ARTEMIS: Detecting Airdrop Hunters in NFT Markets with a Graph Learning System

Chenyu Zhou
The Chinese University of Hong Kong, Shenzhen
Guangdong, China
chenyuzhou1@link.cuhk.edu.cn

Hongzhou Chen
The Chinese University of Hong Kong, Shenzhen
Guangdong, China
hongzhouchen1@link.cuhk.edu.cn

Hao Wu
The Chinese University of Hong Kong, Shenzhen
Guangdong, China
haowu3@link.cuhk.edu.cn

Junyu Zhang
The Chinese University of Hong Kong, Shenzhen
Guangdong, China
Junyuzhang2@link.cuhk.edu.cn

Wei Cai[*]
The Chinese University of Hong Kong, Shenzhen
Guangdong, China
caiwei@cuhk.edu.cn

ABSTRACT

As Web3 projects leverage airdrops to incentivize participation, airdrop hunters tactically amass wallet addresses to capitalize on token giveaways. This poses challenges to the decentralization goal. Current detection approaches tailored for cryptocurrencies overlook non-fungible tokens (NFTs) nuances. We introduce ARTEMIS, an optimized graph neural network system for identifying airdrop hunters in NFT transactions. ARTEMIS captures NFT airdrop hunters through: (1) a multimodal module extracting visual and textual insights from NFT metadata using Transformer models; (2) a tailored node aggregation function chaining NFT transaction sequences, retaining behavioral insights; (3) engineered features based on market manipulation theories detecting anomalous trading. Evaluated on decentralized exchange Blur's data, ARTEMIS significantly outperforms baselines in pinpointing hunters. This pioneering computational solution for an emergent Web3 phenomenon has broad applicability for blockchain anomaly detection. The data and code for the paper are accessible at the following link: doi.org/10.5281/zenodo.10676801.

CCS CONCEPTS

• **Security and privacy** → **Spoofing attacks**; • **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

Airdrop hunters, Web3, NFTs, Graph neural network, Multimodal deep learning

ACM Reference Format:
Chenyu Zhou, Hongzhou Chen, Hao Wu, Junyu Zhang, and Wei Cai. 2024. ARTEMIS: Detecting Airdrop Hunters in NFT Markets with a Graph Learning System. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3589334.3645597

1 INTRODUCTION

Airdrops have become a standard tactic in Web3 business operations, with Decentralized Applications (DApps) distributing tokens to encourage user engagement based on smart contract rules [24]. This practice has spurred the rise of "airdrop hunters," individuals collecting wallet addresses to claim these bountiful token giveaways by interacting with the contracts [10]. While airdrop is beneficial for attracting early DApp users, hunters' self-trading to appear as active participants threatens the ecosystem's integrity and challenges DApps' decentralization goals [22]. DApp teams must balance detecting airdrop hunters without disadvantaging genuine users.

Although airdrops and the corresponding hunters represent an emerging business model and community, relevant research remains scarce. Fan et al. [10]'s research demonstrates identifiable and observable patterns among airdrop hunters' address activities. The simplest example is "transaction loops" cycling assets between their wallets to mimic exchanges. But such straightforward techniques often get flagged by DApps' monitoring systems, prompting airdrop hunters to evolve more sophisticated strategies [1]. This illuminates the limitations of visualizing wallet interactions to detect increasingly complex fraud, falling short of required responsiveness. Moreover, current studies mostly focus on cryptocurrency and ignore the airdrop hunter issue in the NFT context.

There have been some machine learning attempts to detect blockchain fraud behaviors. Among them, graph-based modeling of wallet interactions is a very intuitive approach and has produced many detection frameworks for phishing scams [35], money laundering [23], and bot arbitrage [14]. Consequently, constructing airdrop hunter detection models using machine learning based explicitly on a graphic way is logical. These works offer valuable references for developing our airdrop hunter detection system, but directly adopting them has limitations. Specifically,

[*] Wei Cai is the corresponding author (caiwei@cuhk.edu.cn).

WWW '24, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0171-9/24/05
https://doi.org/10.1145/3589334.3645597

(i) Existing GNN modeling methods cannot accurately characterize transaction paths. Merging multiple edges between the same node pairs in the graph discards critical sequencing data for current airdrop hunter detection. **(ii)** Related works lack the utilization of intrinsic NFT features. Current practices only consider homogeneous cryptocurrency transactions, not accounting for additional information tied to NFTs as traded assets. **(iii)** Absence of tailored feature engineering. With a focus on tracing airdrop hunters in NFT transactions, factors like NFT heterogeneity introduce more noise. More sophisticated feature extraction could bolster modeling effectiveness amidst such intricacies.

Our primary focus revolves around tracking airdrop hunters in NFT transactions, a prevalent trading scenario in Web3. Addressing this, we introduce ARTEMIS: AiRdrop hunTErs detection via a MultImodal and graph learning System. In response to the aforementioned limitations, this system presents three tailored solutions: **(i)** A tailored neighbor sampling and aggregator that chains together multi-hop NFT transaction sequences, incorporating crucial behavioral information. **(ii)** Multimodal feature extraction modules, leveraging Transformer-based pre-trained models to extract visual and textual insights from NFTs. **(iii)** Engineer common NFT price representations and advanced hunter-oriented features based on market manipulation theories and domain knowledge.

In summary, the contributions of this work are:

- We formalize the problem definition of airdrop hunter detection in the NFT market context, and label hunters within Blur marketplace data as a dataset.
- We propose the ARTEMIS, the first systematic airdrop hunter detection based on machine learning. Our system significantly outperforms existing ones for hunter identification. We also introduce tailored strategies during ARTEMIS training to address associated challenges effectively.
- We design and validate multimodal feature extraction, transaction path-based multi-hop neighbor sampling and aggregation, and advanced feature representation modules, which are transferable to downstream tasks and broadly applicable to other NFT or on-chain anomaly detections.

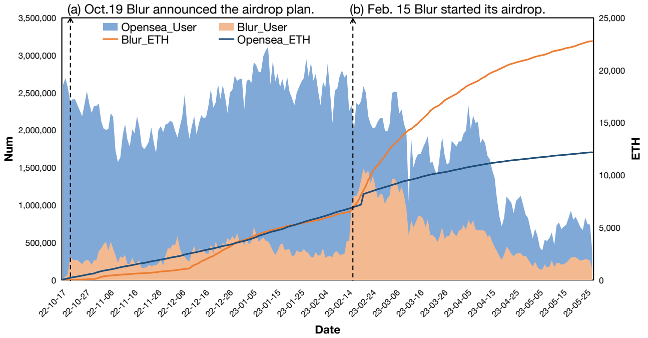


Figure 1: As a late entrant to the market, Blur announced on Oct. 19, 2022, that it would adopt an airdrop strategy, subsequently attracting users and transactions. On Feb. 15, 2023, Blur commenced its airdrop distribution, spurring a surge in daily active users and trade volumes eclipsed OpenSea's.

