CATALOG: CATALOG: Exploiting Joint Temporal Dependencies for Enhanced Phishing Detection on Ethereum

Abstract

Phishing attacks on Ethereum have increased with its growing adoption, creating significant challenges as phishing and non-phishing users often display similar behavior. Additionally, while the network as a whole experiences high activity, individual user behavior is typically sparse, making it difficult to detect phishing patterns. Current methods frequently fail to tackle these challenges and often neglect the temporal sequence of transactions, resulting in data leakage and reduced performance. In this paper, we propose a novel approach that addresses these gaps by focusing on the association of two key aspects: (1) **local temporal behavior fluctuations** of individual users and (2) **deviations from global transaction patterns** within the network. To aim this, we introduce **CATALOG (CApturing joint TemporAl dependencies from LOcal and Global user behaviour)**, a novel representation learning model that jointly captures the local and global behavioral patterns of a user and their correlations by leveraging a dual cross-attention mechanism paired with a bi-directional Masked Language Modelling (MLM) based pipelined transformer framework. Our proposed model simultaneously learns from local behavioral shifts and global market trends along with a contextually enriched embeddings, effectively distinguishing phishing from non-phishing users, while addressing the existing research gaps. Extensive experiments on real-world Ethereum transaction data show that our framework improves phishing detection by 7-8% in F1-Score compared to existing models. Furthermore, it generalizes effectively across Ethereum versions 1.0 and 2.0, demonstrating the robustness of our approach.

CCS Concepts

• **Applied computing** → **Digital cash**; • **Security and privacy** → **Phishing**.

Keywords

Ethereum Transaction Network, Phishing Scams, Representation Learning, Blockchain Security

1 Introduction

The growing adoption of Ethereum [1], fueled by its wide range of applications [2, 14, 17, 25], has attracted a large influx of users seeking to invest in cryptocurrency. Unfortunately, many of these users are unfamiliar with the complexities of the crypto market. As a consequence, they become vulnerable to phishing attacks that compromise their wallets and perform fraudulent activities [3]. The persistent nature of these phishing scams underscores the urgent need for effective detection and prevention mechanisms on Ethereum. However, identifying malicious actors presents significant challenges of *"indistinguishable user behavior"* and *"large scale network and sparse user instances"* resulting from the platform's unique characteristics.

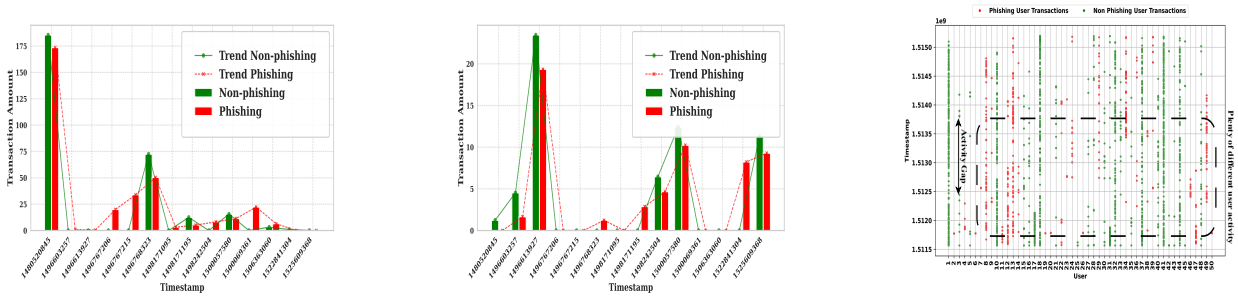
As phishing users often attempt to blend in with non-phishing users by showcasing a normal transaction behaviour over an extended period, thereby building trust within the network [8, 30]. Once the trust is established, they execute malicious activities within a short time interval, after which they either become inactive or revert back to normal behavior. Additionally, user activities on Ethereum are frequently influenced by external factors such as cryptocurrency price fluctuations, market trends, and social media dynamics [37]. For instance, sudden price changes can prompt non-phishing users to alter their transaction frequency or volume, which phishing users may mirror as well. To demonstrate this in a practical scenario, we analyze the temporal variation of a particular real-time average transaction amount of 50 phishing users and 50 non-phishing users of the Ethereum mainnet [15] (from the same time interval), as shown in Figures 1(a) and 1(b). We found that the phishing user exhibits behavioral patterns that are *indistinguishable* to those of non-phishing users. Consequently, this substantial similarity makes it difficult to identify such phishing users. On the other hand, with the rapid growth of user counts on the Ethereum network, reaching 283.63 million unique accounts as of September 23, 2024[1], Ethereum has evolved into a *large-scale* and complex transaction network with substantial temporal variations [11]. Analyzing user behavior in such a large-scale, dynamic network presents significant challenges, especially given the sheer volume of transactions and active users. Although the overall network remains highly active, the activity of individual users tends to be relatively *sparse*. We illustrate this phenomenon by examining a sample of 50 users' behavior over 100 timestamps of real-time Ethereum data as depicted in Figure 1(c) and observed that even within this relatively small dataset, the consecutive user activities demonstrate a visible sparsity. The observations reveal that identifying crucial patterns or abnormalities within this type of data may be more challenging than expected. As a consequence of this highly sparse and dynamic nature of the data, analyzing user behavior over long time intervals often leads to excessive information from different users, making it more difficult to identify the critical patterns of phishing users. Conversely, analyzing shorter time windows may fail to capture temporal behaviors of individual users effectively. These combined factors make detecting phishing behavior particularly difficult in such a dynamic environment.

Despite the extensive research conducted on detecting phishing users in Ethereum, none of the existing approaches singlehandedly address all the challenges mentioned above. To be specific, several

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

[1] https://ycharts.com/indicators/ethereum_cumulative_unique_addresses



(a) As senders. **(b) As receivers.** **(c) Sparse instances of users.**

Figure 1: Temporal variation of average transaction amount of the phishing users and non-phishing users as sender (Figure 1(a)) and receiver (Figure 1(b)) from the same time duration (Timestamp duration for sender plot: 1480532847 to 1507064393; Timestamp duration for receiver Plot: 1480520845 to 1525609368). Figure 1(c) depicts user active instances with respect to time, the activity gaps and user density, for better readability Ethereum hash addresses are represented through numerical digits and the Unix timestamps also are scaled to float. Red dots are phishing users and green dots indicate non-phishing users.

current approaches leverage random subgraph sampling combined with either Graph Neural Networks (GNNs) [4, 18, 19, 29, 31] or feature engineering [8, 24, 36]. While these approaches handle large-scale data with temporal dynamics, their focus on modeling Ethereum transactions as a global network makes it difficult to extract meaningful user-specific embeddings due to the relative sparsity of individual activities. On the other hand, to mitigate sparsity, another line of investigation [13, 29, 32] introduced the notion of user-specific local temporal-transaction networks, focusing more on individual user activities. However, it has been observed that all the above approaches ignore temporal sequence of user activities and transactions in the network, resulting in the inadvertent inclusion of future information during the training process. This oversight leads to **data leakage** [28] concerns, yielding an overly optimistic model performance. Moreover, they often struggle to capture useful user embeddings due to the identical behavioural patterns of both phishing and non-phishing users.

