

Fishing for Phishers: Learning-Based Phishing Detection in Ethereum Transactions

AHOD ALGHURIED, University of Central Florida, USA

ABDULAZIZ ALGHAMDI, University of Central Florida, USA

ALI ALKINOON, University of Central Florida, USA

SOOHYEON CHOI, University of Central Florida, USA

MANAR MOHAISEN, Northeastern Illinois University, USA

DAVID MOHAISEN*, University of Central Florida, USA

Phishing detection on Ethereum has increasingly leveraged advanced machine learning techniques to identify fraudulent transactions. However, limited attention has been given to understanding the effectiveness of feature selection strategies and the role of graph-based models in enhancing detection accuracy. In this paper, we systematically examine these issues by analyzing and contrasting explicit transactional features and implicit graph-based features, both experimentally and analytically. We explore how different feature sets impact the performance of phishing detection models, particularly in the context of Ethereum's transactional network. Additionally, we address key challenges such as class imbalance and dataset composition and their influence on the robustness and precision of detection methods. Our findings demonstrate the advantages and limitations of each feature type, while also providing a clearer understanding of how feature affect model resilience and generalization in adversarial environments.

Additional Key Words and Phrases: Ethereum, Phishing Detection, Transaction Analysis, Machine learning

ACM Reference Format:

Ahod Alghuried, Abdulaziz Alghamdi, Ali Alkinoon, Soohyeon Choi, Manar Mohaisen, and David Mohaisen. 2025. Fishing for Phishers: Learning-Based Phishing Detection in Ethereum Transactions . 1, 1 (April 2025), 23 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Blockchain technology, particularly Ethereum, has revolutionized decentralized transactions, offering secure, transparent, and immutable transaction records [14, 22, 28]. However, as blockchain adoption increases, so does its appeal to cybercriminals, with phishing scams emerging as one of the most prevalent forms of attack [5, 6, 13, 15, 22–24, 37, 38]. Phishing scams, which exploit the trust inherent in blockchain transactions and their associated security challenges, account for millions of dollars in losses annually [9, 16, 32]. The highly publicized phishing attack on Uniswap Labs in 2022, where attackers stole over eight million dollars, serves as a stark reminder of the risks that Ethereum users face [29, 31, 46]. Furthermore, recent findings indicate that illicit activities,

*Corresponding author: David Mohaisen

Authors' addresses: Ahod Alghuried, ah104940@ucf.edu, University of Central Florida, Florida, USA; Abdulaziz Alghamdi, abdulaziz.alghamdi@ucf.edu, University of Central Florida, Florida, USA; Ali Alkinoon, alialkinoon@ucf.edu, University of Central Florida, Florida, USA; Soohyeon Choi, soohyeon.choi@ucf.edu, University of Central Florida, Florida, USA; Manar Mohaisen, Northeastern Illinois University, Chicago, Illinois, USA; David Mohaisen, mohaisen@ucf.edu, University of Central Florida, Florida, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2025 Association for Computing Machinery.

XXXX-XXXX/2025/4-ART \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

including phishing, have stolen over two billion USD from Ethereum users, highlighting the urgent need for effective detection mechanisms [1, 5, 24].

Phishing attacks on Ethereum are distinct from traditional phishing attacks, which typically involve fraudulent websites or emails aimed at stealing sensitive information such as passwords [25, 39, 48]. In contrast, blockchain phishing often exploits the transparency and pseudonymity of blockchain networks, targeting financial assets directly. These attacks are executed through compromised private keys, deceptive wallet addresses, and malicious smart contracts, enabling unauthorized transactions and asset transfers [8, 18, 49]. Moreover, while traditional phishing relies on social engineering to deceive users into providing personal information, Ethereum phishing can be automated using scripts that manipulate smart contracts or intercept transactions without direct interaction with the victim [4, 17]. This methodological shift underscores the automated nature of threats within the blockchain, necessitating advanced countermeasures [7, 35].

Phishing on blockchain networks such as Ethereum exploits the unique complexities of these systems, using techniques that blur the line between legitimate and malicious transactions [3, 33, 34, 49]. The ability of phishing to undermine confidence in blockchain threatens the very foundation of trust and security that these technologies promise. Addressing these risks requires approaches that go beyond traditional methods, incorporating both transactional analysis and advanced machine learning techniques to detect such attacks [21, 36, 40, 45].

Recent advances in phishing detection have increasingly leveraged machine learning models, particularly in blockchain transaction analysis. These models typically rely on explicit transactional features such as transaction values, gas usage, and timestamps [8, 18, 30]. However, while these features offer valuable insights into individual transaction behaviors, they often fail to capture the broader relational and temporal dynamics essential for detecting phishing [2, 22]. Graph-based approaches, which effectively model the Ethereum transaction network as a graph, have emerged as promising direction for phishing detection. These methods focus on implicit features that reveal interactions between addresses, thus enabling the detection of coordinated and rapid transaction patterns indicative of phishing activity [43] (see Figure 1). By utilizing advanced analytical methods, researchers aim to develop algorithms capable of discerning subtle signs of illicit activities amid legitimate transactions, addressing both data volume and the cunning nature of these frauds.

Despite progress in this field, current research often overlooks critical issues related to the robustness and scalability of phishing detection models. For instance, many studies focus on achieving high accuracy without addressing the inherent class imbalance in phishing datasets, where phishing transactions are significantly underrepresented compared to benign ones [8, 12, 49]. Moreover, there is limited exploration of how feature selection impacts the model's ability to generalize across diverse phishing scenarios. These challenges underscore the need for more comprehensive and systematic approaches to phishing detection in Ethereum networks.

Contributions. We address these limitations by developing a phishing detection model that systematically and independently evaluates two distinct feature sets: explicit transactional features and implicit graph-derived behavioral features. In contrast to prior work that often aggregates features without assessing their standalone contributions, we adopt a rigorous comparative methodology to isolate and quantify the impact of each feature type on model performance. Our contributions are as follows: 1) We design and implement a two-stage evaluation framework that contrasts the predictive capabilities of explicit and implicit features in phishing detection, providing insights into their capabilities and limitations. 2) We design a small, focused set of implicit features that describes how Ethereum addresses behave over time, for example, when they send transactions, how often, and on which days. These features go beyond raw transaction values by capturing behavioral patterns that are harder for attackers to fake or hide. 3) We build a Graph Convolutional

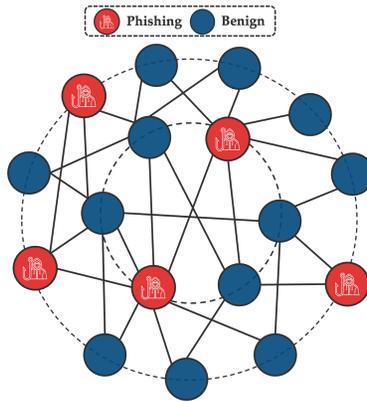


Fig. 1. Illustration of the Ethereum phishing scam network. Phishing addresses (red nodes) are interspersed among benign addresses (blue nodes), exhibiting similar transactional patterns, thereby complicating detection within the broader network structure.

Network (GCN) that learns from the connections between Ethereum addresses by analyzing how they interact over time and across the network. This allows the model to detect suspicious behavior based on both who is connected to whom and how they behave. 4) We evaluate our model on a large dataset of Ethereum transactions, demonstrating significant improvements in phishing detection performance. 5) Our findings show that a small number of carefully selected implicit features can outperform larger sets of basic transactional features used in prior studies, highlighting that effective feature design is more important than quantity, especially in adversarial settings.

Organization. The remainder of this paper is organized as follows. In section 2, we discuss related work on phishing detection. In section 3, we present our data collection process and feature extraction techniques. Our model architecture and experimental setup are outlined in section 4. We present our results in section 5, discussion in section 6, and offer concluding in section 7.

2 RELATED WORK

Phishing detection in Ethereum and other blockchains has attracted considerable research attention as these decentralized systems increasingly become targets for cybercriminals. Various methods have been proposed using explicit transactional and implicit graph-based features. This section reviews the key contributions in the field, organized by feature type, approach, and their relative effectiveness, as summarized in Table 1.

Explicit Transactional Features. Early approaches to phishing detection primarily leverage explicit transactional features extracted directly from blockchain data, such as the number of transactions, gas consumption, timestamps, and transaction values. These features are critical as they provide the initial set of data points that models use to identify potential threats. While these models perform well in detecting clear malicious behavior patterns, they often struggle to capture phishing scams' more complex relational dynamics.

Wen *et al.* [42] applied neural networks to explicit transactional features to identify temporal patterns indicative of phishing scams. Similarly, Kabla *et al.* [18] employed features such as from, input, blockHeight, and timeStamp to differentiate phishing accounts from legitimate ones. Despite their success, these models face challenges when dealing with phishing activities that involve sophisticated interactions between multiple blockchain addresses.

Table 1. A summary of recent phishing detection studies (since 2021), focusing on their employed features (explicit or implicit), methodological approaches (graph-based or machine learning-based), evaluation metrics (accuracy, precision, recall, and F1-score), and the sizes of the datasets used.

