Comparative Graph Analysis on Ethereum: 'The Merge' and Gas Price Prediction

Bowen Fang
UNI:bf2504
bf2504@columbia.edu

Yunjie Qian
UNI:yq2354
yq2354@columbia.edu

William Gu
UNI:wg2400
wg2400@columbia.edu

Abstract

Blockchain-based cryptocurrencies have been one of the most attractive techniques in recent years. We set our sights on Ethereum, which is one of the most popular blockchains. Ethereum finalized 'The Merge' on September 15th, 2022, upgraded from the original PoW (proof of work) mechanism to PoS (proof of stake). Our goal is to analyze this big event for Ethereum through descriptive and predictive analysis. The novelty of our work comes from the fact that few previous works predict gas price through graph-based methods, and few provide data analysis of 'The Merge'. Our project will be useful to researchers, data scientists in Ethereum blockchain analysis.

1. Introduction

The Merge aimed to resolve the disadvantages of ETH 1.0 of low scalability and high energy consumption. By merging the PoS Beacon Chain to the main chain, the Merge reduced about 99.95% Ethereum's energy consumption. Our goal is to analyze this big event for Ethereum. We carry out descriptive analysis comparing data before and after the Merge. As for application, we present a GNN model based on the embedding of transaction networks. We are interested in the Gas Price, which is of great economic value, and 'The Merge' takes one step closer to sharding, which will increase the processing speed of the Ethereum mainnet and lower the transaction fees. The challenge we estimate we might have is that the Ethereum transaction network is heterogeneous and evolves temporally with high velocity.

We're applying both descriptive and predictive methods. First, we try to conduct descriptive and also comparative analysis on transaction graphs. After constructing transaction networks before and after "The Merge", we will compare network metrics, including global metrics like density and centralization, local metrics like cliques, and individual metrics like degree. Based on the comparison results, we will conduct visualization to discern patterns before and after "The Merge." For example, "The Merge" promotes the decentralization of transactions, which was the original intention of blockchain. Also, will "The Merge" bring about potential inflation of cryptocurrency and behavior shifts of users. These are the questions we are interested in.

In addition, we would like to conduct predictive analysis on gas price, which is a key indicator of a transaction and the key part to build advantages for users. Based on prior works, we intend to combine temporal and graphical information to improve prediction performance. To be specific, based on attention mechanisms, we apply graph attention convolution (GATConv) and global pooling to obtain embeddings of the entire graph, and causal Transformer to learn temporal dependencies.

2. Related work

Based on the large and heterogeneous data, there is a considerable amount of research on graph analysis of Ethereum. These studies focus on either descriptive or predictive questions that rely on graph-structured data to represent and solve.

In terms of descriptive analysis, representative studies include Motamed and Bahrak's [6] horizontal comparison of the nodes' and edges' scale in Ethereum's transaction graph with other platforms like Bitcoin, Litecoin, and Dash. Guo et al. [3] found that exhibited a heavy-tailed property and could be fitted with the power-law distribution. At present, a more comprehensive analysis of the properties of Ethereum's transaction network is lacking. Some common but insightful network metrics, such as degree distribution and centralization, have not been involved in the existing research. Moreover, the significance and impact of The Merge, as one of the Ethereum milestones, is still unknown. Given how recent it is, there are few studies on this event that explore and analyze the changes and influences it brought from the perspective of the network.

In terms of predictive analysis, some representative studies include Rawya and Amal's [5] machine learning approach for gas price prediction in Ethereum Blockchain. By

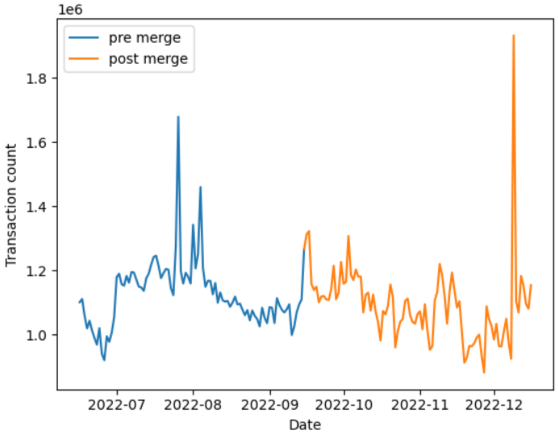


Figure 1. Daily Transactions distribution in the second half of 2022.

using prophet model and deep learning models Long-Short Term Memory model (LSTM) and Gated Recurrent Unit model (GRU) with transaction data, to help users choose an optimal gas price for an Ethereum transaction. However, there is a lack of previous work predicting gas price through graph-based methods, and few provide data analysis since 'The Merge'.

Therefore, we believe that graph analysis and graph-based prediction models of the Ethereum transaction graphs before and after The Merge is of innovative and practical significance.

3. Data

We collect transaction data from Ethereum open sources such as Ethpool and Ethermine, and use Ethereum ETL to manipulate and integrate our data into our database in Google Cloud Platform. Among all data tables, we typically focus on transaction data, we use BigQuery to query and download data within selected time range for description and prediction analysis. There are about 1.1 million transactions generated each day, and we collected about 250 GB over the past few months. The data contains 21 fields, including the Ethereum block information, timestamps, transaction from_address and to_address, fungible tokens (ERC20) and non-fungible tokens (ERC721), transfers, etc. By plotting the correlation matrix, we can observe that gas price affects a series of gas-related data and the transaction result, while the value transacted is very correlated to from_address and to_address, implying different clients having various magnitudes of transactions.