To comprehensively solve these challenges, it is essential to impart contextual insights in the embeddings obtained from the temporal dependencies of users' behaviors, which can effectively differentiate between phishing and non-phishing patterns. Hence, in our work, we have jointly analyzed these two crucial factors: (1) correlation between user's **local behaviour** (*temporal fluctuations in individual user behavior)*, and, **global behavior** *(the deviation of a user's transaction patterns from the average behavior of users across the network during the same period)*; and (2) the correlation between consecutive temporal shifts in a user specific transaction behavior. The main idea is to capture the commonalities and divergences of local and global temporal variations, which are crucial for distinguishing between phishing and non-phishing users, especially when their overall behavior appears to be similar.

To this end, we introduce a novel representation learning framework **CATALOG**, that leverages a dual cross-attention mechanism coupled with a context-sensitive transformer pipeline in order to capture the above aspects. We aim to exploit the latent characteristics of the obtained correlations to identify a phishing user more effectively while mitigating the existing research gaps. More specifically, the Ethereum transaction network is modeled as a series of temporally ordered user-specific ego networks, helping to mitigate sparsity by focusing attention on each user individually. Additionally, the temporal ordering also prevents data leakage. Furthermore, to capture the dynamic variations and correlations between local and global behaviors, a dual cross-attention mechanism is combined with an augmented transformer pipeline. This coupling enhances contextual sensitivity by engraving both recurring transaction patterns and subtle but important behavioral shifts in the representations. As a result, the **CATALOG** provides a standalone solution to Ethereum phishing user detection that single handedly deals with the key challenges and yields enhanced performance. Thus, our major contributions are summarized as follows:

(1) We propose a novel representation learning model, **CATALOG**, which employs a bi-directional masked language model Transformer to derive contextually enriched embeddings.
(2) The model integrates dual cross-attention layers to capture both local and global temporal user behaviors. We introduce one-hot encodings of temporal activity vectors, which serve as positional encodings for phishing classification. To the best of our knowledge, this is the first approach to jointly model local and global behaviors in this way. The code and our dataset will be available at our own anonymous repository[2].
(3) We contribute a new publicly available Ethereum transaction dataset containing 60713 Ethereum 2.0 transactions. Our curated dataset will be useful for the research community to improve the adaptability of fraud detection models within the evolving Ethereum ecosystem. Moreover, robustness experiments confirm that **CATALOG** maintains strong performance across different Ethereum versions.

[2]https://anonymous.4open.science/r/CATALOG-model-and-dataset-D2E6/

(4) Our framework demonstrates a 7-8% improvement in F1-Score over state-of-the-art methods, validated on four real-world Ethereum datasets. Extensive experiments, including ablation studies and sensitivity analysis, underline the model's efficacy in addressing existing research gaps.

The remainder of this paper is structured as follows: Section 2 provides an overview of the related work, followed by a description of the proposed methodology in Section 3. Section 4 details the experimental setup, with the results and analysis presented in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

A plethora of research has focused on detecting phishing scams on the Ethereum blockchain, as observed in [8, 13, 18, 19, 24, 28–31, 36]. These efforts generally fall into two main categories: (1) feature engineering-based approaches [5, 24, 28, 36], which majorly rely on manually crafted features to analyze user attributes, and (2) deep representation learning methods, such as Graph Neural Networks (GNNs) [4, 13, 18, 31], which automate feature extraction and can more effectively handle complex transaction network structures.

While feature engineering approaches tend to overlook crucial structural and temporal dynamics of transaction networks, often resulting in suboptimal performance, representation learning techniques have been shown to address these complexities more effectively [4, 13, 19, 30]. For instance, Wang et al. [29] proposed using ego graphs to process large datasets with GNNs. However, these methods typically rely on static network structures, which can increase preprocessing costs and reduce their ability to adapt to dynamic changes. Similarly, Xia et al. [32] modeled ego graphs with transaction attributes such as timestamp and amount, and introduced relabeling techniques to address class imbalance. Although, they dealt with the sparsity problem, their employed relabeling may be less effective in cases where phishing and non-phishing users exhibit identical behavior, leading to suboptimal results.

Some studies, such as [19, 20], addressed these challenges by utilizing a self-supervised contrastive learning framework. However, inherent issues like network sparsity and fluctuating user behavior continue to restrain model performance. Notably, recent research has made notable advancements by incorporating temporal factors and sequencing into their models, often utilizing a fusion of domain-specific feature engineering and GNN model [6, 22, 33, 36] or tensor-based representation learning [9]. Nevertheless, these methods' reliance on domain-specific features can make them vulnerable to feature re-engineering attacks and less suitable for highly fluctuating network behavior. While Huang et al. [13] introduced a temporal ego-graph approach to address some of these concerns, the challenge of similar user behavior still remains as a significant obstacles. These aforementioned research gaps have motivated us to design a comprehensive solution aimed at improving the detection of phishing users on Ethereum.

3 Methodology

3.1 Problem Definition

We address the problem of phishing user detection in Ethereum by framing it as a node classification task. For a given user $v_a \in V$, with a sequence of transaction activities over a time span $\mathcal{T}$, we represent this as a set of time series weighted directed k-hop ego networks, denoted as: $\mathcal{G}_{v_a} = \{G_{v_a}^{t_i} \mid t_i \in \mathcal{T}\}$. The network $G_{v_a}^{t_i} = (V_{v_a}^{t_i}, E_{v_a}^{t_i}, W_{v_a}^{t_i})$ is defined as follows: $V_{v_a}^{t_i} = \{v_a\} \cup \{v' \mid v'$ transacts within k hops of v_a at time $t_i\}$ is the set of nodes. $E_{v_a}^{t_i} = \{(v', v'') \mid v', v'' \in V_{v_a}^{t_i}, v'$ transacts with $v''\}$ is the set of directed edges within k hops of v_a, $W_{v_a}^{t_i} = \{w_e^{t_i} \mid e \in E_{v_a}^{t_i}\}$ is the set of edge weights denoting the transaction amounts between the users associated with all edges at time t_i. The aim is to learn an objective function $f(\mathcal{G}) : V \rightarrow \mathbb{R}^d$ that maps users to d-dimensional node embeddings based on the set of k hop ego networks $\mathcal{G}$ corresponding to all users, forming the matrix $\mathbf{F} \in \mathbb{R}^{|V| \times d}$. These embeddings are used to classify users as phishing (1) or non-phishing (0).

