Deep Learning Ethereum Token Price Prediction on Dynamic Network and Time Series Analysis

Ziqiao Ao [1]*, Jiayi Li [1], Haoxin Yu [1]

Nature Science Department Duke Kunshan University Kunshan, China
**a36@duke.edu, jl884@duke.edu, hy156@duke.edu*

Abstract

Ethereum has recently surged in popularity, as it can hold various digital tokens and decentralized applications. This paper aims to predict UNI's price in USD through dynamic network analysis and time-series analysis. Previous research in this field rarely considers comprehensive network analysis while predicting token price. This paper puts forward a strengthened Bidirectional LSTM model that includes token economical features and network features. We use Root Mean Squared Error (RMSE) to verify the validity and compare it with other LSTM and GRU models on performance. Lastly, a logarithm difference method for data preprocessing was introduced to resolve the lag problems.

Keywords-*Blockchain; Ethereum ERC20; Network Analysis; Deep Learning; Price Prediction*

1. INTRODUCTION

In recent years, there is a growing trend among scholars in studying cryptocurrencies, such as Ethereum and Bitcoin, based on blockchains. Most previous research mainly focused on Bitcoin, but Ethereum ERC20 tokens are also worth investigating. On the one hand, they can represent some non-physical objects like financial instruments – bonds, stocks [1]; on the other hand, they can also represent physical objects like gold. Modelling and predicting Ethereum token prices are two of the most crucial aspects of studying the Ethereum token dynamic transactions.

For Ethereum ERC20 tokens, all economic and transaction data are public on the blockchain [2], making it possible to establish Ethereum token transaction networks for price prediction. In analyzing complex networks, a commonly used method is to split a network into smaller sub-fractions to study the functionality. In exploring the sub-fractions, traditional graph topological features reflect the essential traits of the whole network [3].

Thus, in the proposed research, we build a one-day interval transaction network, extract topological features, and combine them with economical features to construct a token price prediction model. We put forward a strengthened Bidirectional LSTM model that includes token economic and topological features. In addition, the performance of the proposed model was compared with different types of LSTM models and GRU models to predict token UNI's price. UNI is the primary token for Uniswap DEX (decentralized exchange); Uniswap is the largest DEX in the world in terms of market capitalization at the time.

The paper is organized as the following: Within section 2, the existing methods related to Ethereum token price prediction are concisely introduced and analyzed. Section 3 will first present the data preprocessing of UNI info data and UNI transaction data. Then, the daily dynamic network building and analysis are presented. In addition, we will introduce the variable input combinations and the proposed Bidirectional LSTM model in this section. In section 4, verifying the validity, experimental results for studying the token UNI price are presented. Detailed model comparisons, optimization, and analysis of one-day lag problems are provided in section 4. Eventually, in section 5, some conclusions and possible future directions are presented.

2. LITERATURE REVIEW

To predict the token price, traditional time series models were most frequently used. For example, the Linear Ridge Regression model [4] can be used to analyze multiple regression data that suffer from multicollinearity and can avoid overfitting issues; Autoregressive Integrated Moving Average model (ARIMA), generated by changing fluid time series to stable ones and only regressing its lag value, presents value and the lag value of stochastic error term [5]; Autoregressive Conditional

G. Vilas Bhau et al. (Eds.): MSEA 2022, ACSR 101, pp. 1390–1398, 2023.
https://doi.org/10.2991/978-94-6463-042-8_202

Heteroskedasticity (ARCH) model is frequently used when the error variance of time series follows an autoregressive model [6]. Researchers use machine learning algorithms, such as RNNs and random forest, to predict the token price based on these time series models and token market information.

Neural networks can provide a suitable solution for time series data with time-varying covariables in high-dimension and small signal-to-noise ratio [7]. The structure of recurrent neural networks (RNNs) [1] has periodic links over time, and it can seize the dimensional and time-based information that exceeds the time step through laying information and past state on the internal state of its output [8]. In this manner, as for time series data, RNNs are more appropriate for generation, classification, and prediction.