Then we apply simple Exploratory Data Analysis to the Transaction Data in the second half of 2022, which contains 3 months of data before and after the Merge. In Fig 1, We can observe that the transaction frequency is gradually dropping in the past 6 months, but the reduction is insignificant compared to the huge amount of transactions. Meanwhile, the amplitude of transaction frequency is more obvious after the Merge, and we can obverse larger outliers there.

4. Method

4.1. Descriptive Analysis: Transaction Graphs Comparison

Network-structured data can reflect the interactions between individuals. In order to discover the changes brought about by "The Merge" and interpret the significance of those changes from the individual level, we choose to construct two transaction networks, one before and one after "The Merge," and compare network metrics related to user behavior in Ethereum.

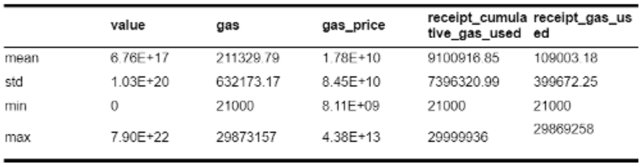
The transaction networks are constructed based on two months' data, August and October, one month before and one month after "The Merge." We regard one month as a period long enough to eliminate contingency and derive more general patterns. According to the field 'from_address' and 'to_address' in the datasets, we construct two directed and weighted graphs. Each edge represents the transaction relationship between two users. By aggregating features from our data, we also track some attributes of the transaction relationships, including the total transaction times, transaction value, and minimum gas price between two users.

Based on the two networks, we compared metrics as follows: node degrees, edge attribute distributions, and graph centralization, which measure the extent to which one or more nodes occupy the central positions of a graph. For node degrees and edge attribute distributions, we draw line charts through the visualization tool d3.js to discern overall patterns and changes. For graph centralization, due to computational complexity reasons, we choose to compute degree centralization among three types of centralization metrics.

Meanwhile, through d3.js, we conduct visualization of two smaller subgraphs, where we can also gain general insights in terms of the transaction structure on Ethereum. The two subgraphs for d3.js visualization are obtained by setting thresholds and filtering according to node degrees and transaction times between nodes, since the entire graphs are too large to be visualized.

The above methods respond to a question, that is, what happened before and after "The Merge." In addition, as a web application, we also provide users with answers to another question in this part, that is, what is happening now and recently on Ethereum.

Here, every day, we obtain the transaction information of the seven days before the current day with Airflow, including total transaction times, transaction values, as well as average and minimum gas prices. Through line charts



	value	gas	gas_price	receipt_cumulative_gas_used	receipt_gas_used
mean	6.76E+17	211329.79	1.78E+10	9100916.85	109003.18
std	1.03E+20	632173.17	8.45E+10	7396320.99	399672.25
min	0	21000	8.11E+09	21000	21000
max	7.90E+22	29873157	4.38E+13	29999936	29869258

Table 1. Description of feature scales. The scales vary vastly.

drawn with d3.js, we present users with timely information on transaction activities on Ethereum.

4.2. Predictive Analysis: Gas Price Prediction

4.2.1 Data Preprocessing

The first step is to aggregate transactions by time. The number of transactions per block can vary dramatically, ranging from 1 transaction to approximately 900 transactions in one single block. The vastly changing size of blocks presents challenges to accurately predicting the minimal gas price in each block. However, the number of transactions against time shows consistency. In practice, it is more important to consider the delay rather than the actual block. Therefore, we aggregate the transactions by a non-overlap sliding window with a length of 2 minutes. 2 and fig 3 compare the distribution of transactions in each block and in a time window of 2 minutes. Fig 2 are transactions count and distribution in each block, fig 3 are transactions count and distribution in a window of 2 minutes. What is observed is that the number of transactions in a consecutive time window has better distribution over that in each block.

The second step is data normalization. The raw data we collected has features on different scales. The gas price and value are both in GWei, and since 1 GWei equals 0.00000000119 ETH, both features contain great values. The value transferred on average is 6.76E+17 and the mean of gas price is 1.78E+10, while the average of gas is 2.11E+5. Data normalization can be crucial to the model's performance. We used Z-score normalization for each feature separately. The third step is to label the data. For each block, only transactions processed were recorded, so that the minimum gas price in each block is the lowest gas price for the transaction to be considered(in this block). After aggregation and normalization, we compute the minimum gas price m_i within each graph i. Then we compute $y_i = \min_{k \in \{i+1, i+2, \ldots, i+l\})} m_k$, which is the minimum of consecutive minimum gas prices.

The last step is to mini-batch along the diagonal for consecutive graphs to create a giant graph. The data loader is built on these giant graphs to preserve the order of sequence.