3.2 Proposed Architecture: CATALOG

Our proposed architecture, **CATALOG**, comprises four key components: *local temporal correlation, contextual enrichment, global temporal correlation, and embedding fusion with classification*. The input Ethereum transaction data is modeled as time series of k-hop weighted directed ego networks, processed using a sliding window with GNN layers to generate user embeddings. These embeddings are then passed through a cross-attention layer to capture *local temporal correlations* between consecutive user embeddings. The attention-based time-series vectors are further processed by a Transformer to provide *contextual enrichment*, capturing users' overall behavioral patterns. Simultaneously, a parallel layer captures *global temporal correlations* by analyzing interactions across users in the network. Finally, the fused features are used for phishing user detection. Figure 2 illustrates the flow of the proposed model, with details of each component described as follows.

3.2.1 Local Correlation Analysis. Given an Ethereum transaction network of n users over m timestamps, we construct k-hop weighted directed ego networks for each user to capture local temporal correlations and dependencies. Since users have sparse local networks, we use a sliding window of size w to focus on relevant transactions with neighbors and detect subtle behavioral shifts across consecutive timestamps. The sliding window improves computational efficiency by processing smaller time windows, which is crucial for large-scale networks like Ethereum, with its high and continuous transaction volumes. It also helps mitigate noise and sparsity by focusing on meaningful data chunks. After preliminary analysis, we selected w based on the sensitivity experiments detailed in Section 5.4. For each user v_a, the sliding window processes w of consecutive timestamps while generating embeddings via a GNN layer:

$$\mathbf{h}_{v_a}^{w_i} = \sigma\left(\sum_{v_b \in \mathcal{N}(v_a)} \mathbf{W}\mathbf{h}_{v_b}^{w_{i-1}} + \mathbf{W}_{v_a}\mathbf{h}_{v_a}^{w_{i-1}} + \mathbf{b}\right) \quad (1)$$

where $\mathbf{h}_{v_a}^{t_{w_i}}$ and $\mathbf{h}_{v_a}^{t_{w_{i-1}}}$ are the hidden representations of v_a at window number w_i and w_{i-1}, respectively and $\mathcal{N}(v_a)$ represents the neighbors of v_a in Equation 1. Thus, this process yields a time series of embeddings: $\{\mathbf{h}_{v_a}^{w_1}, \ldots, \mathbf{h}_{v_a}^{w_m}\}$ for each user with dimension $\mathbb{R}^{16}$. Cross Attention layer [26] is then employed to analyze the relationship between the current and subsequent embeddings of users, enabling the capture of subtle yet significant temporal shifts in a user's local behavior across consecutive time windows.

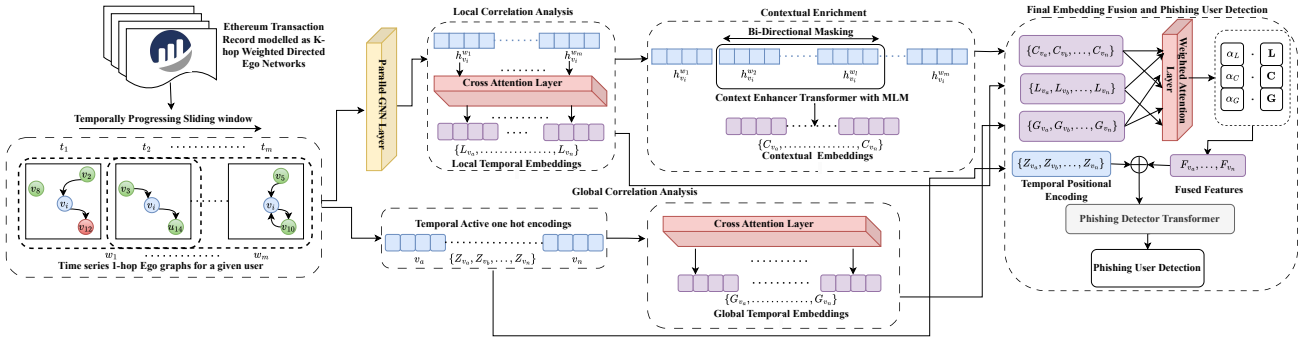


Figure 2: Overall architecture of proposed CATALOG.

In this context, the embedding $\mathbf{h}_{v_a}^{w_i}$ is designated as the query, while the subsequent embedding $\mathbf{h}_{v_a}^{w_{i+1}}$ serves as both the key and value for $i = \{1, \ldots, m\}$. The fundamental idea behind this approach is that cross-attention empowers our model to discern how a user's transactional behavior during the window w_i correlates with their subsequent transactions in window w_{i+1}. This establishes temporal relationships between each user's transactions, thereby facilitating a deeper understanding of how their behavioral patterns evolve over time. The attention scores are then computed using Equation 2, where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are the query, key, and value matrices, and d_k is the dimensionality of the key.

$$\text{Att}(\mathbf{h}_{v_a}^{w_i}, \mathbf{h}_{v_a}^{w_{i+1}}, \mathbf{h}_{v_a}^{w_{i+1}}) = \text{softmax}\left(\frac{(\mathbf{h}_{v_a}^{w_i})^\top \mathbf{W}_Q (\mathbf{h}_{v_a}^{w_{i+1}}) \mathbf{W}_K^\top}{\sqrt{d_k}}\right) \mathbf{h}_{v_a}^{w_{i+1}} \mathbf{W}_V \tag{2}$$

The above obtained Attention score gives us the local correlation embeddings for a given user v_a denoted as, L_{v_a}. This results in the attention-based feature vectors for all users denoted as, $\mathbf{L} \in \mathbb{R}^{n \times \frac{m}{w}}$ (considering n is the number of users, m is the total number of unique timestamps and w is the window size), representing enriched temporal embeddings of their transactional behavior.

3.2.2 Contextual Enrichment. Once local user embeddings (L_{v_a}) are obtained, we aim to enhance them by capturing overall temporal dependencies. While the cross-attention layer effectively extracts correlations between consecutive time windows (e.g., w_i and w_{i+1}), it does not account for long-range bi-directional dependencies (e.g., between w_i and w_{i+x} for $x = \{1, 2, \ldots, m\}$). Since phishing behavior often resembles non-phishing activity before or after fraudulent events, capturing the full behavioral context is crucial for distinguishing between these behaviors.