However, it is not easy to train traditional RNNs with "Real-Time Recurrent Learning" (RTRL) [9] or "Back-Propagation Through Time" (BPTT) [10] methods because of gradient extinction and explosion. As a result, traditional RNNs are not considered appropriate tools for modelling the huge discrete time step dependencies (e.g., more significant than 50) between correlative signals and events. An improvement on the original architecture of RNN can be Long Short-Term Memory (LSTM), which has been used to model time series with long-range correlation and generate more accurate results than traditional neural networks and other models [11].

An existing method combines the network motifs with deep portfolios and LSTM models for Ethereum token price prediction [1]. However, without employing the tool of a topological analysis, the model does not include comprehensive network features as input. In addition, the previous research only compared the proposed model with the regular LSTM model, instead of having various types of neural networks. Furthermore, prior research focused on mainstream cryptocurrencies such as Bitcoin and Ether, but not ERC20 tokens, which is a crucial standard for creating and issuing smart contracts on the Ethereum blockchain.

Our proposed research contained transaction network analysis and applied a particular LSTM model on the price prediction of one of the ERC20 tokens. Our LSTM model took more topological features and critical features extracted from token economic data than the previous methods. Incorporating network analysis, we studied how those factors work within specific transaction networks. Moreover, the proposed study embodied comprehensive model comparison ranging from varieties of LSTM models to GRU models.

3. METHODOLOGY

3.1. Data

3.1.1. UNI economic data preprocessing:

UNI economic data was queried from *CoinMetrics Github* and data is extracted from September 18, 2020 (the starting date for the existence of UNI token prices) to April 24, 2021. Outliers were calculated based on 3∂ principles; we replaced them with the average value in the past five days to get the actual data trend.

The metric of UNI economic features is given in Table. 1:

TABLE 1. METRIC OF UNI ECONOMIC FEATURES

Variable	Description
PriceUSD	The fixed closing price of UNI in USD
SplyFF	Number of available UNI units to trade within one day in the market
TxTfrValMeanUSD	Ratio of total value (USD) of all UNI units to the number of transactions between unique addresses within one day
TxTfrValUSD	Total value (in USD) of all the Uni units been traded between unique addresses within one day
AdrActCnt	Number of unique active addresses in the network.
TxTfrCnt	The sum count of transfers that interval

Then, the correlation between certain features (input variables) and the priceUSD (output variable) was calculated in Table. 2:

TABLE 2. CORRELATION BETWEEN UNI ECONOMIC FEATURES AND PRICEUSD

Econ Features	Correlation Coefficient
	PriceUSD
SplyFF	0.61
TxTfrValMean	0.77
TxTfrVal	0.59
AdrActCnt	0.01
TxTfrCnt	-0.07

According to the table, *SplyFF*, *TxTfrValUSD*, and *TxTfrValMeanUSD* were selected to be the candidates of independent variables due to the high correlation (> 0.5) with price.

3.1.2. UNI transaction data preprocessing:

The transaction data were queried from *Google BigQuery* public dataset *Ethereum_blockchain* from September 18, 2020 to April 24, 2021, and eliminated all missing values, and finally obtained the transaction data of 1,936,030 lines, including 1) ***from_address*** (address of the sender), 2) ***to_address*** (address of receiver), 3) ***value*** (amount of UNI token transferred), and 4) ***block_timestamp*** (transaction time).

As for the *timestamp* variable, the original form of timestamp (*2020-12-24 03:25:54+00:00*) was changed to the form of date (2020-12-24) to match with the data of daily token price.

