WIRE-Wallet: Web3 Integrated Reputation Engine for Ethereum Wallets

1st Kanishk Chhabra
SUCCESS Lab (UIN: 436000700)
Texas A&M University
College Station, TX, USA
kanishkchhabra23@gmail.com

2nd Guofei Gu
SUCCESS Lab
Texas A&M University
College Station, TX, USA
guofei@cse.tamu.edu

***Abstract*—Ethereum wallets form the backbone of Web3 interactions, enabling users to send transactions, deploy contracts, and engage with decentralized applications (DApps). However, the openness of blockchain networks has made them fertile ground for malicious wallet activity, ranging from phishing and airdrop farming to MEV arbitrages and sybil attacks. Taking inspiration from trust scoring algorithms in traditional finance, we introduce WIRE-Wallet, a light and interpretable trust scoring system for Ethereum wallets. Unlike earlier approaches based on third-party APIs or off-chain labels, WIRE-Wallet deduces wallet activity from on-chain transactions directly via the Etherscan API. It deduces a rich feature set — including transaction diversity, counterparty entropy, approval patterns, gas usage, circular transfers, and contract interactions — to characterize wallet activity. We tag wallets with persistent trust scores (0–100) by supervised machine learning models, allowing downstream applications to decide safety and identify anomalies. Our evaluation on 60,000+ tagged wallets shows outstanding model performance (ROC-AUC ¿ 0.98), with understandable feature importance and robust decision boundaries. WIRE-Wallet offers a foundation for explainable, scalable, and metadata-free wallet-level risk determination in decentralized settings.**

***Index Terms*—blockchain, ethereum, web3 reputation, wallet trust score, decentralized security**

I. Introduction

The rapid growth in blockchain technologies transformed the internet as a decentralized space, also called Web3. In all of the various blockchain platforms, Ethereum has emerged as one of the leading ecosystems comprising millions of user wallets, decentralized applications (DApps), smart contracts, and user wallets. Wallets constitute a major doorway in this space through which basic operations like transfers and receptions of tokens, signatures of transactions, interactions with contracts, and identities are managed in digital form.

Despite the open and trustless blockchain environment, illegal activities continue at the wallet level.Monthly attackers utilize wallet addresses for phishing [1], rug pulls [2], MEV (Miner Extractable Value) strategies [3], sybil attacks [4], and other money laundering operations. These illegal activities not only decrease user trust but also have multi-million dollar losses in DeFi, NFT, and DAO domains [5]. Existing defenses are primarily targeted at finding malicious smart contracts [6] or compromised DApps [7], with wallet-based risk assessment yet to be explored.

Moreover, existing approaches rely on costly third-party APIs, hand-curated blacklists, or explainability-lacking and scalability-challenged behavior clustering heuristics. Inspired by the FICO score model in traditional finance, we present WIRE-Wallet, a Web3 Integrated Reputation Engine tailored for Ethereum wallets. WIRE-Wallet takes a transactional approach towards trust scoring based on the entire on-chain transaction history of a wallet, without relying on external APIs, smart contract bytecode, or off-chain metadata.

WIRE-Wallet extracts behavioral traits from transaction patterns like transaction volumes, interaction diversity, token approvals, circular transfers, gas price anomalies, NFT interactions, and bursts within time periods. These are used to train explainable machine learning models which provide a 0-100 continuous trust score reflecting the overall safety and legitimacy of a wallet. In contrast to previous models designed for DApps or contracts [7] [8], our approach directly addresses end-user wallets, thereby being suitable for DeFi platforms, marketplaces, and wallet providers that require risk-aware integrations.

Our contributions of this work are:

- We introduce a light, interpretable, metadata-less trust scoring engine for Ethereum wallets.
- Construct a dataset of 35,000+ labeled wallets from on-chain transaction data and sparse external labels.
- Enforce 30+ transaction-based attributes like entropy, frequency, approval behavior, and contract interactions.
- Compare a few models (Random Forest, Gradient Boosting, MLP) and achieve over 98% ROC-AUC in wallet classification.
- Demonstrate the ability to provide trust scores for unseen wallets solely from transaction behavior.

II. Background and Motivation

A. Wallets, Transactions and User Interactions

Wallets. In Ethereum and other blockchain networks, wallets present the main interface for users to interact with the decentralized environment. Generally, wallets exist in two types: externally owned accounts (EOAs), which are controlled by private keys, and contract accounts, which are governed by smart contract code. EOAs are widely used by individuals and institutions to store digital assets, approve token transfers, and interact with decentralized applications (DApps). Each wallet has a public address and can receive and send Ether or tokens without revealing user identity, which enhances pseudonymity but also increases the risk of abuse. The history of behavior of a wallet, fully represented on the chain, can provide an indication of its age, activity levels, and security posture [9].