Papers	Year	Features	Method	Performance				Number of Instances	
				Acc.	F1-Score	Prec.	Rec.	phishing	benign
Chenet <i>et al.</i> [8]	2021	Implicit	Graph-based	0.57	0.23	0.72	0.14	1,157	2,973,382
Xia <i>et al.</i> [44]	2022	Implicit	Graph-based	-	0.81	0.81	0.82	451	12,834
Kabla <i>et al.</i> [18]	2022	Explicit	ML-based	0.98	0.98	0.98	0.98	5,448	79,216
Li <i>et al.</i> [21]	2022	Implicit	ML-based	0.92	0.81	0.77	0.85	4,932	6,844,050
Wu <i>et al.</i> [43]	2022	Implicit	ML-based	-	0.90	0.92	0.89	1,259	1,259
Fu <i>et al.</i> [12]	2022	Implicit	Graph-based	0.88	0.87	-	-	1,928	1,901
Zhou <i>et al.</i> [49]	2023	Explicit	Graph-based	0.98	0.97	0.96	0.99	1,659	5,805
Lin <i>et al.</i> [24]	2023	Explicit	ML-based	0.82	0.82	0.87	-	301	4,116,315
Li <i>et al.</i> [22]	2023	Implicit	Graph-based	-	0.92	0.91	0.92	5,639	25,000
Cheng <i>et al.</i> [11]	2024	-	ML-based	0.96	0.68	0.62	0.75	7,696	89,318
Liu <i>et al.</i> [25]	2024	Implicit	Graph-based	-	-	-	-	5,363	330,000
Our work	2024	Implicit	Graph-based	0.95	0.95	0.96	0.95	671,865	2,687,460

Lin *et al.* [24] refined phishing detection by incorporating explicit transactional features, although their method still encounters difficulties in identifying complex attack structures. Wang *et al.* [41] focused on ransomware detection in the Bitcoin network using features such as the total number of transactions, value exchanged, and the number of neighboring addresses. However, such an approach may fail to detect more elusive behaviors, especially when attackers obscure their operations using multiple addresses.

Yazdinejad *et al.* [47] combined explicit transactional data with device activity features in decentralized environments to detect cyber threats. Similarly, Kampers *et al.* [19] used features like trading volume, price fluctuations, and transaction frequency to uncover market manipulation tactics such as spoofing and wash trading. However, reliance on explicit features limits the detection of subtler, network-based strategies.

Ngo *et al.* [27] explored the integration of Generative Adversarial Networks (GANs) and anomaly detection to analyze explicit features such as transaction values for phishing detection. While effective in high-dimensional datasets, this method fails to address multi-node phishing attacks. Cheng *et al.* [10] introduced a hybrid model that integrates explicit transactional features with a long short-term memory (LSTM) module to capture evolving asset transfer paths, supported by a graph convolutional network (GCN) to analyze their structural properties.

Implicit Graph-Based Features. In contrast to explicit feature-based methods, implicit approaches focus on capturing the relational dynamics inherent in blockchain networks. These methods frequently employ Graph Neural Networks (GNNs) or similar graph-based models to analyze the network structure between addresses, enabling the detection of more phishing schemes.

Zhou *et al.* [49] introduced an approach called the Edge-Featured Graph Attention Network (EGAT), which leverages both node and edge features, such as transaction values, gas usage, and timestamps, to detect phishing behavior by focusing on the relationships between network nodes. This method uncovers hidden patterns that transactional feature-based models often miss, offering a deeper insight into the complex interplay of network interactions.

Li *et al.* [22] proposed the Transaction Graph Contrast Network (TGC), which utilizes contrastive learning to improve phishing detection by leveraging robust representations of Ethereum addresses within transaction subgraphs. TGC enhances its detection capabilities by introducing node-level and context-level contrast modules, making it particularly effective in large, dynamic networks.

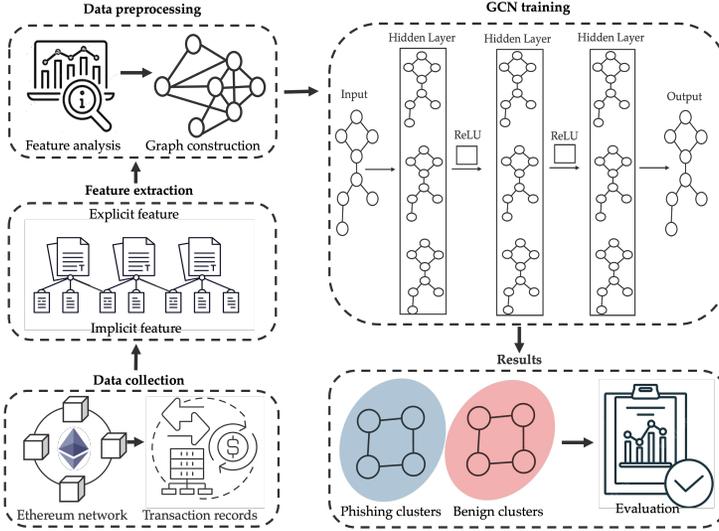


Fig. 2. An illustration of the proposed pipeline, integrating explicit and implicit features from the Ethereum network.

Limitations and Our Contribution. Despite notable progress, existing approaches still face key limitations. Models that rely only on explicit features [18, 42] often miss the broader network context that can reveal important phishing patterns. Conversely, graph-based models [22] may overlook transactional behaviors that are essential for understanding user activity. Our work addresses both issues by first evaluating explicit features to capture direct behavioral patterns, and then analyzing implicit, graph-based features to understand the relational structure between addresses. Unlike prior studies, we evaluate the model in two distinct phases—using explicit and implicit features separately—to better assess their individual impact on performance. We also introduce fine-grained temporal features, such as the time difference between consecutive transactions, which highlight behavioral anomalies that are often missed in existing graph-based approaches. To validate the importance of these features, we use a Random Forest classifier (RF) to assess their predictive value in distinguishing phishing from benign activity. In addition, our study highlights the most informative implicit features. It incorporates a weighted loss function to address the class imbalance, resulting in improved detection performance across multiple metrics.

3 METHODOLOGY

Our pipeline, shown in Figure 2, involves several steps as follows. First, Ethereum transactions are collected and labeled as phishing or benign. Then, both explicit and implicit features are extracted from the data. Weighted loss functions are applied to address the class imbalance. Subsequently, these features are fed into GCN, aggregating information from neighboring nodes to classify addresses. Finally, the model’s performance is evaluated using key metrics. In the following, we elaborate on the various steps in our pipeline.

3.1 Data Collection

The primary goal of data collection was to compile a comprehensive dataset of Ethereum transactions, explicitly focusing on phishing-related and benign activities. This dataset forms the foundation for developing and testing models to detect phishing transactions on the Ethereum network.

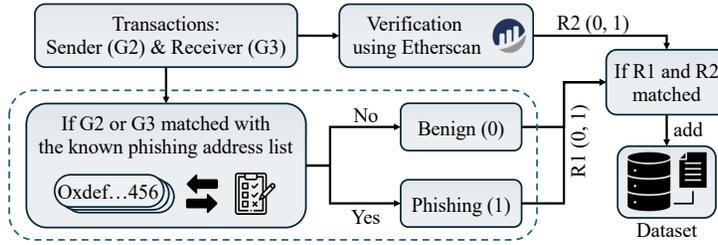


Fig. 3. Illustrating the labeling process. G2 and G3 represent the sender and receiver addresses. R1 and R2 are labeling results. In R1, addresses are first matched against the known phishing list. If a match is found, R2 performs manual verification via Etherscan to ensure labeling accuracy.

The collected data includes transactions associated with known phishing addresses and benign transactions, providing a balanced view for robust model training.

Data Collection Process. To gather the transaction history for each phishing address, the Etherscan API¹ was employed. The API provided access to detailed transactions, including block numbers, timestamps, sender and receiver addresses, and transaction values, which are essential for analyzing blockchain activities. The phishing addresses were sourced from a publicly available dataset on GitHub², compiled initially and utilized in [20]. This dataset includes 7,915 unique Ethereum addresses that have been flagged for their involvement in phishing activities. These addresses were verified using the Etherscan API to confirm their involvement in phishing activities. Each address was cross-referenced against transaction histories and community reports to ensure the accuracy of their classification as malicious. This rigorous process supports our dataset reliability by substantiating the addresses phishing history.

A Python script was designed to automate the data collection process by querying the Etherscan API for transaction data associated with identified phishing addresses. The script systematically extracted details such as block numbers, timestamps, and transaction values, and organized them into a structured CSV file for efficient analysis.

Benign transactions, defined as those not associated with the identified phishing addresses, were carefully collected using the same API parameters and methods. Using the identical API parameters and collection methods, the benign transactions were directly comparable to the phishing transactions regarding data structure and content.

Once the phishing and benign transactions were collected, they were combined into a single dataset. The dataset was labeled to differentiate between phishing and benign transactions. Transactions associated with the 7,915 phishing addresses were labeled as phishing, while those related to other addresses were labeled as benign. The final dataset includes the following features (explained in Table 2): Timestamp (G1), Transaction Hash, Sender Address (G2), Receiver Address (G3), Transaction Value (G4), Gas Used (G7), Gas Price (G6) and Label. The dataset's composition concluded with around 2M benign and around 600K phishing transactions, providing a substantial basis for our phishing detection model.

Labeling Strategy. Each Ethereum transaction in this study was labeled as either phishing (1) or benign (0). The labeling process is illustrated in Figure 3. G2 and G3 represent the sender and receiver addresses, while R1 and R2 are intermediate labeling results. In the first step (R1), an address is matched against the known phishing list compiled from a publicly available dataset used

¹ Accessible at: <https://etherscan.io> (accessed November 2024).