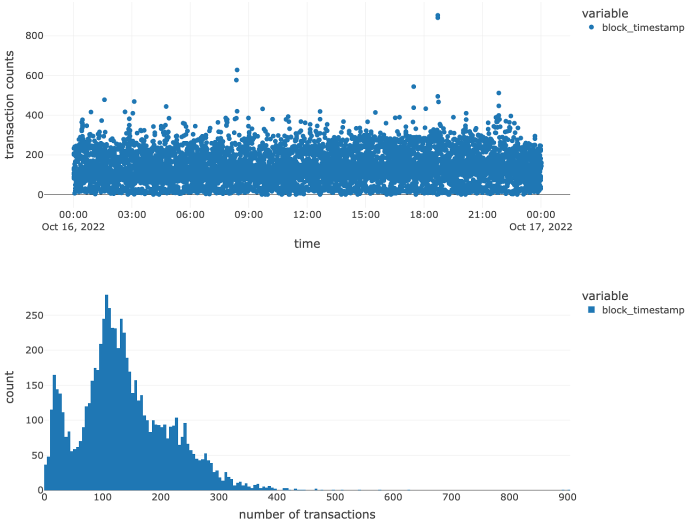


Figure 2. The upper is Number of transactions in each block over time. The lower is Distribution of number of transactions in each block.

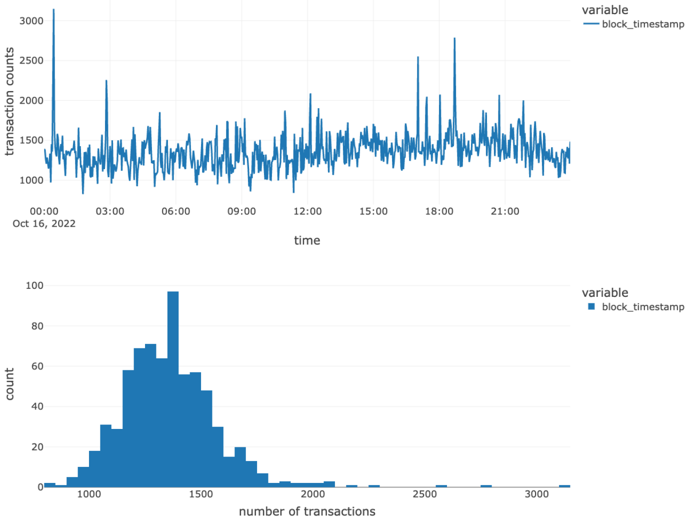


Figure 3. The upper is Number of transactions in a time window of 2 minutes. The lower is Distribution of number of transactions in a time window of 2 minutes.

4.2.2 Modeling: Node2vec

Node2vec [2] learns low-dimensional representations for nodes in a graph by optimizing a neighborhood preserving objective using biased random walks. Given any graph, it can learn continuous feature representations for the nodes, which could then be applied on downstream tasks. The transaction network of Ethereum lacks node features. Although we can generate statistic features, for instance, in/out degree, pagerank, etc, or assign each address(node) a trainable random vector, we finally decided to leverage node2vec method to generate node embedding for the reason that new addresses join the network frequently and a great proportion of addresses appears within 10 times and the majority appears only once. In such conditions, it is infeasible to directly train a random vector without neighborhood information.

4.2.3 Modeling: GAT

The Ethereum transaction network has rich edge features, including value transferred between nodes, the gas consumed during processing and gas price bid for the transaction. To process both the information of node and edge features, we use GATv2 operator from [1] work, which improves GAT [8] so that every node can attend to any other node and thus fix the static attention problem.

$$\mathbf{x}'_i = \alpha_{i,i}\mathbf{\Theta}\mathbf{x}_i + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\mathbf{\Theta}\mathbf{x}_j, \tag{1}$$

Where the edge features is represented as $e_{i,j}$, and the attention coefficient $\alpha_{i,j}$ are computed as:

$$\alpha_{i,j} = \frac{\exp\left(\mathbf{a}^{\top}\mathrm{LeakyReLU}\left(\mathbf{\Theta}[\mathbf{x}_i \,\|\, \mathbf{x}_j \,\|\, \mathbf{e}_{i,j}]\right)\right)}{\sum_{k\in\mathcal{N}(i)\cup\{i\}}\exp\left(\mathbf{a}^{\top}\mathrm{LeakyReLU}\left(\mathbf{\Theta}[\mathbf{x}_i \,\|\, \mathbf{x}_k \,\|\, \mathbf{e}_{i,k}]\right)\right)}. \tag{2}$$

4.2.4 Modeling: Casual Transformer

The last component of the proposed model is the Causal Transformer, which applies the causal mask to the multi-head self-attention module[7].
Let L be the size of an input sequence, the attention mechanism we use is the dot-product attention, which takes $Q, K, V \in \mathbb{R}^{L\times d}$ as inputs and has the form shown in eq.3

$$\begin{aligned} Attention(Q,K,V) &= \tilde{D}^{-1}\tilde{A}V, \\ A &= \exp{(QK^T/\sqrt{d})}, \\ \tilde{A} = tril(A),\ \tilde{D} &= diag(\tilde{A}1_L). \end{aligned} \tag{3}$$

Q, K, V are interpreted as query, key and value and $tril$ computes the lower triangular of the given matrix.

