HasciDB: A Database for Identifying Crypto Sybil Airdrop Hunters

ADELINE WEN, University of Washington, USA
CHUNYANG LI, University of Washington, USA
SHUTONG QU, University of Washington, USA
LEON LENG, University of Washington, USA
WEI CAI*, University of Washington, USA

Cryptocurrency projects give away free tokens to reward early users. These giveaways, called "airdrops," can be worth thousands of dollars per person. However, one person can create many fake accounts and collect the reward many times over. This fraud is called a **sybil attack**. It has drained hundreds of millions of dollars from cryptocurrency projects. Yet researchers have no shared, public database to study this fraud across different projects. We present **HasciDB, the largest open-source airdrop sybil detection database**, covering 2.54 million accounts across 16 projects (2020–2024). For every account, HasciDB checks for telltale signs of cheating. Examples include many accounts sending identical transactions within minutes, or funneling rewards to the same collection address. Each account receives a 0–100 suspicion score. All data and code are open-sourced, and anyone can query the database for free at https://hascidb.org. Across the dataset, 964,034 accounts (38.0%) exhibit suspicious patterns. Cross-project analysis shows that a project's giveaway rules determine how much fraud it attracts. The most vulnerable project lost 87.4% of its tokens to suspicious accounts. The most resistant lost only 8.5%.

CCS Concepts: • **Security and privacy** → **Database and storage security**; **Distributed systems security**; • **Applied computing** → *Electronic commerce*.

Additional Key Words and Phrases: Sybil Detection, Cryptocurrency Airdrop, Blockchain Database, Public Dataset, On-chain Analysis, Cross-project Analysis, Reproducible Research

1 Introduction

Cryptocurrency projects give away free tokens to reward early adopters. In September 2020, the trading platform Uniswap gave every past user tokens worth $1,200–$2,800, instantly making "airdrops" the standard way to bootstrap blockchain communities. But free money attracts fraud. On a blockchain, anyone can create a new account in seconds, for free, with no ID required. A cheater can create a thousand accounts, make each one look like a genuine user, and pocket a thousand rewards. This is called a **sybil attack** [4]. Sybil attacks have drained an estimated hundreds of millions of dollars from airdrops.

Not everyone chasing airdrops is cheating. An **"airdrop hunter"** who uses one account to participate in many projects is just a savvy user. Projects generally welcome this: more real users means a healthier community. The fraud starts when someone controls *many accounts at once*, using scripts to automate what should be individual, independent behavior.

Consider a concrete example. A project announces: "Every account that traded on our platform gets 400 free tokens." A legitimate user has one account and receives 400 tokens. A sybil operator creates 500 accounts, executes one trade from each, and claims 200,000 tokens. To avoid detection, all 500 accounts are funded from the same hidden wallet, execute nearly identical trades within

*Corresponding author.

Research conducted at the UW Decentralized Computing Laboratory, University of Washington.
Authors' Contact Information: Adeline Wen, ywen8@uw.edu, University of Washington, Seattle, Washington, USA; Chunyang Li, chunyang0409@outlook.com, University of Washington, Tacoma, Washington, USA; Shutong Qu, shutongq@uw.edu, University of Washington, Tacoma, Washington, USA; Leon Leng, leonleng@uw.edu, University of Washington, Seattle, Washington, USA; Wei Cai, weicaics@uw.edu, University of Washington, Tacoma, Washington, USA.

minutes, and after receiving their tokens, immediately send everything to a single collection address. These are exactly the patterns HasciDB detects: coordinated transactions, shared funding sources, and rapid token consolidation. Throughout this paper, "sybil" means an account *flagged by our detection pipeline* because its behavior matches these patterns. Some flagged accounts may be innocent power users (false positives), and some real cheaters may be careful enough to avoid detection (false negatives).

In Uniswap's case alone, we estimate that addresses exhibiting sybil-like behavior claimed $170–$400 million in tokens[1]. By 2023–2024, sybil farming became a professional industry. Commercial services now manage thousands of fake accounts using automated scripts.

Despite the scale of this problem, researchers trying to study sybil behavior face three obstacles. **First, no shared definitions**: every project uses its own criteria for what counts as a sybil, so results cannot be compared [12]. **Second, closed data**: commercial analytics companies (Nansen [17], TrustaLabs [20]) sell sybil scores to projects but do not publish their methods. Independent verification is impossible. **Third, single-project scope**: the few public sybil lists (Hop: ~14K addresses [7]; LayerZero: ~800K [11]) cover only one project each. Researchers cannot study patterns that span multiple projects. Previous academic work has changed airdrop rules to deter sybils [8, 21], modeled attacker economics, and built machine-learning detectors [13, 26]. None has produced a reusable, cross-project public database.

This paper fills that gap with **HasciDB**, a large-scale, fully open-source airdrop sybil detection database. Our prior work [12] established the detection methodology: five behavioral indicators and their thresholds, calibrated through expert consensus. This paper's contribution is turning that methodology into a complete, reproducible research infrastructure:

- **The largest public sybil dataset**: 2.54 million addresses across 16 airdrop projects on Ethereum, the most widely used blockchain. The projects span 2020–2024 and cover trading platforms, digital art marketplaces, domain name services, and more. This is 3.2× larger than the biggest existing public list (LayerZero [11]: ~800K addresses).
- **A fully reproducible pipeline**: every label can be regenerated from two public sources. Google BigQuery is a cloud database containing all Ethereum transactions. Etherscan is a public blockchain explorer. All code is open-sourced.
- **A 0–100 risk score**: instead of a binary "sybil" or "clean" label, HasciDB assigns a continuous score. Scores range from 20 (barely above threshold) to 90+ (extreme anomaly across multiple dimensions).
- **A free public query system** at https://hascidb.org: anyone can look up a single address or scan up to 50,000 addresses at once, with results returned in seconds. No login required.

To show what this database makes possible, Section 8 presents cross-project analyses that no single-project dataset could support. A key finding: the way a project distributes tokens determines its vulnerability to sybil exploitation. In the most vulnerable project, suspicious addresses captured 87.4% of tokens. In the most resistant, they captured only 8.5%.

2 Related Work

2.1 Blockchain Sybil Detection Datasets

Several projects have published lists of addresses they identified as sybils after completing their airdrops, but these lists have significant limitations. Messias et al. [16] conducted the first multi-project empirical airdrop analysis (9 projects), finding 66% of recipients rapidly sell tokens, motivating sybil-aware distribution design. Fan et al. [6] characterized airdrop participant roles from altruistic

[1] 144,443 flagged addresses × 400 UNI tokens per address × $3–$7 launch-week price = $173–$404M. This is an order-of-magnitude estimate.

to profit-oriented. Existing public sybil lists are primarily post-airdrop self-audits: Hop [7] (~14K addresses), Arbitrum [1] (~148K), and LayerZero [11] (~800K). These have three limitations: (1) single-project coverage; (2) opaque methods; (3) binary labels only. HasciDB addresses this gap by covering 16 projects with a unified framework, publishing all detection logic and raw indicator values.

2.2 Classical Sybil Defense and Blockchain Forensics

The idea of sybil attacks is not new to blockchain; researchers have studied how to defend against fake identities in online networks for over two decades. Douceur [4] first formalized the sybil attack concept. SybilGuard [24] and SybilLimit [23] use the structure of social connections to identify fake accounts, based on the insight that fake accounts tend to have few genuine friendships. Blockchain forensics research [10, 15, 19] focuses on Bitcoin address clustering and transaction graph analysis. Chen et al. [2] used graph analysis to understand the Ethereum ecosystem. Recent cross-chain bridge attack detection work [22] further reveals distinct transaction patterns across L1 and L2 environments. HasciDB combines these forensic techniques with airdrop-specific behavioral indicators.

2.3 Graph Analysis and Machine Learning Methods

More recently, researchers have applied machine learning to detect sybils by analyzing patterns in transaction data. Liu and Zhu [14] were among the first to propose graph-based sybil detection by modeling airdrop transactions as graphs to identify coordinated account clusters. Zhou et al. [26] proposed Artemis, combining Graph Neural Networks (GNN), a type of machine learning that analyzes relationship patterns, with NFT metadata for airdrop hunter detection (WWW'24). Qin et al. [18] proposed the community-boosting framework ARTEMIX. Liu et al. [13] used subgraph feature propagation combined with LightGBM, achieving high accuracy on 193,701 addresses. Imig et al. [9] applied XGBoost on LayerZero data.

Recent work extends detection further: graph analysis of blockchain ecosystems [2] provides structural and temporal insights into Ethereum transaction networks, and semi-supervised GNN with clustering [25] addresses the sparse-label challenge in blockchain fraud detection. These learning-based methods perform well on specific tasks, but no study has evaluated their cross-project generalization. HasciDB's five raw indicator values can serve directly as node features for GNN methods, and the labeled data from 16 projects supports cross-project transfer learning experiments.

2.4 Proof of Personhood and Positioning of This Work

On-chain behavioral detection targets sybil attack transaction patterns, while off-chain identity verification provides complementary defense-in-depth. Proof-of-Personhood (PoP) systems have become critical infrastructure: World ID [21] employs zero-knowledge proofs, a cryptographic technique that verifies someone is human without revealing their identity, and has been deployed in 40+ countries with thousands of Orb devices and over 16 million verified users; Human Passport [8] provides sybil resistance for over 120 projects through composable identity stamps (digital badges that prove a user has completed various verification steps).

However, PoP systems have inherent limitations: (1) PoP verifies human existence but cannot prevent multi-account operation; (2) identity stamps can be obtained through automation; (3) verification timing may be inconsistent with airdrop snapshots (the point-in-time records that determine who is eligible). In short, PoP answers "is this a real person," while HasciDB answers "is one person controlling many accounts." The two are complementary: PoP provides binary identity

verification, while HasciDB's continuous scoring (0–100) provides behavioral risk quantification; combining both enables graduated response strategies.

3 System Architecture

Before describing each component in detail (Sections 4–7), we give a high-level overview of how HasciDB works end-to-end (Fig. 1).