2 BACKGROUND AND RELATED WORKS

2.1 Blur and Airdrop Hunters

Unlike the read-only Web1 and platform-controlled Web2, Web3 leverages blockchain technologies like smart contracts and cryptocurrencies to put asset ownership back into users' hands [30]. As a vital Web3 application, non-fungible tokens (NFTs) are a new form of digital asset, each representing a unique artwork, certificate, etc., unlike traditional cryptocurrencies such as Bitcoin and Ethereum [5]. Attributing to these traits, the NFT market exploded in 2021, with total market value surging to around $10 billion by early 2023 [28]. In the NFT landscape, decentralized exchanges operating via smart contracts are crucial for bolstering market liquidity and ecosystem growth, which has long been dominated by OpenSea[1] through first-mover advantage. Blur[2] entered the NFT market as an aggregator platform in Oct. 2022, relatively late but empowered by three rounds of token airdrops to incentivize participants. During Blur's second airdrop on Feb. 15th, 2023, over 300 million tokens were distributed (over 10% of the total supply), drawing 115,834 users to surpass OpenSea [36].

Airdrops are a common token distribution approach utilized by Web3 projects like Convex and AAVE, whereby tokens are allocated to users at launch per set criteria to foster long-term holdings or activity [2]. Post-airdrop, Blur's daily active users exploded and then steadily climbed, affirming the immense potential of this Web 3 growth strategy (Figure 1). However, it predictably attracted copious airdrop hunters. Analysts reveal that 50% of Blur's NFT trading volume derives from less than 300 wallets, while 1% of "whales" hold 84% of total value locked in Blur's bid pools [25]. This implies rampant wash trading on Blur, where a traded NFT's buyer and seller are the same airdrop hunter. Such behavior stifles platform growth and triggers market contagion amidst NFTs, jeopardizing overall market health and requiring advanced detection mechanisms.

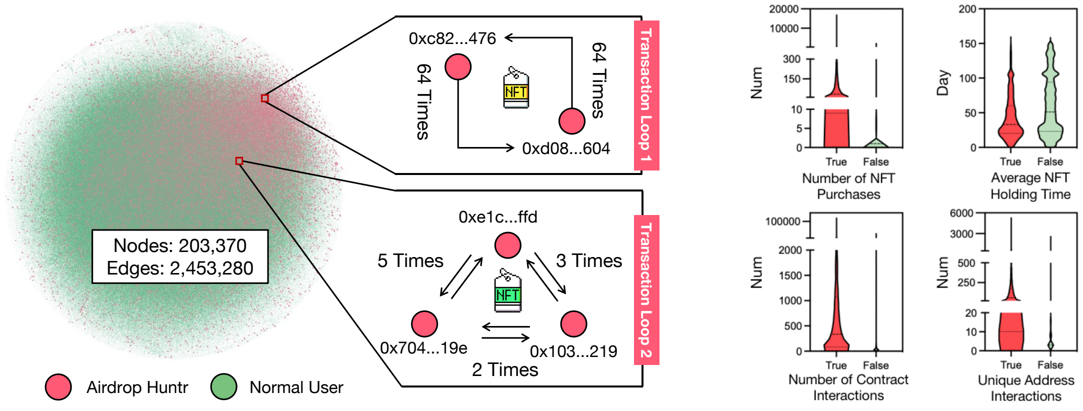
2.2 Graph Learning on Blockchain

Recently, the integration of blockchain and machine learning has garnered a plethora of notable research. This convergence becomes especially pivotal in scenarios such as anti-money laundering, phishing scam detection, and de-anonymization. Given that wallet interactions on the blockchain inherently form a network structure, it offers an ideal landscape for graph representation learning.

In the random walk-based sequence generation, though DeepWalk [27] stands as a hallmark, several advancements have also emerged. Wu et al. devised Trans2Vec [33], integrating transaction timestamps and amounts into a biased random walk process, aiming to capture transaction relationships more authentically. In a similar vein, Lin et al. embarked on a time-weighted random walk approach [18, 19]. Venturing further, Hu et al. considered the heterogeneity of nodes and introduced a "Jump-Stay" temporal-weighted biased walking method [13] for heterogeneous multi-graph modeling, balancing the distribution of diverse node types.

In the domain of GNN, the Graph Convolutional Network (GCN) is a prominent representative [17]. Shen et al., for example, applied

[1] https://opensea.io/
[2] https://blur.io/



(a) (b)

Figure 2: (a) The overview of our dataset. (b) The comparison of airdrop hunters and normal users.

GCN to phishing detection in the blockchain context [29]. Pushing the envelope, Zhou et al. incorporated attention mechanisms, proposing a Hierarchical Graph Attention Network for account de-anonymization [37]. Moreover, Lo et al. unveiled Inspection-L [23], an innovative self-supervised GNN node embedding framework, which achieved state-of-the-art results on the Elliptic money laundering detection dataset. Kanezashi et al. focused on the heterogeneity of nodes, adopted heterogeneous modeling and conducted an exhaustive evaluation of multiple GNN performances [15].

2.3 Transformer Pre-trained Models

Over the past few years, pre-trained models have made significant strides, particularly in Computer Vision (CV) and Natural Language Processing (NLP). Many pivotal advancements in these domains have been achieved by constructing and optimizing pre-trained Transformer models. In the CV arena, the Vision Transformer (ViT) proposed by Dosovitskiy and colleagues leverages the self-attention mechanism of Transformers, demonstrating performance on par with or even surpassing traditional Convolutional Neural Networks [9]. Following closely, Caron and team introduced DINO [4], which is capable of learning visual representations without labels, further propelling the progress in self-supervised learning. Concurrently, in the NLP sphere, the BERT model [7], introduced by Devlin and associates in 2019, utilizes bidirectional Transformers to pre-train extensive text data, offering robust representational learning for downstream tasks. The RoBERTa [21] model is a robustly optimized version of BERT that enhances performance and universality by tweaking BERT's training strategy and data processing workflow. The emergence of these models has enriched the pre-trained resources available for research in the CV and NLP fields, facilitating the evolution of various applications.

Innovative works have pioneered the application of pre-trained models to blockchain-centric tasks, achieving some breakthroughs. For instance, the BERT4ETH [14] model aims to utilize a pre-trained Transformer to detect fraudulent activities on Ethereum, showing significant advantages. In predicting the selling price of NFT, the MERLIN framework [6], employing multimodal deep learning, exhibits remarkable predictive performance. Furthermore, in the realm of smart contract security auditing, research indicates that large language models like GPT-4 and Claude can identify contract vulnerabilities to a certain extent, albeit manual auditors are still required to mitigate false positive rates. These studies unveil the potential of pre-trained models in blockchain applications.

3 DATASET

After defining airdrop hunter detection as our initial objective, we compiled transaction data from the NFT marketplace, Blur, over a designated period and annotated associated addresses.

Data Collection. We utilized the Etherscan API[3] to compile all NFT transaction data and airdrop records related to Blur from Oct. 19, 2022, to Apr. 1, 2023. For traded NFTs, we thoroughly collected metadata, including NFT images, descriptions, and attributes. Adopting previous works' methodology, we leveraged clustering techniques to process transaction information. Through subsequent labeling, we compared airdrop records to identify airdrop hunters meticulously. Subsequently, we sampled varying hunter scales and visualized microscopic transaction paths to validate data reliability.

Data Description. Across the Blur, we acquired 2,453,280 NFT transactions encompassing 203,370 unique user addresses. Total airdrops from Blur's official address[4] were 123,815. Among them, 4,808 (about 4%) were labeled airdrop hunters, the rest regular traders (Appendix A.1). We logged timestamps, type (buy or sell), value (based on ETH token), sending/receiving addresses, NFT collection, and relevant NFT ID for each transaction. For every wallet, we compiled historical transaction and smart contract interaction records. For NFTs themselves, we gathered full metadata for 1,155,947 traded tokens. Figure 2a shows the overview. Simultaneously, we display two simplified real-world examples: In transaction loop 1, two addresses traded the same NFT back and forth 64 times. In loop 2, three addresses reciprocally exchanged a single NFT 10 times.