3.2. Network Analysis

3.2.1. Network Building (Network X):

NetworkX, a python package, was chosen to build, visualize the network and calculate the features of the daily transaction network. The four basic elements of a network are given as:

a)Nodes: Ethereum accounts (ie, external accounts or smart contracts), which is given in our transaction dataset as from_address (seller) and to_address (buyer)

b)Edges: each transaction recorded in the blockchain

c)Weight: number of transactions between addresses

d)Direction: from buyer to sender

A MultiDiGraph that captures both directions and weights for each transaction record were built daily to plot the number of nodes and weighted edges, which is shown in Fig. 1:

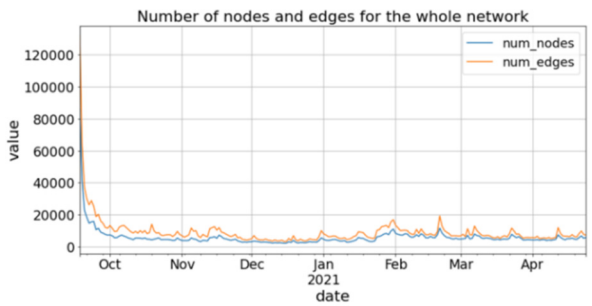

Figure 1. Number of nodes and edges for the whole network

The plot illustrates that the time series of the number of nodes and edges are relatively plain, and no significant changes over time make the network relatively stable.

3.2.2. Feature Extraction:

To analyze how addresses interact with each other, the topological features were computed and extracted from the dataset. Fig. 2 below captures the sample addresses on [2021-03-20].

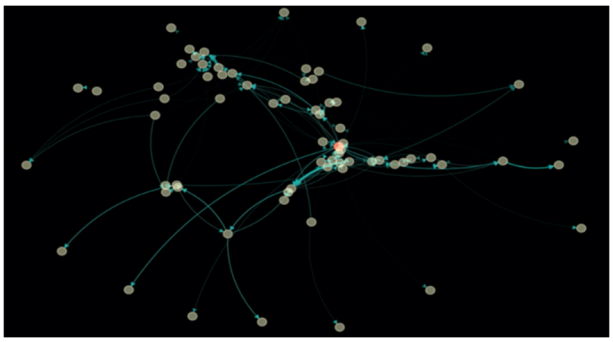

Figure 2. Network for sample addresses on 2021-03-20

Six network features were extracted from the dataset:

a)Degree Centrality: It measures the total weighted edges and the number of directed links between a node and its neighbors. The higher the degree, the more central the node is.

b)Clustering Coefficient: It measures the degree to which nodes in a graph tend to cluster together [12]. Local Clustering coefficient can be calculated by

$$C_i = \frac{2L_i}{k_i(k_i-1)} \quad (1)$$

where k_i denotes the degree of node i and L_i denotes the number of edges between the k_i neighbors of node i. We took the mean and standard deviation of it in this study.

c)Network Modularity: It measures the degree to which the division of a network into communities is [13],

$$Q = \frac{1}{4m}\sum (A_{i,j} - \frac{k_i k_j}{2m}) \quad (2)$$

where $A_{i,j}$ is the adjacency matrix, m represents the number of edges in the network, and k_i denotes the degree of node i. The stronger this division the higher is the value of Q.

d)Eigenvector Centrality: It measures the significance of a node given the significance of its neighbors [14]. $A = (a_{i,j})$ is the adjacency matrix of a network. The eigenvector centrality x_i of node i is given by:

$$x_i = \frac{1}{\lambda}\sum_k a_{k,i}\, x_k \quad (3)$$

where $\lambda \neq 0$ is a constant. In matrix form, we have $\lambda x = xA$.

e)Transitivity: It measures the density of loops of length three (triangles) in a network, it is the ratio of the number of loops of length three and the number of paths of length two.

$$T = 3 \times \frac{N_3}{N_2} \quad (4)$$

where N_3 refers to the number of triangles in the network, N_2 refers to the number of connected triples of nodes in the network.

f)Critical Point: It is defined as the node that has the largest number of transactions and the highest total transaction value. This concept aims to analyze a single node and its influence on prices. Results show that the critical point for each transaction day belongs to the Uniswap ETH pool token holders *"0xd3d2e2692501a5c9ca623199d38826e513033a17",* an account that produces tokens and initiates transfers between individuals.

3.3. Statistical Analysis

3.3.1. Correlation:

Table. 3 shows the correlation between network features and the price, based on which the input variable pairs will be decided.