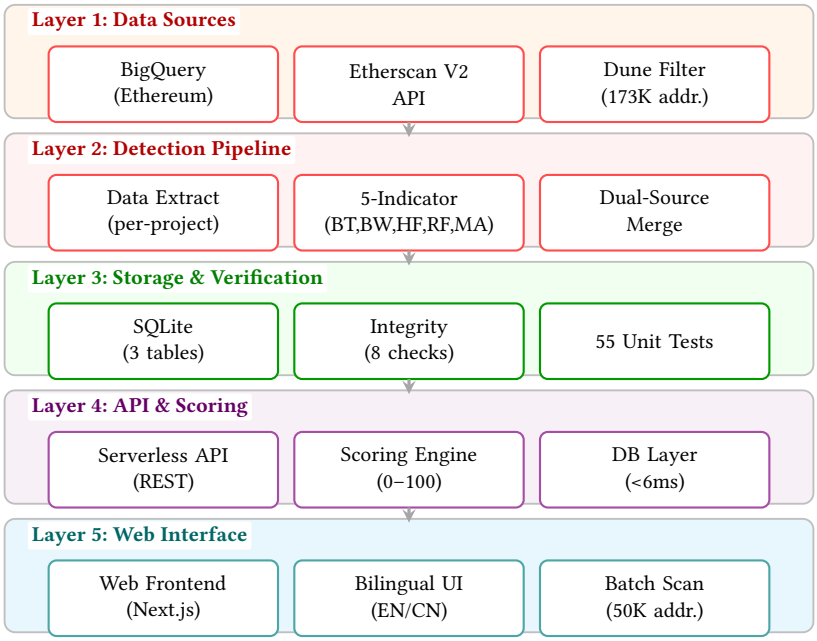


Fig. 1. HasciDB five-layer system architecture: data sources, detection pipeline, storage and verification, API and scoring, and web interface (bottom to top).

Step 1: Collect the raw data. For each of the 16 airdrop projects, we download every transaction involving eligible addresses (those that qualified for the airdrop) from two public sources: Google BigQuery (a cloud database that stores all Ethereum transactions, good for processing millions of records at once) and Etherscan (a public blockchain explorer that provides the complete transaction history for any individual account). We also maintain a filter list of 173,597 known infrastructure addresses (exchanges (where people buy and sell tokens), bridges (which transfer tokens between different blockchains), and automated contracts) from Dune Analytics [5], so that these non-human accounts do not contaminate the analysis.

Step 2: Run the detection pipeline. For every address, the system computes five behavioral indicators (detailed in Section 5). Three indicators check how the account *behaves*: does it send transactions that look scripted? Was it funded by the same wallet as many other accounts (which would suggest common ownership)? Did it suddenly become active only during the airdrop window? Two indicators check where the *money flows*: did the account immediately funnel its tokens to a collection address? Is it part of a circular money-transfer loop (e.g., A sends to B, B sends to C, C sends back to A, suggesting coordinated movement of funds)? Results from BigQuery and Etherscan are cross-checked and merged to maximize accuracy.

Step 3: Store and verify. Results are stored in two databases: a single downloadable file (SQLite, ~1.4 GB) so any researcher can run analyses locally without setting up a server, and a cloud database

(PostgreSQL) for the public query system. Eight automated integrity checks run after every pipeline execution to ensure the two copies stay consistent (Section 6).

Step 4: Score and serve. Each flagged address receives a 0–100 risk score (Section 6.3) that quantifies how suspicious it is. The score, along with all raw indicator values, is served through a web interface at https://hascidb.org where users can look up individual addresses, scan batches of up to 50,000, and explore interactive visualizations of cross-project sybil patterns (Section 7).

4 Dataset Construction

4.1 Project Selection

We need to know which accounts were eligible for each airdrop and which ones actually collected their tokens. This information comes from different sources depending on the project. HasciDB covers 16 Ethereum L1 airdrop projects (Table 1), selected by: (1) having a well-defined eligible address set or on-chain traceable claim events; (2) covering 6 protocol categories: DEX, NFT marketplaces, domain name services, public goods funding, Decentralized Finance (DeFi), and restaking; (3) spanning 2020–2024 to capture sybil behavior evolution. Ethereum L1 was chosen because it hosts the majority of high-value airdrops in this period with fully public transaction data; 9 of ~25 major mainnet airdrops were excluded due to non-reproducible eligibility criteria. PENGU is a cross-chain airdrop: users hold NFTs on Ethereum and claim tokens on Solana. Because cross-chain address mapping is unavailable, the eligible set covers only the Ethereum-side reproducible portion (15 ERC-721 contracts + MOG ERC-20 holders); see Section 9.

4.2 Eligible Address Set Extraction

How each project's eligible addresses are obtained depends on what data the project published. We classify the 16 projects into three groups.

Official Merkle trees (3 projects). Uniswap, ParaSwap, and Badger published complete Merkle proof trees containing every eligible address (Uniswap via IPFS, ParaSwap and Badger via GitHub with on-chain verified roots). We use these directly without modification.

SQL-reproducible from BigQuery (8 projects). Eight projects have eligible sets that can be reproduced by running the corresponding SQL script on BigQuery: 1inch (router callers plus MerkleDistributor claimants), LooksRare and X2Y2 (OpenSea trading volume from Wyvern exchange event logs, with LooksRare requiring ≥3 ETH), ApeCoin (BAYC/MAYC/BAKC NFT ownership replayed at the snapshot block), dYdX (MerkleDistributor claim events across all reward epochs), Gitcoin (GTC token claim events combined with BulkCheckout donation records), Ampleforth (all AMPL token holders and transactors before the snapshot), and Blur S1 (Blur Exchange contract users plus confirmed claimants). For dYdX and Ampleforth, the reconstructed sets are intentionally broader than the projects' original allocations to maximize sybil detection coverage: dYdX includes all reward epochs (~74K vs. ~64K original traders), Ampleforth includes all token holders (~87K vs. ~75K original). Blur S1 includes only Blur Exchange contract users (~147K); Blur's Care Package points thresholds were never published, so the true eligible set is unknown (see Table 1 footnote g).

Not reproducible from public data (4 projects). Four projects determined eligibility through off-chain systems that BigQuery cannot capture. The published CSV files (verified by SHA-256 checksums) are the authoritative source; the accompanying SQL scripts document an on-chain approximation but do not produce the final eligible sets:

- **Blur S2**: eligibility was based on the same centralized points system as S1, with undisclosed thresholds. The published eligible set (157,788 addresses) is derived from a curated research dataset, not from the SQL query.

Table 1. Overview of 16 airdrop projects (sorted by sybil rate, descending)

Project	Type	Year	Eligible	Claimants	Sybils	Rate
Blur S1[g]	NFT Marketplace	2023	—	124,632	111,216	—
1inch	DEX Aggregator[e]	2020	108,451	68,692	72,996	67.3%
Uniswap	DEX	2020	252,802	221,578	144,443	57.1%
LooksRare	NFT Marketplace	2022	187,160	91,358	102,441	54.7%
ApeCoin	NFT/DAO[a]	2022	17,190	15,069	9,068	52.8%
ENS	Domain Service	2021	137,689	100,295	60,192	43.7%
Gitcoin	Public Goods	2021	23,878	14,607	9,991	41.8%
Ether.fi	Liquid Staking[c]	2024	98,638	87,240	36,981	37.5%
X2Y2	NFT Marketplace	2022	900,362	35,658	296,604	32.9%
dYdX	DEX	2021	73,952	73,952	22,385	30.3%
Blur S2[g]	NFT Marketplace	2023	—	53,494	49,768	—
Badger[h]	DeFi/BTC	2020	—	7,887	8,422	—
ParaSwap	DEX Aggregator[e]	2021	19,999	15,308	2,596	13.0%
Ampleforth	Elastic Supply[d]	2020	87,296	41,989	10,173	11.7%
EigenLayer	Restaking[f]	2024	243,229	218,015	23,909	9.8%
PENGU[b]	NFT/Cross-chain	2024	52,202	—	2,849	5.5%
Total		2020–24		**1,169,774**	**964,034**	

[a]DAO: Decentralized Autonomous Organization.
[b]PENGU eligible addresses cover only the Ethereum side (15 ERC-721 contracts + MOG holders). The full airdrop covered ~7M addresses (ETH/SOL/BTC). Tokens were claimed on Solana; no on-chain claimant data is available on Ethereum.
[c]Liquid Staking: staking tokens to receive tradeable derivative tokens.
[d]Elastic Supply: token total supply automatically expands or contracts based on market price.
[e]DEX Aggregator: pools liquidity from multiple decentralized exchanges to find the best price.
[f]Restaking: reusing already-staked assets to secure additional protocols for extra yield.
[g]Blur S1/S2 used centralized points systems (Care Packages / loyalty scores) with unpublished thresholds. The true eligible set is unknown; sybil detection was performed on the set of on-chain Blur Exchange users ∪ claimants (S1: 147,435; S2: 157,788). Sybil rate is not reported because the denominator is an approximation, not the true eligible population.
[h]Badger published 19 eligibility categories but never released its Merkle tree. The eligible set cannot be fully reconstructed; sybil detection was performed on 31,786 addresses derived from partial category reconstruction.

- **ENS**: the ENS DAO published a Merkle tree incorporating off-chain factors (active reverse records, domain expiry dates) that cannot be replicated from on-chain data alone.
- **Ether.fi**: tokens were allocated via a proprietary points system (deposit amount, lock duration, time-weighting). Our SQL covers eETH/weETH staking deposits but not the point multipliers.
- **EigenLayer**: allocation used a proprietary formula (staked amount × time × multipliers) with internal sybil filtering. Our SQL captures staking and delegation events but not the exact allocation logic.

PENGU covers only the Ethereum side (15 NFT collections + MOG token holders), as discussed in Table 1 footnote b. All 16 extraction scripts are open-sourced.

4.3 Claim Data and Exchange Filtering

To find out which eligible addresses actually collected their tokens, we look at the blockchain's permanent transaction records. Specifically, actual claimants are extracted by parsing on-chain Claimed and Transfer event logs (structured records that the blockchain automatically creates whenever tokens change hands). Before computing indicators, we filter out 173,597 known infrastructure addresses (centralized exchange hot/cold wallets, DEX router and pool contracts, bridge contracts, MEV bots, and other automated contracts) sourced from Dune Analytics. Without this filter, infrastructure addresses cause systematic false positives: for example, a Binance hot wallet

that funds thousands of user withdrawals would inflate every recipient's BW score, and users who sell their claimed tokens on an exchange would make the exchange deposit address look like a sybil consolidation receiver (inflating RF). Dune was chosen because it provides the largest publicly available, community-maintained blockchain infrastructure label set; commercial alternatives (Chainalysis, Nansen) offer larger label sets but are closed-source and non-reproducible. The RF indicator is computed only for addresses that have claimed (RF=0 for non-claimants), because unclaimed addresses have no post-claim token flow to analyze; consequently, sybil addresses that qualified but never claimed can only be detected via the four remaining indicators.