² Accessible at: <https://github.com/YNclusk/scamsonethereum> (accessed November 2024).

in prior work [20]. If a match is found, a second step (R2) involves manual verification through the Etherscan API to confirm phishing activity and reduce labeling errors.

Only transactions involving addresses directly (1-hop) connected to verified phishing addresses were labeled as phishing. This conservative approach minimizes false positives, though it may overlook some phishing behaviors. For example, if address "0xabc...123", listed in the phishing dataset, sends funds to "0xdef...456", that transaction is labeled as phishing.

For benign labels, we used addresses with no known history of phishing, scams, or related activity. Although some benign addresses may later be flagged as malicious, the risk is minimized by selecting historically clean addresses. For example, if address "0x111...aaa" sends funds to "0x222...bbb", and neither address appears in the phishing list, the transaction is labeled as benign. To ensure consistency and fairness, both phishing and benign transactions were collected using the same Etherscan API and stored in a unified data format.

Data Balancing. A weighted loss function was employed to address the significant class imbalance in the dataset, where phishing transactions constituted only 7.63% of the total transactions. This approach involved assigning higher weights to the underrepresented phishing class and lower weights to the more prevalent benign class. This weighting strategy compels the model to focus more on phishing, despite their relative scarcity, thereby improving the model's sensitivity to phishing. The loss function is defined as follows:

$$\mathcal{L} = - \sum_{i=1}^N w_{y_i} \cdot \log p(y_i) \quad (1)$$

where N is the number of nodes, $y_i \in \{0, 1\}$ is the true class label of node i , $p(y_i)$ is the model's predicted probability for the true class, and w_{y_i} is the weight assigned to the class.

A weighted loss function (Equation 1) was employed to address the problem of class imbalance in the dataset, as phishing nodes occur much less frequently compared to benign nodes. This method increases the penalty for incorrectly classifying phishing nodes, encouraging the model to give special attention to these rare but important cases. As a result, the model avoids overly favoring the majority class and improves its ability to detect phishing behavior effectively. The weighted loss function was selected over alternatives like focal loss because it is simpler to implement, provides better stability during training, and does not require tuning additional parameters. Furthermore, it naturally aligns with graph neural network models, allowing effective learning from imbalanced data without adding unnecessary complexity. This choice is particularly valuable in phishing detection tasks, where overlooking phishing nodes typically causes more harm than mistakenly flagging benign ones. Ultimately, the use of a weighted loss function helps the model to become more sensitive to the minority class, leading to more balanced and accurate results.

3.2 Feature Selection Rationale

The selection of features for this study was guided by a combination of observational insights and established research on blockchain behaviors [8, 25, 49]. This informed approach ensures that the features we focus on are both indicative of phishing activities and reflective of the unique dynamics within blockchain transactions.

Transaction Volume (G4). Transaction volume is a critical indicator in phishing detection. Phishing transactions often involve volumes that are either significantly higher or lower than those of typical benign transactions. High-value transactions may be executed with the intent to quickly drain compromised wallets, capitalizing on the rapid execution capabilities of blockchain technologies. Conversely, attackers might also distribute funds in numerous smaller transactions to mimic routine user behavior, thereby evading detection systems that are tuned to spot large, irregular

transfers. This dichotomy in transaction sizes provides crucial signals for distinguishing between benign and malicious activities within the network.

Gas Usage (G7). Manipulation of gas usage is a common tactic in phishing schemes, utilized to optimize the execution success of malicious transactions. High gas fees are often prioritized to ensure fraudulent transactions are processed swiftly, outpacing any reactive security measures. On the other end of the spectrum, lower-than-average gas fees can be indicative of an attacker's strategy to minimize operational costs during extensive phishing campaigns that require the execution of numerous transactions. By analyzing patterns in gas usage, our model can identify deviations from the norm that suggest underlying phishing attempts.

Timing Features (G1). The timing of transactions offers profound insights into user behavior on the blockchain. Phishing operations frequently exhibit abnormal timing patterns—such as sudden bursts of high-intensity activity followed by extended periods of dormancy—that starkly contrast with the more uniform transaction timing of regular users. These anomalies in transaction timing are vital for identifying potential phishing activities, as they often reflect the opportunistic nature of attacks and the subsequent attempts to hide illicit actions within normal traffic flows.

Node Connectivity (Implicit Features). The relational dynamics between nodes, or addresses, in the blockchain provide a ground for detecting phishing. Patterns such as repetitive transactions with certain clusters of addresses or the sudden emergence of transactions with new, previously unrelated nodes can signal the operation of controlled accounts engaged in phishing. These connectivity patterns, especially when they deviate from typical user behavior, are strong indicators of coordinated malicious activities.

The efficacy of implicit features in our phishing detection framework hinges on their ability to uncover subtle yet consistent anomalies in transaction patterns and inter-node relationships. The theoretical underpinnings and empirical validations from previous studies bolster our reliance on graph-based features to robustly detect anomalies in network security [8, 21, 44]. Moreover, the inherent transparency of blockchain transactions allows for a comprehensive analysis of these relational patterns, rendering these implicit features especially powerful for uncovering and understanding the sophisticated strategies employed in phishing attacks. This detailed exploration not only aids in effectively identifying phishing activities but also enhances our understanding of the interaction paradigms within Ethereum's complex system.

3.3 Feature Extraction

This work employs a GCN to detect phishing addresses on the Ethereum blockchain by testing the model with two feature sets: explicit transactional features and then implicit graph-based features. The objective is to compare which feature set is more effective in improving the model's performance and delivering better results. Exploring these different aspects allows us to understand the impact of each feature type on the predictive capabilities of the model, providing insights into optimizing detection for enhanced accuracy and efficiency.

Explicit Features. The explicit features refer to the direct, transaction-specific attributes extracted from the raw Ethereum data, providing key insights into the fundamental properties of each transaction. These features include critical fields such as transaction timestamp (G1), sender (G2) and recipient (G3) addresses, value transferred (G4), and gas-related features (G5, G6, G7). The goal of leveraging explicit features is to capture fundamental transactional behavior, such as the timing, volume, and cost-efficiency of transactions, which can provide early indicators of phishing. This direct approach to feature extraction helps in quickly assessing transaction integrity and forming a preliminary defense against phishing tactics.

Table 2. Summary of explicit and implicit features transactions on the Ethereum network.

T	Features Used	G	Explanation
Explicit Features	TimeStamp	G1	The timestamp of the transaction
	From	G2	The sender's address
	To	G3	The recipient's address
	Value	G4	The amount of Ether transferred
	Gas	G5	The amount of gas provided for the transaction
	GasPrice	G6	The price of gas (in Wei) used for the transaction
	GasUsed	G7	The actual amount of gas used in the transaction
Implicit Features	From_tx_cnt	G2	Number of transactions initiated by address
	To_tx_cnt	G3	Number of transactions received by address
	Total_val_sent	G4	Total Ether sent by address
	Total_val_rec'd	G4	Total Ether received by address
	Avg_gas_sent	G7	Average gas for transactions sent
	Avg_gas_rec'd	G7	Average gas for transactions received
	Mean_hour_sent	G1	Average hour of day when transactions are sent
	Mean_hour_rec'd	G1	Average hour of day when transactions received
	Std_hour_sent	G1	Standard deviation of hour transactions sent
	Std_hour_rec'd	G1	Standard deviation of hour transactions received
	Avg_time_bw_tx	G1	Average time b/w consecutive transactions sent
	Min_time_bw_tx	G1	Minimum time b/w consecutive transactions sent
	Max_time_bw_tx	G1	Maximum time b/w consecutive transactions sent
	Tx_duration	G1	Duration b/w first and last transactions of address
	Wd_tx_ratio_sent	G1	Proportion of transactions sent on weekends
Wd_tx_ratio_rec'd	G1	Proportion of transactions received on weekends	

Implicit Features. To capture the deeper structural and temporal dynamics within the Ethereum transaction, implicit features were extracted from the transaction graph. Unlike explicit features, which focus on individual transaction details, implicit features highlight patterns from the interactions between nodes (addresses) over time, offering a broader perspective on behavior. These features are essential for understanding the complex network relationships and potential collusion among addresses that are often characteristic of phishing.

The implicit features analyze how nodes interact within the network, uncovering patterns like bursts of rapid transactions followed by inactivity, often associated with phishing activities. Time-based features such as the average time between transactions (G1), the standard deviation of transaction times (G1), and identifying irregular transaction patterns common in phishing schemes.

A crucial component of implicit features is the examination of node relationships. For instance, repetitive or coordinated behaviors, such as frequent small-value transfers, can indicate phishing clusters. Considering node behavior within the entire transaction graph allowed for identifying more subtle patterns linked to phishing. Temporal characteristics also proved valuable in distinguishing phishing from benign addresses. Metrics such as transaction time intervals and the duration between a node's first and last transactions helped detect anomalies in timing. A detailed description of the explicit and implicit features is provided in Table 2. These analyses provide a deeper insight into the tactics employed by attackers, enabling more effective prevention and mitigation strategies.