To enrich the embeddings with the temporal dependencies, we employ a Transformer model with Bi-directional Masked Language Modeling (MLM) [7]. The bi-directional masking approach predicts embeddings by analyzing both past and future contexts, capturing two-way temporal coherency. While phishing and non-phishing users may exhibit similar overall behavior, their subtle behavioral fluctuations should follow a coherent pattern and a two way long-term dependency. By masking the a certain embedding to predict its subsequent, and vice versa, the model improves its perceptions of the overall temporal relationships. This method enhances user behaviour consistency and context thereby enabling the model identify phishing and non-phishing patterns more accurately. For a user v_a, the embeddings sequence is $\mathbf{H}_{v_a} = \{\mathbf{h}_{v_a}^{w_1}, \ldots, \mathbf{h}_{v_a}^{w_m}\}$, and $\mathbf{P}_{v_a}$ are the positional encodings defined in Equation 3.

$$\mathbf{P}_{v_a}(w_i, 2j) = \sin\left(\frac{w_i}{10000^{\frac{2j}{d}}}\right), \quad \mathbf{P}_{v_a}(w_i, 2j+1) = \cos\left(\frac{w_i}{10000^{\frac{2j}{d}}}\right) \tag{3}$$

where d is the embedding dimension and j is the index. We apply the transformer's multi-head attention mechanism considering the Query, Key and Values calculation using Equation 4.

$$Q = (\mathbf{H}_{v_a} + \mathbf{P}_{v_a})\mathbf{W}_Q, \quad K = (\mathbf{H}_{v_a} + \mathbf{P}_{v_a})\mathbf{W}_K, \quad V = (\mathbf{H}_{v_a} + \mathbf{P}_{v_a})\mathbf{W}_V \tag{4}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are learned projection matrices, and d_k is the dimensionality of the key. The final concatenations of the attention heads are formalized as Equation 5,

$$\mathbf{H'}_{v_a} = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_O \tag{5}$$

with each head computing $\text{head}_i = \text{Attention}(Q_i, K_i, V_i)$, followed by output projection via $\mathbf{W}_O$. The transformer aggregates the obtained embeddings into a combined one $\mathbf{C}_{v_a}$ (for node v_a) thereby capturing the holistic temporal contextual patterns. Thus, for each node we follow this process to generate the contextually enriched embeddings.

3.2.3 Global Correlation Analysis. Simultaneously, with the extraction of local correlations, we examine the global correlation of a user's temporal behavior, which is essential for assessing an individual's activities in comparison with the other active users during the same period. This analysis is vital to understand behavioral changes concerning the market trends; for instance, during a Ether price hike, an increase in activity across all users is expected. Capturing these crucial correlations is imperative for effective distinction of phishing and non-phishing patterns thereby highlighting the necessity of exploring a global perspective. To comprehend this global correlation, we initially derive a temporal activity vector that encapsulates the overall temporal activity of a user within the network. To depict each user's temporal activity over a given period, we use one-hot encoding. Let m be the total unique timestamps, and $\mathcal{T}_{v_a} = \{t_1, t_2, \ldots, t_l\}$ represent the timestamps where user v_a is active, with $l \leq m$. We define a vector $\mathbf{z}_{v_a} \in \{0, 1\}^m$ in Equation 6

as:

$$\mathbf{z}_{v_a}(t_i) = \begin{cases} 1, & \text{if } v_a \text{ is active at } t_i, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The resulting vector $\mathbf{z}_{v_a}$ has '1's for timestamps where v_a is active, providing a temporal activity representation. Following the above, we compute the global temporal correlation between users using a cross-attention mechanism using one-hot encoded temporal vectors $\mathbf{Z} \in \mathbb{R}^{n \times m}$ (with m as the number of unique timestamps). In this context, the cross-attention layer is designed to capture deviations in an individual user's behavior relative to the overall behavior of other users in the network. Specifically, this layer helps to analyze how a particular user deviate from the general trend, which may be indicative of suspicious or unusual activity. For user v_a, the query is $\mathbf{z}_{v_a}$, and the key and value are the one-hot vectors of all other users in the same time period, $\mathbf{z}_{v_b}, \ldots, \mathbf{z}_{v_n}$. The corresponding global attention is computed in Equation 7 as:

$$\text{Att}(\mathbf{z}_{v_a}, \mathbf{z}_{v_b}, \mathbf{z}_{v_b}) = \text{softmax}\left(\frac{(\mathbf{z}_{v_a})^\top \mathbf{W}_Q (\mathbf{z}_{v_b}) \mathbf{W}_K^\top}{\sqrt{d_k}}\right) \mathbf{z}_{v_b} \mathbf{W}_V \tag{7}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are projection matrices, and d_k is the dimensionality of the key. This results in an attention score vector for each user, that captures global temporal patterns across all users with respect to the average behaviour of other users in the network. The global correlation vector is denoted as, $\mathbf{G} \in \mathbb{R}^{n \times (n-1)}$.

3.2.4 Embedding Fusion and Classification. In order to capture the inter-dependencies between the three main aspects in terms of the user's local temporal embeddings with their contextual relations along with the global behavior to classify phishing and non-phishing users, we apply a weighted attention layer. The components $\mathbf{L}_{v_a}$ (local embeddings), $\mathbf{C}_{v_a}$ (contextual transformer embeddings) and $\mathbf{G}_{v_a}$ (global temporal attention) are fused together using weighted attention, yielding the final embedding $\mathbf{F}_{v_a}$. The role of the weighted attention layer is to balance and prioritize the influence of each component based on their relevance to the task of detecting phishing users. This ensures that the final embedding incorporates comprehensive information from both short-term and long-term behavioral patterns along with their correlations thereby improving the model's ability to make accurate predictions.

We compute attention scores α_L, α_C, and α_G for each embedding component using a softmax over their respective query projections. The associated projection vectors are learned during the training process, with the weight matrices w_L, w_C, and w_G corresponding to the local, contextual, and global embeddings, respectively and formalized as Equations 8, 9, and 10.

$$\alpha_L = \frac{\exp(w_L^\top \mathbf{L}_{v_a})}{\exp(w_L^\top \mathbf{L}_{v_a}) + \exp(w_C^\top \mathbf{C}_{v_a}) + \exp(w_G^\top \mathbf{G}_{v_a})} \tag{8}$$

$$\alpha_C = \frac{\exp(w_C^\top \mathbf{C}_{v_a})}{\exp(w_L^\top \mathbf{L}_{v_a}) + \exp(w_C^\top \mathbf{C}_{v_a}) + \exp(w_G^\top \mathbf{G}_{v_a})} \tag{9}$$

$$\alpha_G = \frac{\exp(w_G^\top \mathbf{G}_{v_a})}{\exp(w_L^\top \mathbf{L}_{v_a}) + \exp(w_C^\top \mathbf{C}_{v_a}) + \exp(w_G^\top \mathbf{G}_{v_a})} \tag{10}$$

These scores represent the relative importance of each embedding component in the final fused representation. The final fused embedding $\mathbf{F}_{v_a}$ is the weighted sum of the three components, using the attention scores α_L, α_C, and α_G as the weights depicted in Equation 11.

$$\mathbf{F}_{v_a} = \alpha_L \mathbf{L}_{v_a} + \alpha_C \mathbf{C}_{v_a} + \alpha_G \mathbf{G}_{v_a} \tag{11}$$