5 Detection Pipeline

5.1 Problem Definition

In plain terms: for every account in every project, we compute five numbers that measure how suspicious its behavior is. If any of these numbers exceeds a threshold set by domain experts, the account is flagged as a potential sybil. For readers who prefer a formal statement:

Given a set of airdrop projects $P = \{p_1, p_2, \ldots, p_{16}\}$ and each project's eligible address set $A(p_j)$, the global address set is $A = \bigcup_j A(p_j)$. For each (address, project) pair (a_i, p_j), we define five indicator functions $f_k : A \times P \to \mathbb{R}^{\geq 0}$ ($k \in \{\text{BT}, \text{BW}, \text{HF}, \text{RF}, \text{MA}\}$), each computing a non-negative real value from on-chain transaction data. Given the expert-consensus thresholds (BT≥5, BW≥10, HF≥0.80, RF≥0.50, MA≥5), the labeling rule is:

$$\text{sybil}(a_i, p_j) = 1 \iff \exists k : f_k(a_i, p_j) \geq \tau_k \tag{1}$$

That is, OR-trigger: any single indicator reaching or exceeding its threshold results in a sybil label (is_sybil = 1). This dataset adopts the labels produced by this construction as the primary sybil labels, but these labels have not been verified against external ground truth. Users may apply stricter rules based on the published raw indicator values and continuous scores.

End-to-end example. To preview how the full system works: for a real Uniswap address (`0x457da487ceee7d89bdc85848d405158a77a44bc9`), the pipeline detects 1,661 coordinated trading events, funding from a shared wallet, 91% activity concentration in the airdrop window, 100% token funneling, and 298 circular fund paths. All five indicators exceed their thresholds, producing a risk score of 81.4/100 (Critical). Section 7 walks through this address in detail with screenshots of the query system output.

5.2 Five-Indicator Dual-Axis Framework

HasciDB does not define its detection signals ad hoc. Instead, it operationalizes the indicator framework established in our prior Delphi consensus study [12], which was designed specifically to answer what should count as "suspicious enough" in airdrop sybil detection when no accepted ground truth exists.

That Delphi study recruited 10 external experts with blockchain security or airdrop analysis experience (authors excluded from scoring) and used two anonymous questionnaire rounds. Round 1 collected independent judgments about each candidate indicator's appropriateness and threshold on a 1–5 Likert scale; Round 2 returned the Round 1 statistical summaries (median and interquartile range) and allowed experts to revise their assessments. Final thresholds are the Round 2 medians, and the retained indicators all reached median ≥ 4.0/5.0, IQR ≤ 1, and consensus level ≥ 80% (expert selection criteria and full protocol in [12]). We use Delphi expert consensus [3] because this domain lacks independently verified labels, so purely data-driven threshold optimization would rest on circular assumptions. Researchers can still use HasciDB's published raw indicator values to apply stricter or alternative classification rules.

From that broader Delphi framework, HasciDB instantiates five indicators: Batch Transaction (BT), Batch Wallet (BW), High Frequency (HF), Rapid Funneling (RF), and Multi-Address Cycling (MA). These five are the concrete signals used throughout the database, API, and risk-scoring pipeline (Table 2).

Why these five out of a larger candidate pool? The Delphi study evaluated additional signals as well, including temporal regularity, gas-price clustering, contract interaction diversity, and several counter-evidence indicators such as account age and identity credentials. BT, BW, HF, RF, and MA were selected because they received the strongest expert support while covering distinct manifestations of sybil cost-saving behavior: BT captures scripted bursts of near-identical transactions; BW captures shared seeding from a common parent wallet; HF captures accounts whose activity is concentrated almost entirely in the farming window; RF captures post-claim token consolidation to a single collector; and MA captures cyclical fund routing across multiple addresses. The set is therefore intentionally focused rather than exhaustive, and HasciDB's modular pipeline allows additional indicators to be appended later without changing the existing ones.

Rationale for dual-axis partitioning. Once these five indicators are fixed, they fall into two natural groups. BT, BW, and HF describe *how* an address behaves, so they form the *operational axis*. RF and MA describe *how funds move across addresses*, so they form the *fund-flow axis*. This partition reflects two common sybil strategies: some operators reveal themselves primarily through coordinated behavior, while others reveal themselves through fund-linkage patterns. Organizing the framework along these two orthogonal axes clarifies why the five indicators complement rather than duplicate one another.

Table 2. Five-indicator detection framework

Indicator	Full Name	Axis	Thr.	Algorithm Core
BT	Batch Transaction	Ops	≥ 5	Detects coordinated multi-address behavior sending similar amounts to the same target within 10-minute windows
BW	Batch Wallet	Ops	≥ 10	Traces multiple new addresses funded by the same source within 30 days
HF	High Frequency	Ops	≥ 0.80	Computes the ratio of transactions within a 180-day window to total transactions, identifying concentrated activity[†]
RF	Rapid Funneling	Fund	≥ 0.50	Detects whether tokens flow to a unified collection address within 30 days after airdrop claim
MA	Multi-Address	Fund	≥ 5	Traces fund cycling paths that return to the original address within 2–3 hops

[†]HF uses a fixed 180-day window. This duration was chosen because 12 of the 16 projects have eligibility windows shorter than 180 days, making it a practical upper bound that preserves HF's semantic consistency. For projects with eligibility periods exceeding 180 days (e.g., ENS: 731 days, Ampleforth: 638 days), the 180-day window covers only ∼25% of the eligibility period, shifting HF's semantics from "activity concentration" to "whether activity is concentrated in a specific time segment"; HF results for these projects should be interpreted cautiously.

The following elaborates on each indicator's detection logic with design rationale for key parameters:

BT (Batch Transaction) algorithm. For each address, collect outgoing transactions (value > 0), filtering infrastructure addresses (exchanges, bridges, known contracts). Apply a sliding window of 5 consecutive transactions: a window qualifies if time_span ≤ 600 s and not all values are round ETH amounts. In essence, we create a unique signature for each group of transactions so that transactions going to the same destination for approximately the same amount get the same signature. For each qualifying window, compute a fingerprint, a compact hash (specifically `FARM_FINGERPRINT`) over the sequence of (to_address || value_band) tuples, where value_band = `ROUND(LOG10(value)`

`* 10)` yields logarithmic tolerance bands (~25% per band, accommodating gas-price fluctuations while excluding unrelated transactions). Addresses sharing the same fingerprint hash are grouped into a batch cluster. BT_score = count of distinct other addresses sharing any hash with the target; BT ≥ 5 triggers. The 10-minute (600 s) window is calibrated to automated batch scripts: median completion for five script-driven transactions is ~90 s, but congestion can extend delays to ~10 minutes; a shorter window (5 min) misses congestion-delayed batches, while a longer one (30 min) absorbs coincidental transactions.

RF (Rapid Funneling). The funneling recipient is required to have received tokens from at least 3 distinct source addresses, distinguishing sybil funneling behavior (one address collecting tokens from multiple controlled wallets) from normal bilateral transfers (e.g., a single transfer to a friend or exchange). A threshold of 2 would generate excessive false positives (any address receiving two inbound transfers would be flagged), whereas 3 effectively filters for concentrated funneling patterns.

MA (Multi-Address Cycling). The pipeline enumerates cycles by querying the full `ethereum.transactions` table (not restricted to eligible addresses). Two cycle types are detected: 2-hop (A→B→A) and 3-hop (A→B→C→A), each within a 30-day window. Each return leg must carry ≥80% of the outgoing value, accommodating gas losses (0.1%–1%) and DEX slippage. 1-hop cycles (self-transfers) are excluded as trivially detectable; 4+ hops face $O(n^k)$ combinatorial explosion and are empirically dominated by legitimate DeFi routing (decentralized exchanges often route a single trade through multiple intermediate pools, creating cycles that look suspicious but are actually normal). The 80% tolerance is deliberately conservative: strict matching (95%) would miss DEX-routed cycles, while permissive thresholds (50%) would mislabel normal activity. MA_score = count of valid 2-hop + 3-hop cycles, capped at 10,000 (~2,000× the threshold of 5) to prevent path explosion in high-connectivity addresses. Fewer than 0.1% of addresses reach this cap.

BW definition. "Same funding source" is defined as: address X received its first ETH transfer from address Y within 30 days of X's creation (i.e., Y is X's direct funder). If address Y funds ≥10 distinct addresses in this manner within 30 days, all funded addresses receive BW ≥10, triggering the threshold. This definition traces only 1-hop direct funding relationships without recursive propagation; transfers from known exchanges and bridge contracts (via the Dune filter set) are excluded.

30-day window for BW/RF/MA. All three indicators use a 30-day window, corresponding to the typical operational cycle of airdrop sybil campaigns: operators generally complete wallet funding, eligibility interactions, and profit funneling within one month. A 30-day window is sufficient to capture rapid operational behavior while excluding noise from legitimate users' long-term asset adjustments.

Classification rule (single-axis OR-trigger):

$$\text{ops_flag} = (\text{BT} \geq 5) \vee (\text{BW} \geq 10) \vee (\text{HF} \geq 0.80) \tag{2}$$

$$\text{fund_flag} = (\text{RF} \geq 0.50) \vee (\text{MA} \geq 5) \tag{3}$$

$$\text{is_sybil} = \text{ops_flag} \vee \text{fund_flag} \tag{4}$$

Why flag an account if *any single* indicator is suspicious? An alternative would be to require multiple indicators to agree before flagging (an AND rule). We deliberately chose the more permissive OR rule for three reasons. First, real sybil operators use diverse strategies: some leave traces in transaction patterns but not in fund flows, others in fund flows but not in transaction patterns. An AND rule would miss every attacker who is careful in one dimension. Second, HasciDB is designed as a research database, not a final verdict. By casting a wide net, we give researchers the raw data to apply their own, stricter rules. Third, every threshold is already set at a high bar

(>95th percentile of all addresses), so even a single trigger represents genuinely unusual behavior. We acknowledge one limitation: the MA indicator has many addresses clustered near its threshold (scores of 4–6), making MA-only flags less confident than others. This is one reason we provide continuous scores: users who want higher confidence can simply require score ≥ 30 (two indicators) instead of ≥ 20 (one indicator). Section 8 shows how different classification rules affect the dataset size.