Feature Analysis and Selection. To identify the most distinguishing features for classifying phishing and benign nodes, we utilized statistical analysis and Random Forest (RF). Random Forest is an ensemble learning method that constructs multiple decision trees during training and determines the output class based on the majority vote among the trees. It is widely used due to its robustness, interpretability, and ability to handle high-dimensional data. Statistical analysis was first employed to compare key features between phishing and benign nodes, such as transaction amounts, gas usage,

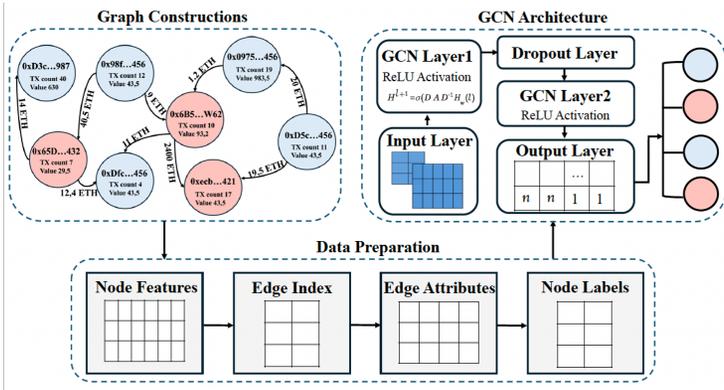


Fig. 4. Construction of a directed Ethereum transaction graph and its transformation into PyTorch Geometric inputs. The resulting data is processed by a GCN to classify addresses as phishing or benign.

and transaction timing. This helped uncover significant behavioral differences between the two types of nodes. The RF classifier was then applied to rank the importance of the extracted features. This dual approach ensures a comprehensive understanding of feature relevance, enhancing the model's accuracy by focusing on the most predictive attributes. RF provides a quantifiable measure of feature importance, highlighting the most critical features for distinguishing phishing nodes from benign ones. This method validated the statistical analysis findings and helped refine the feature set by prioritizing attributes that proved to be key indicators of phishing behavior.

Data Preprocessing. Before applying the model, several preprocessing steps were performed to ensure the data was in an appropriate format and range for effective and robust model training. Namely, these steps included normalization and scaling. These preprocessing efforts align data from diverse sources and scales, enhancing model training and performance predictability.

① *Normalization:* To alleviate the bias due to different feature scales, the Min-Max scaling was applied to the features. This scaling ensures that all features are transformed into a $[0,1]$ range, which then aids in improving convergence during training. Normalization is particularly important in handling outliers and reducing skewness in data distribution. The `MinMaxScaler` from `scikit-learn` was utilized to rescale each feature based on its minimum and maximum values within the dataset.

② *Scaling:* The features were categorized into explicit (e.g., timestamp, value) and implicit features (e.g., transaction frequency, behavioral patterns). After scaling, these features were integrated into a node feature matrix that was used as input for the GCN model. This structured approach to feature integration facilitates more effective learning by the neural network, optimizing the detection of complex phishing patterns.

3.4 Graph Construction

A directed graph was constructed using NetworkX [26], where each node represents an Ethereum address and each edge represents a transaction between two addresses. Figure 4 illustrates the full pipeline, including graph construction, data preparation, and the GCN architecture used for address classification. This graph captures the underlying structure of the Ethereum transaction network and allows for a detailed analysis of how Ethereum addresses interact. Understanding these interactions is essential for visualizing complex network dynamics and serves as a foundation

for analytical models. After construction, the graph is transformed into a format compatible with PyTorch Geometric for training the GCN. The conversion involves:

- (1) **Node Features.** A matrix where each row corresponds to the feature vector of a node, representing the explicit and implicit attributes of the Ethereum addresses. This matrix supplies the necessary data for the GCN to evaluate each node based on its distinct characteristics.
- (2) **Edge Index.** A tensor representing the directed edges between nodes, indicating the relationships between the sender and the receiver addresses (edge). This tensor is vital for the GCN to recognize and utilize the connections between nodes, facilitating effective feature propagation through the network.
- (3) **Edge Attributes.** A label indicating whether the node is involved in phishing activity (1 for phishing, 0 for benign). These labels are imperative for training the model to accurately classify nodes based on their transactional behaviors and associations.
- (4) **Node Labels.** A label indicating whether the corresponding labeled node is phishing 1 or benign 0. This classification supports the supervised learning process, guiding the GCN in generating precise predictive outcomes.

Model Architecture. A GCN was implemented to classify Ethereum addresses as phishing or benign. The GCN aggregates features from neighboring nodes and propagates information through the graph, enabling the model to learn embeddings for each node by considering both its features and those of its neighbors. This architecture takes advantage of network connectivity and feature sets to uncover subtle indicators of malicious activities that traditional methods might miss.

- (1) **Input Layer.** The node feature matrix, where each node is represented by a vector of explicit features. This layer is the entry point for data into the GCN, setting the foundation for complex pattern detection.
- (2) **Hidden Layers.** Multiple GCN layers that apply graph convolutions to propagate information between neighboring nodes. Each GCN layer updates the representation of a node by aggregating the features of its neighbors. These layers refine the raw data into actionable insights, crucial for the detection process.
- (3) **Activation Functions.** ReLU (Rectified Linear Unit) activation is applied after each hidden layer to inject non-linearity into the model, enhancing its capability to model complex relationships.
- (4) **Dropout.** Dropout regularization is strategically applied to the hidden layers to effectively prevent overfitting, especially in the presence of highly imbalanced data. This technique ensures that the model remains generalizable and effective against various forms of data variance.
- (5) **Output Layer.** The output layer uses a softmax activation function to produce the final classification (phishing or benign) for each node. This layer determines the ultimate classification outcome, translating learned embeddings into definitive labels.

Graph Construction. The Ethereum transaction network was modeled as a directed graph $G = (V, E)$, where:

- ✧ **Nodes.** V represents the nodes corresponding to Ethereum addresses. Each node is enriched with various features, either explicit or implicit, capturing important aspects of the transaction behavior.
- ✧ **Edges.** E represents the directed edges corresponding to transactions between addresses. Each edge connects a sender node u to a receiver node v , representing a transaction from u to v , with additional attributes.

Graph Convolutional Layers. The core of the GCN model consists of graph convolutional layers, which operate on the adjacency matrix A of the transaction graph and the feature matrix X , with a feature vector for each node. The GCN aggregates the features of each node's neighbors, propagating the aggregated information through multiple layers. This enables the model to learn implicit features that capture the relationships between addresses.

The **layer-wise propagation rule** for the GCN is given by:

$$H^{(l+1)} = \sigma \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad (2)$$

Where A is the adjacency matrix of the graph, representing connections between nodes, D is the degree matrix, where each diagonal element represents the degree of the corresponding node, $H^{(l)}$ is the hidden state at layer l , representing the node embeddings at that layer, $W^{(l)}$ is the weight matrix for layer l , which is learned during training, and σ is an activation function, ReLU in this case, applied element wise.

Evaluation Metrics. The performance of the model was assessed using the following key metrics, each defined mathematically.

- ① *Accuracy*: Accuracy measures the overall proportion of correctly classified phishing and benign nodes. It is defined as: $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$, where TP (True Positives) are correctly identified phishing, TN (True Negatives) are correctly identified benign, FP (False Positives) are benign misclassified as phishing, and FN (False Negatives) are phishing misclassified.
- ② *Precision*: Precision evaluates the accuracy of the phishing node predictions, focusing on minimizing false positives. It is defined as: $\text{Precision} = \frac{TP}{TP+FP}$. Precision reflects the proportion of correctly identified phishing nodes out of all nodes predicted as phishing.
- ③ *Recall*: Also known as sensitivity or true positive rate, recall measures the model's ability to detect actual phishing nodes. It is defined as: $\text{Recall} = \frac{TP}{TP+FN}$. Recall emphasizes the proportion of actual phishing nodes that were correctly identified by the model.
- ④ *F1-Score*: The F1-score provides a balanced measure of precision and recall as the harmonic mean of the two and is useful when there is an imbalance between the classes. The F1-score is defined as: $\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$. This metric is valuable in assessing performance when both precision and recall are critical.

3.5 System Model

This work presents a behavior-based detection system that operates at the Ethereum address level. Rather than focusing on transactional attributes, the system is built around implicit behavioral characteristics—subtle, often hidden patterns that are difficult for adversaries to forge or predict. The goal is to distinguish phishing addresses from benign ones by modeling how addresses behave over time and within their transactional context.

The system starts by constructing a directed transaction graph, where each node represents an Ethereum address, and each edge represents a transaction between two addresses. This graph is the basis for learning behavioral features. Each node is embedded with fine-grained behavioral signals derived from its historical activity, including how frequently it transacts, the timing and variability of its transactions, and its interaction rhythm with other addresses.

Key features include average inter-transaction times, weekend activity ratios, and standard deviation in transaction hours—attributes that together form a behavioral fingerprint of each address. These patterns reveal how an address typically operates and whether its behavior aligns more closely with legitimate users or with patterns common to phishing activity. For instance, some phishing addresses stayed inactive for long periods and then started sending a series of low-value transactions during early morning hours (e.g., 2–4 AM UTC). These transactions were usually sent

to new or unfamiliar addresses, which may suggest testing or hiding funds. These actions may not seem suspicious. However, when combined with short gaps between transactions, regular timing, or sudden changes in gas prices, the overall pattern can point to automated or hidden activity that is less common among normal user behavior.

To capture these relationships, the system applies GCN to model these behavioral relationships by aggregating feature information from neighboring nodes in the transaction graph. This mechanism allows the model to learn from the features of each address in isolation and the transactional context in which it operates, such as repeated interactions with certain nodes, timing irregularities, or sudden shifts in activity patterns. Additionally, the final embedding for each address is passed through a classification layer that outputs a phishing probability score. Addresses exceeding a fixed threshold are classified as phishing, enabling proactive intervention. The system is also designed to operate passively on historical transaction data without relying on future information. Classification decisions are made at the address level based on past behavior within a fixed observation window, making the system suitable for deployment in near-real-time detection scenarios.

