*Grenze International Journal of Engineering and Technology, June Issue*

# Hybrid Deep Learning Approach for Dynamic Ethereum Fraud Detection

K.R.Nandhashree[1], Atchaya Kumar S[2], Sethu Sridharan S[3] and Dharshaan M[4]
[1]Professor, Department of Cyber Security, SRM Valliammai Engineering College
Email: nandhashreekr.cys@srmvalliammai.ac.in
[2-4]UG Scholar, Department of Cyber Security, SRM Valliammai Engineering College
Email: atchayakumar456@gmail.com, sethutals14@gmail.com, dharshaanmsd@gmail.com

***Abstract*— Ethereum, a prominent block chain platform, facilitates digital asset transactions through its decentralized network. However, the decentralized nature of Ethereum also opens avenues for fraudulent activities, presenting challenges to network security and integrity. This work addresses the problem of Ethereum transaction fraud by proposing an advanced fraud detection system. Initially, it outlines the characteristics of Ethereum transactions and the associated risks of fraudulent behavior within the network. Subsequently, it highlights the detrimental impacts of fraudulent transactions, including financial losses and diminished trust among users. To combat these challenges, the work presents a comprehensive solution in the form of a Hybrid Model for Ethereum Fraud Prediction (HMEFP). The proposed HMEFP model integrates a Customized Dense Deep Learning Model and an LSTM-based recurrent neural network, which collectively learn intricate patterns and temporal dependencies within Ethereum transactions. Through a fusion mechanism, the model combines the strengths of both architectures, enhancing its fraud detection capabilities The results demonstrate the potential of the HMEFP to safeguard the integrity and security of the Ethereum network by accurately identifying fraudulent transactions.**

***Index Terms*— Blockchains, Feature extraction, Smart contracts, Generative adversarial networks, Abnormal transactions,**

## I. INTRODUCTION

The rise of blockchain technology, particularly Ethereum, has transformed financial transactions, enabling decentralized networks for asset transfer, smart contracts, and decentralized applications. However, Ethereum's decentralized nature presents security challenges, including fraudulent activities such as phishing, Ponzi schemes, and smart contract vulnerabilities. Addressing Ethereum fraud requires innovative solutions, combining technologies like machine learning and blockchain analysis. This study aims to develop a comprehensive fraud detection system for Ethereum, enhancing security and trust within the network. Through empirical analysis, the research seeks to contribute to Ethereum's integrity as a leading blockchain platform.

## II. RELATED WORKS

[1] Within the field of blockchain research, a number of creative applications have surfaced and garnered a lot of interest. But there are financial hazards associated with the rise of phishingschemes, particularly in the

*Grenze ID: 01.GIJET.10.2.205*

cryptocurrency space like Ethereum. Our study frames phishing detection as a node classification problem, treating accounts and transactions as nodes and edges to address this. We provide a technique that uses an autoencoder and Graph Convolutional Network to reliably identify phishing accounts, which works on a variety of Ethereum datasets. [2] The dynamism of transaction networks are typically difficult for currentgraph embedding approaches to depict, especially in Ethereum-like systems. We address this by introducing T-EDGE, a Temporal Weighted Multidigraph Embedding method that takes multiplicity and edge dynamics into account. With the integration of weighted edge and temporal data, T-EDGE provides a full model of dynamic transaction networks. T-EDGE's superiority in node categorization is demonstrated in experiments with Ethereum data, highlighting the advantages of temporal-aware techniques in time-sensitive transaction network analysis. [3] Our research presents a novel detection technique to mitigate the growingthreat of phishing scams on Ethereum and other blockchain platforms. We identify phishing scams by crawling labelled addresses and using trans2vec network embedding to analyse transaction records. Our approach, which uses oneclass SVM for classification, works well indetecting phishing schemes on Ethereum, and trans2vec outperforms other techniques in termsof feature extraction. [4] Existing detection technologies are inadequate given the growth of Ponzi schemes inside smart contracts, a critical worry for blockchain investors. To address this,we introduce a novel strategy that combines enhanced LightGBM algorithm with bytecode characteristics and opcode frequencies. Tests conducted using actual Ethereum data show notable increases in accuracy, providing a more effective way to spot Ponzi schemes and lowerthe danger of investing. [5] Our paper presents Heterogeneous Graph Transformer Networks (S_HGTNs), in response to the urgent demand for anomaly detection in smart contracts, especially on Ethereum. S_HGTNs provide strong feature representation for anomaly detectionthrough the usage of transformer-derived relationship matrices and the construction of a Heterogeneous Information Network (HIN). Empirical findings demonstrate the model's superiority over conventional techniques in detecting anomalous contracts and reducing the likelihood of financial fraud.