5.3 Dual-Source Detection and Merge Strategy

Why dual sources are necessary. BigQuery computes BT, BW, and HF from a pre-extracted transaction table (`target_txs`) scoped to project-relevant block ranges. This enables batch processing of hundreds of thousands of addresses in minutes, but the pre-extracted subset may not include every address's complete lifetime transaction history. The consequence is twofold: HF can be inflated because the denominator (total lifetime transactions) is underestimated when older transactions fall outside the queried range, and BT/BW can miss coordinated transactions or funding relationships that occurred outside the block window. Etherscan indexes by address and returns the complete transaction sequence from creation to present regardless of block range, correcting these gaps at the cost of slower per-address retrieval.

Primary engine. BigQuery extracts raw transaction data into temporary tables and computes all five indicators in parallel. **Verification source.** The Etherscan module (6 API keys, rate-limited at 4.8 req/s per key) rechecks BT, BW, and HF using each address's complete chain history.

Merge strategy. BT and BW take the maximum of both sources: either source may detect batch coordination or shared funding that the other missed, and taking the maximum ensures no omissions. HF is corrected by Etherscan's complete history where available, since BigQuery's incomplete denominator tends to overestimate this ratio. RF and MA use BigQuery only: these indicators query the full public `token_transfers` and `transactions` tables directly (not the pre-extracted subset), and the standard Etherscan API does not return internal transactions that MA's cycle detection requires.

6 Database Schema

6.1 Schema Design

HasciDB stores its results in a database with three tables, each designed for a different type of question. The first table holds the most detailed data (one row per address per project); the second summarizes each address across all projects; the third captures relationships between addresses. This structure lets researchers ask both narrow questions ("is this specific address suspicious in Uniswap?") and broad questions ("which addresses are flagged across five or more projects?").

sybil_results (main table, 2,539,857 rows): Each row represents a single (address, project) pair, the most granular unit of analysis. In database terms, the columns are: the five raw indicator values (bt_score, bw_score, hf_score, rf_score, ma_score), derived boolean flags (ops_flag, fund_flag, is_sybil), the trigger count (n_indicators), and claim data (claimed, amount). This is the primary table for per-project analysis: the same address may appear in up to 16 rows (one per project), with potentially different indicator values and classification outcomes in each. For example, an address might trigger BT in 1inch (where it batch-swapped) but show clean behavior in ENS (where it simply held a domain).

address_profiles (aggregation view, 1,807,415 rows, computed on demand): One row per unique address, aggregating cross-project performance. Fields include how many projects the address was eligible for, how many it was flagged in, which projects specifically, and which indicator triggered most often. This table enables the cross-project analyses presented in Section 8: researchers can

query which addresses are flagged across multiple projects, compute participation breadth statistics, and identify the "broad-spectrum" operator population.

Graph index tables (optional, generated by a helper script): `sybil_clusters` groups same-project sybils by indicator values; `cross_project_edges` summarizes cross-project behavior of multi-project addresses.

Table 3 provides the complete column specification of the `sybil_results` table. Every field is computed deterministically from on-chain data and stored in simple formats that can be queried directly.

Table 3. Column specification of the `sybil_results` table

Column	Type	Description
address	TEXT (PK)	Ethereum address (0x-prefixed, checksummed)
project	TEXT (PK)	Project slug (e.g., "uniswap")
bt_score	REAL	Batch Transaction raw value
bw_score	REAL	Batch Wallet raw value
hf_score	REAL	High Frequency ratio [0,1]
rf_score	REAL	Rapid Funneling ratio [0,1]
ma_score	REAL	Multi-Address Cycling count
ops_flag	BOOL	BT≥5 OR BW≥10 OR HF≥0.80
fund_flag	BOOL	RF≥0.50 OR MA≥5
is_sybil	BOOL	ops_flag OR fund_flag
n_indicators	INT	Count of triggered indicators [0–5]
claimed	BOOL	Whether address claimed tokens
amount	REAL	Token amount claimed (NULL if N/A)

Primary key is (address, project). Composite indexes on (project, is_sybil) and (address) support the two most common query patterns: per-project filtering and cross-project address lookup.

6.2 Data Integrity Verification

Reproducibility requires not only open-source code but verifiable data integrity. HasciDB runs 8 automated checks after every pipeline execution, verifying that column types match, no duplicate records exist, all values fall within valid ranges, and the two database copies (SQLite and PostgreSQL) agree. These checks complete in under 12 seconds on the full 2.54M-row dataset; any failure halts the pipeline and identifies the affected rows.

6.3 Scoring Engine

A binary label is insufficient: an address barely triggering one indicator (BT=5) is far less threatening than one triggering all five at extreme values (BT=15,287, RF=100%). HasciDB therefore computes a **0–100 risk score** for every address. The 0–100 scale was chosen for intuitive interpretability (aligning with percentage-based mental models) and sufficient granularity for seven risk tiers without requiring decimal precision.

The score is the sum of two sub-scores:

- **Part A (breadth, 0–50 points):** Of the five indicators, how many exceeded their threshold? More triggers = more suspicious.
- **Part B (severity, 0–50 points):** For each triggered indicator, how far above the threshold is it? Farther = more severe.

Why 50/50? Breadth and severity are two independent dimensions of suspicion. An address that triggers all five indicators but each barely above threshold, and an address that triggers only one

but at an extreme value (e.g., BT=50,000), are both high-risk sybils, but in different ways: the former suggests a wide-net automation script, while the latter indicates deep manipulation of a single dimension. If breadth were weighted 80% and severity only 20%, the latter attack pattern would be underestimated, and vice versa. Equal weighting ensures both patterns receive sufficiently high scores, and is a robust default in the absence of externally verified labels. Researchers may adjust the weight ratio for specific use cases. Empirically, the 50/50 split produces good discrimination: the Gini coefficient (a standard measure of how spread out the scores are; 0 means all identical, 1 means maximally varied) of the score distribution among flagged addresses is 0.34, indicating meaningful stratification rather than a degenerate distribution where all flagged addresses cluster at one score value. For context, a Gini of 0 would mean every flagged address receives the same score (no discrimination), while 1 would mean all scores cluster at two extremes. A Gini of 0.34 is comparable to moderate income inequality in developed economies, indicating that the scoring system successfully spreads addresses across a meaningful range rather than collapsing them into a single cluster. This level of dispersion confirms that the 50/50 breadth-severity split produces actionable differentiation: project teams can set a threshold at any point along the score continuum and expect a substantively different population above versus below it.
Total score = Part A + Part B, maximum 100 (Fig. 2). **20 is the sybil boundary**: an address with exactly one triggered indicator and zero excess scores exactly 20; below 20 = non-sybil, 20 or above = sybil, fully consistent with the binary label.

The system maps scores to seven risk tiers (Table 4). Seven tiers align with the Part A plateaus at $n = 0$–5; fewer tiers would conflate single-trigger borderline cases with multi-trigger severe ones. In practice, the R1 tier (1–19) is empty because non-flagged addresses almost universally score exactly zero; R1 is retained for future extensibility as additional indicators may produce near-threshold cases.

Table 4. Seven-tier risk classification system

Score	Risk Level	Action
0	R0 Clean	Full trust
1–19	R1 Low	Monitor
20–29	R2 Medium	Manual review
30–49	R3 High	Restrict privileges
50–69	R4 Very High	High-prob. sybil
70–89	R5 Critical	Multi-indicator severe excess
90–100	R6 Extreme	All indicators extreme

Part A: Count the red flags.

Of the five indicators, how many have raw values at or above their threshold? Call this count n. The higher n is, the higher the Part A score:

Red flags n	0	1	2	3	4	5
Part A score	0–19	20	35	42	47	50

Notice the critical jump: when $n = 0$ (no red flags), the maximum is 19; it *never* reaches 20. The moment $n = 1$, Part A jumps to 20, guaranteeing the sybil boundary. Subsequent increments decrease (+20, +15, +7, +5, +3) because the first trigger (crossing the sybil boundary) is the strongest signal; each additional trigger adds diminishing evidence. These increments produce gradually increasing Part A scores (20, 35, 42, 47, 50), maximizing score discrimination in the R2–R3 range where 92.8% of flagged addresses concentrate; a linear mapping would compress most flagged

addresses into a single tier, reducing discriminative utility. Two endpoint constraints anchor the scale: n=1 must yield 20 (binary-label consistency) and n=5 must yield 50 (equal room for Part B); the intermediate values were selected from this family to maximize tier separation in the R2–R3 range.

When $n = 0$, the score is not necessarily zero. If an indicator is close to its threshold, it earns an "approach score" as an early warning. The formula is: per-indicator approach = 4×raw value/threshold (the coefficient 4 derives from $19/5 \approx 3.8$, rounded up: it ensures each indicator contributes just under 4 points, so even five simultaneous near-threshold values cannot reach the sybil boundary of 20), summed across all five and capped at 19. For example, if BT's threshold is 5 and the raw value is 4, the approach score is $4 \times 4/5 = 3.2$.

Part B: Measure how far above the line.

Part A only cares *whether* an indicator exceeded its threshold, not *by how much*. Part B fills this gap: for each triggered indicator, it measures how far above the threshold the raw value is, and converts that into a 0–10 score:

$$\text{per-indicator score} = 10 \times \frac{\text{raw value} - \text{threshold}}{\text{ceiling} - \text{threshold}} \tag{5}$$

The "ceiling" caps each indicator to prevent extreme outliers (e.g., BT=50,000) from monopolizing the score. For count-based indicators (BT=500, BW=200, MA=500), the ceiling is the approximate 99.9th percentile among flagged addresses, rounded (99.9th ensures <0.1% truncation; ceilings differ across indicators because their empirical distributions differ). For ratio-based indicators (HF, RF), the ceiling is simply their natural maximum of 1.00. Values above the ceiling are treated as equal to the ceiling. Each indicator contributes at most 10 points; all five together cap at 50.

Scoring non-claimants. Addresses that never claimed tokens have RF = 0 by definition (Section 4), so RF can never trigger for them. Their theoretical maximum score is 87 (Part A = 47 for n=4, plus Part B = 40 for four indicators) rather than 100. We do not normalize for this asymmetry because it reflects a genuine difference in realized harm: non-claimants did not extract tokens, so the funneling dimension of their threat is objectively zero, not merely unmeasured. A lower ceiling appropriately reflects their lower economic impact.