[3] https://etherscan.io/
[4] 0xf2d15c0a89428c9251d71a0e29b39ff1e86bce25

Statistical Analysis. Our analysis of airdrop hunters (labeled "True") and normal participants (labeled "False") unearthed two primary airdrop hunter strategies. Initially, hunters created "transaction loops" across their wallets to forge genuine-looking transactions. Recently, their strategies evolved as detection systems became more sophisticated [20]. As Figure 2b shows: 1. Airdrop hunters had more pronounced extremes in NFT purchases, indicating a lack of interest in specific NFTs and a strategy to exploit market asymmetries.. 2. Hunters typically held NFTs for shorter periods (36 days) than normal users (53 days), aligning with a profit-driven approach. Notably, our study relies not solely on this single metric but employs it as a contributory signal within a robust framework. 3. Their smart contract interactions show higher activity levels than regular users. 4. Airdrop hunters also had a significantly higher distribution of unique addresses transacted with than regular users. These behavioral differences support using a GNN model for detecting airdrop hunters through a comprehensive feature analysis.

Brief Conclusion. With Blur's initial lenient airdrop rules and lack of hunter detection, we observed hunters routinely employ transaction loops to inflate airdrop eligibility. However, as Blur refined its airdrop policies and instituted logic to deter these basic tactics, hunters had to adopt more intricate strategies. Similarly to other Web3 projects, comprehensive detection via conventional means (e.g., rules-based filtering on structural features) becomes very challenging in this situation [10]. Nonetheless, given hunters' consistent underlying motivation to maximize their airdrop acquisition, we posit that multi-dimensional analysis of address attributes, transaction patterns, and traded asset (in our study, the NFT) characteristics using GNNs may uncover unique collective on-chain behavior to identify hunters amidst complexity effectively.

4 MOTIVATION

Our work is the first to systematically detect airdrop hunters using graph-based machine learning in NFT trading contexts. Several critical insights motivated our design:

Graph Representations of Blockchain: Previous blockchain graph modeling traded off between GNNs and random walks. GNNs fail to capture transaction sequences adequately, crucial for identifying hunters via trade paths, whereas random walks preserve sequences but lack GNNs' modeling depth. We desire both strengths, capitalizing on NFT traceability and prioritizing sequential neighbors during graph neighbor sampling.

Unique NFT Attributes: NFT heterogeneity presents opportunities for more discerning models. Intuitively, high-quality NFT records are more reliable, while hunters may manipulate low-quality ones. We posit NFT visual and textual traits as critical for assessing value and incorporate NFT feature extraction to combine quality cues with other signals to evaluate transaction legitimacy.

Advanced Pricing Features: Unlike fungible tokens, each NFT has unique pricing, complicating pattern detection from transaction values. Therefore, more sophisticated characteristics are needed to capture market manipulation traits accurately. We referred to Benford's law and the roundness detection of transaction tail numbers, which is widely used in market manipulation detection [8], to extract higher-order features from transaction prices to determine whether a transaction occurred "naturally" or by hunters.

The development of the multimodal feature module is inspired by research showing the significant impact of NFT image and text features on pricing, emphasizing the representational power [6]. Furthermore, GNN's application in Ethereum phishing detection underscores the importance of incorporating detailed features for effective modeling [15]. Hence, our goal is to integrate blockchain technologies, NFT attributes, and user behavior insights into a comprehensive detection system, distinguishing airdrop hunters.

5 ARTEMIS

For detecting airdrop hunters, we propose ARTEMIS: AiRdrop hunTErs detection via a MultImodal and graph learning System. ARTEMIS uniquely combines advanced feature representation, multimodal extraction (including visual and textual attributes), and transaction path-based neighbor sampling and aggregation. The unique melding of these techniques for NFT airdrop hunters highlights our work's core novelty. In this section, we will elucidate our design rationale and introduce the various modules of ARTEMIS.

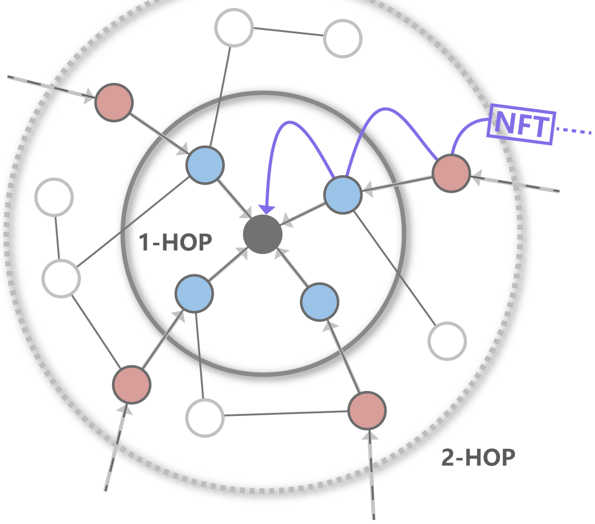


Figure 3: Neighbor sampling based on transaction paths. Blue nodes are randomly sampled 1-hop nodes that have direct NFT transactions with the center node. Red nodes are 2-hop nodes that trace the corresponding NFT transaction paths. This process can be extended to K depth.

5.1 Graph Sampling and Aggregation

In this subsection, we describe the core module within ARTEMIS, which entails an enhancement of the aggregation function in graph neural networks. We leverage the transaction paths of NFTs as a guide for neighbor sampling and node information aggregation. Unlike random sampling, our algorithm prioritizes sampling along the NFT transaction paths, ensuring that the generated embeddings can capture the context of transactions, and obtain ample information. This sampling algorithm aligns with our design philosophy of characterizing node embeddings through transaction paths.

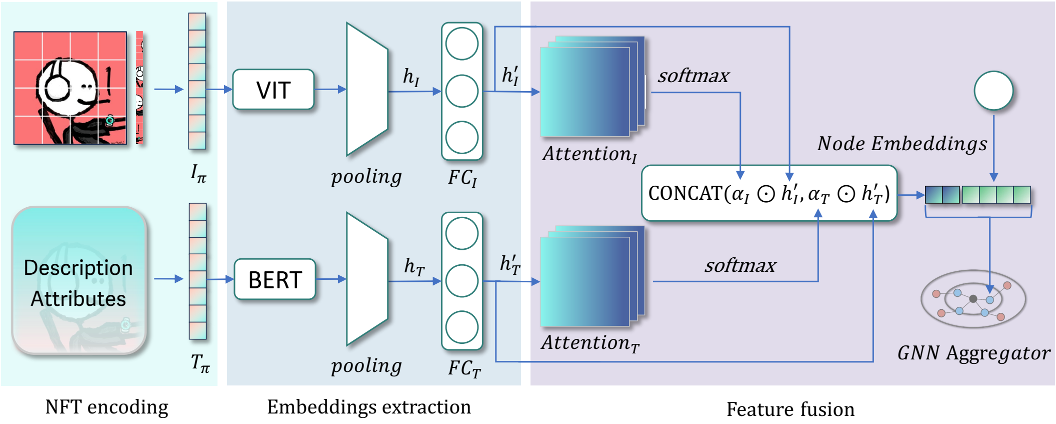


Figure 4: NFT multimodal feature extraction module. Fine-tune pretrained BERT and ViT to generate embeddings from NFT's image and text descriptions. Then fusioned embeddings are involved in the training of downstream graph model

