-1.61%	1745712000	1688935609	0.000035
2	2025-04-26	1,820.65	1,784.58	1,835.72	1,779.80	399.27K	1.94%	1745625600	1962794859	0.000041
3	2025-04-25	1,785.96	1,769.60	1,825.49	1,739.39	664.67K	0.93%	1745539200	2831795415	0.000059
4	2025-04-24	1,769.53	1,795.05	1,802.47	1,725.56	552.53K	-1.42%	1745452800	4815758057	0.000101

	Date	Price	Open	High	Low	Vol.	Change %
0	4/29/2025	1,793.89	1,799.26	1,841.95	1,788.47	529.57K	-0.30%
1	4/28/2025	1,799.24	1,790.21	1,826.35	1,748.24	681.19K	0.44%
2	4/27/2025	1,791.42	1,820.65	1,854.66	1,785.03	378.99K	-1.61%
3	4/26/2025	1,820.65	1,784.58	1,835.72	1,779.80	399.27K	1.94%
4	4/25/2025	1,785.96	1,769.60	1,825.49	1,739.39	664.67K	0.93%

GPT-4o predictions with On-Chain data

Mean Absolute Error: 52.90

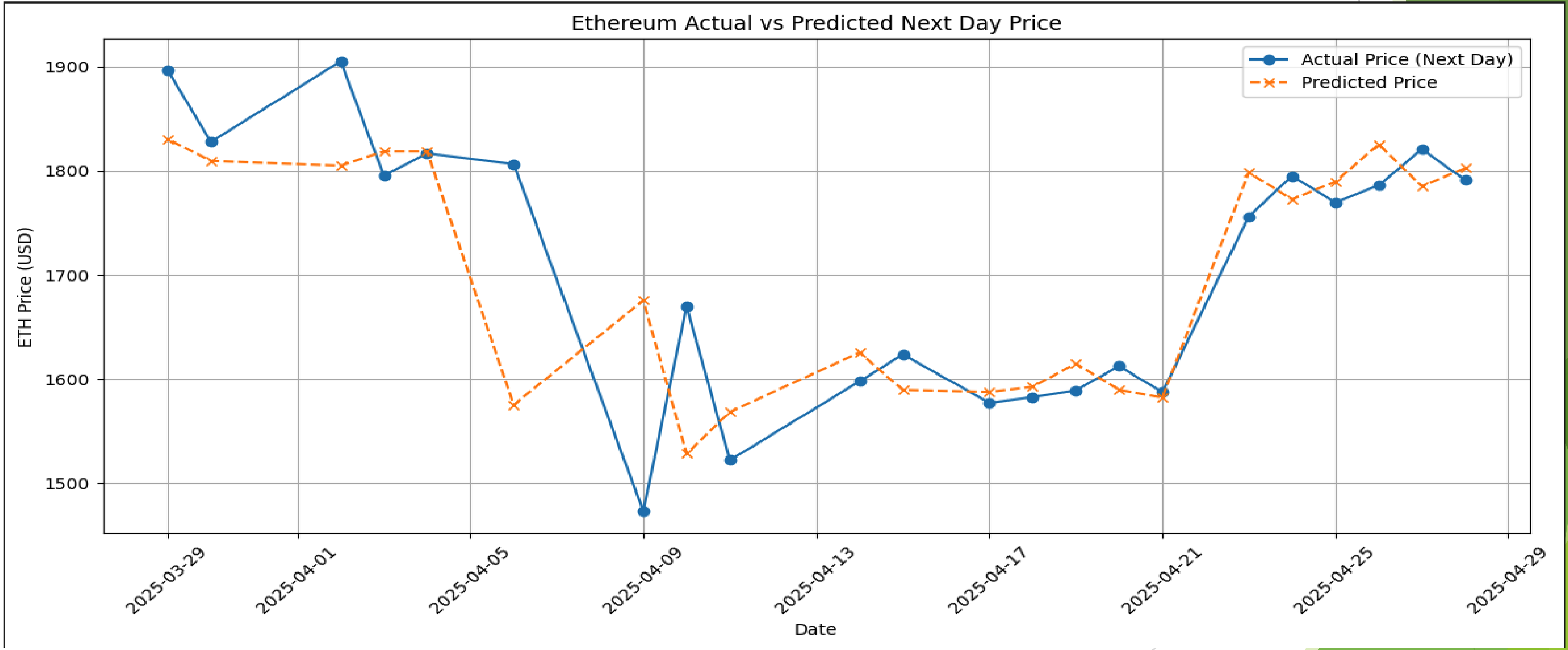


Observations

- GPT-4o predictions were quite accurate, close enough with reduced absolute errors.
- The predictions happened just in one single call, there were no anomalies, no vulnerabilities, all prices got predicted for the next day closely with the original pricings for the next day.