TABLE 3. CORRELATION BETWEEN NEYWORK FEATURES AND PRICEUSD

Network Features	Correlation with PriceUSD
Degree Centrality MEAN	-0.52
Degree Centrality STD	-0.75
Cluster Mean	-0.66
Cluster STD	-0.61
Modularity	0.73
Transitivity	0.05
Eigen Vector Mean	-0.55
Eigen Vector STD	-0.24
Critical Point Mean	-0.33
Critical Point Degree	-0.21

3.3.2. Input Variable Pairs:

From the calculation, we finally included all the UNI economic features: *priceUSD, SplyFF, TxTfrValMeanUSD, TxTfrValUSD* (named as [U]) and network features: *Modularity, Degree Centrality, Clustering Coefficient, Eigen Mean*, which have a correlation value larger than 0.5 with the price.

Hence, we design our initial input pairs in Table. 4 as follows:

TABLE 4. INPUT PAIRS FOR MODEL COMPARISONS

Group ID	Input Variables
UNI Economic Features (U)	[priceUSD, SplyFF, TxTfrValMeanUSD, TxTfrValUSD]
Group 1	DCmean +[U]
Group 2	DCstd + [U]
Group 3	Clustermean + ClusterSTD +[U]
Group 4	Modularity +[U]
Group 5	EigMean +[U]
Group 6	DCmean +DCstd +Clustermean +Clusterstd +[U]

3.4. Prediction Model

3.4.1. Introduction to Bi-directional LSTM:

The Bidirectional LSTM model can be regarded as a revised augmentation of the simple LSTM model. For the issues of sequence classification types, the BI-LSTM model can help promote the execution of the model. It encodes the sequence of inputs in two directions: forward direction & backward direction. Then, it joints the results from both LSTMS of two directions at each timestep [15].

3.4.2. Model Building:

The following Table. 5 includes the parameters chosen for our Bidirectional LSTM model:

TABLE 5. PARAMETERS CHOSEN FOR THE BIDIRECTIONAL LSTM MODEL

Parameter	Value
Hidden Unit (Hidden Size)	100
Activation	Sigmoid
Optimizer	Adam
Learning Rate	0.1
Loss Function	MSE
Epoch	300
Batch_Size	64

4. Performance Evaluation and Discussion

4.1. Evaluation Methods

To evaluate the performance of our applied models, we used Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actural_i)^2}{N}} \quad (5)$$

Training loss score and validation loss score (both based on MSE) were utilized to compute the model's error during optimization.

4.2. Bi-directional Training Results

Bidirectional LSTM model is utilized to train on four window sizes, three train-test split modes [1], and 7 input variable combinations in a total of 84 combinations, all the combinations are given as:

a)Window Size: 4,5,6,7

b)Train-test Split: 9:1, 8:2, 7:3

c)Input pairs: (Table 4 in 3.3.2)

The average RMSE (average three times) for each combination is summarized in Table. 6 below:

TABLE 6. Average RMSE for Different Input Pairs

		Input Pair Groups						
train/test	WL	*[U]*	*1*	*2*	*3*	*4*	*5*	*6*
9--1	4	3.22	2.56	4.14	3.27	2.77	2.59	4.59
	5	2.74	2.28	2.33	2.27	2.18	2.94	3.36
	6	3.03	2.57	2.47	4.28	2.36	2.46	3.26
	7	3.34	2.48	2.10	4.17	3.22	3.40	2.60
8--2	4	2.88	3.05	2.30	3.97	3.95	1.90	2.62
	5	2.58	2.29	2.64	3.67	2.08	2.10	2.27
	6	2.40	3.43	3.05	3.56	2.36	2.72	3.13
	7	2.27	2.65	2.31	2.00	2.13	2.44	3.27
7--3	4	4.81	9.85	8.83	10.0	8.06	9.59	8.52
	5	3.39	8.79	4.38	8.07	3.30	3.62	7.11
	6	9.88	3.40	8.57	4.78	3.90	9.37	4.55
	7	3.91	6.08	4.94	3.93	6.94	4.29	5.60

The results demonstrated that:

1) Train-test split: when ratio = 8:2, performed best.