5.1.1 ***neighbor Sampling.*** Showing as Figure 3, we define a graph $G = (V, E)$, where nodes V represent transactions addresses, and E represents NFT transactions. Our forward propagation algorithm generates embeddings for each node. We assume the model has been pre-trained, with fixed parameters including the aggregation function and weight matrices. At each depth, nodes aggregate information from neighbors. For the first hop, any neighbor can be sampled. For subsequent hops, the sampling is based on the transaction paths: if node V_0 (central node) transacted NFT_a with node V_1 (one-hop neighbor), then while sampling two-hop neighbors for V_0, with V_1 as the intermediate, only nodes that transacted NFT_a are sampled. This ensures that the sampled neighborhood is a random subset with nodes sharing the same NFT transaction history and can be extended to multi-hop neighbors.

5.1.2 ***Embedding Generation.*** Upon completing the neighbor sampling, each node updates its representation not only using its own current representation but also incorporating information from its neighbors. To achieve this, we concatenate the current representation of the node with the aggregated vector from its neighborhood. This concatenated vector is then passed through a weight matrix for a linear transformation, followed by a nonlinear activation function σ, such as ReLU, to obtain the new node representation. In our subsequent work, we further incorporate NFT features into this node representation. These NFT features are derived from our multimodal feature extraction module, with more details to be discussed in the next subsection.

5.1.3 ***neighborhood Definition and Computation Strategy.*** For efficiency and consistency, we adopt a fixed-size strategy during neighbor sampling. Specifically, for any node v, its neighborhood $N(v)$ is defined as a fixed-size subset obtained by uniform sampling from the set of nodes connected, denoted as $\{u \in V : (u, v) \in E\}$. In each iteration of forward propagation, we perform uniform neighbor sampling anew for every node.

5.2 NFT Multimodal Feature Extraction

We introduce the NFT Multimodal Feature Extraction module within ARTEMIS (Figure 4). As the transaction targets for airdrop hunters, each NFT with unique image and description. We use pre-trained vision models (ViT) and pre-trained language models (BERT) to extract their visual and textual features. Then, we fuse these two types of representations to generate a unified embedding, which then participates in the graph model's aggregation module. For the pre-trained models, we fine-tune them using a public large-scale NFT dataset [26], and we verify the performance difference between fine-tuned and non-fine-tuned models in the experimental section.

Input Data Representation. Consider a series of NFT datasets $\mathcal{D}$, where each data object $\pi \in \mathcal{D}$ is composed of a pair: an image I and its corresponding text description T. Both the image and text are initially transformed into token sequences, represented as:

$$I_\pi = [i_1, i_2, \ldots, i_n] \tag{1}$$

$$T_\pi = [t_1, t_2, \ldots, t_n] \tag{2}$$

Text Learning. For a given NFT π, we employ a Transformer-based Pre-trained Language Model (PLM) – BERT, to perform deep contextualization of the token sequence of the text part T, mapping it to a d_T-dimensional space:

$$\text{Embed}_T = \text{BERT}(T_\pi) \tag{3}$$

where $\text{Embed}_T \in \mathbb{R}^{m \times d_T}$.

Using a pooling function, we obtain an embedding vector representing the entire text:

$$h_T = \text{pooling}(\text{Embed}_T) \tag{4}$$

where $h_T \in \mathbb{R}^{d_T}$. In this context, the polling function produces a special token, [CLS], which represents the entirety of the input. The same is true for the pooling function that follows.

Image Learning. Similarly, for the image part of the NFT, we utilize a Transformer-based Pre-trained Vision Model (PVM) – ViT, for processing:

$$\text{Embed}_I = \text{ViT}(I_\pi) \tag{5}$$

where $\text{Embed}_I \in \mathbb{R}^{n \times d_I}$.

Using a pooling function, an embedding vector representing the entire image is obtained:

$$h_I = \text{pooling}(\text{Embed}_I) \tag{6}$$

where $h_I \in \mathbb{R}^{d_I}$.

Feature Fusion. To obtain a fused representation of both text and image for each NFT, we first pass h_T and h_I through two different fully connected layers for dimensionality reduction:

$$h'_T = FC_T(h_T) \tag{7}$$

$$h'_I = FC_I(h_I) \tag{8}$$

where $h'_T \in \mathbb{R}^{d'_T}$ and $h'_I \in \mathbb{R}^{d'_I}$.

Subsequently, we utilize a self-attention mechanism to compute the weights of these two embeddings:

$$\alpha_T = \text{softmax}(Attention(h'_T)) \tag{9}$$

$$\alpha_I = \text{softmax}(Attention(h'_I)) \tag{10}$$

The final fused representation consists of a concatenation of the weighted embeddings:

$$h_A = \text{concat}(\alpha_T \odot h'_T, \alpha_I \odot h'_I) \tag{11}$$

where $h_A \in \mathbb{R}^{d'_T + d'_I}$.

Subsequently, the NFT embeddings are concatenated with the address embeddings and are utilized in the downstream graph neural network training. For the complete processes, please refer to Sections 5.1 and 5.2, and Algorithm 1 in Appendix A.2.

5.3 Advanced Features

This section elaborates on the effective features we constructed during the modeling process, along with our insights and some tests regarding these features.

5.3.1 *Market Manipulation Price Features.*

Each NFT carries a unique value associated with it, posing a challenge for the model to extract generalizable information, especially from the prices of NFTs as it's hard for the model to directly learn potential market patterns, necessitating more sophisticated feature extraction techniques. We hypothesize that the activities of airdrop hunters are essentially market manipulation behaviors and validated this using two tests: Benford's Law and the rounding test of transaction prices (see Appendix A.3 for test results). Benford's Law utilizes the leading digit of price datasets to detect market manipulation. Specifically, the probability of the leading digit d (where $d \in \{1, 2, ..., 9\}$) should be given by the following formula:

$$P(d) = \log_{10}(d+1) - \log_{10}(d) \tag{12}$$

Similarly, under market manipulation, certain trailing digits in prices may appear more frequently than would be expected in a random distribution. Inspired by these theories, we extracted the leading and trailing non-zero digits of prices as features to characterize the naturalness of transactions.

5.3.2 *Asset Turnover Features.*

Through our observation of the simplistic strategy "transaction loop" previously, the trading strategies of airdrop hunters imply that their wallets often have higher asset turnover rates and multiple buyback behaviors. We extracted the average holding duration of NFT assets for each wallet, and those NFTs still held are calculated based on the time from purchase to the present. Similarly, we counted the average holding occurrences for each wallet concerning NFT assets.

5.3.3 *Wallet Activity Features.*

The number of interactive addresses can help us understand the activity level of a wallet and its connections with other users. The ratio of transaction count to interactive address count reveals the transaction exclusivity of the wallet, and often, multiple wallets held by airdrop hunters stand out in this metric. Due to the complex airdrop computation rules, interactions with contracts without generating transactions could also lead to airdrops, hence we accounted for the number of contract calls for each wallet to augment the information.