These fused embeddings $\mathbf{F} \in \mathbb{R}^{n \times 32}$ are then fed to the transformer for the final phishing user detection. As transformer [26] can capture complex patterns and relationships within sequential data, we also utilize transformer to perform our binary classification task to better exploit the temporal embeddings. Furthermore, we demonstrate the efficacy of transformer in the classification task with rigorous experimentation which is illustrated in Appendix A. Here, the one-hot encoded temporal activity vector $\mathbf{z}_{v_a}$ serves as the positional encoding for the transformer coupled with a classification layer, added to the fused embedding. While the fused embedding captures user behavior over time, benefits from the one-hot encoding by providing an explicit time anchor. Specifically, this acts as a form of *time marker* or tag that enriches the continuous embeddings for discretization thereby helping the transformer understand the temporal progression of user actions.

$$\mathbf{H}_{v_a} = \mathbf{F}_{v_a} \oplus \mathbf{z}_{v_a} \tag{12}$$

The output embedding is passed through a classification layer with softmax activation function and binary cross entropy loss function detailed in Equations 13 and 14 as,

$$\hat{y}_{v_a} = \text{softmax}(\mathbf{H}_{v_a} \mathbf{W}_c) \tag{13}$$

$$\mathcal{L}_{v_a} = -\left[y_{v_a} \log(\hat{y}_{v_a}) + (1 - y_{v_a}) \log(1 - \hat{y}_{v_a})\right] \tag{14}$$

4 Experiments

In this section, we perform empirical evaluations to demonstrate the effectiveness of the proposed framework. Specifically, we aim to answer the following research questions:

- **RQ1**: To what extent is the proposed approach effective in identifying phishing addresses within the Ethereum transaction network?
- **RQ2**: What is the individual impact of each component of CATALOG (i.e., Local Correlation, Global Correlation, and Contextual Enrichment) on the overall detection performance?
- **RQ3**: The ability of our model to ensure robustness and handling the existing research gap of Data Leakage?

4.1 Dataset Details

We consider three publicly available Ethereum transaction datasets from [4], [31], and [35], accessible via XBlock[3], to evaluate the performance of CATALOG. These datasets primarily focus on transactions from Ethereum 1.0, which may not fully reflect the current state of the network following the introduction of Ethereum 2.0 (up to 2022). The earlier datasets may contain outdated phishing patterns, limiting the pertinence of models trained on them in detecting recent fraudulent activities. To address this, we contribute a new dataset that captures recent Ethereum 2.0 transactions from August 01, 2024, to October 09, 2024. This updated dataset crucial for enhancing the accuracy and adaptability of fraud detection models within the evolving Ethereum ecosystem, offering up-to-date data on phishing and non-phishing activity. Following the data collection

[3] https://xblock.pro/#/

Table 1: Detailed dataset statistics.

Dataset	Transactions	Phishing	Non-phishing
D_1 [4]	15 million	1165	2.89 million
D_2 [31]	84489	1259	37533
D_3 [35]	1048576	1660	312179
D_4 [**Ours**]	60713	270	20194

method of Li et al. [18] we include 57 flagged phishing addresses from Etherscan[4] and their first-order neighborhood transactions, as well as 57 active non-phishing addresses with their 2,000 most recent transactions, totaling 60,686 transactions, with 270 phishing and 20,194 non-phishing users (considering the neighbors). The dataset is publicly available, and we denote them as D_1, D_2, D_3, and D_4. A $1:1$ ratio was used to address class imbalance, with scalability experiments in Section 5.3 showing CATALOG's robustness to imbalance. All four dataset provides details on Ethereum transactions, including the sender, receiver, amount, and timestamps. Each sender and receiver is labeled as either phishing or non-phishing. Comprehensive statistics can be found in Table 1.

4.2 Reproducibility Setup

For improved reproducibility, we present a detailed experimental setup for CATALOG. The reproducibility materials along with the novel dataset are available here[2]. We employ a $k = 1$ for the ego networks sliding window approach with a size of $w = 3$, processing three consecutive timestamps simultaneously. A two-layer GNN processes ego networks, with hidden dimensions of 64 and 128, followed by ReLU activation and mean aggregation. To capture local correlations, we apply a cross-attention layer with Query, Key, and Value dimensions of $\mathbb{R}^{16}$, generating local attention scores. A three-layer Transformer with four attention heads and a feed-forward network (hidden size 256, ReLU activation) produces contextually enriched embeddings. The Query, Key, and Value dimensions are $\mathbb{R}^m$, where m is the number of unique timestamps, resulting in global attention vectors. The final fused embeddings are fed into a phishing detection Transformer and classification is performed using softmax activation and Binary Cross-Entropy loss, with a 70:30 train-test split.

4.3 Baselines

For rigorous experimentation, we evaluate our model against five baseline models selected based on their novelty and relevance, namely Trans2Vec [31], TTAGN [18], Bert4Eth [12], Expanded Feature space [8], PEAEGNN [13]. Additionally, we compare our approach with four SOTA GNN models GCN [16], GAT [27], GIN [34], Dy-GCN [21], and SOTA node embedding methods Deepwalk [23], and Node2Vec [10]. To ensure a fair comparison, all baselines were evaluated using the same classifier and hyperparameters.

Evaluation metrics: We employ three conventional evaluation metrics to thoroughly assess the performance of CATALOG : (1) **Recall**: measures the proportion of actual phishing users correctly detected amongst all users detected as phishing and wrongly detected as non-phishing ; (2) **Precision**: indicates the proportion of correctly classified phishing users amongst all users detected as phishing; and (3) **F1-Score**: provides a balanced evaluation by combining both Precision and Recall.

[4]https://etherscan.io/

5 Results & Analysis

We demonstrate the efficiency of CATALOG through a set of extensive experiments discussed as follows. Due to space constraints we have added experiments on different classifiers Appendices A and the limitations of the model in Appendix B.

5.1 Baseline Comparison

To address RQ1, we evaluate the performance of the baseline methods for detecting phishing scams on Ethereum, as detailed in Table 2. Several insights appear, with CATALOG consistently outperforming other models across all three metrics. Specifically, on the D_3 dataset, our model yields 0.90 Precision, 0.86 Recall, and 0.88 F1-Score. Although Trans2Vec and our CATALOG show comparable performance in terms of precision for the D_1 and D_2 datasets, CATALOG achieves a 2% improvement in Recall, resulting in an overall better F1-Score of 0.91 and 0.90, respectively. CATALOG's higher Recall provides better network security and user trust, as the impact of undetected phishing is far more severe, making our model more relevant for detecting such potential threats. Notably, on the novel D_4 dataset, CATALOG achieves the best results, while TTAGN surpasses Trans2Vec in Recall. Expanded Features and PEAEGNN perform similarly, with F1-Scores around 0.85, whereas Bert4Eth shows the weakest performance, with just 0.46 Recall. These findings highlight the model's ability to capture both local and global network features, unlike models like Trans2Vec and TTAGN, which struggle with similar behavior patterns in phishing and non-phishing users.