## III. Architectural Diagram

Several essential modules make up the implementation of the Ethereum transaction fraud detection system. First, data collection entails obtaining sender and recipient addresses, quantities, timestamps, and gas fees associated with Ethereum transactions from a Kaggle database. The obtained data is then cleaned and prepared by the data preparation module, which also handles scaling characteristics and missing values. The pre-processed data is split into training and test sets via data splitting. Feature selection makes use of methods such as Spearman correlation analysis to find pertinent features. Next, in order to efficiently evaluate Ethereum transaction data, the Hybrid Model for Ethereum Fraud Prediction (HMEFP) is constructed by combining an LSTM-based recurrent neural network with a Customized Dense Deep Learning Model. Optimizing parameters and assessing using measures like accuracy and binary cross-entropy loss are part of training the HMEFP model. In order to evaluate the trained model's capacity for generalization and efficacy in identifying fraud, prediction entails applying it to test data. Lastly, the model's accuracy, precision, recall, and F1-score are evaluated by the performance evaluation module, which also provides guidance for any necessary changes to enhance performance.

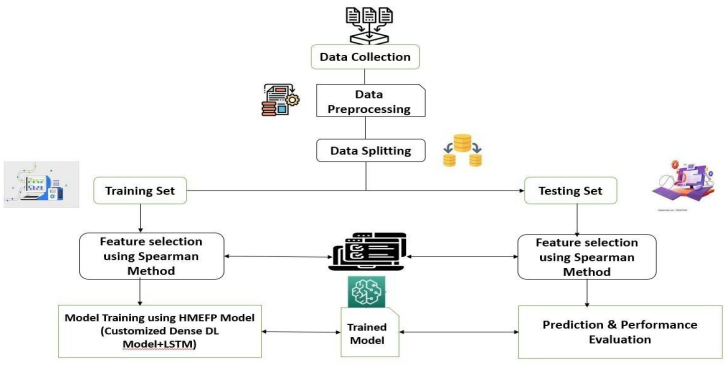



Fig 1 – HMEFP Model Hybrid - Integration of Customized Dense DL Model & LSTM

## IV. Material Implementation

### *A. Data Loading and Exploring*

In order to provide a broad and representative dataset, the Ethereum fraud detection process started with the acquisition of Ethereum transaction data from multiple sources. The dataset included many different kinds of

transactions, such as token swaps, transfers, and interactions with smart contracts. The goal of data gathering activities was to provide an extensive foundation for fraud detection research by capturing the diversity and complexity of transactions within the Ethereum network.