5.3.4 *Acquisition of Airdrop Tokens.*

This is a crucial post hoc feature that directly correlates to whether an address belongs to airdrop hunters. Airdrop hunters employ a series of strategies with the explicit aim of acquiring airdrop tokens from events. We aim to construct a real-time model for detecting airdrop hunters rather than post hoc inductions. Therefore, the unannotated ARTEMIS in the subsequent experimental sections does not encapsulate this feature. We only mention and analyze this feature in the ablation study subsection and conduct relevant analyses.

5.4 Training Strategies

In this subsection, we primarily introduce the training strategies tailored for ARTEMIS and explain the purpose of these strategies.

Power Law Distribution. The blockchain transaction addresses often follow a power-law distribution, meaning that a small number of high-frequency accounts appear massively in transactions. We tested the blur market address distribution and found it follows a power-law distribution (Appendix A.4). From a graph construction perspective, this implies that some nodes act as super-nodes, possessing many edges. These super-nodes, during training, can affect the feature representations of other nodes.

To mitigate this impact during training, we employed:

Inverse Frequency Sampling. We aim to reduce the probability of sampling super-nodes during neighbor sampling to ensure effective learning. Since the 2-hop and beyond neighbor sampling is based on NFT transaction paths, here we only consider the initial neighbor sampling. We calculate the degree for each node's neighbors: degree(V_i), rank the neighbor nodes in ascending order based on their degrees $r(V_i)$, and then compute the sampling probability:

$$P_{\text{sample}}(N_i) = \frac{\exp(-\beta \cdot r(N_i))}{\sum_j \exp(-\beta \cdot r(N_j))} \tag{13}$$

where β is a hyperparameter, and j iterates over all neighbor nodes.

In this formula, we employ the exponential function to emphasize the sampling priority of nodes ranked higher (i.e., with smaller degrees). The hyperparameter β determines the extent of this emphasis: a larger β value will grant significantly higher sampling probabilities to the few nodes with the smallest degrees, while a smaller β will lead to a smoother distribution. The impact of this hyperparameter on model performance will be discussed in the subsequent experimental sections.

Batch Balance. We adopt a fixed quantity of neighbor sampling for training to ensure a balanced number of positive and negative samples in each batch. Employing a fixed neighbor count aims to reduce computational load and alleviate the influence of super-nodes, while balancing samples between batches aims to mitigate biases brought about by dataset imbalance.

6 EXPERIMENTS

6.1 Experimental Setup

Task Description. Experiments are conducted on the dataset described in Section 3 with the objective of detecting airdrop hunters, formulated as a binary classification problem where airdrop hunters are considered as the positive class. For experimental purposes, the dataset is split into training and validation sets in a ratio of 9:1. The evaluation metrics adopted are the precision, recall, and F1 score for positive samples. Briefly, Precision quantifies the proportion correctly predicted as the positive class, Recall depicts the proportion of actual positive class correctly predicted, while the F1 score is the harmonic mean of the two. Therefore, we primarily use the F1 score to compare the comprehensive performance of models.

Baselines. In this experiment, the ARTEMIS model is utilized and compared against three types of baseline models: **(1)** Methods on structured data like SVM [31] and LightGBM [16] cannot use edge information, so classification is based on nodes. **(2)** Methods based on graph random walks like DeepWalk [27] and Node2Vec [11], which take advantage of both the graph structure and node features. **(3)** Methods based on Graph Neural Networks like GCN [17], GraphSAGE [12], GAT [32], and GIN [34].

We meticulously optimized the hyperparameters of each baseline model through techniques such as grid search, adjusting learning rates, batch sizes, and other critical parameters for peak performance on our dataset. We recognize the challenges presented by the inherent data structure limitations of the baseline models. For example, models like DeepWalk were limited to using graph topological information, unlike GNN-based models which could utilize a broader range of data dimensions. We were diligent in maintaining comparison fairness, mindful of these inherent differences.

Implementation. The number of layers k in the graph neural network is treated as an experimental variable. The neighbor sample size is 8. The batch size is configured to 256 with a dropout ratio of 50%. In the NFT feature extraction module, ViT-base (patch16-224) is used as the pre-trained visual model (PVM), and BERT-base-uncased is employed as the pre-trained language model (PLM). A 12-layer Transformer encoder is set up with the hidden layer size $d_I = d_T$ defaulted to 768, and 12 attention-heads are used.

Baseline models setup: For methods based on random walks (DeepWalk and Node2Vec), the number of walks is 20, the walk length is 5, and the context size is 10. For all methods based on Graph Neural Networks, the number of GNN layers is 2, the neighbor sample size is 8, the batch size is 256, and the dropout ratio is 50%.

6.2 Performance Comparison

Each experiment was conducted five times consecutively, averaging the results of the experiments:

In Table 1, we compare various methods for identifying airdrop hunters. Even without considering blockchain network topology, SVM (F1 = 0.629) and LightGBM (F1 = 0.680) have decent performance, showing the robustness of traditional machine learning methods. Random walk-based methods, DeepWalk and Node2Vec, yield moderate F1 scores of 0.496 and 0.500, highlighting the limitations of only using transaction path topologies. Among GNN techniques, GIN stands out with an F1 score of 0.776, surpassing GCN, GraphSAGE, and GAT. ARTEMIS outshines all, achieving the highest Precision and F1 (0.820 and 0.826, respectively). Compared with GNNs, even with a slightly lower Recall, the notable improvement of Precision and F1 underscores ARTEMIS's efficacy, mainly because it incorporates specific NFT information, enhancing the ability to discern between airdrop hunters and active traders and more accurately avoid mistaken identification.

Table 1: Comparison for Airdrop Hunters Detection

Method	Precision	Recall	F1
SVM [31]	0.744	0.544	0.629
LightGBM [16]	0.793	0.597	0.680
DeepWalk [27]	0.567	0.501	0.496
Node2Vec [11]	0.620	0.502	0.500
GCN [17]	0.648	0.896	0.752
GraphSAGE [12]	0.562	0.934	0.701
GAT [32]	0.464	0.873	0.579
GIN [34]	0.680	0.903	0.776
ARTEMIS	**0.820**	**0.833**	**0.826**

Table 2: Evaluating Non-real-time Improvement

Method	Precision	Recall	F1
ARTEMIS	0.820	0.833	0.826
w/ airdrop count	**0.834**	**0.836**	**0.835**

As Table 2, the enhancement to the model imparted by the post-event feature Airdrop Count. Given that our objective is to devise a real-time model, this post-event feature is utilized here solely for comparison. The performance of our current model does not significantly fall short when juxtaposed with the state attained with post-event features, thus preliminarily affirming that our model satisfactorily fulfills the requirement of real-time operation.

Table 3: Impact of Aggregation Depth K on ARTEMIS

Method	Precision	Recall	F1
ARTEMIS_1	0.783	0.774	0.778
ARTEMIS_2	0.814	0.827	0.820
ARTEMIS_3	**0.820**	**0.833**	**0.826**
ARTEMIS_4	0.803	0.812	0.807

Table 3 illustrates the impact of selecting the depth parameter K for neighbor sampling and aggregation while keeping other parameters fixed. Notably, the model's performance shows an upward trend as the depth of neighbor aggregation increases from 1 to 3. However, when the depth reaches 4, there is a slight performance decrease. This suggests that most valuable information can be distilled within three layers of neighbor aggregation, and increasing K beyond 3 starts to introduce more noise into the model.