2) Window length: when WL = 5, performed best generally.

3) Input variables:

a) Adding network topological features to input improved model performance, which implies that those factors have specific benefits for price prediction.

b) Group 4 (*modularity, priceUSD, SplyFF, TxTfrValMeanUSD, TxTfrValUSD*) had the best performance relatively, showing that the token price has a higher correlation with the tendency towards clustering of transaction accounts.

c) Including all network features hampered performance due to the "multicollinearity" between variables [16].

4.3. Model Comparisons

1)For different models: Bidirectional LSTM model is compared with the other 6 models in performance, below gives an introduction to the models:

a)Vanilla LSTM: It is the simplest of the LSTM models, with only a single hidden layer of LSTM units and an output layer for prediction.

b)The Stacked LSTM model: It is an LSTM model with multiple hidden LSTM layers stacked one on top of another.

c)GRU: It is a new generation of RNN that is very similar to LSTM. However, instead of using a unit state, the GRU uses a hidden state to transfer information [17].

2)For combination: Input, window, and train-test split combination that performed generally great (RMSE) in Bidirectional LSTM is chosen for model comparison:

a)Input: Modularity, priceUSD, SplyFF, TxTfrValMeanUSD, TxTfrValUSD

b)Window size: 5

c)Train-test split: 8:2

7 different models are run 3 times each and their average RMSEs are summarized in the Table. 7 below:

TABLE 7. RMSEs and Loss for different models

Models	Evaluation Methods

	RMSE	*Train loss converges?*	*Validation loss converges?*
Vanilla LSTM	2.6728	Yes	Yes
Vanilla LSTM-dropout	3.0308	Yes	Yes
Stacked LSTM	3.1366	Yes	Yes
Bidirectional LSTM	2.0756	Yes	Yes
Bidirectional LSTM-dropout	3.5562	Yes	Yes
GRU	2.8478	Yes	Yes
GRU-dropout	3.5097	Yes	Yes

The summary evaluation table above demonstrated that:

- For all the seven models we trained, both train loss and validation loss function converge well, suggesting that the training effect improves with the Epochs' increase and reaches stability and saturation.
- According to Vanilla LSTM, Bidirectional LSTM, and GRU, adding a dropout (0.2) layer makes the model perform worse than not. Since we only have less than 200 data sizes for training, dropping out is likely to let the model omit some essential features in price, leading to a worse fit. Therefore, in this case of the UNI token, we do not add dropout layers into our model.

Comparing the average RMSE for the seven models, we have Bidirectional LSTM performs the best generally, with the lowest average RMSE.

4.4. Lag Problem

4.4.1. Log Difference Method:

The predicted value by Bidirectional LSTM model and ground truth for price is shown in Fig. 3:

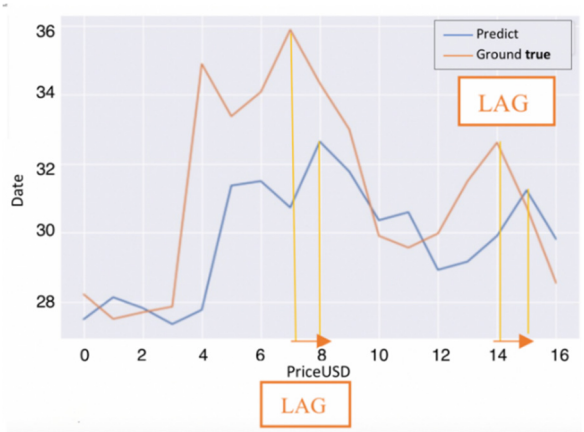

Figure 3. Predicted value and ground truth *(BI-LSTM)*

From the above section's output, the predicted value at time t is often the actual value at time t-1; the natural value curve lags the expected value curve. This is resulted from the autocorrelation [18] of our time series data, making it not stationary. To eliminate autocorrelation, **the first log difference between t and t-1 price was taken**. In this way, our regression goal could be the difference between the current moment and the last moment.