The system functions as a behavioral detection framework by combining graph-based learning with implicit temporal and behavioral features. This enables it to flag previously unseen phishing addresses based on their behavior over time and within the network structure. This approach is particularly effective at detecting emerging or stealthy threats that do not match the known phishing profiles but still exhibit suspicious behavior.

3.6 Threat Model

This work assumes a threat model where the attacker is a regular participant in the Ethereum network with no privileged access. The adversary cannot see system internals or influence the detection model directly. Instead, the attacker operates by creating and controlling multiple addresses, all of which interact with the blockchain through normal transactions.

The attacker's main goal is to carry out phishing campaigns by tricking users into sending funds to addresses under the attacker's control. To avoid detection, the attacker may try to mimic benign transaction behavior. For example, they may send low-value transactions, choose normal gas prices, or maintain long idle periods before starting activity. They can also spread activity across several addresses to make their actions less noticeable.

However, the detection system is designed to capture behavioral patterns that are difficult to fake. The system relies on implicit features, such as transaction frequency, average time between transactions, gas usage patterns, and transaction timing variability. That reflects how addresses behave over time. These features are based on the statistical behavior observed across historical activity. As a result, they are harder for attackers to manipulate without leaving detectable traces. For instance, an attacker might register a new address and leave it inactive for several days. Then, during early morning hours (*e.g.*, 3:00–4:00 AM UTC), the address suddenly begins to send multiple small transactions to unfamiliar addresses. The attacker increases the gas price to prioritize these transactions. While each individual action might appear normal, the system detects the combination of unusual timing, abrupt change in behavior, and tightly spaced transactions as a suspicious pattern.

Because the model learns from a wide set of implicit features and considers the address's historical and relational behavior in the graph, it can flag such addresses as phishing, even if the attacker tries to blend in. The use of graph-based learning further strengthens detection by aggregating information from neighboring nodes, making it harder for the attacker to isolate their actions.

In summary, the assumed adversary is capable of imitating surface-level activity, but cannot easily avoid exposing behavioral inconsistencies. The system's use of implicit temporal and structural features enables it to detect phishing strategies that would be missed by methods relying only on explicit or static information.

Table 3. Training and testing dataset sizes.

Dataset	Phishing Nodes	Benign Nodes	Edges
Training Set	537,492	2,149,968	1,234,355
Testing Set	134,373	537,492	72,848
Total	671,865	2,687,460	1,307,203

4 EXPERIMENTAL SETUP

This section outlines the steps taken to evaluate the performance of the GCN model in detecting phishing activities on the Ethereum blockchain. The experiment was divided into three main phases: (1) data preparation, (2) model training and testing, and (3) performance evaluation. The goal of the experiment was to test the efficacy of explicit versus implicit feature sets in distinguishing between phishing and benign transactions. This comparative analysis aims to identify the most impactful features and refine the model's predictive capabilities for real-world applications.

4.1 Data Preparation

Data Collection. Ethereum transaction data was collected using the Etherscan API, comprising 671,865 phishing transactions from 7,915 phishing addresses and 2,687,460 benign transactions. This resulted in a final dataset of 3,359,325 transactions. The comprehensive dataset allows for a robust analysis of phishing patterns and aids in the development of a nuanced understanding of transaction behaviors.

Data Cleaning and Labeling. The dataset was thoroughly cleaned by removing duplicates, null values, and irrelevant transactions. Each transaction was carefully labeled as either phishing or benign. The label distribution was as follows: around 600K phishing and around 2M benign transactions. This step ensures the accuracy and reliability of the training and testing datasets, providing a solid foundation for the subsequent machine learning tasks. The final dataset included important features such as Block Number, Timestamp, Transaction Hash, Sender Address, Receiver Address, Transaction Value, Gas Used, Gas Price, and the phishing or benign label.

Data Splitting. The dataset, consisting of phishing and benign addresses, was divided into training and testing using an 80/20 split to ensure proportional representation and effective model training. This split strategy supports the model's ability to generalize well to new, unseen data while ensuring that it is robustly trained on a substantial portion of the available data. The specifics of the split are shown in the Table 3.

Statistics. The training set contains around 500K phishing transactions and around 2M benign transactions, connected by around 1M transaction edges. This large number of edges facilitates a detailed network analysis, enhancing the model's ability to learn from complex relational data. The testing set comprises around 100K phishing transactions and around 500K benign transactions, with 70K edges.

Class Balancing. We employed class weighting during training due to the significant class imbalance, where phishing nodes make up a small portion of the overall dataset. This method adjusts the model's focus, ensuring that less frequent but critical phishing cases are not overlooked. This approach assigns higher weights to phishing nodes, ensuring the model pays more attention to identifying phishing addresses.

Normalization. Min-Max scaling was applied to the explicit features (e.g., transaction value, gas used, gas price), normalizing their values between 0 and 1. This preprocessing step ensures that all

features contribute equally to the training process, preventing any single feature from dominating due to differences in scale.

4.2 Feature Engineering

Both explicit and implicit features were tested separately to determine their individual contributions to enhancing phishing detection performance. Initially, the model was evaluated using only explicit features to understand their baseline. Subsequently, implicit features were applied in isolation to assess any improvements or changes in detection capabilities. This separate evaluation allows us to precisely compare the impacts of each type of feature on the model's performance.

Implicit Features. Various implicit features were extracted from Ethereum transaction data to characterize each node's behavior within the transaction network. The following key features were derived from the raw transaction data:

- (1) **Transaction Frequency.** The number of transactions initiated by a node (transactions initiated) and the number of transactions received by a node (transactions received) are closely monitored. This measure helps to identify nodes with unusually high or low activity, which can be clearly indicative of either central hubs in legitimate operations or potential points of compromise in fraudulent schemes.
- (2) **Total Transaction Value.** The total Ether value sent and received by each node (total value sent, total value received). Monitoring the flow of significant sums can help flag nodes that are potentially involved in money laundering or the unauthorized transfer of funds as part of phishing scams.
- (3) **Gas Usage.** The average gas used by each node for sending and receiving transactions (average gas used for sending, average gas used for receiving) is recorded. Patterns in gas usage can provide vital clues about nodes prioritizing transactions to strategically facilitate quick settlements, a common tactic in phishing to effectively avoid detection.
- (4) **Time-Based Features** Time-related patterns were captured by computing the mean and standard deviation of transaction hours for sent and received transactions. These metrics include the mean transaction hour for sending, the standard deviation of transaction hour for sending, the mean transaction hour for receiving, and the standard deviation of transaction hour for receiving. By analyzing these figures, we can discern not only the typical hours during which users are most active but also the variability in their transaction times, indicating their temporal transaction habits. This helps in understanding the regularity or randomness of user activities in terms of time, facilitating insights into user behavior and system usage trends.
- (5) **Transaction Duration.** The time difference between a node's first and last transaction (duration), as well as the average, minimum, and maximum time intervals between transactions (average time between transactions, minimum time between transactions, maximum time between transactions) were calculated to capture temporal transaction patterns. Examining the frequency and regularity of transactions over time allows for the detection of irregular patterns that deviate from normal user behavior, often associated with phishing.
- (6) **Weekend Transaction Ratio.** The proportion of transactions initiated or received during weekends (weekend transaction ratio for sending, weekend transaction ratio for receiving) was extracted, as phishing addresses may follow distinct temporal patterns compared to benign addresses. By determining the ratio of weekend transactions, it is possible to identify anomalies or consistent trends that clearly differentiate suspicious activities from normal behaviors. This metric helps pinpoint deviations in transactional behavior typically unseen during the regular working week, thus providing a clearer perspective on potentially malicious operations.

To capture more complex interactions and patterns within the transaction network that may not be directly evident from explicit features, graph-based techniques were employed to extract implicit features. Specifically, GCN was applied to the Ethereum transaction graph, where nodes represent Ethereum addresses and edges represent transactions. Through the GCN layers, each node's embedding was iteratively updated by aggregating information from its neighboring nodes, capturing the structural and relational properties of the graph. These implicit features reflect the deeper, latent patterns of node connectivity and behavior within the network, which are critical in distinguishing phishing nodes from benign ones. This method enhances the model's capability to discern complex patterns of interaction within the Ethereum transaction graph, improving its effectiveness in detecting and addressing potential phishing threats based on network behavior.

5 RESULTS AND ANALYSIS

In this section, we present the GCN's performance when trained and evaluated on explicit and implicit features. We assess the model's effectiveness using key metrics such as precision, recall, and F1-score, focusing on distinguishing phishing from benign nodes in the Ethereum network. To identify the most distinguishing features, we employ statistical analysis and RF classifier.

5.1 Distinguishing Features

We conducted a detailed statistical analysis of key features in transaction data to accurately differentiate phishing from benign nodes. Table 4 summarizes the features most indicative of phishing behavior, providing insights into the patterns distinguishing these nodes.

Characteristics. We found that the phishing nodes send significantly higher transaction amounts than benign nodes, with an average of 8.27×10^{19} compared to 3.72×10^{19} . Despite similar Max values, phishing nodes show greater variability, making this a key differentiating feature. Phishing nodes also receive more value, averaging 3.88×10^{19} , compared to 1.73×10^{19} for benign nodes. While benign has higher Max values, phishing exhibits more consistent high-value behavior.