Two examples. Example 1: BT=5 (at threshold), all others zero. Part A: 1 flag → 20 points. Part B: excess = 0. **Total** = 20, R2 (Medium). **Example 2:** BT=630, RF=1.00, others zero. Part A: 2 flags → 35. Part B: BT capped at ceiling → 10; RF = 10. **Total** = 55, R4 (Very High).

Fig. 2 illustrates the scoring decomposition with a concrete example: an address triggering four of five indicators (BW not triggered) receives Part A = 47, plus Part B severity bonuses totaling 21.4, for a score of 68.4 (R4 Very High).

7 Query System

A database is only useful if people can actually query it. Most research datasets require downloading large files and writing custom analysis scripts, which limits adoption. HasciDB therefore provides a free public web interface at https://hascidb.org that lets anyone explore the data without downloading or programming. The system serves three user groups: (1) airdrop project teams performing pre-distribution filtering via batch scan; (2) researchers studying sybil behavior through single-address deep dives; and (3) individual users verifying address reputation. No wallet connection or authentication is required. Performance: <6 ms database round-trip, <50 ms P99 latency.

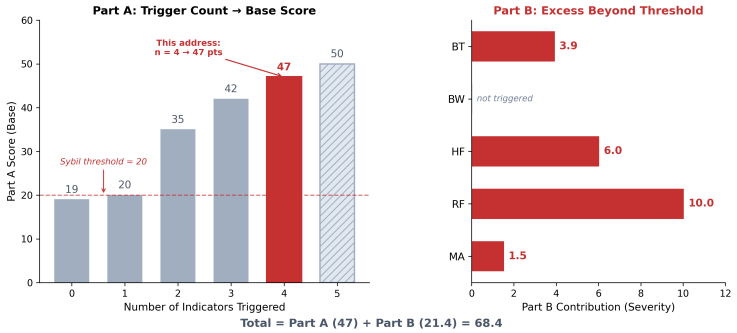


Fig. 2. Two-part scoring decomposition illustrated with a concrete example (4 of 5 indicators triggered). **Left (Part A):** the red bar highlights this address at n=4 triggers (Part A = 47); the hatched bar at n=5 is not reached. The dashed line marks the sybil boundary at 20. **Right (Part B):** four triggered indicators contribute severity bonuses proportional to their excess above threshold (0–10 each); BW (not triggered) is shown as hatched. Total = 47 + 21.4 = 68.4 (R4 Very High).

7.1 Single Address Scan

The single-address scan (Fig. 3a–3b) is the primary entry point for detailed analysis (producing output similar to the end-to-end example in Section 5). It accepts any 42-character Ethereum address and returns a multi-section report within 50 ms, illustrated here with a high-risk address (score 72/100, R5 Critical, flagged as SYBIL in 13 of 14 projects). The report has four components:

Overview panel (Fig. 3a, top): displays the composite sybil score (0–100) as a circular ring, the corresponding risk tier (R0–R6), the number of triggered indicators (n/5), and the number of projects the address participated in (with sybil/clean breakdown).

Cross-project radar chart (Fig. 3a, bottom): a five-axis radar chart overlays indicator profiles across all projects the address participated in. Each colored polygon represents one project; the dashed ring marks the detection threshold. Polygons protruding beyond the ring indicate triggered indicators. Sybil addresses typically show similar radar shapes across projects, while legitimate users exhibit varied profiles.

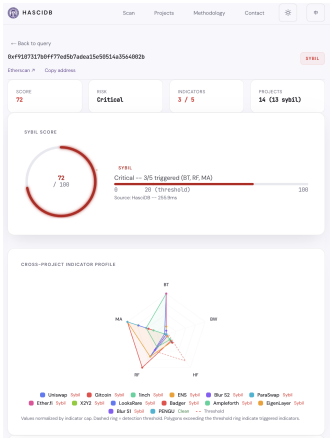
Indicator detail panel (Fig. 3b): groups the five indicators by detection axis (Operations Axis: BT, BW, HF; Fund-Flow Axis: RF, MA). For each indicator, the panel shows the raw value, a progress bar relative to the cap, and the Part B scoring contribution.

Project history table (Fig. 3c): lists all projects the address participated in, each row showing project name, protocol type, claim status, per-project sybil score, and classification (SYBIL/CLEAN), sorted by score descending.

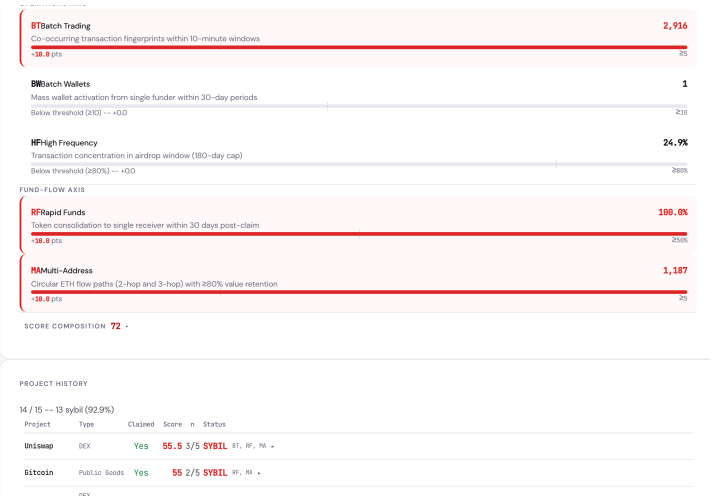
In this example, radar polygons protrude beyond the threshold ring on BT and RF axes in nearly every project, a hallmark of automated sybil operations. The indicator detail panel provides the quantitative evidence trail with dual-axis grouping.

7.2 Batch Scanning and Screening

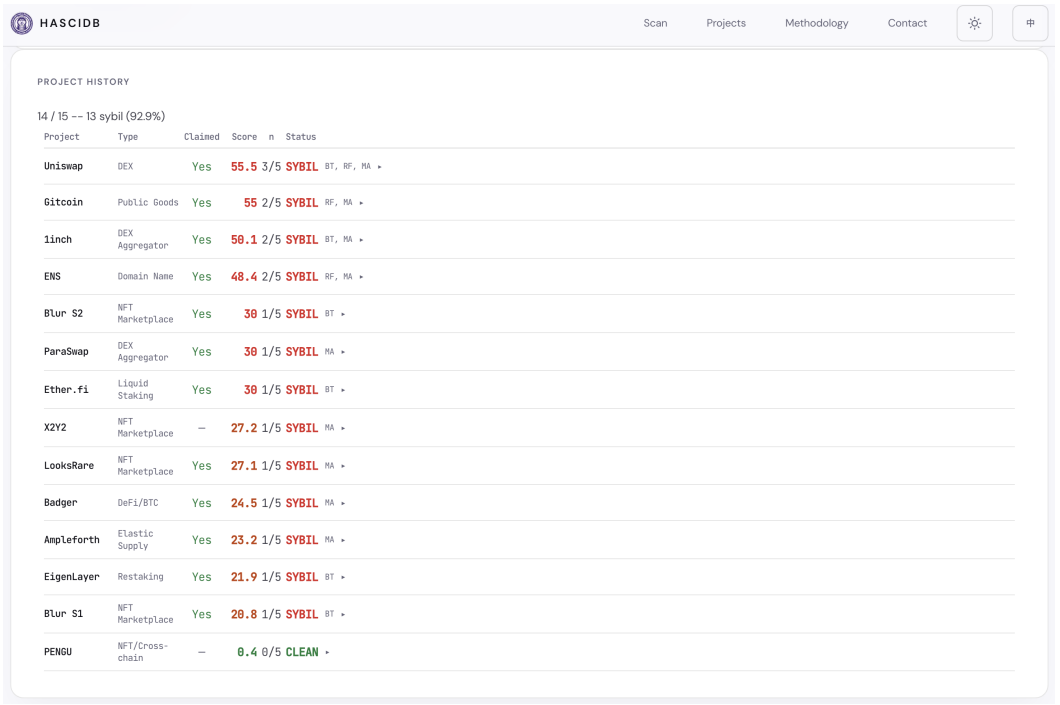
For project teams performing pre-distribution filtering, HasciDB supports batch scanning of up to 50,000 addresses per submission (sized to cover typical pre-distribution filtering batches while maintaining sub-12-second response times on the shared Supabase instance). Users can either paste newline-separated addresses into a text area or upload a CSV file. The batch endpoint processes addresses in parallel against the Supabase PostgreSQL backend. Results are returned as a risk-annotated dashboard with summary statistics, score distribution by risk tier, per-address filtering



(a) Overview: score ring (72/100, Critical) and radar chart across 14 projects.



(b) Indicator details: BT=2,916, RF=100%, MA=1,187.



(c) Project history table: 13 of 14 projects marked SYBIL (scores 21.9–55.5), one CLEAN.

Fig. 3. Single address scan: (a) overview panel with score ring (top) and five-axis radar chart (bottom); (b) indicator detail panel grouped by detection axis; (c) per-project history with scores, trigger counts, and classification status.

(All/Sybil/Clean tabs), and downloadable CSV exports. Typical batch latency: <3 seconds for 10,000 addresses, <12 seconds for 50,000 addresses.

The batch scan workflow is designed for a practical pre-distribution use case: a project team exports its candidate airdrop list, uploads it to HasciDB, and receives a risk-annotated version within seconds. The team can then apply custom filtering rules: for example, excluding all addresses with score ≥ 50 (R4+), manually reviewing addresses with scores 20–49 (R2–R3), and auto-approving addresses with score < 20 (R0–R1). This tiered response strategy leverages the continuous scoring to balance false-positive risk against sybil exposure, rather than applying a single binary cutoff.

A concrete example: suppose a project team preparing to distribute tokens uploads their 30,000 candidate addresses via batch scan. Within seconds, HasciDB returns a risk-annotated list. The team discovers that 4,200 addresses (14%) score above 50 (high risk) and decides to exclude them outright. Another 8,100 addresses (27%) score 20–49 (moderate risk) and are flagged for manual review by a three-person committee, who inspect radar charts and indicator breakdowns to make case-by-case decisions. The remaining 17,700 addresses (59%) score below 20 and are auto-approved. This graduated workflow, enabled by continuous scoring, replaces the crude binary approach of either blocking all suspicious addresses or ignoring the problem entirely. It also produces an auditable decision trail: the project can later justify each inclusion or exclusion by pointing to the specific indicator values and scores, reducing the risk of community backlash from perceived unfairness.