Further, to stabilize the training procedure, the LayerNorm is moved into the residual block and before the attention mechanism[9]. Fig 4 is a visualization of the architecture of the causal Transformer in the proposed model. The node features are processed layer by layer. A fully connected layer with tanh activation is applied for the Transformer's output for the reason that the target is Z-score normalized and could be negative.

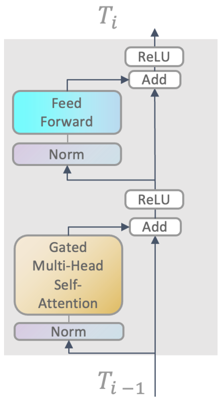


Figure 4. One layer of causal Transformer.

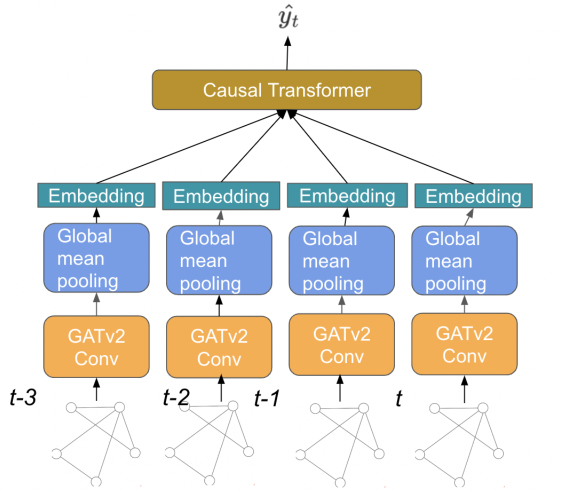


Figure 5. The structure of proposed model.

4.2.5 Proposed model: ETHGT

We proposed a novel model named ETHGT, whose structure is shown at fig 5. The model apples the same Graph Attention Convolution Blocks for a consecutive of graphs, the extracted features are passed through global mean pooling to get the embedding for the transaction graph. The embedding from each timestep is then fed to causal Transformer. The output is a single float number which represents the estimate lowest gas price in the next serval minutes.

5. System Overview

Fig 6 is an overview of the proposed system. For descriptive analysis of transaction networks, the sizes of the two obtained datasets are 33.83 gigabytes (GB) and 32.91 gigabytes (GB), respectively. In order to process data and

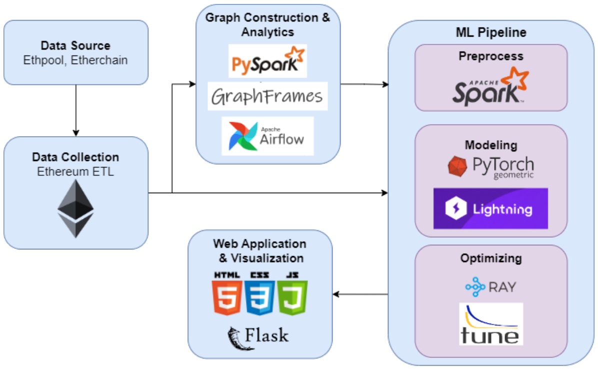


Figure 6. An overview of the system structure.

construct networks at that scale, we apply the GraphFrame package in PySpark. GraphFrame inherits the functions of GraphX and uses Spark DataFrame as its backend. Therefore, it is capable of constructing and storing large-scale graphs, which is in line with our goal for this part. Meanwhile, as mentioned above, Airflow is part of our system for fetching timely transaction information.

Considering the huge amount of data, we decide to start by picking some dates before and after the Merge, and compare its key features such as transaction addresses for descriptive analysis using Big Query. For predictive analysis, we'll first process the data with scikit-learn, which provides rich features in data preprocessing. Then we leverage the Pytorch geometric to build the Node2Vec model for node embedding. The proposed predictive model is written in Pytorch and with the help of Pytorch Lightning, the model could scale up easily. Our model is optimized with Ray Tune, which is capable of large-scale hyperparameter tuning, and finally, we build web application with Flask for data visualization.

6. Experiment

6.1. What Happened Before and After The Merge: Comparative Analysis of Pre-Merge and Post-Merge Graphs

After constructing the Pre- and Post-Merge graphs, we conducted comparative analysis of the two graphs by comparing graph metrics of different levels, including individual metrics like in-/out-degrees and edge attribute distributions, and global metrics like degree centralization.

6.1.1 Individual Metrics

In terms of individual node metrics, we compared the degree distribution of the Pre- and Post-Merge graphs.

Fig 7 plots the Cumulative Distribution Function (CDF) of node degrees of the two graphs. Fig 8 plots the Cumulative Distribution Function (CDF) of the weighted node degrees, namely total transaction times of a user, of the two graphs.

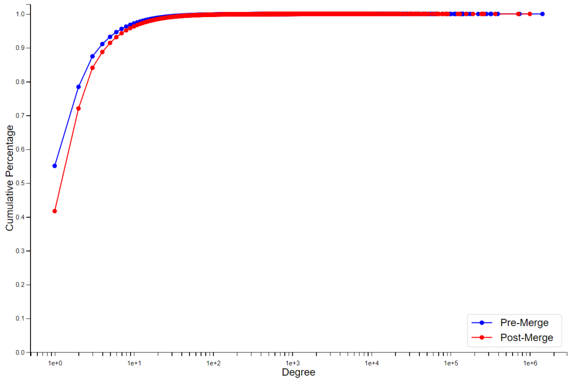


Figure 7. Cumulative Distribution Function (CDF) of Node Degrees in Transaction Graphs (Log Scale).