Although benign nodes generally use more gas on average (73,375 vs. 44,467 for phishing nodes), phishing nodes display higher outliers, with Max values reaching 2.52×10^7 , clearly indicating sporadic spikes in gas usage. Phishing nodes exhibit longer intervals between transactions, averaging 1.71×10^5 seconds, while benign nodes average -5.16×10^4 seconds. Phishing nodes also show significantly higher variance in the hours they receive transactions (1.00 vs. 0.37 for benign nodes), reflecting more irregular and unpredictable behavior.

The most significant differences between phishing and benign nodes lie notably in transaction volume and timing. Phishing nodes send and receive considerably larger amounts, show higher variability in gas usage, and typically have longer intervals between transactions, making these key indicators essential for distinguishing them from benign nodes.

Random Forest Classifier for Feature Importance. RF classifier was applied to accurately quantify the most important features for distinguishing phishing from benign nodes. The RF analysis identified the total value sent as the most important feature (score of 0.40), aligning perfectly with the statistical analysis. The average gas used when sending transactions (score of 0.31) also emerged as a critical feature. Timing-related features such as the average time between transactions further supported the classification of phishing nodes. These feature importance scores are represented in Figure 5, which highlights the top 10 most influential implicit features for distinguishing phishing behavior, as ranked by the RF model.

Table 4. Comparison of implicit features for phishing vs. benign nodes in terms of mean, max, and standard deviation (Std). Highlighted cells indicate notably higher phishing statistics.

Feature	Phishing Nodes			Benign Nodes		
	Mean	Max	Std	Mean	Max	Std
G2/G3	5.20	20,393	107.65	3.40	20,393	64.53
G4 (Sent)	8.27×10^{19}	4.61×10^{24}	1.26×10^{22}	3.73×10^{19}	4.61×10^{24}	7.09×10^{21}
G4 (Received)	3.88×10^{19}	1.55×10^{24}	4.52×10^{21}	1.74×10^{19}	2.33×10^{24}	3.65×10^{21}
G7 (Sent)	44,467.31	25,258,280	149,682.80	73,375.15	7,600,027	120,198.40
G7 (Received)	1,289.49	89,612	5,111.78	782.73	89,612	4,245.39
G1 (Mean hour sent)	11.99	23	6.08	11.63	23	6.01
G1 (Std hour sent)	7.99	264.50	23.77	10.71	264.50	27.94
G1 (Std hour received)	1.00	264.50	8.01	0.38	264.50	5.24
G1 (Avg time between tx)	171,726.90	222,295,100	7,434,316	-51,675.76	270,628,700	10,037,440
G1 (Weekend tx ratio sent)	0.29	1	0.42	0.24	1	0.39
G1 (Weekend tx ratio recvd)	0.02	1	0.11	0.01	1	0.09

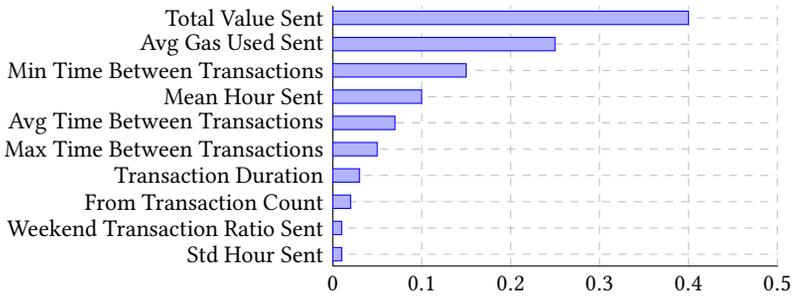


Fig. 5. Top 10 important features based on Random Forest.

Takeaway

The combination of high transaction volumes and irregular gas usage and timing provides the strongest basis for distinguishing phishing nodes. These findings highlight the value of combining statistical analysis with machine learning to identify suspicious behavior in Ethereum networks.

5.2 Performance on Explicit Features

In our first experiment, we trained the GCN using only explicit features, which included basic transactional details such as value, gas used, and timestamps. The model achieved an overall accuracy of 0.79, showing reasonable success in classifying benign nodes. However, its performance in detecting phishing nodes was poor, as reflected by a recall of zero, indicating a complete failure to correctly identify phishing.

Table 5 presents the performance metrics. The low recall for phishing nodes suggests that relying on explicit features limits the model’s ability to capture the nuanced and coordinated behaviors that characterize phishing activities. While the model performed well in classifying benign nodes with a precision of 0.79, it struggled with phishing detection, as evidenced by a precision of only 0.76 and an F1-score of 0.01 for phishing nodes. This result highlights the need for more complex features that can better distinguish between benign and phishing behaviors.

Table 5. Performance of GCN model using explicit for Phishing and Benign Transactions.

Metric	Benign	Phishing	Weighted Avg
Precision	0.79	0.76	0.79
Recall	1.00	0.00	0.79
F1-Score	0.89	0.01	0.70

Table 6. Performance of GCN model using implicit for Phishing and Benign Transactions.

Metric	Benign	Phishing	Weighted Avg
Precision	0.98	0.25	0.96
Recall	0.97	0.33	0.95
F1-Score	0.97	0.28	0.95

5.3 Performance on Implicit Features

The second experiment expanded the feature set by incorporating implicit features derived from the transaction graph. These features capture more complex relational and behavioral patterns, such as transaction frequency and node interactions, providing a richer representation of the Ethereum network's transactional dynamics. When implicit features were included, the model's overall accuracy improved significantly to 0.95, and the recall for phishing nodes increased to 0.33, (see Table 6). This improvement indicates that implicit, graph-based features are more effective at capturing the underlying behaviors associated with phishing activities, which often involve coordinated and rapid transactions between nodes.

The classification report for implicit features underscores the effectiveness of incorporating network-level information into the model. Precision for benign nodes remained high at 0.98, while phishing node precision, although still low at 0.25, shows a marked improvement compared to the experiment with explicit features. The F1-score for phishing nodes increased to 0.28, reflecting a better balance between precision and recall, and further demonstrating the utility of implicit features in detecting phishing activities.

Takeaway

The results of both experiments highlight the limitations of explicit features and the advantages of incorporating implicit, graph-based features. While phishing detection remains challenging, the improvements with implicit features offer promising directions for future detection on blockchain.

5.4 Comparative Evaluation

To evaluate our approach, we compare our model against recent phishing detection systems as summarized in Table 1. These baselines traverse various machine learning, feature types, and experimental settings, enabling a multi-faceted assessment of our method's effectiveness.

Different Learning Mechanisms. Our method is based on GCN, which allows learning over Ethereum's transaction graph. In contrast, many prior works adopt alternative learning mechanisms, such as classical machine learning (*e.g.*, decision trees, random forests, XGBoost) or deep learning on tabular data. For example, Kabla *et al.* [18] used ML classifiers over explicit transactional features and reported high F1-scores (0.98) on a relatively small dataset of 5,448 phishing instances. Similarly, Cheng *et al.* [11] explored hybrid models using LSTM and ML components. However, these approaches rely on limited features and datasets, with weaker generalization in large or imbalanced settings. Our GCN model, by comparison, achieved an F1-score of 0.95 and precision of

0.96 on over 600K phishing cases, demonstrating better performance and scalability in learning from structural context.

Different Feature Types. The majority of prior studies fall into two camps: those using explicit features (*e.g.*, transaction value, timestamp, gas) and those using implicit, graph-derived features (*e.g.*, connectivity, behavior patterns). Zhou *et al.* [49] and Kabla *et al.* [18] achieved strong performance with explicit features and ML models, but their datasets were significantly smaller and less diverse. Conversely, works like Li *et al.* [22] and Wu *et al.* [43] leveraged implicit features in graph-based or ML-based models. Our study builds upon the latter category but further refines implicit features using temporal patterns and behavioral distributions, such as transaction inter-arrival times and weekend activity ratios. This richer representation contributes to our higher recall (0.95) compared to others, such as Chen *et al.* [8] (recall: 0.14) and Li *et al.* [22] (recall: 0.92).

Same Feature Class, Different Instantiations. Among studies utilizing implicit features, our work distinguishes itself through the granularity and temporal depth of the extracted features. Prior studies such as Li *et al.* [22] relied on node- or subgraph-level embeddings using contrastive learning, while Fu *et al.* [12] explored address linkages without temporal decomposition. In contrast, our implicit features were handcrafted to reflect detailed transaction behavior over time. These include statistical metrics such as average, minimum, and maximum inter-transaction intervals; gas usage patterns across sending and receiving behaviors; and distributional metrics like the proportion of weekend activity. We also analyzed behavioral rhythms through time-of-day statistics (mean and standard deviation of transaction hours), providing temporal signatures that distinguish phishing nodes from benign ones. Importantly, we combined this with statistical validation and feature importance ranking using Random Forests to systematically identify and prioritize the most predictive attributes. These refined implicit features enabled our GCN to capture nuanced behavioral patterns that are often subtle or obscured in generic graph representations. As shown in Figure 5, the most influential features include total value sent, average gas used when sending, and time-based metrics like minimum time between transactions and standard deviation of send hours. These distinctions set our model apart from other graph-based systems using implicit features but less expressive or temporally coarse representations.