7.3 Project Listing and Methodology Reference

The query system includes a dedicated project listing page that displays all 16 covered projects with their metadata (protocol type, airdrop year, eligible/sybil counts, sybil rate), sortable by any column. In practice, the page functions as a compact comparison table: readers can immediately see which projects concentrate suspicious activity, which have higher claimant conversion, and how protocol category correlates with sybil prevalence. A companion methodology reference page provides interactive documentation of the five-indicator framework, serving as an embedded reference manual: any flagged address holder can trace the exact detection logic that produced their score.

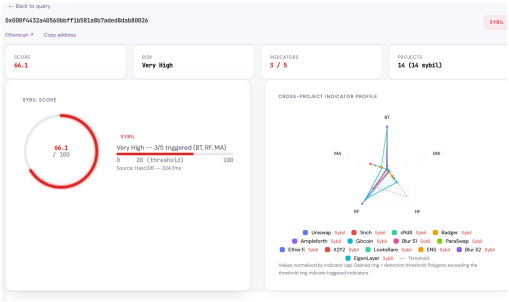
7.4 Sybil Overlap Network

The homepage features an interactive network visualization of cross-project sybil address overlap. Nodes represent projects, sized by sybil count and colored by protocol category; edges represent shared sybil addresses. In the live interface, the layout naturally clusters projects with high mutual overlap: a dense NFT cluster (X2Y2, LooksRare, Blur S1/S2), a DEX cluster (Uniswap, 1inch, dYdX), and peripheral positioning of EigenLayer/Ether.fi (2024, capital-staking eligibility). This text-only summary captures the same analytical takeaway as the omitted figure: overlap is not random, but structured by protocol similarity and user acquisition incentives. The interactive version remains available at https://hascidb.org.

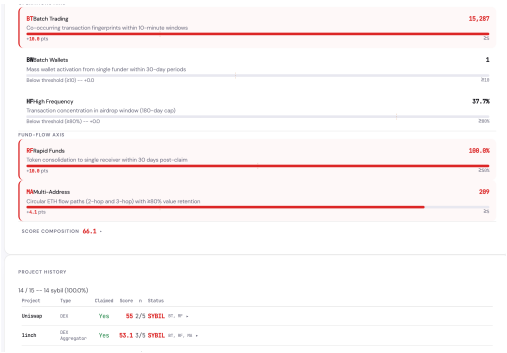
7.5 Preliminary Case Studies

Three preset cases (accessible via one-click buttons on the homepage) demonstrate the scoring engine's discriminative power across fundamentally different sybil strategies: a broad-spectrum operator (Case A), a targeted operator (Case B), and a genuinely active clean user (Case C).

7.5.1 Case A: Cross-Project Sybil ("Broad-Spectrum" Wide-Net). This address (Fig. 4a) triggers BT=15,287, RF=100%, and MA=289 while BW and HF remain below thresholds, demonstrating selective evasion of certain detection dimensions. The 14 radar polygons show remarkably consistent



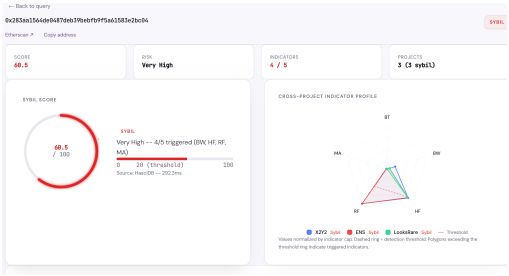
(a) Case A overview: score 66.1, 3/5 triggered.



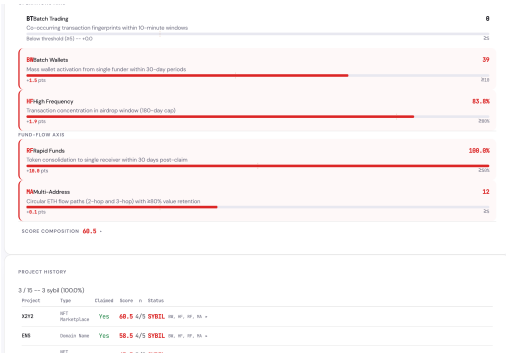
(b) Case A detail: BT=15,287, RF=100%.

Fig. 4. Case A (broad-spectrum sybil): overview with radar chart (left) and indicator breakdown (right).

shapes (BT and RF axes protruding in every project), reflecting a rigid automated strategy applied uniformly across protocols. This "wide-net" archetype prioritizes volume over stealth.



(a) Case B overview: score 60.5, 4/5 triggered.

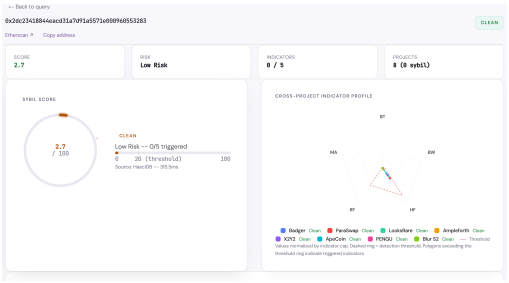


(b) Case B detail: BW=39, HF=83.8%.

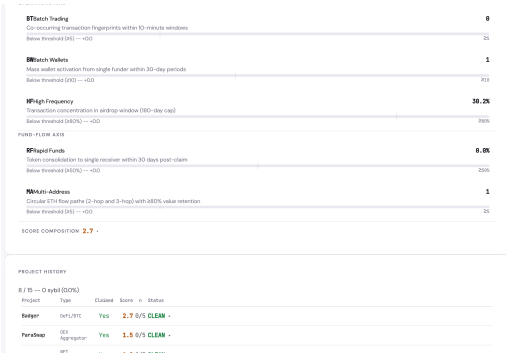
Fig. 5. Case B (targeted sybil): overview with radar chart (left) and indicator breakdown (right).

7.5.2 Case B: Extreme Sybil ("Targeted" Multi-Dimensional Anomaly). Case B (Fig. 5a): despite a comparable score (60.5 vs. 66.1), the behavioral profile is fundamentally different: BW=39 (wallet-farm creation), HF=83.8% (airdrop-only address), RF=100%, but critically BT=0, avoiding the BT detector entirely. This contrast demonstrates strategy-agnosticism: Case A's "wide-net" approach and Case B's "targeted" approach receive comparable scores through entirely different indicator combinations.

7.5.3 Case C: High-Activity Clean Address (Control). This control address (Fig. 6a) participates in eight projects across DEX, NFT, and DeFi categories, yet triggers 0/5 indicators (score 2.7). All values reflect normal usage: BT=0, BW=1, HF=30.2%, RF=0%, MA=1. The score of 2.7 derives entirely from the approach mechanism (a small bonus given to accounts whose indicators are close to but below the flagging threshold, as an early warning), proving that participation breadth alone does not trigger sybil flagging.



(a) Case C overview: score 2.7, 0/5 triggered.



(b) Case C detail: all below thresholds.

Fig. 6. Case C (clean control): overview with radar chart (left) and indicator breakdown (right).

8 Evaluation

8.1 Dataset Scale and Indicator Distribution

This section examines what the database reveals. As a reminder, the five indicators are: BT (co-ordinated batch transactions), BW (shared wallet funding), HF (activity concentrated in airdrop windows), RF (rapid token funneling to a single address), and MA (circular money flows). How widespread is suspicious behavior? Does it look the same across different types of projects? And does the scoring system actually distinguish different levels of risk?

HasciDB contains 2,539,857 eligible address records, of which 964,034 are flagged as sybil (38.0%). After deduplication, there are 1,807,415 unique addresses. Sybil rates vary significantly across projects (Fig. 7): highest at 1inch (67.3%, among projects with known eligible sets), lowest at PENGU (5.5%). The variation across projects reflects protocol type and airdrop design; three representative cases illustrate the pattern:

High: 1inch (67.3%). The highest sybil rate in the dataset stems from 1inch's permissive eligibility design: any address that performed a single swap on the 1inch aggregator qualified. This near-zero barrier enabled automated scripts to register thousands of addresses with minimal cost (estimated <$1 per address in gas costs). The BT indicator dominates at 83%, consistent with batch-executed swaps, a pattern where a script rapidly sends near-identical transactions from multiple addresses within tight time windows. The combination of low entry cost and high per-address reward (~$600 at claim price) created an economically rational sybil opportunity.

Medium: ENS (43.7%). The Ethereum Name Service required holding a .eth domain ($5–$50/year depending on name length), creating a moderate capital barrier. This barrier shifted the dominant sybil strategy from batch transactions to token funneling: RF triggers at 73.5% among flagged addresses, indicating that sybil operators registered cheap .eth domains across many addresses, then consolidated claimed ENS tokens to a single collection address within 30 days. The capital requirement was sufficient to deter the lowest-cost scripted farming (unlike 1inch) but insufficient to prevent operators willing to invest $50–$500 in domain registrations.

Low: EigenLayer (9.8%). The lowest sybil rate reflects EigenLayer's capital-intensive eligibility: users needed to stake ETH (minimum ~0.01 ETH but allocation was proportional to stake amount). Capital-weighted allocation makes address splitting counterproductive: distributing 10 ETH across 10 addresses yields 10 small allocations that sum to less than a single 10 ETH stake's allocation, because the allocation formula rewards larger individual stakes. The 9.8% flagging rate primarily

reflects operators who staked meaningful amounts across a small number of addresses rather than broad-scale farming.