We conducted comparative tests on the impact of the hyperparameter β in Frequency Inverse Order Sampling on model performance. The experimental results in Table 4 demonstrate that the

Table 4: Impact of Frequency Inverse Order Sampling

Method	Precision	Recall	F1
ARTEMIS(β=1.0)	**0.820**	**0.833**	**0.826**
ARTEMIS(β=0.1)	0.743	0.735	0.739
ARTEMIS(β=0.5)	0.780	0.785	0.782
ARTEMIS(β=2.0)	0.795	0.800	0.797

choice of β has a significant effect on model performance. When $\beta = 0.1$, the strategy degenerates into something approximating random sampling. When $\beta = 1.0$, ARTEMIS achieves the best performance on all metrics. This observation highlights the advantage of moderate non-linearity in capturing the intrinsic structure of the data. Further comparison shows that either larger or smaller values of β (such as $\beta = 2.0$ and $\beta = 0.1$) lead to a decline in overall performance. This might suggest that either overly aggressive or conservative non-linearity is not applicable on this dataset.

6.3 Ablation Study

Table 5: Ablation Study for Different Modules

Method	Precision	Recall	F1
ARTEMIS	0.820	0.833	0.826
Ablation study of NFT multimodal modules			
w/o fine-tuning	0.816	0.829	0.822
w/o Image Embeddings	0.804	0.818	0.811
w/o Text Embeddings	0.812	0.827	0.819
w/o NFT Multimodal	0.797	0.810	0.803
Ablation study of other modules			
w/o Adv. Features	0.801	0.817	0.809
w/o Trade Neighb. Aggr.	0.798	0.803	0.800

Table 5 presents the ablation study. Initially, there was a positive contribution from each module to the results, with a noticeable decline when removing anyone. Secondly, the neighbor aggregation based on transactions plays the most crucial role; the F1 score drops by about 0.26 when this module is omitted. The advanced features also significantly impact the performance with successful deep characterization of NFT transactions. Furthermore, we demonstrate the effectiveness of the NFT feature extraction module under various conditions. Fine-tuning has not shown sufficient effectiveness, which is reasonable considering the pre-trained model already possesses strong feature representation capabilities. Among Image and Text Embeddings, the image information proves to be more critical, aligning with our intuition that images hold more importance in NFTs. We interpreted the multimodal module in Appendix A.5.

7 DISCUSSION AND FUTURE WORK

Accurately identifying airdrop hunters is inherently challenging, considering the blurred distinction between professional airdrop hunters and active benign users. The official Blur platform was widely criticized for its aggressive banning policy towards airdrop hunters, which inadvertently harmed many legitimate users [3]. In this challenging context, ARTEMIS achieves state-of-the-art performance with a precision of 0.820 and a recall of 0.833.

ARTEMIS innovates by leveraging pre-trained models on visual (ViT) and textual (BERT) data for the unique NFT airdrop hunter challenge in Web3, where ViT emphasizes the importance of visual elements in this arena, distinct from conventional cryptocurrency analysis. Additionally, the enhanced graph neural network system is tailor-made for blockchain transactions. Our system provides a holistic analytical framework. This aligns with the WebConf community's interests in web technology, user behavior, and data analytics, advancing security and demonstrating the relevance of traditional web technologies in evolving blockchain contexts.

Regarding the practical application of the model, we acknowledge that using ARTEMIS as an automatic detector might be too aggressive. However, ARTEMIS can be an excellent auxiliary tool for identifying airdrop hunters, aiding analysts in making quicker and more efficient judgments and decisions. Detecting and relieving airdrop hunters involves the intrinsic trade-offs between potential benefits and costs for the stakeholders, necessitating rigorous game-theoretic modeling. We also recognize the necessity to demonstrate the generalizability of ARTEMIS to other NFT marketplaces, such as OpenSea and Rarible, to provide a more encompassing assessment. Our subsequent research will delve into such analyses, enriching the discourse around counter-strategies and reinforcing the transferability and robustness of ARTEMIS.

8 CONCLUSION

This work represents the first step in building a deep learning system to detect airdrop hunters, a critical and emerging problem with implications for Web3 ecosystem health and future research directions of the WWW community. We formalize the novel task of airdrop hunter detection and benchmark the performance of baseline models. Through compiling on-chain data from NFT trading markets, we propose ARTEMIS, a multimodal graph neural network system tailored for this task. ARTEMIS contains three primary design modules and accompanying training strategies to address data distribution challenges. Subsequent experiments demonstrate the model's superiority over various baselines, including traditional machine learning, random walk-based, and GNN methods, with ablation studies discussing each component's importance, especially the NFT text and image information brought with the multimodal module. In the future, we will extend our analysis data from other marketplaces to provide a more encompassing assessment of the ARTEMIS system. Moreover, tracing NFT transaction paths and extracting multimodal NFT representations and generalized advanced features could transfer to other potential NFT-based machine-learning tasks. We provide one of the first specialized computational solutions for this frontier domain.

ACKNOWLEDGMENTS

This work is supported in part by the Open Topics of Key Laboratory of Blockchain Technology and Data Security, The Ministry of Industry and Information Technology of the People's Republic of China and in part by Shenzhen Science and Technology Program (Grant No. JCYJ20210324124205016).

REFERENCES

[1] Darcy WE Allen. 2023. Crypto Airdrops: An Evolutionary Approach. *Available at SSRN 4456248* (2023).

[2] Darcy WE Allen, Chris Berg, and Aaron M Lane. 2023. Why airdrop cryptocurrency tokens? *Journal of Business Research* 163 (2023), 113945.

[3] Blur. 2022. *Wash traders were filtered out!* Blur. Retrieved Feb 6, 2024 from https://twitter.com/blur_io/status/1600262957816254466

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9650–9660.

[5] Hongzhou Chen, Haihan Duan, Maha Abdallah, Yufeng Zhu, Yonggang Wen, Abdulmotaleb El Saddik, and Wei Cai. 2023. Web3 Metaverse: State-of-the-art and vision. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 4 (2023), 1–42.

[6] Davide Costa, Lucio La Cava, and Andrea Tagarelli. 2023. Show me your NFT and I tell you how it will perform: Multimodal representation learning for NFT selling price prediction. In *Proceedings of the ACM Web Conference 2023*. 1875–1885.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] Andreas Diekmann and Ben Jann. 2010. Benford's law and fraud detection: Facts and legends. *German economic review* 11, 3 (2010), 397–401.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[10] Sizheng Fan, Tian Min, Xiao Wu, and Wei Cai. 2023. Altruistic and Profit-oriented: Making Sense of Roles in Web3 Community from Airdrop Perspective. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–16.

[11] 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.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[13] Jiahao Hu, Mingpei Cao, Xizhe Zhang, Xiong Zhang, and Yuesheng Zhu. 2023. Temporal Weighted Heterogeneous Multigraph Embedding for Ethereum Phishing Scams Detection. In *2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 1208–1213.

[14] 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.

[15] Hiroki Kanezashi, Toyotaro Suzumura, Xin Liu, and Takahiro Hirofuchi. 2022. Ethereum Fraud Detection with Heterogeneous Graph Neural Networks. *ArXiv* abs/2203.12363 (2022). https://api.semanticscholar.org/CorpusID:247619169

[16] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[17] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[18] Dan Lin, Jiajing Wu, Qi Yuan, and Zibin Zheng. 2020. Modeling and understanding ethereum transaction records via a complex network approach. *IEEE Transactions on Circuits and Systems II: Express Briefs* 67, 11 (2020), 2737–2741.