Dataset Scale and Generalization. Our dataset includes over 671K phishing and 2.6M benign transactions, making it one of the largest used in this domain. Many prior works use fewer than 10K phishing samples. Despite scale and class imbalance, our model maintains high precision, recall, and F1-score, showing strong generalization.

6 DISCUSSION

Performance of Explicit Features. In the initial experiment, where only explicit transactional features such as transaction value, gas usage, and timestamps were used, the model exhibited limited ability to distinguish phishing from benign nodes. While these features effectively identified benign transactions, they fell short of capturing the intricate behaviors typical of phishing scams, resulting in a high rate of false negatives. This observation suggests that while explicit features can identify clear-cut cases of normal transactions, they do not provide the nuanced understanding required to detect more deceptive phishing strategies. As such, this highlights the limitations of relying solely on explicit data for detecting phishing, as these features offer only a surface-level view of transaction patterns.

Impact of Implicit Features. When the GCN was tested with implicit features derived from the Ethereum transaction graph, a marked improvement in phishing detection was observed. Implicit features, which capture the relationships and transactional dynamics between nodes, enabled

the model to better identify the complex, coordinated activities common in phishing scams. This improvement in detection capability illustrates the critical role of network context in identifying fraud, which is often missed when analyzing transactions in isolation. These graph-based features allowed the model to consider transaction patterns. This underscores the value of focusing on the broader network structure rather than isolated transaction details. However, despite these improvements, the recall for phishing nodes remained at 0.33, indicating that a significant proportion of phishing activities went undetected due to the inherent complexity of phishing patterns, which our current model struggles to fully capture. Graph-based models like GCNs generally require more computational resources compared to traditional machine learning methods, but this cost is often justified by their improved ability to capture complex relational patterns.

Feature Comparison and Model Insights. The comparison between the two experiments demonstrates the distinct roles of explicit and implicit features in phishing detection. While explicit features are useful for understanding basic transactional properties, they lack the depth needed to capture sophisticated behaviors. In contrast, implicit features, which account for interactions and transaction sequences within the network, offer a more detection capability. The GCN's performance with implicit features highlights the importance of relational data for detecting phishing activities.

Addressing Class Imbalance. A key challenge throughout the experiments was the significant class imbalance, with phishing transactions being vastly outnumbered by benign ones. To mitigate this, a weighted loss function was employed, which improved recall for phishing nodes, especially in the experiment using implicit features. However, despite these improvements, the model still faced issues with false positives, suggesting that further refinement is needed to enhance precision while maintaining high recall.

Clarifying Performance Comparisons. We understand that comparing our results to prior studies can be misleading if the datasets or evaluation methods differ. In Section section 5, we only included these comparisons to give general context, not as a direct benchmark. Our model was trained and tested on a much larger dataset that we built independently, and we reported both per-class and weighted metrics to give a full picture of the model's performance. Still, we agree that detecting phishing is the most important part, and we have been transparent about the limitations of our recall. In the future, we plan to use shared datasets or re-run baseline models on our data to allow for more consistent and fair comparisons.

Limitations and Future Directions. Although the use of implicit features improved phishing detection, the model's precision for phishing nodes remains lower than desired, indicating room for optimization. Additionally, the limitations of explicit features highlight the need for approaches to feature engineering. To address the challenge of detecting a higher proportion of phishing, future efforts will focus on refining the model to improve recall, ensuring fewer malicious transactions are missed while maintaining precision. Additionally, future work could explore integrating advanced techniques such as attention mechanisms or temporal graph networks to further refine phishing detection capabilities. Addressing these challenges could enhance both precision and robustness, especially in real world with imbalanced data. Another important direction is adapting the model to work with streaming Ethereum transactions, allowing it to support real-time phishing detection as transactions occur. Exploring these new methodologies could provide the breakthrough needed to advance the state of phishing detection on the Ethereum blockchain. We also plan to explore adversarial training to better capture subtle phishing behaviors and improve recall.

7 CONCLUSION

This study investigated phishing detection on the Ethereum blockchain using GCNs with explicit and implicit features. While explicit features like value and gas usage provided a basic foundation,

they were insufficient for detecting the complex behaviors of phishing attacks. Implicit, graph-based features significantly improved detection by capturing relationships between addresses and broader network patterns. Addressing class imbalance with a weighted loss function enhanced the model's recall for phishing nodes, but challenges with precision remain, indicating a need for further refinement. Our results highlight the importance of using implicit features for more robust phishing detection. Future work should focus on improving precision and exploring advanced techniques like attention mechanisms and temporal graph networks. In summary, implicit features are essential for detecting phishing activities, and addressing class imbalance will be key to developing more effective detection on blockchain networks.

REFERENCES

- [1] Ayodeji Adeniran, Kieran Human, and David Mohaisen. 2024. Dissecting the Infrastructure Used in Web-based Cryptojacking: A Measurement Perspective. *CoRR* abs/2408.03426 (2024). <https://doi.org/10.48550/ARXIV.2408.03426>
- [2] Ashar Ahmad, Muhammad Saad, Mostafa A. Bassiouni, and Aziz Mohaisen. 2018. Towards Blockchain-Driven, Secure and Transparent Audit Logs. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MobiQuitous 2018, 5-7 November 2018, New York City, NY, USA*. ACM, 443–448. <https://doi.org/10.1145/3286978.3286985>
- [3] Ahod Alghuried and David Mohaisen. 2024. Simple Perturbations Subvert Ethereum Phishing Transactions Detection: An Empirical Analysis. In *The 25th International Conference Information Security Applications, WISA*. 1–12.
- [4] Massimo Bartoletti, Stefano Lande, Andrea Loddo, Livio Pompianu, and Sergio Serusi. 2021. Cryptocurrency Scams: Analysis and Perspectives. *IEEE Access* 9 (2021), 148353–148373. <https://doi.org/10.1109/ACCESS.2021.3123894>
- [5] Shaista Bibi. 2019. Cryptocurrency world identification and public concerns detection via social media: student research abstract. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC*. ACM, 550–552.
- [6] Federico Cernera, Massimo La Morgia, Alessandro Mei, and Francesco Sassi. 2023. Token Spammers, Rug Pulls, and Sniper Bots: An Analysis of the Ecosystem of Tokens in Ethereum and in the Binance Smart Chain (BNB). In *32nd USENIX Security Symposium*. USENIX Association, 3349–3366.
- [7] Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu. 2021. A Survey on Ethereum Systems Security: Vulnerabilities, Attacks, and Defenses. *ACM Comput. Surv.* 53, 3 (2021), 67:1–67:43.
- [8] Liang Chen, Jiaying Peng, Yang Liu, Jintang Li, Fenfang Xie, and Zibin Zheng. 2021. Phishing Scams Detection in Ethereum Transaction Network. *ACM Trans. Internet Techn.* 21, 1 (2021), 10:1–10:16.
- [9] Weili Chen, Xiongfeng Guo, Zhiguang Chen, Zibin Zheng, and Yutong Lu. 2020. Phishing Scam Detection on Ethereum: Towards Financial Security for Blockchain Ecosystem. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*. ijcai.org, 4506–4512.
- [10] Ling Cheng, Feida Zhu, Yong Wang, Ruicheng Liang, and Huiwen Liu. 2023. Evolve Path Tracer: Early Detection of Malicious Addresses in Cryptocurrency. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD*. ACM, 3889–3900.
- [11] Ling Cheng, Feida Zhu, Yong Wang, Ruicheng Liang, and Huiwen Liu. 2024. From Asset Flow to Status, Action, and Intention Discovery: Early Malice Detection in Cryptocurrency. *ACM Trans. Knowl. Discov. Data* 18, 3 (2024), 50:1–50:27.
- [12] Bingxue Fu, Xing Yu, and Tao Feng. 2022. CT-GCN: a phishing identification model for blockchain cryptocurrency transactions. *Int. J. Inf. Sec.* 21, 6 (2022), 1223–1232.
- [13] Letterio Galletta and Fabio Pinelli. 2024. Explainable Ponzi Schemes Detection on Ethereum. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC*, Jiman Hong and Juw Won Park (Eds.). ACM, 1014–1023.
- [14] Bowen He, Yuan Chen, Zhuo Chen, Xiaohui Hu, Yufeng Hu, Lei Wu, Rui Chang, Haoyu Wang, and Yajin Zhou. 2023. TxPhishScope: Towards Detecting and Understanding Transaction-based Phishing on Ethereum. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS*. ACM, 120–134.
- [15] Hwanjo Heo, Seungwon Woo, Taeung Yoon, Min Suk Kang, and Seungwon Shin. 2023. Partitioning Ethereum without Eclipsing It. In *30th Annual Network and Distributed System Security Symposium, NDSS*. The Internet Society.
- [16] Lars Hornuf, Paul P Momtaz, Rachel J Nam, and Ye Yuan. 2023. Cybercrime on the ethereum blockchain. (2023).
- [17] Keith S. Jones, Miriam E. Armstrong, McKenna K. Tornblad, and Akbar Siami Namin. 2021. How social engineers use persuasion principles during vishing attacks. *Inf. Comput. Secur.* 29, 2 (2021), 314–331. <https://doi.org/10.1108/ICS-07-2020-0113>
- [18] Arkan Hammoody Hasan Kabla, Mohammed Anbar, Selvakumar Manickam, and Shankar Karuppayah. 2022. Eth-PSD: A Machine Learning-Based Phishing Scam Detection Approach in Ethereum. *IEEE Access* 10 (2022), 118043–118057.