| | Avg min between sent tnx | Avg min between received tnx | Time Diff between first and last (Mins) | Sent tnx | Received Tnx | Number of Created Contracts | Unique Received From Addresses | Unique Sent To Addresses | min value received | max value received | ... | ERC20 max val sent | ERC20 avg val sent | ERC20 min val sent contract | ERC20 max val sent contract | E avg cont |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 844.26 | 1093.71 | 704785.63 | 721 | 89 | 0 | 40 | 118 | 0.000000 | 45.806785 | ... | 1.683100e+07 | 271779.920000 | 0.0 | 0.0 | |
| 1 | 12709.07 | 2958.44 | 1218216.73 | 94 | 8 | 0 | 5 | 14 | 0.000000 | 2.613269 | ... | 2.260809e+00 | 2.260809 | 0.0 | 0.0 | |
| 2 | 246194.54 | 2434.02 | 516729.30 | 2 | 10 | 0 | 10 | 2 | 0.113119 | 1.165453 | ... | 0.000000e+00 | 0.000000 | 0.0 | 0.0 | |
| 3 | 10219.60 | 15785.09 | 397555.90 | 25 | 9 | 0 | 7 | 13 | 0.000000 | 500.000000 | ... | 9.029231e+03 | 3804.076893 | 0.0 | 0.0 | |
| 4 | 36.61 | 10707.77 | 382472.42 | 4598 | 20 | 1 | 7 | 19 | 0.000000 | 12.802411 | ... | 4.500000e+04 | 13726.659220 | 0.0 | 0.0 | |
| 5 | 9900.12 | 375.48 | 20926.68 | 2 | 3 | 0 | 2 | 1 | 0.724148 | 4.813776 | ... | 0.000000e+00 | 0.000000 | 0.0 | 0.0 | |
| 6 | 69.46 | 629.44 | 8660.35 | 25 | 11 | 0 | 9 | 20 | 0.049000 | 2.650000 | ... | 0.000000e+00 | 0.000000 | 0.0 | 0.0 | |
| 7 | 1497.39 | 176.84 | 319828.05 | 213 | 5 | 0 | 3 | 3 | 0.118490 | 2.000000 | ... | 0.000000e+00 | 0.000000 | 0.0 | 0.0 | |
| 8 | 0.00 | 0.00 | 496.62 | 1 | 1 | 0 | 1 | 1 | 2.000000 | 2.000000 | ... | 0.000000e+00 | 0.000000 | 0.0 | 0.0 | |
| 9 | 2570.59 | 3336.01 | 30572.70 | 8 | 3 | 0 | 2 | 4 | 0.100000 | 40.000000 | ... | 0.000000e+00 | 0.000000 | 0.0 | 0.0 | |

Fig 2 – Dataset the first x number of rows when head 10

*B. Data Preprocessing*

After gathering and importing the data, a number of preparation procedures were carried out to clean and ready the dataset for analysis. This required standardized data formats and addressing missing values. In order to maintain consistency among features, data scaling is used.