The distribution, autocorrelation, partial autocorrelation plots and for UNI price before and after *log_difference* processing are shown in Fig. 4 below.

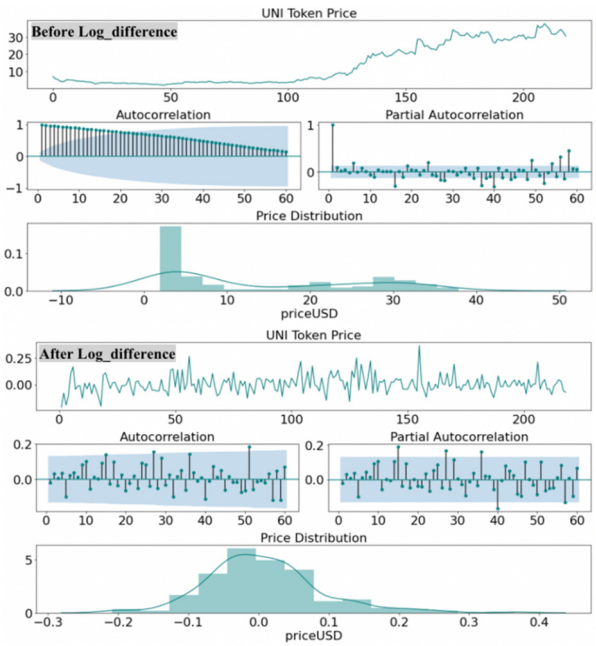

Figure 4. Statistical test resulst for UNI token price with and without log_difference preprocessing

The Dickey-Fuller statistical test results are shown in Table. 8 below:

TABLE 8. P-VALUE FOR UNI TOKEN PRICE WITH AND WITHOUT KOG DIFFERENCE PREPROCESSING

	Before Log Difference	*After Log Difference*

	Preprocessing	*Preprocessing*
P-Value	0.9411	0.00000

The graphs and statistics illustrate that after the first *log_difference*, the upward trend and autocorrelation were eliminated, making the vibration dependent purely on day-to-day changes. P-value for Dickey-Fuller statistical test [19] decreased to zero, indicating that the series became stationary after processing.

4.4.2. Training results and comparison:

After log_difference preprocessing, we trained the previously chosen input pairs with our Bidirectional LSTM model. A dropout layer (dropout = 0.3) was added to solve the potential overfitting problem. The predicted price graphs under the MinMacScalar method and log_difference method are shown in Fig. 5:

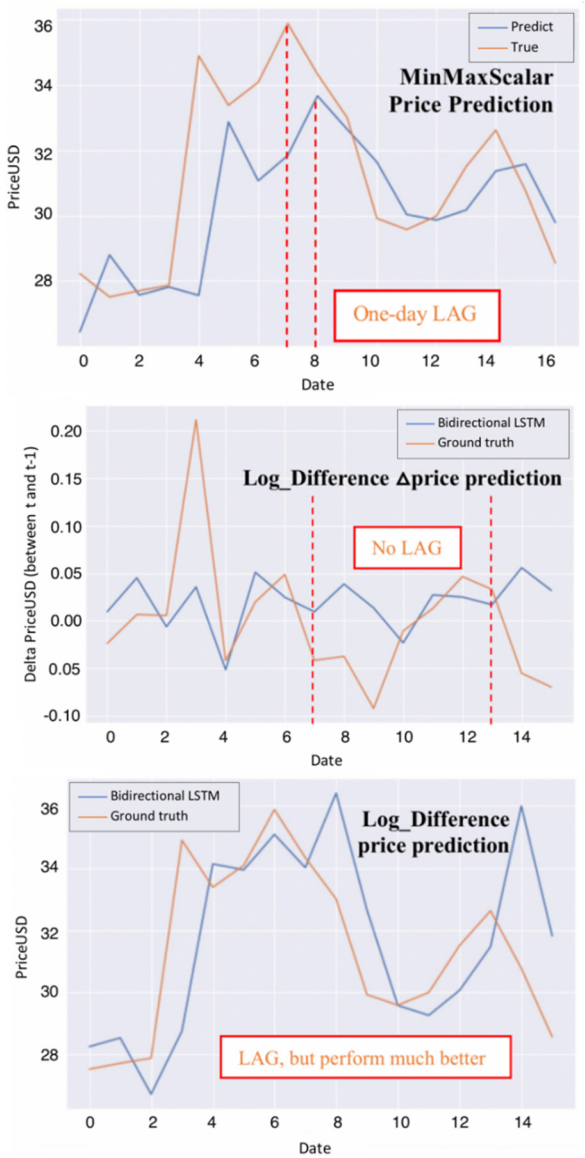

Figure 5. Predicted values and ground truth plots with different preprocessing methods

The RMSE under the MinMacScalar method and log_difference method are shown in Table. 9:

TABLE 9. RMSEs WITH DIFFERENT PREPROCESSING METHODS

RMSE (price)	*MinMaxScalar*	*Log Difference*
Train-Test Split = 9:1	2.1781	2.2013
Train-Test Split = 8:2	2.0756	2.0689

From the above graphs and statistics, it could be concluded that after *log_difference* preprocessing:

- Lag problem was solved for △price prediction. The predicted and actual △price trend is consistent, suggesting that *log_difference* preprocessing could provide a relatively accurate prediction result for the price difference.
- Lagging still bothers the prediction of price value. However, it could be observed that the predicted value is generally closer to the actual value, as is reflected from the graph, but the lag problem still existed. This is probably caused by the accumulation of errors in forecasting the △price, which is amplified when converted into the price, thus presenting a lag problem.

4.4.3. Discussion for the existence of the lag problem

- Limitation from the size of the training dataset.
- Room for model trials and improvement. Ensembling GRU, Vanilla LSTM, and BI-LSTM with a painfully weighted average method was tried, but the prediction effect has not been significantly improved.
- Enlarge the model input pool. It is concluded that network characteristic data can improve prediction accuracy, but there are probably more critical independent variables that need further exploration. Specific research measures in the future will be described in the *Future Work* section.

5. CONCLUSION AND FUTURE WORK

5.1. Conclusion

- Adding topological network features to input variables could generally improve the model performance (lower RMSE), suggesting that the

characteristics of the UNI transaction network affect its price to a certain degree. Among the five network features we tested; **modularity** had the best effect on UNI token price prediction relatively.

- Including all network features hampered prediction results due to the "multicollinearity" between features.
- Among the seven models, our proposed **Bidirectional LSTM** performs the best on UNI price prediction.
- The instability, upward trend, and the autocorrelation of input data series lead to a one-day lag between the real and the predicted price. *Log_difference* is adapted to eliminate the movement and the autocorrelation, the lag problem was solved when predicting price difference, and the accuracy (RMSE) for price prediction was improved.

5.2. Future Work

- Explore other influential input variables. Potential input variables include other closely related ERC20 tokens' price, BTC price, ETH price, etc.; Unexpected events such as the number of COVID-19 daily new cases; Social and government influences news, policies, etc.
- Optimize the prediction model and explore other potential deep learning models after adding more input variables to further deal with the lag problem.
- Resolve the problem of technical constraints. Topological features including Closeness Centrality, Betweenness Centrality, etc., are expensive to calculate. Possible solutions include running on a more powerful GPU, segmenting addresses based on activeness, and combining each weighted category's influence on prediction. By adding more network features as input variables, we expect our model to have better performance.

REFERENCES

[1] Chen, Y., and H. K. T. Ng. "Deep Learning Ethereum Token Price Prediction with Network Motif Analysis." IEEE Xplore, November 1, 2019. https://doi.org/10.1109/ICDMW.2019.00043.

[2] Somin, Shahar, Goren Gordon, and Yaniv Altshuler. "Network Analysis of ERC20 Tokens Trading on Ethereum Blockchain." *Unifying Themes in Complex Systems IX*, 2018, 439–50. https://doi.org/10.1007/978-3-319-96661-8_45.