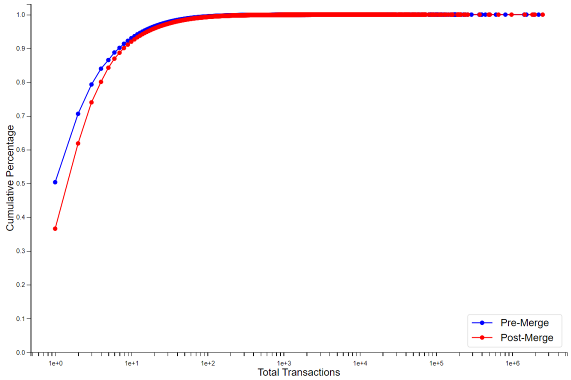


Figure 8. Cumulative Distribution Function (CDF) of Users' Transaction Times / Weighted Node Degrees (Log Scale).

The two figures present a similar pattern. It can be seen that at a rather small (weighted) degree, the Post-Merge red lines are always below the Pre-Merge blue lines, which suggests that the proportion of users who conduct few transactions or conduct transactions with few users is shrinking. Overall, users are more active in developing relationships with other users in Ethereum after The Merge. In fact, the average degree of nodes rose from 4.33 before The Merge to 4.86 after The Merge, while the average transaction times of users rose from 8.93 to 10.35.

In terms of individual edge metrics, we compared the total transaction value and minimum gas price of the Pre- and Post-Merge Graphs. Fig 9 and 10 plots the Cumulative Distribution Function (CDF) of the total transaction value and minimum gas price before and after The Merge.

In Fig 9, the red and blue lines are almost overlapping. However, the Post-Merge red line is still slightly below the

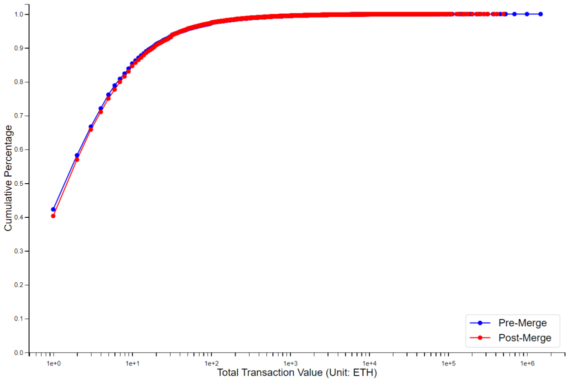


Figure 9. Cumulative Distribution Function (CDF) of Transaction Values between Users (Log Scale).

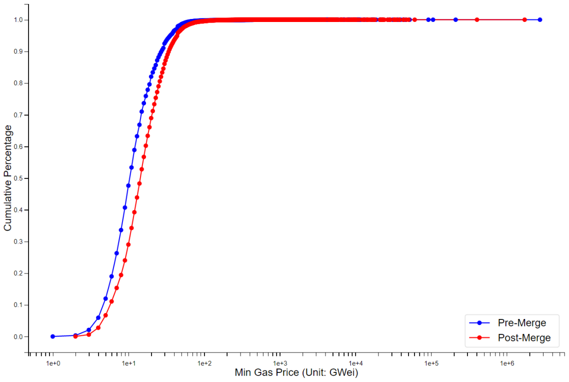


Figure 10. Cumulative Distribution Function (CDF) of Minimum Gas Price between Users (Log Scale).

Pre-Merge blue line at the beginning, indicating that after The Merge, users are less likely to conduct transactions of fewer values. The trend is verified by the decreased average transaction value from the Pre-Merge 4.730 to the Post-Merge 4.851 (unit: Ether).

In Fig 10, it can be seen that at a low price, the Post-Merge red line is below the Pre-Merge blue line, which suggests that the proportion of transactions with low gas price decreased after The Merge. The trend is verified by the increased average gas price from the Pre-Merge 14.99 to the Post-Merge 19.11 (unit: GWei). This indicates that after The Merge, there may be a growing extra cost paid for every transaction.

To sum up, the comparison of those individual-level network metrics reveals a potential behavior shift of users. Users now become more engaged in blockchain transactions on Ethereum, developing relationships with users more widely and frequently on the platform. Meanwhile, the value of transactions between users is generally on the rise. However, despite the rising figures of transaction times and values, users are also paying much more extra money, namely the rising gas price, to get engaged in transactions.

6.1.2 Global Metrics

In terms of global graph metrics, we compared the centralization of the Pre- and Post-Merge Graph. In network science, centrality measures the positional property at the node level, while centralization measures that at the graph level. In specific, graph centralization measures the difference in node centrality and reflects the extent to which one or more nodes occupy the central positions of the graph.