User Interactions. Wallets do not exist in isolation; their behavioral imprint is established by interacting with other users, DApps, DeFi protocols, NFT marketplaces, or token contracts. These interactions can be in the form of sending/receiving ETH, approving tokens for spending, or invoking smart contract methods. The diversity, frequency, and context of these interactions are all robust predictors of a wallet's intent and legitimacy. For instance, wallets that only interact with known mixers or flash loan contracts can be considered anomalous [12]. Conversely, high counterparty diversity, usage consistency across time, and active participation in popular DApps can all be indicative of trustworthiness. Such patterns are crucial to grasp for constructing an interpretable and meaningful reputation score for wallets — a concept that we intend to develop further in WIRE-Wallet.

Transactions. An Ethereum transaction is a signed data payload triggering a change in blockchain state. It may be sending ETH, calling a smart contract, approving tokens, or deploying new contracts. A transaction has several crucial fields that collectively specify its intention and cost. These are:

- Receiver and Sender Addresses: Source and destination identifiers.
- Nonce: Order counter to prevent transaction order and block replay attacks.
- Gas Price and Gas Limit: Determine the fee for the transaction and resource limit the sender is willing to pay.
- Gas Used: Amount of gas actually used for the execution.
- Input Data: Formatted contract function and arguments (if any).
- Contract Address (if created): Completed for contract deployment transactions.
- Timestamp: The timestamp when the transaction was added to a block. Timestamps are required to perform behavioral analysis. For example, excessive gas prices, persistent contract calls, or anomalous bursts of transactions may indicate exploitative bot behavior or money laundering activity patterns [10], [11].

B. Malicious Behaviors in Wallets and Accounts

Although blockchain guarantees data transparency and immutability, it certainly does not preclude malicious use by users or wallets. Misbehavior in Ethereum wallets comes in the most sophisticated, impact-heavy but ostensibly innocuous behaviors such as phishing, laundering, airdrop farming, sybil attacks, and transactions within fraudulent or hacked smart contracts. Such behaviors not only target individual users independently, but also disrupt the integrity of decentralized ecosystems systemically through manipulation of reputation scores or draining of liquidity.

One of the most common malicious activities is phishing wallets, where they impersonate official applications or tokens to deceive users into signing illegal transactions. They will usually sleep until triggered by an incoming transfer and then drain funds immediately through scripted routines. Another type includes bot wallets: computer-controlled accounts that execute high-frequency trades, fronrunning, and gas sniping strategies to exploit latency and block confirmation time [13]. These wallets can be identified by anomalies such as excessively stable gas prices, abnormally high transaction throughput rates, or routine transaction spikes against constrained counter parties.

More advanced attacks are based on money laundering schemes that involve circular transactions and self-transfers to mask the source of funds. Such wallets can transfer ETH or tokens through a sequence of intermediate accounts, occasionally back to the original source or with privacy-protecting smart contracts such as Tornado Cash [14]. However, Sybil wallets are used to masquerade as a multitude of users on behalf of one or multiple actors, artificially inflating the reputation of DApps or manipulating decentralized governance protocols [15].

Airdrop hunting is an emerging strategy too, whereby consumers create or purchase large numbers of low-activity wallets to be able to claim incentive token airdrops. Such behavior creates skewed network measures and undermines the validity of regular community engagement [**?**]. In trust scores, identification of such wallet behaviors is critical so as not to inflate reputation improperly and reduce the risk of the system in the broader Ethereum space.

C. Limitations of Prior Work

Several projects and tools have emerged in recent years aiming to assess wallet safety or reputation in decentralized environments. These include commercial platforms like Webacy and research-inspired systems such as the Blockchain Bureau, as well as other academic proposals that analyze user behaviors or assign reputation scores. However, despite their innovation, these tools face key limitations in terms of methodology, scalability, transparency, and adaptability.

Webacy, for instance, offers wallet safety ratings based on criteria like recent activity, wallet connectivity, and usage patterns. However, its scoring mechanism is largely opaque, proprietary, and lacks detailed feature-level insights for each wallet. It functions as more of a trust-signal layer than a robust behavioral analysis engine. Moreover, it doesn't open-source its scoring logic, making reproducibility and academic validation challenging [17].