| Null Value Checking Result | | After Cleaning | |
|---|---|---|---|
| avg val sent | 0 | avg val sent | 0 |
| min value sent to contract | 0 | min value sent to contract | 0 |
| max val sent to contract | 0 | max val sent to contract | 0 |
| avg value sent to contract | 0 | avg value sent to contract | 0 |
| total transactions (including tnx to create contract | 0 | total transactions (including tnx to create contract | 0 |
| total Ether sent | 0 | total Ether sent | 0 |
| total ether received | 0 | total ether received | 0 |
| total ether sent contracts | 0 | total ether sent contracts | 0 |
| total ether balance | 0 | total ether balance | 0 |
| Total ERC20 tnxs | 885 | Total ERC20 tnxs | 0 |
| ERC20 total Ether received | 885 | ERC20 total Ether received | 0 |
| ERC20 total ether sent | 885 | ERC20 total ether sent | 0 |
| ERC20 total Ether sent contract | 885 | ERC20 total Ether sent contract | 0 |
| ERC20 uniq sent addr | 885 | ERC20 uniq sent addr | 0 |
| ERC20 uniq rec addr | 885 | ERC20 uniq rec addr | 0 |
| ERC20 uniq sent addr.1 | 885 | ERC20 uniq sent addr.1 | 0 |
| ERC20 uniq rec contract addr | 885 | ERC20 uniq rec contract addr | 0 |
| ERC20 avg time between sent tnx | 885 | ERC20 avg time between sent tnx | 0 |
| ERC20 avg time between rec tnx | 885 | ERC20 avg time between rec tnx | 0 |
| ERC20 avg time between rec 2 tnx | 885 | ERC20 avg time between rec 2 tnx | 0 |
| ERC20 avg time between contract tnx | 885 | ERC20 avg time between contract tnx | 0 |
| ERC20 min val rec | 885 | ERC20 min val rec | 0 |
| ERC20 max val rec | 885 | ERC20 max val rec | 0 |
| ERC20 avg val rec | 885 | ERC20 avg val rec | 0 |
| ERC20 min val sent | 885 | ERC20 min val sent | 0 |
| ERC20 max val sent | 885 | ERC20 max val sent | 0 |
| ERC20 avg val sent | 885 | ERC20 avg val sent | 0 |
| ERC20 min val sent contract | 885 | ERC20 min val sent contract | 0 |
| ERC20 max val sent contract | 885 | ERC20 max val sent contract | 0 |
| ERC20 avg val sent contract | 885 | ERC20 avg val sent contract | 0 |
| ERC20 uniq sent token name | 885 | ERC20 uniq sent token name | 0 |
| ERC20 uniq rec token name | 885 | ERC20 uniq rec token name | 0 |
| ERC20 most sent token type | 897 | FLAG | 0 |
| ERC20_most_rec_token_type | 907 | dtype: int64 | |
| FLAG | 0 | | |

Fig 3- Turning object variables into 'category' dtype for more computation efficiency

*C. Data Splitting*

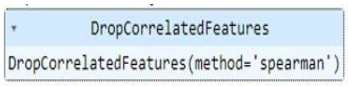
To aid in the creation and assessment of the model, the preprocessed dataset was split into separate training and test sets. The bulk of the data required to train the fraud detection model was found in the training set, whilst the test set functioned as a separate dataset for model assessment.

```
training set size: (6992, 38), testing set size: (2997, 38)
```

Fig 4- Data Splitting Testing set 30% , Training set 70%

*D. Develop Defense Model*

By locating and choosing the most pertinent features from the Ethereum transaction dataset, feature selection is essential to improving the model's performance. According to our investigation, Spearman correlation analysis reduces noise and redundancy in the data, successfully identifying correlated features and improving the prediction performance of the model.




**Correlated features to drop and correlated features set**

{'x16', 'x27', 'x26', 'x37', 'x15', 'x23', 'x32', 'x31', 'x12', 'x29', 'x28', 'x17', 'x18', 'x35', 'x36', 'x10', 'x20', 'x19', 'x7', 'x34'}
[{'x17', 'x2'}, {'x3', 'x7'}, {'x9', 'x12', 'x19', 'x10'}, {'x13', 'x18'}, {'x16', 'x14', 'x15', 'x20'}, {'x31', 'x27', 'x29', 'x22', 'x37', 'x32', 'x23'}, {'x24

**After correlating features dropping, train and test datasets shape**

X_train shape after correlation: (6992, 18)
X_test shape after correlation: (2997, 18)

Fig 5- Feature Selection using the Spearman Correlation Coefficient Technique

*E. Feature Selection*

An LSTM-based recurrent neural network and a customized dense deep learning model were combined to create the Hybrid Model for Ethereum Fraud Prediction (HMEFP). The advantages of both architectures were merged in this hybrid method to efficiently analyze Ethereum transaction data and identify fraudulent activity. The Customized Dense Deep Learning Model's dense layers were able to identify intricate patterns in the data, and the LSTM component improved the model's ability to detect fraud by capturing temporal relationships in sequential transactions.