[3] Savić, Miloš, Mirjana Ivanović, and Lakhmi C. Jain. "Fundamentals of Complex Network Analysis." *Intelligent Systems Reference Library*, May 11, 2018, 17–56. https://doi.org/10.1007/978-3-319-91196-0_2.

[4] Chen, Ting, Zihao Li, Yuxiao Zhu, Jiachi Chen, Xiapu Luo, John Chi-Shing Lui, Xiaodong Lin, and Xiaosong Zhang. "Understanding Ethereum via Graph Analysis." *ACM Transactions on Internet Technology* 20, no. 2 (May 25, 2020): 1–32. https://doi.org/10.1145/3381036.

[5] ADAM, HAYES. "Autoregressive Integrated Moving Average (ARIMA)." Investopedia, 2019. https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp.

[6] Sato, Takaki, and Yasumasa Matsuda. "Spatial Autoregressive Conditional Heteroskedasticity Models." *JOURNAL of the JAPAN STATISTICAL SOCIETY* 47, no. 2 (December 28, 2017): 221–36. https://doi.org/10.14490/jjss.47.221.

[7] Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer Feedforward Networks Are Universal Approximators." *Neural Networks* 2, no. 5 (January 1989): 359–66. https://doi.org/10.1016/0893-6080(89)90020-8.

[8] Glorot, Xavier, and Yoshua Bengio. "Understanding the Difficulty of Training Deep Feedforward Neural Networks." *PMLR*, March 31, 2010, 249–56. http://proceedings.mlr.press/v9/glorot10a.html.

[9] WILLIAMS, RONALD J., and DAVID ZIPSER. "Experimental Analysis of the Real-Time Recurrent Learning Algorithm." *Connection Science* 1, no. 1 (January 1989): 87–111. https://doi.org/10.1080/09540098908915631.

[10] Bianconi, G., and Mahesh Agrawal. "Predicting Bitcoin Transactions with Network Analysis." www.semanticscholar.org, 2017. https://www.semanticscholar.org/paper/Predicting-Bitcoin-Transactions-with-Network-Bianconi-Agrawal/a6ef96ca65597e38299e0d20e737b50eb4e2d176.

[11] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long Short-Term Memory." *Neural Computation* 9, no. 8 (November 1997): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

[12] Holland, Paul W., and Samuel Leinhardt. "Transitivity in Structural Models of Small Groups." *Comparative Group Studies* 2, no. 2 (May 1971): 107–24. https://doi.org/10.1177/104649647100200201.

[13] Ji, Xiaonan, Raghu Machiraju, Alan Ritter, and Po-Yin Yen. "Examining the Distribution, Modularity, and Community Structure in Article Networks for

Systematic Reviews." *AMIA Annual Symposium Proceedings* 2015 (2015).

[14] Golbeck, Jennifer. *Analyzing the Social Web*. Amsterdam: Elsevier, 2013.

[15] Graves, Alex, Santiago Fernández, and Jürgen Schmidhuber. "Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition." *Lecture Notes in Computer Science*, 2005, 799–804. https://doi.org/10.1007/11550907_126.

[16] Stephanie. "Multicollinearity: Definition, Causes, Examples." Statistics How To, September 22, 2015. https://www.statisticshowto.com/multicollinearity/.

[17] Dutta, Aniruddha, Saket Kumar, and Meheli Basu. "A Gated Recurrent Unit Approach to Bitcoin Price Prediction." *SSRN Electronic Journal*, 2019. https://doi.org/10.2139/ssrn.3514069.

[18] Legendre, Pierre. "Spatial Autocorrelation: Trouble or New Paradigm?" *Ecology* 74, no. 6 (September 1993): 1659–73. https://doi.org/10.2307/1939924.

[19] Cheung, Yin-Wong, and Kon S. Lai. "Lag Order and Critical Values of the Augmented Dickey–Fuller Test." *Journal of Business & Economic Statistics* 13, no. 3 (July 1995): 277–80. https://doi.org/10.1080/07350015.1995.10524601.