Similarly, Blockchain Bureau focuses on ranking wallets using basic metrics like transaction count, average value, and tag-based classifications (e.g., miner, airdrop hunter, whale). Although intuitive, this approach falls short in detecting nuanced malicious behaviors such as circular laundering or contract exploit patterns. It also treats wallets as static entities rather than dynamic participants in evolving interaction networks, missing temporal variations in behavior [18].

More broadly, existing research efforts on reputation systems—especially those adapted from traditional finance or federated identity—tend to rely heavily on historical tags (e.g., scam labels) or external metadata (e.g., ENS names, contract labels). These systems lack adaptability to new or low-activity wallets and struggle to evaluate unlabeled or cold-start accounts [19]. Furthermore, some scoring models use binary classifications (e.g., scam vs. non-scam) without capturing the nuances needed for a continuous trust spectrum, limiting their applicability to fine-grained risk management in Web3 environments.

Finally, most prior tools and models are limited in explainability and user accessibility. They often fail to offer clear breakdowns of how specific wallet behaviors influence the final score. This lack of interpretability poses challenges for adoption by everyday users and developers seeking actionable insights for interaction safety.

These gaps motivate the design of WIRE-Wallet, which aims to build a transparent, extensible, and behavior-driven trust engine that generalizes across wallet types and does not rely on paid APIs or proprietary data sources. By focusing on on-chain transaction histories alone, we remove dependencies on third-party heuristics, thus enabling reproducible and decentralized wallet evaluation.

D. Motivating Case

An existing case that depicts the need for wallet-level trust scores is the July 2023 Multichain attack when approximately $126M was drained from different chains via administratively breached controls. Although the initial attack targeted smart contracts, secondary attacks involved attacker-controlled wallets trading with victims, laundering the money in consecutive self-directed hops before landing on CEX accounts [20]. Traditional DApp-based reputation systems did not recognize or mark these wallets in time since their behavior looked like innocent user activity on the surface.

In another case, the Inferno Drainer-as-a-Service campaign rapidly onboarded phishing victims' funds with stealthy-like activities—speaking to known contracts, mimicking legitimate behavior, and staying clear of usual blacklists [21]. Such criminal wallets evaded early detection as they lacked behavior-based scoring, indicating the shortfalls of binary tagging or metadata-only approaches.

These scenarios bring to the fore the need for an always-on, behavior-driven reputation system that has the capability to follow wallet patterns along flows of transactions, token interactions, and asset transfers—even on brand-new or unlabelled addresses. WIRE-Wallet strives to close the gap through interpretable, score-based trust estimates that dynamically change with shifting wallet behavior.

III. Related Work

The area of blockchain security has seen much research in contract-level and DApp reputation systems, but wallet-level trust analysis is relatively less explored. Various past systems and research have attempted to address Web3 reputation and security, with much of the effort being placed on smart contract behavior or project metadata.

A. DApp and Contract Reputation

The original WIRE system introduced a behavior-based reputation engine for smart contracts and decentralized applications (DApps), evaluating deployed contracts based on bytecode similarity, transaction behavior, and on-chain activity without relying on source code availability [22]. Other platforms like CertiK, Forta, and DeFiSafety perform audits and monitoring of smart contracts, but these systems remain focused on DApps or protocols, leaving individual user accounts unmonitored [23], [24].

B. Wallet Blacklisting and Threat Intelligence

Most recent anti-phishing and scam-detection tools such as ScamSniffer, CryptoScamDB, and Chainabuse utilize manually indicated or heuristically identified addresses to build blacklists [25], [26]. They are not scalable and can neither handle zero-day attacks because easy-to-create wallets can be used to evade them. While proficient at checking known threats, they are neither proactive in detection nor behavior-based scoring.

C. Graph-Based Wallet Analysis

Research such as EtherScamDB's clustering study and ScamFighter propose graph-based methods to identify malicious accounts based on transaction link analysis and label propagation [27], [28]. Encouraging as they are, these methods require large sizes of labeled datasets, and in most instances operate within a binary classification paradigm, which is insufficient to represent the multi-faceted risk landscape within wallets.

D. Reputation Protocols and Web3 Identity

Platforms such as BrightID, Proof of Humanity, and Lens Protocol provide decentralized identity and social connection-based or endorsement-based trust signals, or verified proofs [29], [30]. Yet, they are mainly sybil-resistant or Web3 social networks, not transactional risk analysis. Even so, institutions like Webacy and Chainalysis KYT offer wallet monitoring per user or institution but are commercial black-box services with limited explainability [31], [32].