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| Input (InputLayer) | [(None, 18)] | 0 | [] |
| dense (Dense) | (None, 512) | 9728 | ['Input[0][0]'] |
| dropout (Dropout) | (None, 512) | 0 | ['dense[0][0]'] |
| dense_1 (Dense) | (None, 256) | 131328 | ['dropout[0][0]'] |
| dropout_1 (Dropout) | (None, 256) | 0 | ['dense_1[0][0]'] |
| dense_2 (Dense) | (None, 128) | 32896 | ['dropout_1[0][0]'] |
| Input (InputLayer) | [(None, 18, 1)] | 0 | [] |
| dropout_2 (Dropout) | (None, 128) | 0 | ['dense_2[0][0]'] |
| lstm (LSTM) | (None, 100) | 40800 | ['input[0][0]'] |
| dense_3 (Dense) | (None, 64) | 8256 | ['dropout_2[0][0]'] |
| dense_5 (Dense) | (None, 512) | 51712 | ['lstm[0][0]'] |
| dropout_3 (Dropout) | (None, 64) | 0 | ['dense_3[0][0]'] |
| dropout_4 (Dropout) | (None, 512) | 0 | ['dense_5[0][0]'] |
| dense_4 (Dense) | (None, 64) | 4160 | ['dropout_3[0][0]'] |
| dense_6 (Dense) | (None, 64) | 32832 | ['dropout_4[0][0]'] |
| concatenate (Concatenate) | (None, 128) | 0 | ['dense_4[0][0]', 'dense_6[0][0]'] |
| dense_7 (Dense) | (None, 1) | 129 | ['concatenate[0][0]'] |

Total params: 311841 (1.19 MB)
Trainable params: 311841 (1.19 MB)
Non-trainable params: 0 (0.00 Byte)

Fig 6- HMEFP Model Building using Hybrid - Integration of Customized Dense DL Model & LSTM

*F. Train the HMEFP Model*

Utilizing the binary cross-entropy loss function and the Adam optimizer, the HMEFP was trained on the assigned training set. In order to minimize the loss function and maximize performance, the model iteratively changed its parameters throughout training. The learning rate, dropout rates, and layer sizes were adjusted as hyperparameters to enhance convergence and avoid overfitting. The training procedure comprised identifying patterns from the dataset's sequential and organized components, guaranteeing thorough fraud detection abilities.

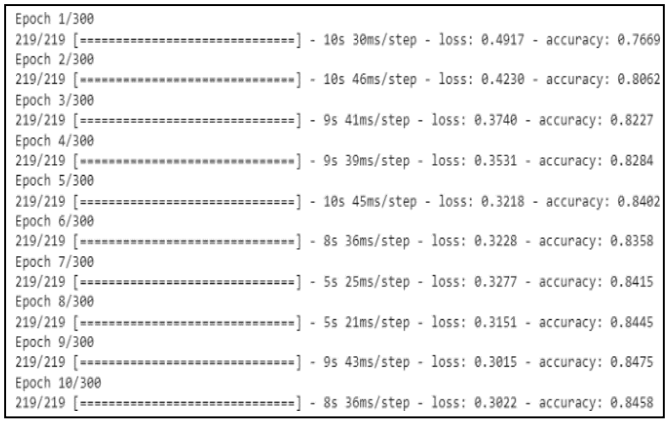



Fig 7- The output for HMEFP model

## V. Conclusion & Future Work

An important development in the area of fraud detection inside the Ethereum network is the proposed Hybrid Model for Ethereum Fraud Prediction (HMEFP). The HMEFP efficiently captures the sequential and structured components of Ethereum transaction data by combining an LSTM-based recurrent neural network with a customized dense deep learning model, enabling extensive fraud detection capabilities. The model is able to learn complex patterns and temporal relationships present in Ethereum transactions thanks to the LSTM components' unique architecture and dense layers. After testing and assessment, the HMEFP shows encouraging results in terms of correctly detecting Ethereum fraud. The model's performance metrics—precision, recall, accuracy, and F1-score—show how well it can differentiate between authentic and fraudulent transactions.