Here, we adopt degree centralization in our comparative analysis, which indicates the difference in the degrees of nodes. The smaller the value is, the more decentralized the graph is, and vice versa. The computation formula for degree centralization is as follows:

$$C^d = \frac{\sum_i c^{d*} - c_i^d}{\max \sum_i c^{d*} - c_i^d} = \frac{\sum_i c^{d*} - c_i^d}{(n-1)(n-2)}$$
$$\text{where } c_i^d = \text{degree centrality of node } i,$$
$$c^{d*} = \text{maximum degree centrality of all nodes.}$$

The computation result shows that degree centralization exhibited a quite large extent of decrease after The Merge. Degree centralization falls from the Pre-Merge 0.093 to the Post-Merge 0.076. This result indicates that users of Ethereum did participate in transaction activities more equally after The Merge, since a smaller proportion of nodes occupy the central positions. The Merge, which was intended to promote finance decentralization, may have been successful in doing so.

6.1.3 Overall Graph Pattern

Through d3.js, we conduct visualization of two smaller subgraphs of the original Pre- and Post-Merge graphs, where we can discern patterns and gain insights in terms of the overall transaction structure on Ethereum. The subgraphs are obtained through setting thresholds of at least 30 degree and at least 10 transaction times, and filtering the original graphs.

Figure 11 and 12 shows that there is no significant change in terms of the overall transaction structure on Ethereum. The transaction graph is divided into several components. Among them, the largest component will have an obvious central node, and many other nodes around it will develop transaction relationships with it. A possible guess is that this node is an important financial institution, who attracts many individual or organizational users to conduct transaction with it on Ethereum. In terms of the other components of the graph, some have a center and present a ”core-periphery” structure, while others are more decen-

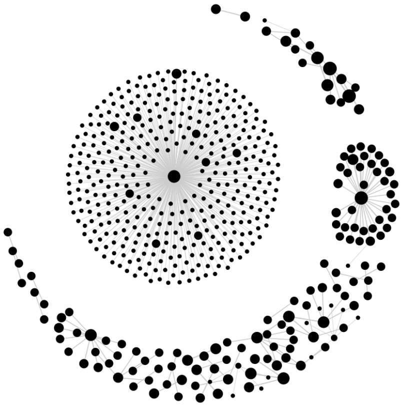


Figure 11. Visualization of the Pre-Merge Graph (Node size represents its transaction times).

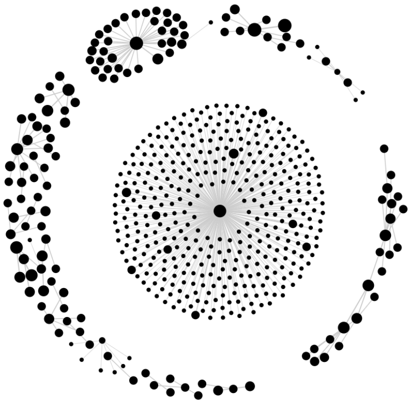
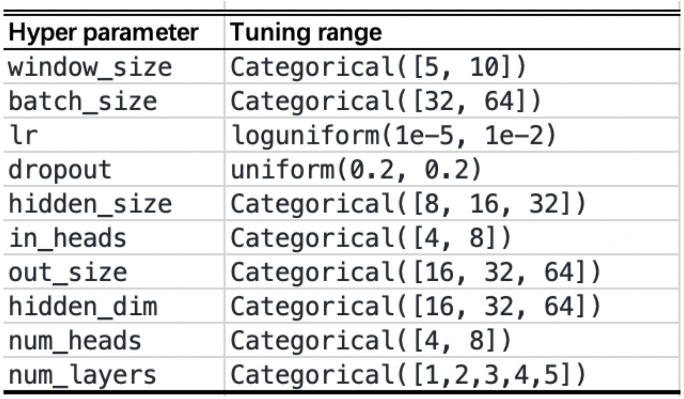
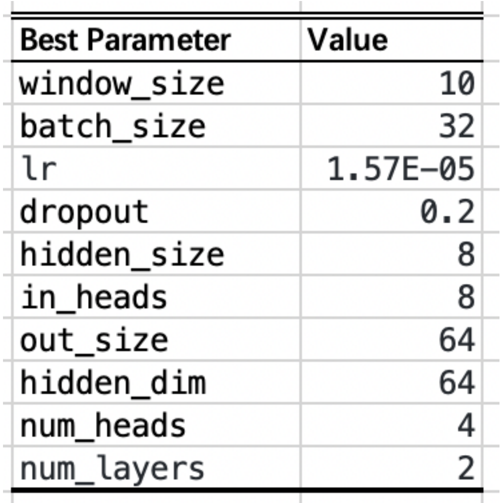


Figure 12. Visualization of the Post-Merge Graph (Node size represents its transaction times).



Hyper parameter	Tuning range
window_size	Categorical([5, 10])
batch_size	Categorical([32, 64])
lr	loguniform(1e-5, 1e-2)
dropout	uniform(0.2, 0.2)
hidden_size	Categorical([8, 16, 32])
in_heads	Categorical([4, 8])
out_size	Categorical([16, 32, 64])
hidden_dim	Categorical([16, 32, 64])
num_heads	Categorical([4, 8])
num_layers	Categorical([1,2,3,4,5])

Table 2. Hyper parameter tuning range.