Furthermore, our proposed model also excels in representing dynamic graphs, outperforming static models such as GCN, GAT, and GIN, which focus on static properties. Even when compared to Dy-GCN, CATALOG achieves higher Recall, as Dy-GCN's reliance on neighborhood aggregation results in lower Recall (0.55) due to non-phishing users in phishing user neighborhoods [8]. Similarly, static models like Node2Vec and DeepWalk perform poorly, underscoring their limitations in dynamic networks. CATALOG surpasses these methods by around 20% across all metrics, proving its strength in leveraging temporal dependencies in Ethereum transaction data for phishing detection. Overall, our proposed model outperforms all baselines by effectively capturing both local and global features and temporal dependencies, leading to superior phishing detection.

5.2 Ablation Study

To address RQ2 and evaluate the contribution of each component within the CATALOG framework, we conducted an ablation study, with results shown in Table 3. This study isolates the impact of each module. First, using only local correlations with a temporal layer resulted in a significantly lower Recall of 0.71, indicating that local temporal dynamics alone fail to capture the broader structure, leading to suboptimal representations. Similarly, using only global correlations produced an even lower Recall of 0.65, revealing that global temporal dependencies alone are insufficient for effective phishing detection due to network sparsity. When we

Table 2: Performance comparisons of CATALOG with the baseline methods for three Ethereum transaction datasets [Best results are bold, and second best is underlined].

Datasets →	D_1			D_2			D_3			D_4		
Model	***Precision***	***Recall***	***F1-Score***	***Precision***	***Recall***	***F1-Score***	***Precision***	***Recall***	***F1-Score***	***Precision***	***Recall***	***F1-Score***
Trans2Vec [31]	**0.92**	0.87	**0.89**	**0.91**	0.87	0.88	0.88	0.83	0.85	0.84	0.81	0.82
TTAGN [18]	0.80	0.85	0.82	0.79	0.83	0.81	0.86	0.82	0.84	0.80	0.83	0.81
Bert4Eth [12]	0.58	0.46	0.52	0.55	0.43	0.48	0.61	0.49	0.54	0.53	0.47	0.50
Expanded Features [8]	0.90	0.84	0.87	0.86	0.83	0.85	0.84	0.81	0.82	0.82	0.80	0.81
PEAEGNN [13]	0.87	0.84	0.86	0.85	0.81	0.83	0.86	0.82	0.84	0.84	0.81	0.82
Deepwalk [23]	0.57	0.51	0.54	0.55	0.49	0.52	0.59	0.51	0.54	0.51	0.44	0.47
Node2Vec [10]	0.54	0.41	0.46	0.51	0.43	0.46	0.51	0.46	0.48	0.53	0.42	0.47
GCN [16]	0.54	0.47	0.50	0.49	0.45	0.47	0.53	0.46	0.49	0.52	0.43	0.47
GAT [27]	0.55	0.49	0.52	0.53	0.48	0.50	0.54	0.50	0.52	0.56	0.51	0.53
GIN [34]	0.61	0.58	0.59	0.52	0.56	0.54	0.54	0.58	0.56	0.56	0.53	0.54
DyGCN [21]	0.78	0.74	0.76	0.76	0.72	0.74	0.79	0.74	0.76	0.74	0.71	0.72
CATALOG	0.91	**0.88**	**0.91**	0.89	**0.91**	**0.90**	**0.90**	**0.86**	**0.88**	**0.91**	**0.87**	**0.89**

Table 3: Ablation study results to measure individual contributions of the components for phishing user detection.

Components	*Precision*	*Recall*	*F1-Score*
Local Emb.	0.75	0.71	0.73
Global Emb.	0.71	0.65	0.68
Contextual Emb.	0.80	0.76	0.0.78
Local+Global	0.79	0.79	0.77
Local+Contextual	0.84	0.81	0.82
CATALOG	**0.92**	**0.89**	**0.91**

combined both local and global correlations, Recall improved by 9%, highlighting the benefit of integrating both aspects. The contextual embedding module achieves a Recall of 0.76, demonstrating that capturing the overall temporal context significantly enhances detection accuracy. Combining local correlations with contextual enhancement resulted in a further 10% increase in performance. Finally, when all three components—local correlation, global correlation, and contextual enrichment—were combined in the full CATALOG model, Recall improved by 14%, demonstrating the complementary strengths of each module.

5.3 Effectiveness Study

To address RQ3, we perform a robustness evaluation of CATALOG, assessing its effectiveness in mitigating data leakage along with a scalability analysis, which are detailed below.

5.3.1 ***Robustness Study:*** To evaluate the robustness of the proposed model across different datasets, we conduct an experiment where the classifier was trained on embeddings generated from D_1 [15] and tested on embeddings from three other datasets: D_2 [31], D_3 [35], and D_4 **[Ours]**. A 70:30 train-test split was employed, ensuring that the training and testing sets contained entirely distinct users. As shown in Table 4, CATALOG consistently performs well under varying data conditions. This experiment highlights the model's ability to produce high-quality user representation vectors that jointly capture both local and global behavioral patterns, enhancing phishing detection across different Ethereum datasets.

5.3.2 ***Mitigating the Dataleakage:*** We evaluate CATALOG's effectiveness using a range of temporally sequenced train-test splits, focusing on its ability to mitigate data leakage—an issue often overlooked in current research. As previously noted, many models display inflated performance due to data leakage. To ensure rigorous evaluation, we implement a sequential train-test split analysis. The model is trained on consecutive 100 days of transaction data and tested on various sequential test sets spanning 10, 15, 20, 25, and 30 days. Crucially, the training and testing sets are strictly non-overlapping to prevent any temporal leakage. The performance of the proposed model is compared to the baseline model, Trans2Vec. As shown in Table 5, the baseline's performance declines noticeably, while CATALOG consistently maintains strong results across all test sets, confirming its robustness in preventing data leakage.

Table 4: Robustness experiment of CATALOG on different training and testing data distribution.

Test Data	*Precision*	*Recall*	*F1-Score*
D_2 [31]	0.91	0.89	0.90
D_3 [35]	0.90	0.87	0.88
D_4 **[Ours]**	0.91	0.87	0.89

Table 5: Comparison of the performances of CATALOG and Trans2Vec (Second best results) on data leakage handling.

Test size	*Models*	*Precision*	*Recall*	*F1-Score*
10	Trans2Vec	0.71	0.67	0.69
	CATALOG	**0.84**	**0.81**	**0.82**
15	Trans2Vec	0.73	0.68	0.70
	CATALOG	**0.85**	**0.82**	**0.83**
20	Trans2Vec	0.73	0.71	0.72
	CATALOG	**0.85**	**0.83**	**0.84**
25	Trans2Vec	0.74	0.72	0.73
	CATALOG	**0.88**	**0.85**	**0.86**
30	Trans2Vec	0.75	0.71	0.72
	CATALOG	**0.91**	**0.88**	**0.89**

5.3.3 ***Scalability Study:*** As outlined in the dataset description, a significant class imbalance exists between phishing and non-phishing transactions. Previous studies have addressed this via undersampling [5, 29, 31, 32] or oversampling [8, 24]. However, after transaction sampling, the class imbalance was effectively managed,

Table 6: Scalability analysis of CATALOG.