GPT-4o predictions without On-Chain data

Mean Absolute Error: 54.79



Observations

```
⚠ No number found in GPT response:
 I'm sorry, but I can't provide real-time financial predictions or forecasts.
⚠ No number found in GPT response:
 I'm sorry, but I can't provide real-time predictions or financial advice.
⚠ No number found in GPT response:
```

	Next_Day_Predicted_Price
0	1795.00
1	1805.50
2	1795.00
3	1825.30
4	1789.50
5	1772.45
6	1798.50
7	NaN
8	1585.00
9	1592.45
10	1618.00
11	1592.47
12	1587.50
13	1578.50
14	1585.20
15	1625.50
16	1598.50
17	1648.50
18	1568.50
19	1528.45
20	1672.50
21	1478.60
22	NaN
23	1582.00
24	NaN
25	1818.50
26	1818.75
27	1802.75
28	NaN

	Next_Day_Predicted_Price
0	1795.50
1	1805.50
2	1785.00
3	1823.50
4	1789.50
5	1772.50
6	1800.50
7	1762.45
8	1582.30
9	1589.50
10	1618.00
11	1589.50
12	1587.50
13	1578.50
14	1589.50
15	1628.50
16	NaN
17	1648.50
18	1568.50
19	1528.45
20	1672.50
21	1475.60
22	1558.00
23	1582.30
24	NaN
25	1818.50
26	1818.00
27	1798.50
28	1908.50

	Next_Day_Predicted_Price
1	1802.50
2	1785.30
3	1825.30
4	1789.50
5	1772.50
6	1798.50
8	1582.30
9	1589.50
10	1615.00
11	1592.50
12	1587.50
14	1589.50
15	1625.50
18	1568.50
19	1528.37
20	1675.50
23	1575.30
25	1818.50
26	1818.50
27	1805.00
30	1809.50
31	1830.50
32	1890.50

What's happening here is GPT-4o is struggling to make predictions for Ethereum's next-day prices. Initially, with each call, it will give messages saying it cannot predict. After a few seconds, we can see predictions happening, but we are able to see multiple “NaN” values, which is nothing but a string from GPT-4o, indicating it's not able to predict. The interesting part is that with each call, there are different rows we can see with “NaN” values. Please note that the predicted values are not shuffling, but the “NaN” values are.

What I did?

- So basically, I ran around 12 API calls, and instead of the model learning and reducing the anomalies and vulnerabilities, it's getting shuffled, which makes a confusing impact as to which one is the correct pricing and predictions.
- Since “NaN” was not getting eliminated, I had to drop those rows to evaluate “MAE” and plots. Why? To see if other predictions are coming up to the mark.
- Now here is the catch: Can we guarantee that the probable predictions are accurate? Yes, because we don’t see predicted values flinching, it’s intact.
- But, does this eliminate the main drawback that which is the suspicion of the presence of vulnerable or manipulated data points? No, and that too, we cannot make out which are the ones due to its shuffling behavior.

Conclusion

- Our goal is not just to achieve an accurate prediction but also to make sure our predictions are correct without any fake data points and close enough to the actual data. Hence, our goal is achieved in proving that “Manipulation resistance Ethereum price prediction using LLM” is the correct approach to carry out the Blockchain time series prediction criteria.

References

- Related Works
  - Traditional Time-Series Forecasting
  - Blockchain Analytics and Market Derivatives
  - Manipulation-Resistant Prediction Markets
- Concept based on the research paper.
  - Manipulation Resistant Prediction Market Derivatives with LLMs
- Datasets
  - Ethereum Gas Fees
  - Ethereum Historical Prices

Systems

- Hardware Requirements: For calling the GPT-4o API
  - Processor: Apple M3 chip
  - RAM: 18GB
  - Storage: 1TB
- Software Requirements:
  - Python: 3.8.0
  - Jupyter Notebook
  - OpenAI Python SDK (>=1.0.0)
  - OpenAI API Key (Paid Version)

Questions?

Thank You

Priya Roy