## References

[1] Liang Chen, Jiaying Peng, Yang Liu, Jintang Li, Fenfang Xie, and Zibin Zheng. 2020. Phishing Scams Detection in Ethereum Transaction Network. ACM Trans. Internet Technol. 21, 1, Article 10 (dec 2020), 16 pages. https://doi.org/10.1145/ 3398071

[2] Dan Lin, Jiajing Wu, Qi Yuan, and Zibin Zheng. 2020. T-edge: Temporal weighted multidigraph embedding for ethereum transaction network analysis. Frontiers in Physics 8 (2020), 204.

[3] Jiajing Wu, Qi Yuan, Dan Lin, Wei You, Weili Chen, Chuan Chen, and Zibin Zheng. 2020. Who are the phishers? phishing scam detection on ethereum via network embedding. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2020).

[4] Yunyi Xie, Jie Jin, Jian Zhang, Shanqing Yu, and Qi Xuan. 2021. TemporalAmount Snapshot MultiGraph for Ethereum Transaction Tracking. In Blockchain and Trustworthy Systems, Hong-Ning Dai, Xuanzhe Liu, Daniel Xiapu Luo, Jiang Xiao, and Xiangping Chen (Eds.). Springer Singapore, Singapore, 133–146.

[5] Arnold L, Brennecke M, Camus P, Fridgen G, Guggenberger T, Radszuwill S, Rieger A, Schweizer A, Urbach N (2019) Blockchain and initial coin offerings: blockchain's implications for crowdfunding. In: Business transformation through Blockchain. Palgrave Macmillan, Cham, pp 233–272

[6] Li X, Whinston AB (2020) Analyzing cryptocurrencies. Inf Syst Front 22(1):17–22

[7] Liu J, Serletis A (2019) Volatility in the cryptocurrency market. Open Econ Rev 30(4):779– 811

[8] Mock W, Corps N (2019) CSCI49379: Intro to Blockchain (Syllabus)

[9] Malik H, Manzoor A, Ylianttila M, Liyanage M (2019) Performance Analysis of Blockchain based Smart Grids with Ethereum and Hyperledger Implementations. In: 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, pp 1–5

[10] Bartoletti M, Carta S, Cimoli T, Saia RJFGCS (2020) Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact. Futur Gener Comput Syst 102:259–277

[11] Chen W, Zheng Z, Ngai EC-H, Zheng P, Zhou YJIA (2019) Exploiting blockchain data to detect smart ponzi schemes on ethereum. IEEE Access 7:37575–37586

[12] Aziz R, Verma C, Srivastava N (2015) A weighted-SNR feature selection from independent component subspace for nb classifi-cation of microarray data. Int J Adv Biotec Res 6:245–255

[13] Chen W, Zheng Z, Cui J, Ngai E, Zheng P, Zhou Y (2018) Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. In: Proceedings of the 2018 World Wide Web Conference, pp 1409–1418

[14] Chow SS, Choo KKR, Han J (2021) Editorial for accountability and privacy issues in blockchain and cryptocurrency. Futur Gener Comput Syst 114:647–648

[15] Zhang Y, Wenqiang Y, Ziyu L, Salman R, Huaihu C (2021) Detecting Ethereum Ponzi Schemes Based on Improved LightGBM Algorithm. IEEE Transactions on Computational Social Systems
[16] Liu L, Tsai W-T, Bhuiyan MZA, Peng H, Liu MJFGCS (2021) Blockchain-enabled fraud discovery through abnormal smart contract detection on Ethereum. Futur Gener Comput Syst 6:133–137
[17] Hu T, Liu X, Chen T, Zhang X, Huang X, Niu W, Lu J, Zhou K, Liu Y (2021) Transaction- based classification and detection approach for Ethereum smart contract. Inf Process Manag 58(2):102462
[18] Farrugia S, Ellul J, Azzopardi G (2020) Detection of illicit accounts over the Ethereum blockchain. Exp Syst Appl 150:113318