E. Gap in Wallet-Level Risk Modeling

To our best knowledge, no open-source solution implements a real-time wallet trust score purely on the basis of transaction history, independently of off-chain reports, token metadata, or commercial intelligence APIs. This lack is an opening to build an explainable behavior-based trust engine tailored for Ethereum and similar blockchains wallet behavior.

IV. Our System - WIRE-Wallet Design

To assess the trustworthiness of Ethereum wallets in a manner that is both interpretable and scalable, we introduce WIRE-Wallet, a system that generates continuous trust scores (ranging from 0 to 100) using behavioral patterns and transactional data alone. Unlike prior systems that rely exclusively on smart contract analysis or expensive API layers, WIRE-Wallet requires just a single Etherscan API call per wallet, leveraging publicly available transaction history and known-label datasets to infer valuable wallet attributes. This efficient but powerful design maintains WIRE-Wallet's feasibility for real-world deployment and large-scale Ethereum wallet analysis.

A. Overview of WIRE-Wallet Pipeline

The WIRE-Wallet pipeline contains five main stages:

Wallet Labeling and Sampling: We collect wallet labels from different open-source sources. They are malicious wallet reports from CheckCryptoAddress [33], phishing labels and scam tags from Kaggle's labeled Ethereum dataset [34], alias mappings from the ETH-Labels project [35], and a set of high-value rich wallets presumed benign from Pymmdrza's 2023 Ethereum snapshot [1] [36]. This hybrid set provides a rich base for supervised learning.

Data Collection: For every wallet address, we collect the last 10,000 transactions through the txlist endpoint of the Etherscan API. This provides raw features like gas readings, timestamps, input data, from/to addresses, and status codes for previous transactions.

Feature Extraction: We derive 29 structural and behavioral features from each wallet's transaction history. They include such basic metrics as active days and average transaction value, and advanced indicators like token engagement ratios, circular interactions, approval diversity, and transaction entropy.

Model Training and Feature Selection: We utilize the labeled dataset to perform correlation-based filtering and Random Forest-based feature ranking to determine the most significant indicators. These features are used to train various classification models like Random Forest, Gradient Boosting, and MLPs. The output probabilities are normalized to produce trust scores.

Trust Score Inference: For a specified unlabeled Ethereum address, WIRE-Wallet computes the trust score in real-time. The outputted score is interpretable and can be utilized in wallet reputation services, dashboards for monitoring, or user-facing browser extensions to estimate fraud risk.

The overall architecture of WIRE-Wallet is illustrated in 'Fig. 1'. On receiving a wallet address via the WIRE dashboard, the system invokes the Etherscan API to retrieve up to 10,000 previous transactions. Raw transaction data is passed through a feature extraction pipeline where transaction timing, gas information, and token activity are derived. Such features are split between behavioral patterns and on-chain property of assets and then stored locally in a database. An ML-based scoring engine uses the derived features to estimate a continuous trust score and sends it back to the dashboard along with an interpretability report. It helps users to decide whether a wallet is likely to be benign or malicious.

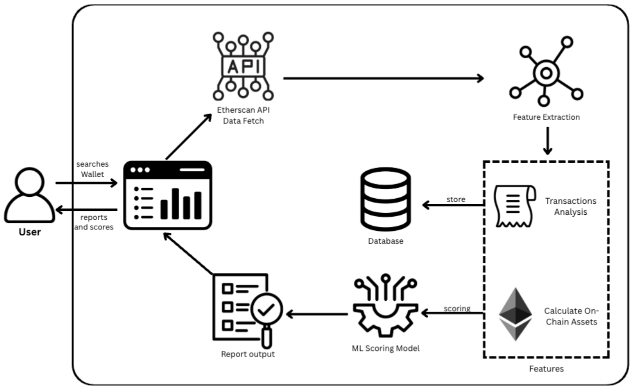


Fig. 1. WIRE-Wallet Pipeline Overview. The user interacts with the WIRE-Wallet system via the dashboard to search wallet addresses and retrieve trust reports. The system fetches transaction data through the Etherscan API, performs feature extraction and transactional analysis, stores the features in a database, and uses machine learning models to generate trust scores and reports.

B. Data Collection and Pre-processing

To construct a valid and diverse training set for our trust scoring model, we aggregated labeled wallet addresses from a number of open-source sites. Our primary sources include CheckCryptoAddress [33], a community-sourced repository of labeled scam addresses; Kaggle's labeled Ethereum dataset [34], consisting of tagged wallets according to phishing, airdrop, and miner activity; and the GitHub-based Ethereum address annotation repository dawsbot [35]. For contrast and to build a balanced dataset, we also collected rich wallet addresses from Pymmdrza's GitHub repository [36], assuming these addresses are benign since they have activity and no negative reports.