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

[20] Derek Liu, Francesco Piccoli, Katie Chen, Adrina Tang, and Victor Fang. 2023. NFT Wash Trading Detection. *arXiv preprint arXiv:2305.01543* (2023).

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[22] Zheng Liu and Hongyang Zhu. 2022. Fighting Sybils in Airdrops. *arXiv preprint arXiv:2209.04603* (2022).

[23] Wai Weng Lo, Gayan K Kulatilleke, Mohanad Sarhan, Siamak Layeghy, and Marius Portmann. 2023. Inspection-L: self-supervised GNN node embeddings for money laundering detection in bitcoin. *Applied Intelligence* (2023), 1–12.

[24] Christos A Makridis, Michael Fröwis, Kiran Sridhar, and Rainer Böhme. 2023. The rise of decentralized cryptocurrency exchanges: Evaluating the role of airdrops and governance tokens. *Journal of Corporate Finance* 79 (2023), 102358.

[25] David Marsanic. 2023. *Blur Airdrop: Just 23 Users Received More Than $1 Million in BLUR Each*. Retrieved Oct 4, 2023 from https://dailycoin.com/blur-airdrop-23-users-got-more-than-1-million-in-blur/

[26] Matthieu Nadini, Laura Alessandretti, Flavio Di Giacinto, Mauro Martino, Luca Maria Aiello, and Andrea Baronchelli. 2021. Mapping the NFT revolution: market trends, trade networks, and visual features. *Scientific Reports* 11, 1 (Oct. 2021), 20902. https://doi.org/10.1038/s41598-021-00053-8 Number: 1 Publisher: Nature Publishing Group.

[27] 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.

[28] NFTGo Research. 2023. *NFT Market Report – Q1 2023*. NFTgo. Retrieved Sep 4, 2023 from https://nftgo.io/research/nft-insights/nft-market-report-q1-2023-en/

[29] Jie Shen, Jiajun Zhou, Yunyi Xie, Shanqing Yu, and Qi Xuan. 2021. Identity inference on blockchain using graph neural network. In *Blockchain and Trustworthy Systems: Third International Conference, BlockSys 2021, Guangzhou, China, August 5–6, 2021, Revised Selected Papers 3*. Springer, 3–17.

[30] Dan Sheridan, James Harris, Frank Wear, Jerry Cowell Jr, Easton Wong, and Abbas Yazdinejad. 2022. Web3 challenges and opportunities for the market. *arXiv preprint arXiv:2209.02446* (2022).

[31] Vladimir Vapnik. 1999. *The nature of statistical learning theory*. Springer science & business media.

[32] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat* 1050, 20 (2017), 10–48550.

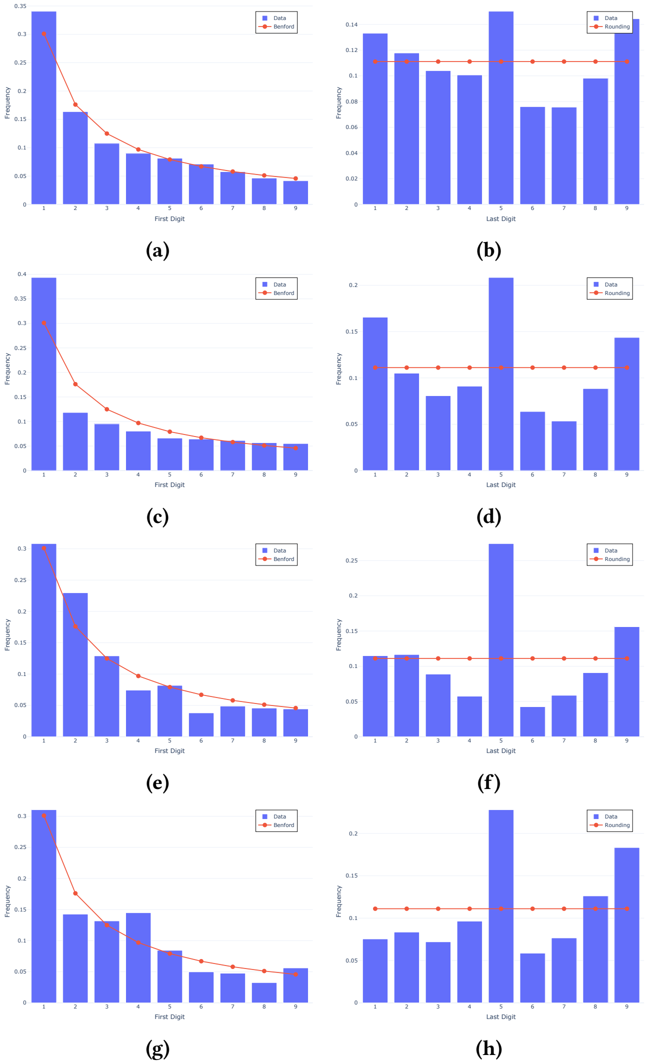
[33] Jiajing Wu, Qi Yuan, Dan Lin, Wei You, Weili Chen, Chuan Chen, and Zibin Zheng. 2020. Who are the phishers? phishing scam detection on ethereum via network embedding. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52, 2 (2020), 1156–1166.

[34] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks? arXiv:1810.00826 [cs.LG]

[35] Dunjie Zhang, Jinyin Chen, and Xiaosong Lu. 2021. Blockchain phishing scam detection via multi-channel graph classification. In *Blockchain and Trustworthy Systems: Third International Conference, BlockSys 2021, Guangzhou, China, August 5–6, 2021, Revised Selected Papers 3*. Springer, 241–256.

[36] Joel Zhao. 2023. *What BLUR Did To Surpass Opensea*. Retrieved Oct 4, 2023 from https://chaindebrief.com/what-blur-did-to-surpass-opensea

[37] Jiajun Zhou, Chenkai Hu, Jianlei Chi, Jiajing Wu, Meng Shen, and Qi Xuan. 2022. Behavior-aware account de-anonymization on ethereum interaction graph. *IEEE Transactions on Information Forensics and Security* 17 (2022), 3433–3448.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5: (a) Distribution of the first digits in NFT transaction prices on the Blur marketplace. (b) Distribution of the last digits in NFT transaction prices on the Blur marketplace. (c) Distribution of the first digits in NFT transaction prices on the Lookrare marketplace. (d) Distribution of the last digits in NFT transaction prices on the Lookrare marketplace. (e) Distribution of the first digits in NFT transaction prices on the Opensea marketplace. (f) Distribution of the last digits in NFT transaction prices on the Opensea marketplace. (g) Distribution of the first digits in NFT transaction prices on the X2Y2 marketplace. (h) Distribution of the last digits in NFT transaction prices on the X2Y2 marketplace.