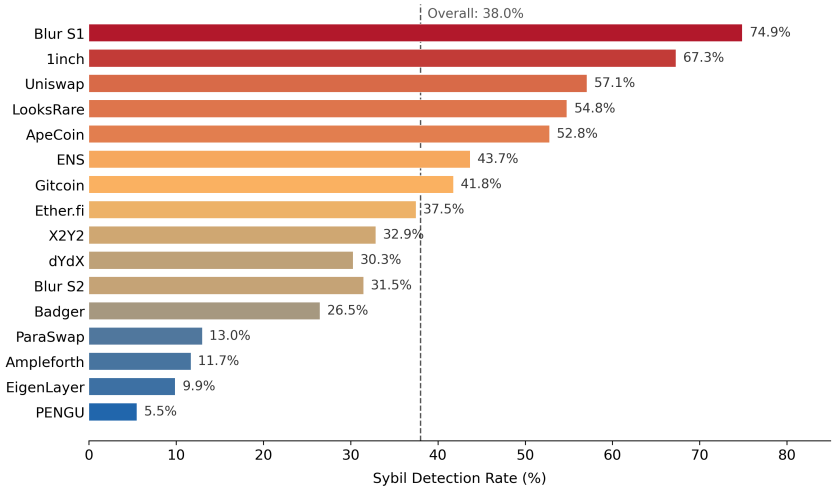


Fig. 7. Sybil detection rates across 16 projects (eligible-based), sorted by rate descending. Rates for Blur S1/S2 and Badger are computed on approximated eligible sets (see Table 1). The vertical dashed line marks the aggregate rate (38.0%). Caution: cross-project rate comparisons are affected by eligible-set construction differences; see text for claimant-based rates.

These three cases illustrate a general pattern: the easier it is to create qualifying accounts, the higher the sybil rate. When qualification requires only a single transaction (1inch), automation makes sybil farming nearly free. When it requires holding an asset over time (ENS), the cost rises enough to shift the dominant strategy from mass account creation to token consolidation. When rewards are proportional to capital invested (EigenLayer), splitting funds across multiple accounts is actually counterproductive, making sybil attacks economically irrational. This gradient from activity-based to capital-based allocation design is the single most actionable finding for projects designing future airdrops: by tying rewards to sustained capital commitment rather than discrete interactions, projects can structurally discourage sybil behavior without relying on post-hoc detection.

Comparability caveat: eligible vs. claimant sybil rates. The sybil rates above are computed over each project's *eligible* address set, but eligible sets are constructed differently across projects (Table 1 footnotes). Projects with official Merkle trees (Uniswap, ENS) have precise eligible sets, while projects with broad reconstruction rules (X2Y2: all OpenSea users with no volume threshold) include many addresses that may never have been truly eligible. This inflates the denominator and dilutes the sybil rate: X2Y2's eligible-based rate is 32.9%, but among addresses that actually claimed tokens it rises to 93.1%. To verify that the mechanism-design finding is not an artifact of eligible-set construction, we recompute sybil rates using only *claimants* (addresses that collected tokens on-chain) as the denominator. The pattern holds and strengthens: 1inch claimant sybil rate is 83.1% (activity-based), ENS is 53.6% (asset-based), and EigenLayer is 8.9% (capital-based). The activity→capital gradient persists regardless of denominator choice, confirming it reflects genuine economic incentives rather than eligible-set artifacts. We recommend that future cross-project comparisons use claimant-based rates for apples-to-apples analysis, reserving eligible-based rates for within-project detection coverage assessment.

8.1.1 Score Distribution. Among the 964,034 flagged addresses, the vast majority fall in the lower score tiers (Table 5). Recall the seven risk tiers from Table 4: R0 (score 0, clean) through R6 (score 90+, extreme). Most flagged addresses fall in the lower tiers:

- **R2 (20–29, 41.4%) and R3 (30–49, 50.3%)** together account for 91.7%. Most sybil addresses trigger only one or two indicators: one triggered indicator gives a base score of 20 and two gives 35; since most barely exceed their thresholds, total scores mostly fall in the 20–49 range.
- **R4+ (50 and above) accounts for 8.3%**, indicating that addresses triggering three or more indicators with significant excess are uncommon. Only 1,244 addresses (0.1%) reach R5 (70+), and R6 (90+) is empty because scoring 90 requires both near-maximum Part A ($n \geq 4$) and high Part B, which is extremely rare in practice.
- **R1 (scores 1–19) contains 616,923 clean addresses** (24.3% of all records) whose indicator values fall below triggering thresholds but are nonzero. The approach-score function awards up to 4 points per indicator for below-threshold values, capped at 19.

Table 5. Score distribution across risk tiers

Score	Tier	Count	%
0	R0	958,640	37.7[a]
1–19	R1	616,923	24.3[a]
20–29	R2	399,218	41.4[b]
30–49	R3	484,793	50.3[b]
50–69	R4	79,039	8.2[b]
70–89	R5	1,244	0.1[b]
90–100	R6	0	0

[a]Of all 2,539,857 records.
[b]Of 964,034 flagged.

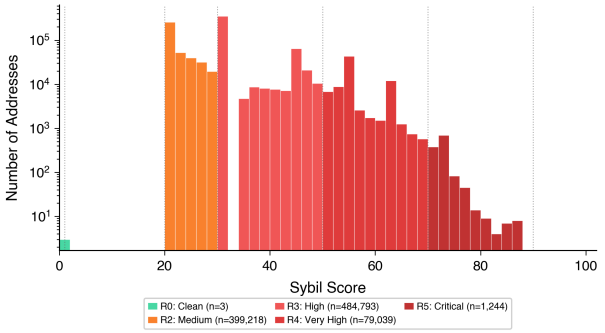


Fig. 8. Score distribution histogram (log-scale y-axis). Bimodal peaks at 20 and 35 correspond to n=1 and n=2 triggered indicators. R2+R3 account for 91.7%; R5 is rare (0.1%); R6 is empty.

The two prominent peaks at scores 20 and 35 in Fig. 8 correspond to Part A base scores for n=1 and n=2 triggered indicators. The practical implication is that the scoring system stratifies the flagged population into actionable tiers. The R2 tier (41.4%) represents borderline cases suited for manual review; R3 (50.3%) shows stronger evidence; R4+ (8.3%) represents highest-confidence detections with converging multi-dimensional evidence.

8.2 What Happens Under Stricter Rules?

To evaluate the sensitivity boundaries of the OR-trigger rule, we compare five classification strategies (Table 6): OR rule (any indicator triggers classification), AND rule (all axes must trigger simultaneously, meaning at least one behavioral indicator AND at least one money-flow indicator must both be suspicious), and k-of-5 rules (requiring at least k indicators to be suspicious before flagging).

The OR rule detects 4.5× more than $k \geq 2$, and **7.2× more than the AND rule**. This gap primarily stems from 78.0% of sybils (751,883 / 964,034) being triggered by a single indicator only.

Implication: The 78.0% single-indicator majority is a deliberate consequence of the OR rule's high-recall design. Users requiring higher confidence should filter to $n \geq 2$ (212,151 addresses, 22.0%) or use continuous scores.

Table 6. Classification strategy ablation

Classification Rule	Sybils	Rate	vs. OR
OR (current rule)	964,034	38.0%	1.00×
$k \geq 2$	212,151	8.4%	0.22×
AND (Ops AND Fund)	134,447	5.3%	0.14×
$k \geq 3$	28,881	1.1%	0.03×
$k \geq 4$	831	0.03%	<0.01×
$k = 5$	9	~0%	~0×

The steady drop from 964,034 (OR) to 9 (k=5) addresses demonstrates that the 38.0% rate is not an artifact of a few dominant indicators, but the aggregate of five largely independent detection dimensions. Comparing the AND rule to the $k \geq 2$ rule: two-thirds of addresses with multiple suspicious indicators are suspicious in *both* dimensions (behavior and money flow). The other third are suspicious in only one dimension (e.g., two behavioral indicators but no money-flow anomaly). Researchers can select their desired operating point along this spectrum based on false-positive versus false-negative tolerance.

8.3 Cross-Project Analysis

This subsection leverages HasciDB's unique multi-project coverage to analyze sybil behavior across project boundaries. By tracking the same address across 16 projects, we can observe how operators distribute their activity and whether behavioral patterns persist or adapt across different project types.

8.3.1 Participation Breadth and Sybil Probability. Cross-project data reveals a clear upward trend: the more projects an address participates in, the more likely it is to be flagged (Fig. 9). Table 7 quantifies this relationship across five participation tiers (bins reflect the empirical distribution: 77.3% of addresses participate in a single project, so higher bins are progressively aggregated to maintain statistically meaningful sample sizes).

Table 7. Cross-project participation breadth and sybil rate

Projects Participated	Addresses	Sybils	Rate
1	1,333,877	374,086	28.0%
2	327,695	186,013	56.8%
3–4	118,175	99,689	84.4%
5–7	25,135	24,628	98.0%
8+	2,533	2,531	**99.9%**
Total	1,807,415	686,893[a]	38.0%

[a]Deduplicated count of unique addresses flagged as sybil at least once.

Key observation: 99.9% of 2,533 addresses in 8+ projects are flagged at least once (9+: 100%). This extreme ratio is partly attributable to OR-rule statistical accumulation (each additional project provides five more independent threshold tests). Under a simple null model assuming independent per-indicator trigger probabilities equal to the observed marginals, the expected flagging rate for 8-project addresses would be ~85%[2], suggesting the observed 99.9% exceeds the null by ~15

[2]Null model: if the per-project, per-indicator trigger probability is p_k (observed marginal), the probability of *not* being flagged in any of m projects is $\prod_{j=1}^{m} \prod_{k=1}^{5} (1 - p_k)$. With m=8 and observed marginals, this yields ~15% clean probability, i.e., ~85% flagged.

percentage points. In other words, even purely random indicators would flag about 85% of addresses in 8+ projects just by chance. The observed 99.9% is 15 percentage points higher, suggesting a real behavioral signal on top of the statistical accumulation.

Stepwise pattern: The flagging rate rises from single-project (28.0%) to two-project (56.8%), then accelerates through 84.4% (3–4 projects) to 98%+ (5+ projects). This suggests two distinct populations: "targeted" operators (1–2 projects, selective) and "broad-spectrum" operators (5+, automated). However, the OR rule's cumulative nature means each additional project provides five more threshold tests (an address in 16 projects faces $5 \times 16 = 80$ independent tests), so some statistical accumulation is expected even for moderately active legitimate users. Among addresses participating in 2+ projects, 43.8% show mixed status (flagged in some projects, clean in others), with BT and RF having the highest cross-project flip rates (53% and 52% of addresses change from triggered to non-triggered or vice versa across projects).