- [19] Olaf Kampers, Abdulhakim Ali Qahtan, Swati Mathur, and Yannis Velegarakis. 2022. Manipulation detection in cryptocurrency markets: an anomaly and change detection based approach. In *SAC '22: The 37th ACM/SIGAPP Symposium on Applied Computing*. ACM, 326–329.
- [20] Author Name Kimber. 2023. Scams on Ethereum. (2023). <https://github.com/YNclusk/scamsonethereum>
- [21] Sijia Li, Gaopeng Gou, Chang Liu, Chengshang Hou, Zhenzhen Li, and Gang Xiong. 2022. TTAGN: Temporal Transaction Aggregation Graph Network for Ethereum Phishing Scams Detection. In *WWW '22: The ACM Web Conference 2022*. ACM, 661–669.
- [22] Sijia Li, Gaopeng Gou, Chang Liu, Gang Xiong, Zhen Li, Junchao Xiao, and Xinyu Xing. 2023. TGC: Transaction Graph Contrast Network for Ethereum Phishing Scam Detection. In *Annual Computer Security Applications Conference, ACSAC*. ACM, 352–365.
- [23] Shucheng Li, Runchuan Wang, Hao Wu, Sheng Zhong, and Fengyuan Xu. 2023. SIEGE: Self-Supervised Incremental Deep Graph Learning for Ethereum Phishing Scam Detection. In *Proceedings of the 31st ACM International Conference on Multimedia, MM*. ACM, 8881–8890.
- [24] Zhutian Lin, Xi Xiao, Guangwu Hu, Bin Zhang, Qixu Liu, and Xiapu Luo. 2023. Phish2vec: A Temporal and Heterogeneous Network Embedding Approach for Detecting Phishing Scams on Ethereum. In *20th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON*. IEEE.
- [25] Jieli Liu, Jinze Chen, Jiajing Wu, Zhiying Wu, Junyuan Fang, and Zibin Zheng. 2024. Fishing for Fraudsters: Uncovering Ethereum Phishing Gangs With Blockchain Data. *IEEE Trans. Inf. Forensics Secur.* 19 (2024), 3038–3050.
- [26] NetworkX. 2024. NetworkX. <https://networkx.org/>. (2024). Accessed: 2024-11.
- [27] Cuong Phuc Ngo, Amadeus Aristo Winarto, Connie Khor Li Kou, Sojeong Park, Farhan Akram, and Hwee Kuan Lee. 2019. Fence GAN: Towards Better Anomaly Detection. In *31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI*. IEEE, 141–148.
- [28] Pawel Pinio, Roman Batko, and Dagmara Lewicka. 2024. Between Theory and Value Transactions: A Multifaceted Exploration of Relevance and Resilience of Decentralised Autonomous Organisations. In *Proceedings of the 2024 7th International Conference on Software Engineering and Information Management, ICSIM*. ACM, 42–48.
- [29] Muhammad Saad, Ashar Ahmad, and Aziz Mohaisen. 2019. Fighting Fake News Propagation with Blockchains. In *7th IEEE Conference on Communications and Network Security, CNS 2019, Washington, DC, USA, June 10-12, 2019*. IEEE, 1–4. <https://doi.org/10.1109/CNS.2019.8802670>
- [30] Muhammad Saad, Afsah Anwar, Ashar Ahmad, Hisham Alasmary, Murat Yuksel, and David Mohaisen. 2022. *RouteChain*: Towards Blockchain-based secure and efficient BGP routing. *Comput. Networks* 217 (2022), 109362. <https://doi.org/10.1016/J.COMNET.2022.109362>
- [31] Muhammad Saad, Songqing Chen, and David Mohaisen. 2021. SyncAttack: Double-spending in Bitcoin Without Mining Power. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*. ACM, 1668–1685. <https://doi.org/10.1145/3460120.3484568>
- [32] Muhammad Saad, Jinchun Choi, DaeHun Nyang, Joongheon Kim, and Aziz Mohaisen. 2020. Toward Characterizing Blockchain-Based Cryptocurrencies for Highly Accurate Predictions. *IEEE Syst. J.* 14, 1 (2020), 321–332. <https://doi.org/10.1109/JSYST.2019.2927707>
- [33] Muhammad Saad, Joongheon Kim, DaeHun Nyang, and David Mohaisen. 2021. Contra-_s: Mechanisms for countering spam attacks on blockchain's memory pools. *J. Netw. Comput. Appl.* 179 (2021), 102971. <https://doi.org/10.1016/J.JNCA.2020.102971>
- [34] Muhammad Saad and David Mohaisen. 2023. Three Birds with One Stone: Efficient Partitioning Attacks on Interdependent Cryptocurrency Networks. In *44th IEEE Symposium on Security and Privacy, SP*. IEEE, 111–125.
- [35] Muhammad Saad, Laurent Njilla, Charles A. Kamhoua, Joongheon Kim, DaeHun Nyang, and Aziz Mohaisen. 2019. Mempool optimization for Defending Against DDoS Attacks in PoW-based Blockchain Systems. In *IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2019, Seoul, Korea (South), May 14-17, 2019*. IEEE, 285–292. <https://doi.org/10.1109/BLOC.2019.8751476>
- [36] Muhammad Saad, Laurent Njilla, Charles A. Kamhoua, and Aziz Mohaisen. 2019. Countering Selfish Mining in Blockchains. In *International Conference on Computing, Networking and Communications, ICNC 2019, Honolulu, HI, USA, February 18-21, 2019*. IEEE, 360–364. <https://doi.org/10.1109/ICNC.2019.8685577>
- [37] Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles A. Kamhoua, Sachin Shetty, DaeHun Nyang, and David Mohaisen. 2020. Exploring the Attack Surface of Blockchain: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* 22, 3 (2020), 1977–2008.
- [38] Marie Vasek and Tyler Moore. 2015. There's No Free Lunch, Even Using Bitcoin: Tracking the Popularity and Profits of Virtual Currency Scams. In *Financial Cryptography and Data Security - 19th International Conference, FC (Lecture Notes in Computer Science)*, Vol. 8975. Springer, 44–61.
- [39] Yun Wan, Feng Xiao, and Dapeng Zhang. 2023. Early-stage phishing detection on the Ethereum transaction network. *Soft Comput.* 27, 7 (2023), 3707–3719. <https://doi.org/10.1007/S00500-022-07661-0>

- [40] Jinhuan Wang, Pengtao Chen, Xinyao Xu, Jiajing Wu, Meng Shen, Qi Xuan, and Xiaoniu Yang. 2022. TSGN: Transaction Subgraph Networks Assisting Phishing Detection in Ethereum. *CoRR* abs/2208.12938 (2022).
- [41] Kai Wang, Michael Tong, Jun Pang, Jitao Wang, and Weili Han. 2024. XRAD: Ransomware Address Detection Method based on Bitcoin Transaction Relationships. *ACM Trans. Web* (2024).
- [42] Tingke Wen, Yuanxing Xiao, Anqi Wang, and Haizhou Wang. 2023. A novel hybrid feature fusion model for detecting phishing scam on Ethereum using deep neural network. *Expert Syst. Appl.* 211 (2023), 118463.
- [43] Jiajing Wu, Qi Yuan, Dan Lin, Wei You, Weili Chen, Chuan Chen, and Zibin Zheng. 2022. Who Are the Phishers? Phishing Scam Detection on Ethereum via Network Embedding. *IEEE Trans. Syst. Man Cybern. Syst.* 52, 2 (2022), 1156–1166.
- [44] Yijun Xia, Jieli Liu, and Jiajing Wu. 2022. Phishing Detection on Ethereum via Attributed Ego-Graph Embedding. *IEEE Trans. Circuits Syst. II Express Briefs* 69, 5 (2022), 2538–2542.
- [45] Yufeng Xu, Lun Zhang, Turan Vural, Peng Qian, Yanbin Wang, Yuqing Fan, Ming Li, Xueyan Tang, and Zheng Cao. 2023. STFNet: Spatio-Temporal Fusion Network to Detect Ethereum Phishing Scams. In *Proceedings of the 2023 7th International Conference on Electronic Information Technology and Computer Engineering, EITCE*. ACM, 599–605.
- [46] Mingxuan Yao, Runze Zhang, Haichuan Xu, Shih-Huan Chou, Varun Chowdhary Paturi, Amit Kumar Sikder, and Brendan Saltaformaggio. 2024. Pulling Off The Mask: Forensic Analysis of the Deceptive Creator Wallets Behind Smart Contract Fraud. In *IEEE Symposium on Security and Privacy, SP*. IEEE, 2236–2254.
- [47] Abbas Yazdinejad, Ali Dehghantanha, Reza M. Parizi, Mohammad Hammoudeh, Hadis Karimipour, and Gautam Srivastava. 2022. Block Hunter: Federated Learning for Cyber Threat Hunting in Blockchain-Based IIoT Networks. *IEEE Trans. Ind. Informatics* 18, 11 (2022), 8356–8366.
- [48] Qi Yuan, Baoying Huang, Jie Zhang, Jiajing Wu, Haonan Zhang, and Xi Zhang. 2020. Detecting Phishing Scams on Ethereum Based on Transaction Records. In *IEEE International Symposium on Circuits and Systems, ISCAS*. IEEE, 1–5.
- [49] Xuanchen Zhou, Wenzhong Yang, and Xiaodan Tian. 2023. Detecting Phishing Accounts on Ethereum Based on Transaction Records and EGAT. *Electronics* 12, 4 (2023).