Phishing to Non-phishing ratio	*Precision*	*Recall*	*F1-Score*
1 : 1	**0.92**	**0.89**	**0.90**
1 : 1.5	**0.92**	**0.89**	**0.90**
1 : 2	**0.92**	0.88	0.89
1 : 3	0.91	0.88	0.90
1 : 4	0.90	0.88	0.89
1 : 5	0.89	0.86	0.87
1 : 6	0.87	0.84	0.86
1 : 7	0.85	0.81	0.83

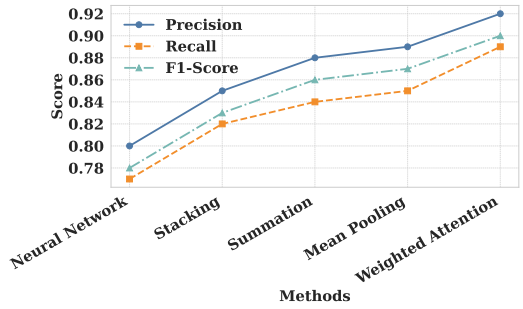


Figure 3: Performance on different Feature Fusion.

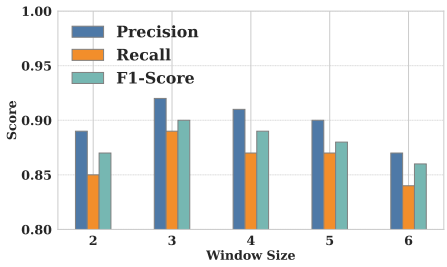


Figure 4: Experiment on different window size

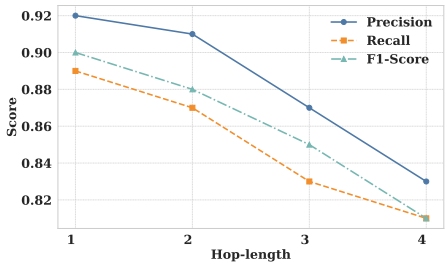


Figure 5: Performance variation on different hops.

as shown in earlier experiments. To assess CATALOG's scalability, we conduct an experiment starting with a 1:1 phishing-to-non-phishing ratio, gradually increasing the number of non-phishing transactions to observe performance changes. As shown in Table 6, our model maintains stable results until the ratio reached 1:5, beyond which a slight performance drop occurred due to the growing imbalance affecting classifier training. These results indicate that the proposed model scales well up to a certain phishing-to-non-phishing ratio, with minor declines afterward and it is suitable for real-world deployment.

5.4 Sensitivity Analysis

5.4.1 ***Different Feature Fusions****:* We experimented with several feature fusion techniques to combine the three embeddings, including element-wise summation, simple concatenation, neural networks, and weighted attention mechanisms. As shown in Figure 3, while element-wise summation emerged as the second-best performer, the weighted attention mechanism consistently outperformed the others. This superior performance is likely due to the attention mechanism's ability to assign learned weights according to the underlying data distribution, allowing it to extract more informative and meaningful embeddings.

5.4.2 ***Different Window Size****:* To determine the optimal sliding window size, we conducted experiments with window sizes ranging from $\{2, \ldots, 6\}$. The results, presented in Figure 4, indicate that a window size of three yields the best performance. Beyond this point, there is a noticeable decline in performance, can be attributed to the sparse activity of users. Therefore, we have selected a window size of three for our analysis.

5.4.3 ***Different Hop-Length****.* To determine the optimal hop length for k-hop ego graphs, we experiment with hop lengths from 1 to 4. The results showed that 1-hop performed the best, with 2-hop yielding the second best performance, followed by a significant performance drop beyond that. This decline is likely due to network sparsity and the oversmoothing issue in GNNs. Hence, we set the hop length as 1 to ensure optimal performance.

6 Conclusion

This paper introduces CATALOG, a novel representation learning framework designed to improve Ethereum phishing user detection. By addressing inherent challenges like data leakage, network sparsity, and identical user behavior, the framework enhances phishing detection with its innovative dual cross-attention layer and bi-directional MLM-based transformer pipeline. Unlike existing approaches, CATALOG jointly captures both local and global temporal dependencies while enriching user representations with contextual information, offering a more robust solution. The framework's validation on four real-world transaction datasets shows considerable performance gains, surpassing baseline models with a 7-8% increase in the F1-score. The improved performance and high scalability of CATALOG makes it suitable for real-world deployment on the back-end of Etherscan, where it can early flag suspicious accounts based on their behavioral patterns, thereby enhancing the security of Ethereum network. While CATALOG enhances phishing detection on Ethereum, it faces some limitations due to the high training time complexity and memory demand of transformers that makes it computationally expensive, particularly for long sequences. Hence, our future work will focus on optimizing the model for improved real-time performances.

References

[1] Vitalik Buterin. 2014. A next-generation smart contract and decentralized application platform. *white paper* 3, 37 (2014), 2–1.

[2] R Aroul Canessane, N Srinivasan, Abinash Beuria, Ashwini Singh, and B Muthu Kumar. 2019. Decentralised applications using ethereum blockchain. In *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, Vol. 1. IEEE, 75–79.

[3] Chainalysis. 2023. Crime Report. https://go.chainalysis.com/2023-crypto-crime-report.html.

[4] 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

[5] Weili Chen, Xiongfeng Guo, Zhiguang Chen, Zibin Zheng, and Yutong Lu. 2020. Phishing Scam Detection on Ethereum: Towards Financial Security for Blockchain Ecosystem.. In *IJCAI*. 4506–4512.

[6] Zhen Chen, Jia Huang, Shengzheng Liu, and Haixia Long. 2024. Multiscale Feature Fusion and Graph Convolutional Network for Detecting Ethereum Phishing Scams. *Electronics* 13, 6 (2024), 1012.

[7] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] Medhasree Ghosh, Dyuti Ghosh, Raju Halder, and Joydeep Chandra. 2023. Investigating the impact of structural and temporal behaviors in ethereum phishing users detection. *Blockchain: Research and Applications* (2023), 100153.

[9] Medhasree Ghosh, Raju Halder, and Joydeep Chandra. 2024. SpaTeD: Sparsity-Aware Tensor Decomposition-Based Representation Learning Framework for Phishing Scams Detection. *IEEE Transactions on Computational Social Systems* (2024), 1–15. https://doi.org/10.1109/TCSS.2024.3462552

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

[11] Li Guo, Wolfgang Karl Härdle, and Yubo Tao. 2024. A time-varying network for cryptocurrencies. *Journal of Business & Economic Statistics* 42, 2 (2024), 437–456.