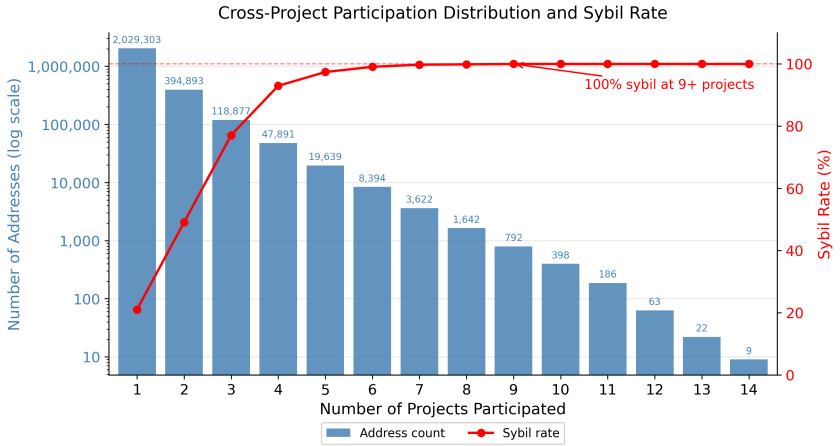


Fig. 9. Cross-project participation distribution (dual-axis). Blue bars (left axis, log scale): address count; red line (right axis): sybil rate. Addresses in 8+ projects are universally flagged. Two populations emerge: "targeted" (1–2 projects, <50%) and "broad-spectrum" (5+, >98%).

9 Discussion

Scope. The analyses below *characterize* behavioral patterns in the dataset. They do not validate detection accuracy. Without independently verified ground truth, precision and recall remain outside this work's scope. The 38.0% aggregate flagging rate falls within the 20–40% range commonly cited by commercial analytics platforms [17, 20]. The stepwise increase with participation breadth (28% single-project to 100% for 9+) implies a genuine behavioral signal rather than random noise.

9.1 Toward Ground Truth Construction

HasciDB currently operates as a *behavioral characterization dataset*: the 38.0% flagging rate reflects detection by behavioral indicators, not independently verified sybil confirmation. Building toward a validated ground truth is a primary goal for the next phase of this project.

Three complementary paths are planned: (1) integrating counter-evidence indicators from [12] (account age, activity history, identity credentials) to identify and reclassify likely false positives; (2) using the $n \geq 2$ multi-indicator subset (22.0% of flagged addresses) as a higher-confidence seed, since multiple independent anomalies are unlikely to co-occur by chance in legitimate users; and

(3) expert-labeling a stratified random sample of flagged and clean addresses to directly estimate precision and recall.

HasciDB ensures reproducibility at three levels: **data-level** (SHA-256 checksums for byte-level verification), **pipeline-level** (open-source with declarative YAML configurations), and **result-level** (deterministic scoring from published indicator values). The exchange filter list (173,597 addresses, March 2026 snapshot) will be updated semi-annually with versioned releases.

9.2 Known Limitations and Ethical Considerations

Dataset-level: (1) L1 only (no L2/cross-chain), though four indicators (BT/BW/HF/MA) are chain-agnostic; (2) no verified ground truth, so precision/recall cannot be computed directly; (3) PENGU cross-chain mapping unavailable; (4) exchange filter (173,597 addresses, March 2026) requires periodic updates.

Scoring engine: The 50/50 weight split is the simplest design satisfying score $\geq 20 \Leftrightarrow$ is_sybil $=$ 1; 93.5% of addresses retain the same tier under 60/40 or 70/30 alternatives. RF can only be computed after tokens are distributed, making it unsuitable for pre-distribution prediction.

Framework-inherited [12]: (1) Four counter-evidence indicators (account age, activity history, identity credentials, NFT holdings) are not yet integrated; without them, active legitimate users may be misflagged. (2) BW/MA may mis-associate innocent fund recipients with sybil clusters. (3) RF fails on intermediate-token funneling and Solana's account model.

Ethical considerations. HasciDB publishes only address-level aggregates (no personal identifiers or off-chain data). Mislabeling risk is mitigated by: (1) continuous scoring distinguishing high-confidence (R4+) from borderline (R2) cases; (2) full detection logic transparency enabling evidence-based rebuttals; (3) open-source data allowing stricter thresholds or counter-evidence integration. Labels should be treated as risk signals requiring human review, not definitive determinations for automated enforcement.

10 Conclusion and Future Work

HasciDB is the first open-source database that applies a unified sybil detection framework across 16 cryptocurrency airdrop projects and 2.54 million addresses. Unlike prior work that studies one project at a time, HasciDB makes cross-project comparison possible, revealing patterns that no single-project analysis can see. As airdrops distribute billions of dollars in tokens each year and sybil manipulation erodes their fairness, the field needs exactly this kind of shared, reproducible infrastructure.

The single most important finding is that how a project designs its airdrop matters far more than how sophisticated its detection is. Projects that reward interaction count exhibit sybil rates above 67%, while those that require staked capital see rates below 10%. This order-of-magnitude gap, observable only because HasciDB spans 16 projects under identical criteria, suggests that mechanism design is a stronger defense than any post-hoc filter.

The next step is validating HasciDB's labels through expert-labeled samples that enable direct measurement of precision and recall, establishing HasciDB as a verified ground-truth benchmark. Beyond that, expanding coverage to Layer-2 and non-Ethereum chains will test the framework's generality, and releasing HasciDB as a shared benchmark will let the community compare detection methods (graph-based, gradient-boosted, and hybrid) on common ground. The dataset is open-sourced at https://hascidb.org. We envision HasciDB as a living benchmark that grows with the ecosystem, enabling the shift from adversarial detection to sybil-resistant mechanism design.

References

[1] Arbitrum Foundation. 2023. Sybil Detection for Arbitrum Airdrop. GitHub repository. https://github.com/

ArbitrumFoundation/sybil-detection

[2] Ting Chen, Zihao Li, Yuxiao Zhu, Jiachi Chen, Xiapu Luo, John C.-S. Lui, Xiaodong Lin, and Xiaosong Zhang. 2020. Understanding Ethereum via Graph Analysis. *ACM Transactions on Internet Technology* 20, 2 (2020), 1–32.

[3] Norman Dalkey and Olaf Helmer. 1963. An Experimental Application of the Delphi Method to the Use of Experts. *Management Science* 9, 3 (1963), 458–467. doi:10.1287/mnsc.9.3.458

[4] John R. Douceur. 2002. The Sybil Attack. In *International Workshop on Peer-to-Peer Systems (IPTPS)*. Springer, 251–260. doi:10.1007/3-540-45748-8_24

[5] Dune Analytics. 2024. Community-Curated Blockchain Labels and Dashboards. https://dune.com. 173,597 infrastructure addresses used in this work; snapshot March 2026.

[6] 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 (CHI '23)*. ACM, New York, NY, USA, Article 551. doi:10.1145/3544548.3580766

[7] Hop Protocol. 2022. Hop Airdrop Distribution. GitHub repository. https://github.com/hop-protocol/hop-airdrop

[8] Human Technology. 2024. Human Passport: Proof of Personhood and Sybil Resistance for Web3. https://human.tech. Acquired from Gitcoin Passport, protecting $430M+ across 120+ projects.

[9] Scott Imig, Thuat Paven Do, Duc Phan Thanh, and Long Ta. 2025. XGBoost for Sybil Detection on Public Blockchains: A Case Study of LayerZero Interactors. *SSRN preprint* (2025).

[10] Harry Kalodner, Malte Möser, Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, and Arvind Narayanan. 2020. BlockSci: Design and Applications of a Blockchain Analysis Platform. In *Proceedings of USENIX Security '20*. USENIX, 2721–2738.

[11] LayerZero Labs. 2024. Addressing Sybil Activity. Medium. https://medium.com/layerzero-official/addressing-sybil-activity

[12] Chunyang Li, Hongzhou Chen, and Wei Cai. 2026. From Slang to Standards: Consensus-Driven Airdrop Hunter Definition as a Baseline for Cryptocurrency Ecosystem Security and Governance. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (Barcelona, Spain) (CHI '26)*. ACM, New York, NY, USA. doi:10.1145/3772318.3790777 Accepted at CHI'26.

[13] Qiangqiang Liu, Qian Huang, Frank Fan, Haishan Wu, and Xueyan Tang. 2025. Detecting Sybil Addresses in Blockchain Airdrops: A Subgraph-based Feature Propagation and Fusion Approach. In *Proceedings of IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, Pisa, Italy.

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

[15] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. 2013. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In *Proceedings of the ACM Internet Measurement Conference (IMC)*. ACM, 127–139.

[16] Johnnatan Messias, Vabson Oliveira, and Mateus Vasconcelos. 2023. Airdrops: Giving Money Away Is Harder Than It Seems. *arXiv preprint arXiv:2312.02752* (2023).

[17] Nansen. 2024. On-Chain Analytics Platform. https://www.nansen.ai. Accessed: March 2026.

[18] Yuyang Qin, Tengfei Ma, Hongzhou Chen, and Haihan Duan. 2024. ARTEMIX: A Community-Boosting-Based Framework for Airdrop Hunter Detection in the Web3 Community. *Blockchain* 2, 2 (2024), 0010. doi:10.55092/blockchain20240010

[19] Dorit Ron and Adi Shamir. 2013. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography and Data Security (FC '13)*. Springer, 6–24.

[20] TrustaLabs. 2024. On-Chain Trust and Safety. https://trustalabs.ai. Accessed: March 2026.

[21] Worldcoin Foundation. 2025. World ID: Privacy-Preserving Proof of Personhood. https://whitepaper.world.org. 16M+ verified users, thousands of Orbs in 40+ countries (as of 2025).

[22] Jiajing Wu, Kaixin Lin, Dan Lin, Bozhao Zhang, Zhiying Wu, and Jianzhong Su. 2025. Safeguarding Blockchain Ecosystem: Understanding and Detecting Attack Transactions on Cross-chain Bridges. In *Proceedings of the ACM Web Conference 2025 (WWW '25)*. ACM.

[23] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. 2008. SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks. In *Proceedings of IEEE S&P '08*. IEEE, 3–17.

[24] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. 2006. SybilGuard: Defending Against Sybil Attacks via Social Networks. In *Proceedings of ACM SIGCOMM '06*. ACM, 267–278.

[25] Hao Zhang, Yunfei Li, Tianqing Zhu, Changzhou Han, Sheng Wen, and Yang Xiang. 2025. A Scalable and Semi-supervised Approach for Ethereum Fraud Detection Using Graph Neural Networks and K-Nearest Neighbors Clustering. *Journal of Information Security and Applications* 95 (2025), 104255. doi:10.1016/j.jisa.2025.104255

[26] 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)*. ACM, 1824–1834.