Best Parameter	Value
window_size	10
batch_size	32
lr	1.57E-05
dropout	0.2
hidden_size	8
in_heads	8
out_size	64
hidden_dim	64
num_heads	4
num_layers	2

Table 3. Hyper parameter tuning range.

tralized and every node enjoy quite equal status in the transaction structure.

6.2. What is Happening Now and Recently: Daily Transaction Monitoring

As a web application, we also provide users with answers to another question, what is happening now and recently on Ethereum. With Airflow, we schedule a task per day. The task is to obtain the transaction data of the week before the current day, including the total transaction times, values, as well as the average and minimum gas price of transactions per day.

To be specific, we combine PySpark with Airflow to complete the task of obtaining transaction data. Because PySpark does not support multiple sessions, we encapsulate the task of obtaining and storing transaction data in one single function in Airflow. The logic and execution flow of this Airflow function are shown in Fig 13 as a directed acyclic graph (DAG).

First, we fetch data from our dataset using BigQuery and store it in BigQuery as well. Then, we process the data to get edge and node data successively, and write them to BigQuery. In the next step, we get a subgraph by setting thresholds to filter nodes and edges in the original graph. Further on, we carry out visualizations of distributions of daily transaction times, total value, and gas price with line charts in HTML. Finally, we store the visualization charts to Google Cloud Storage bucket in HTML format.

The HTML files in GCS bucket can be connected to the front-end of our system, showing users the transaction information as well as potential changes on Ethereum in the recent week.

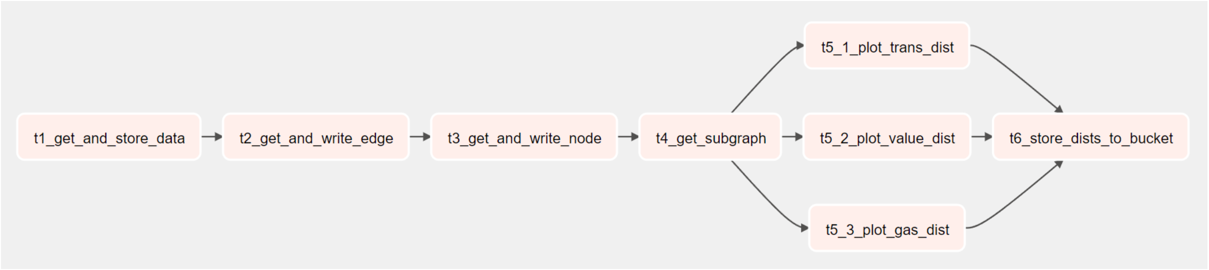


Figure 13. The Execution Flow of the Airflow Function.

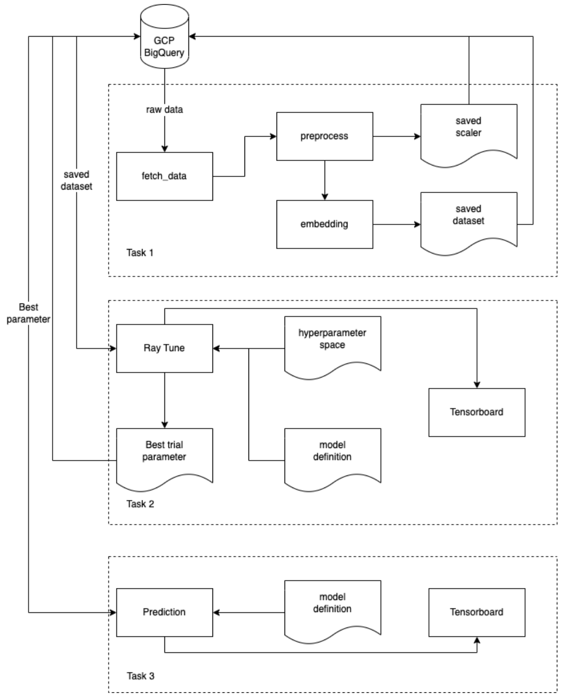


Figure 14. The workflow of training and tuning the proposed model ETHGT.

6.3. What May Happen Next: Gas Price Prediction

Fig 14 is the visualization of our training and tuning work flowx. We present the experiment result in this section. We use population-based training(PBT)[4] scheduler to efficiently search for best set-up. PBT trains and optimises a series of networks at the same time, allowing the optimal set-up to be quickly found. The parameter space is shown at Table 2 and the best parameter we found is displayed at Table 3

To evaluate the performance of the proposed model, we used Mean Square Error(MSE) $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$. After 500 episodes, we observed that the train loss curve fig 16 and validation loss curve fig 17 both drop as the training continues. We apply the trained model to predict on the test dataset (fig 18), where we discovers that the predicted value is close to the real value and has the similar trend as the real value. Therefore, the model can provide informative gas price prediction for users.