A APPENDIX

A.1 Airdrop Hunters Labeling

In our data processing phase, we referred to the identification process of Fan et al. [10] and others for the preliminary clustering identification of airdrop hunters. Our clustering algorithm is derived from a similarity graph, where each address's features are obtained from its graph representation as well as wallet characteristics. We then employed agglomerative hierarchical clustering (AHC) to cluster the addresses, using the silhouette coefficient to select the optimal number of clusters. After completing the clustering, we conducted a further verification process. We invited three experienced on-chain data analysts to review and label the clustering results. This process aimed to ensure the accuracy and reliability of the clustering outcomes. During the labeling stage, we combined the airdrop records from Blur Season 1 and the transaction history of the wallets to annotate each major category of the clusters. Additionally, the data analysts conducted manual discrimination to identify and exclude outliers in each category. Finally, we adopted a consensus-based approach for classifying wallet addresses. If a wallet address did not reach a 60% consensus threshold in being labeled as an "Airdrop Hunter", we categorized it as a regular user.

A.2 Sampling and Aggregation by Transaction

Algorithm 1 Sampling and Aggregation by Transaction Paths

```
Require: Graph $G(V, E)$ with edge attributes for NFTs
Require: Node features $\{x_v, \forall v \in V\}$
Require: Depth $K$
Ensure: Vector representations $Z = \{z_v, \forall v \in V\}$
    for $k = 1$ to $K$ do
        for all $v$ in $V$ do
            if $k == 1$ then
                $N_v^k \leftarrow$ inverse_frequency_sample($G$.neighbors($v$))
            else
                for all $u$ in $N_v^{k-1}$ do
                    NFT $\leftarrow$ edge_attribute($G, v, u$)
                    $N_u^k \leftarrow$ sample($\{w|$
                    $w \in G$.neighbors$(u) \wedge$
                    edge_attribute$(G, u, w)$ = NFT$\}$)
                end for
            end if
            $h_v^k \leftarrow$ AGGREGATE$(\{h_u^{k-1} \oplus h_{NFT}, \forall u \in N_v^k\})$
            $h_v^k \leftarrow \sigma(W^k \times$ CONCAT$(h_v^{k-1}, h_v^k))$
        end for
    end for
    for all $v$ in $V$ do
        $z_v \leftarrow h_v^K$
    end for
    return $Z$
```

A.3 Market manipulation detection

Benford's Law indicates that in numerical data sets, lower digits (1-3) are more common as leading digits than higher ones (8-9), challenging the expectation of equal frequency. This principle is widely used in forensic accounting and fraud detection to identify anomalies suggesting number manipulation.

On the other hand, the Last Digit Rounding Law highlights the human tendency to round numbers, often leading to a disproportionate number of figures ending in specific digits, especially 0 or 5. Similar to Benford's Law, an unusually high occurrence of numbers ending in these rounded digits in financial or other data can hint at potential rounding or data manipulation.

The left half of Figure 5 shows the Benford's Law test for each market, while the right half displays the distribution of the last digit. The image indicates that the market distribution deviates somewhat from the expected distribution of Benford's Law. Additionally, the last digits are not as uniformly rounded as expected, suggesting a potential for market manipulation to some extent.

A.4 Address frequency power-law detection

In Figure 6, we illustrate an empirical test to ascertain whether the distribution of addresses in the blockchain follows a power law, a common characteristic observed in various networked systems. The axes are plotted on a logarithmic scale to better discern the relation. The x-axis denotes the rank of addresses, which is determined by the frequency of their occurrences, while the y-axis represents the said frequency of occurrences. In a system following a power law distribution, a linear relationship is expected on a log-log plot, as exhibited by the data in the figure. This linear trend suggests that there are a few addresses (the "head" of the distribution) that occur very frequently, while the majority of addresses (the "tail") occur much less frequently. This distribution characteristic is crucial as it highlights the existence of 'hubs' or highly connected nodes, a feature common in many real-world networks.

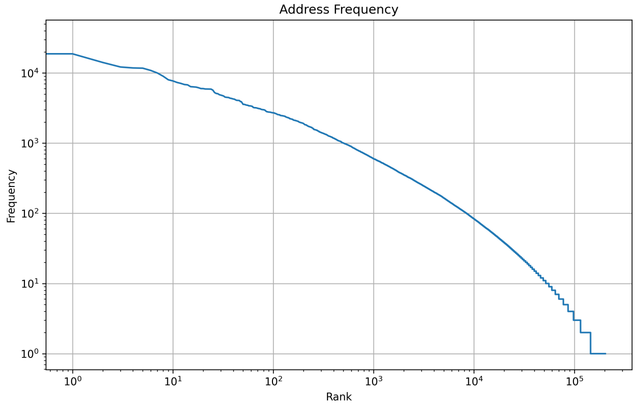


Figure 6: Address Power Law Distribution Test. The x-axis represents Rank while the y-axis represents Frequency, with both axes on a logarithmic scale. When the addresses' frequency and ranks approximate a straight line on a log-log plot, it suggests that the distribution may follow a power law.

A.5 Multimodal Attention Mechanisms

An intuitive explanation is that if an NFT exhibits transaction volumes inconsistent with its popularity, it is more likely to be involved in fraudulent transactions by airdrop hunters. In Figure 7, we selected four examples to explore the intuitive interpretations underlying the multimodal module:

We compared two PFPs (Profile Picture) NFTs, Azuki (ID=6660) and BONER (ID=2032). We believe that the image and text features of Azuki provide positive information to the model and prevent it from being targeted by airdrop hunters. Specifically, the text's average attention is concentrated on the top 5 words describing the NFT's traits, including its rarity and visual appeal. In contrast, BONER is preferred by airdrop hunters, whose descriptions contain provocative terms like "vibe" and "ethereum", and the image attention successfully identifies the common imagery of this collection.

Furthermore, we compared functional NFTs. Land (ID=19213) is a functional NFT from a game where 20,000 fixed genesis blocks were released. The image and text features accurately identify the map's rare resources (e.g., wood, grass, and water), crucial for subsequent gameplay. Flur Alpha (ID=3949) originates from an investment community's NFT, granting access to an internal discord channel for investment insights. The image feature represents the card's main characteristics, while the text feature focuses on the NFT's functionality. However, compared to Land, the ambiguous characteristics and value of Flur Alpha make it a target for airdrop hunters.

Regarding the intuition behind the utility of multimodal features, we believe that the multimodal features of an NFT encompass its value standard and popularity. Intuitively, if an NFT has a transaction volume incongruent with its popularity, it is more likely to be involved in spurious transactions by airdrop hunters. Our model relies not merely on the presence of a single indicator but on the aggregation of signals, which collectively enhance prediction. This holistic perspective aids in the detection of NFT airdrop hunters.

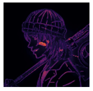
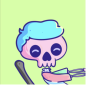
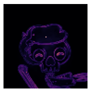
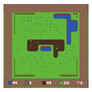
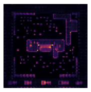
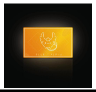
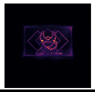
NFT Name & ID	Type	Original Image	Heads Mean Attention	Attention Top 5 words	Hunter Target
Azuki ID: 6660	PFP			1. banner 2. brown 3. chill 4. Azuki 5. hair	No
BONER ID: 2032	PFP			1. Boner 2. vibe 3. mfer 4. nerd 5. ethereum	Yes
Land ID:19213	Utility			1. Wood 2. Grass 3. Genesis 4. Land 5. community	No
Flur Alpha ID:3949	Utility			1. alpha 2. passes 3. holder 4. discord 5. own	Yes

Figure 7: Analysis of Multimodal Attention Mechanisms