[12] Sihao Hu, Zhen Zhang, Bingqiao Luo, Shengliang Lu, Bingsheng He, and Ling Liu. 2023. Bert4eth: A pre-trained transformer for ethereum fraud detection. In *Proceedings of the ACM Web Conference 2023*. 2189–2197.

[13] Hexiang Huang, Xuan Zhang, Jishu Wang, Chen Gao, Xue Li, Rui Zhu, and Qiuying Ma. 2024. PEAE-GNN: Phishing Detection on Ethereum via Augmentation Ego-Graph Based on Graph Neural Network. *IEEE Transactions on Computational Social Systems* (2024).

[14] Giacomo Ibba, Sabrina Aufiero, Silvia Bartolucci, Rumyana Neykova, Marco Ortu, Roberto Tonelli, and Giuseppe Destefanis. 2024. Mindthedapp: a toolchain for complex network-driven structural analysis of ethereum-based decentralised applications. *IEEE Access* (2024).

[15] Kaggle. 2020. Kaggle Datasets. https://www.kaggle.com/xblock/ethereum-phishing-transaction-network.

[16] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

[17] Xi Tong Lee, Arijit Khan, Sourav Sen Gupta, Yu Hann Ong, and Xuan Liu. 2020. Measurements, analyses, and insights on the entire ethereum blockchain network. In *Proceedings of The Web Conference 2020*. 155–166.

[18] 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 *Proceedings of the ACM Web Conference 2022*. 661–669.

[19] 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 *Proceedings of the 39th Annual Computer Security Applications Conference*. 352–365.

[20] 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*. 8881–8890.

[21] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2019. Dynamic Graph Convolutional Networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*.

[22] Bofeng Pan, Natalia Stakhanova, and Zhongwen Zhu. 2024. EtherShield: Time-interval Analysis for Detection of Malicious Behavior on Ethereum. *ACM Transactions on Internet Technology* 21, 1 (2024), 1–30.

[23] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[24] Farimah Poursafaei, Reihaneh Rabbany, and Zeljko Zilic. 2021. SigTran: Signature Vectors for Detecting Illicit Activities in Blockchain Transaction Networks.. In *PAKDD (1)*. Springer, 27–39.

[25] Vishnu Prasad Ranganthan, Ram Dantu, Aditya Paul, Paula Mears, and Kirill Morozov. 2018. A decentralized marketplace application on the ethereum blockchain. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 90–97.

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

[27] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

[28] Yun Wan, Feng Xiao, and Dapeng Zhang. 2023. Early-stage phishing detection on the Ethereum transaction network. *Soft Computing* 27, 7 (2023), 3707–3719.

[29] Jinhuan Wang, Pengtao Chen, Shanqing Yu, and Qi Xuan. 2021. Tsgn: Transaction subgraph networks for identifying ethereum phishing accounts. In *International Conference on Blockchain and Trustworthy Systems*. Springer, 187–200.

[30] Yixian Wang, Zhaowei Liu, Jindong Xu, and Weiqing Yan. 2022. Heterogeneous Network Representation Learning Approach for Ethereum Identity Identification. *IEEE Transactions on Computational Social Systems* (2022).

[31] 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).

[32] Yijun Xia, Jieli Liu, and Jiajing Wu. 2022. Phishing Detection on Ethereum via Attributed Ego-Graph Embedding. *IEEE Transactions on Circuits and Systems II: Express Briefs* 69, 5 (2022), 2538–2542. https://doi.org/10.1109/TCSII.2022.3159594

[33] Chang Xu, Rongrong Li, Liehuang Zhu, Xiaodong Shen, and Kashif Sharif. 2024. EWDPS: A Novel Framework for Early Warning and Detection on Ethereum Phishing Scams. *IEEE Internet of Things Journal* (2024).

[34] Keyulu Xu, Weiyue Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[35] Zihao Yuan, Qi Yuan, and Jiajing Wu. 2020. Phishing detection on Ethereum via learning representation of transaction subgraphs. In *Proc. BlockSys*, Vol. 1267. Springer Singapore, 178–191.

[36] Jiale Zhang, Hao Sui, Xiaobing Sun, Chunpeng Ge, Lu Zhou, and Willy Susilo. 2024. GrabPhisher: Phishing Scams Detection in Ethereum via Temporally Evolving GNNs. *IEEE Transactions on Services Computing* (2024).

[37] Lin Zhao, Sourav Sen Gupta, Arijit Khan, and Robby Luo. 2021. Temporal Analysis of the Entire Ethereum Blockchain Network. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 2258–2269. https://doi.org/10.1145/3442381.3449916

A Different Classifier

To conduct a thorough evaluation, we assess the performance of CATALOG using several commonly used classifiers, such as MLP, SVM, Random Forest, XGBoost, and LightGBM, as presented in Figure 6. In this experiment, we replaced the phishing detector transformer with these classifiers to directly compare their ability to identify phishing users, thus highlighting the advantages of the transformer. Also, this experiment demonstrate the overall quality of the embeddings as well. The results show that all classifiers performed effectively, with the Transformer achieving the highest Recall of 0.89. Notably, LightGBM exhibited a comparable performance to the Transformer, whereas MLP and SVM yields the poorest performances. With the attention mechanisms, Transformers can effectively analyze the temporal dependencies and contextual information in user behavior, enabling more accurate detection of subtle phishing activities. This capability allows for a deeper understanding of user interactions over time, ultimately enhancing the model's effectiveness in identifying potential threats within the Ethereum network. This evaluation affirms both the robustness of the obtained embeddings by CATALOG and the reliable performance of the embeddings across various classifiers, while also reinforcing the superiority of the Transformer in detecting phishing users.

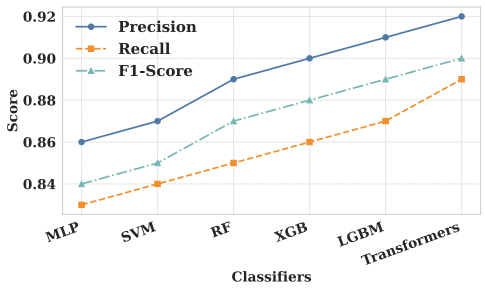


Figure 6: Performance of CATALOG on different classifiers.

B Limitations and Future work

Despite CATALOG's improvements in phishing user detection on Ethereum, several limitations arise due to the use of transformer models. Transformers have high training time complexity, largely due to the self-attention mechanism, making them computationally expensive, particularly for long sequences. Their substantial memory requirements also present challenges for scaling to larger models. Additionally, transformers demand large amounts of labeled data for optimal performance, which is often scarce in real-world settings. Furthermore, their interpretability is limited, as the decision-making process within the deep attention layers can be difficult to decipher, leading to a "black box" nature in some applications. To address these limitations, our future work will focus on model optimization to reduce computational and memory overhead, potentially through more efficient attention mechanisms or model pruning techniques. These improvements aim to enhance the model's real-time reliability and usage.