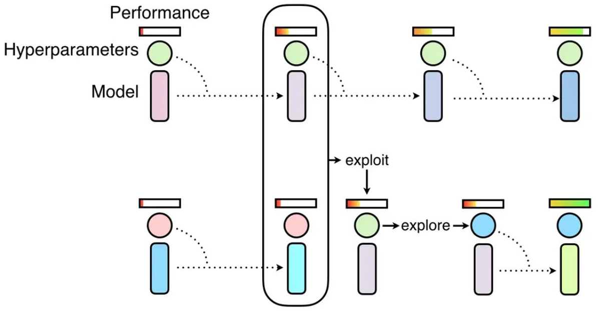


Figure 15. Population Based Training of neural networks starts like random search, but allows workers to exploit the partial results of other workers and explore new hyperparameters as training progresses.

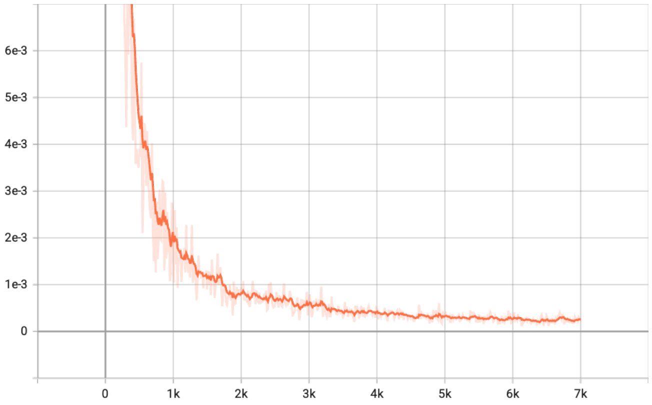


Figure 16. Train loss curve of ETHGT.

7. Conclusion

In this project, we have conducted both descriptive and predictive analysis of the transaction network on Ethereum,

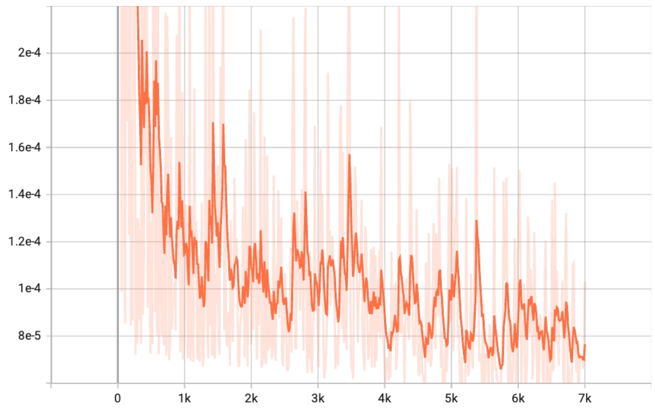


Figure 17. Val loss curve of ETHGT.

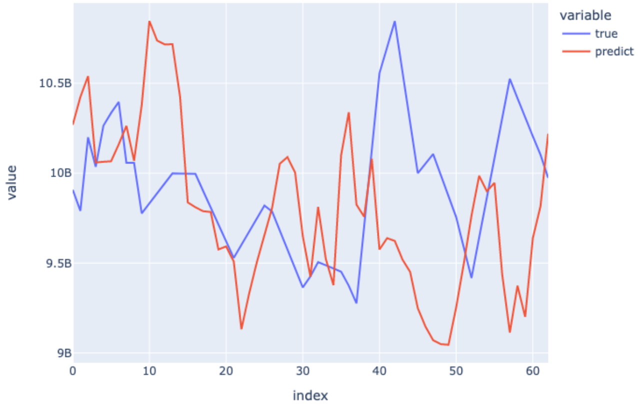


Figure 18. Prediction on the test dataset.

providing insights on what happened before and after ’The Merge’, what is happening recently, and what may happen next in terms of gas price, which is the key to a successful transaction.

In terms of descriptive analysis, we carry out comparative analysis on two months’ transaction graphs. Having constructed graphs through PySpark, GraphFrames, and BigQuery, we compare graph metrics including node degrees, edge attributes, and graph centralization. We reveal that there may be a behavior shift for users. Users now become more engaged in blockchain transactions after The Merge on Ethereum, but they are also paying much more extra money to get engaged in transactions.

Moreover, we adopt Airflow to schedule an everyday task of obtaining the data and exhibiting the trends of transactions in the recent week, helping users to monitor transaction information and make related decisions.

In terms of predictive analysis, we proposed the model ETHGT for gas price prediction based on the transaction graph through Ray and Torch. By leveraging the Ray Tune PBT scheduler, we searched the best set-up on scale and trained an efficient model to predict the lowest gas price in the next 10 minutes, which realizes a MSE below 8e-5 and thus is competitive with other models.

References

[1] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021. 4

[2] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016. 3

[3] Dongchao Guo, Jiaqing Dong, and Kai Wang. Graph structure and statistical properties of ethereum transaction relationships. *Information Sciences*, 492:58–71, 2019. 1

[4] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017. 8

[5] Rawya Mars, Amal Abid, Saoussen Cheikhrouhou, and Slim Kallel. A machine learning approach for gas price prediction in ethereum blockchain. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 156–165. IEEE, 2021. 1

[6] Amir Pasha Motamed and Behnam Bahrak. Quantitative analysis of cryptocurrencies transaction graph. *Applied Network Science*, 4(1):1–21, 2019. 1

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[8] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 4

[9] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020. 4