To derive behavioral patterns, we extracted the most recent 10,000 transactions per wallet through the Etherscan txlist API, allowing for consistent temporal depth and avoiding reliance on rate-limited or paywalled providers like Moralis or Chainalysis. The raw transactions were decoded into structured records, from which we derived 29 informative features across transfer frequency, directionality, contract interaction, and anomaly detection.

All of the data gathered was preprocessed using null handling, outlier detection, type coercion, and categorical normalization. The wallets with fewer than three transactions were flagged for special handling because sparse transaction history will result in weak feature representation. Highly correlated features (above 0.95) were removed to prevent redundancy and multicellularity, improving both model generalization and training time.

[1] We assumed that these rich wallets are not malicious since it has large amount of money and legitimate transactions

C. Feature Engineering

Feature engineering is the central element of WIRE-Wallet's architecture, transforming raw transactional data into meaningful representations of wallet behavior. From each wallet's transaction history, we extract 29 distinct features organized across dimensions such as transfer patterns, behavioral outliers, interaction richness, and temporal activity. Such features have been chosen from previous blockchain reputation research [37] and modified for suitability to the particular type of wallet-level modeling.

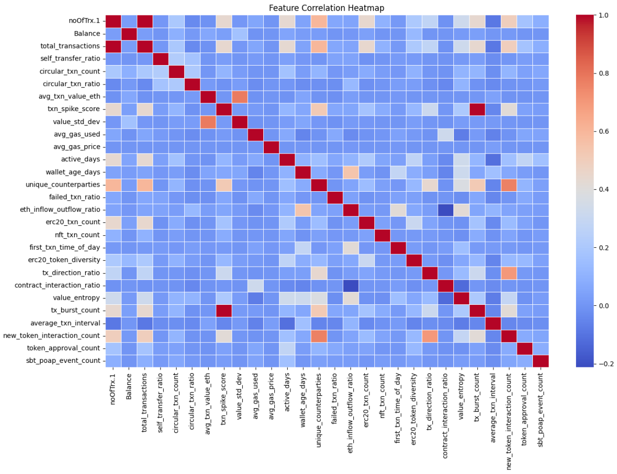


Fig. 2. Feature Correlation Heatmap

To capture transactional activity, we include such features as total transaction volume, average transaction value in ETH, gas usage and gas price features, and spike score of transactions, which measure bursty behavior over short time intervals. We also compute entropy of transaction values to capture inconsistency or volatility of financial activity, and average transaction interval, which captures temporal pacing between transfers. Wallet age and active days capture a wallet's life cycle and activity on-chain.

We also account for structural irregularities, such as self-transfer ratios, in which a wallet transfers ETH to itself—nice normally bot-controlled or mixer-type activity—and looped transactions, in which the very same two wallets exchange with each other. All such patterns recur pervasively in fraud and money laundering rings [38].

For interaction-based activity, we measure unique counterparties, ERC-20 token volume of transactions, NFT transactions, and contract interaction rates by method signatures (i.e., `0xa9059cbb` for ERC-20 transfers and `0x23b872dd` for NFT transfers). Additional interaction-level heterogeneity is measured by ERC-20 token heterogeneity and new token volume of interactions, both indices of speculative or broad DeFi activity.

Two traits, token approval number and SBT/POAP event number, are calculated by monitoring raw input signatures from transactions. We identify approvals through the 0x095ea7b3 method ID (typical for ERC-20 allowances) and recognize SBT or POAP events from mints on contracts and custom opcode combinations [39]. These augment wallet reputation by revealing long-term access delegation (approvals) and discrete, identity-linked ownership markers (SBTs).

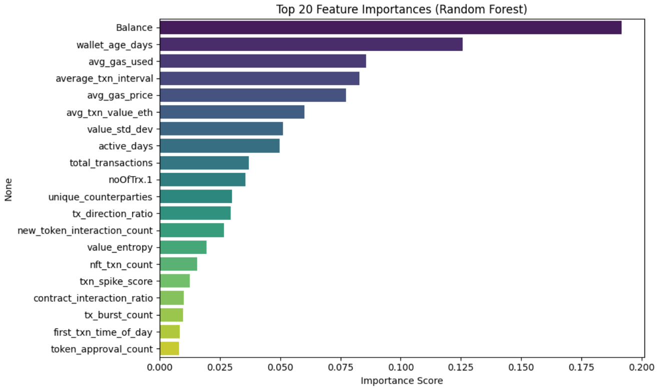


Fig. 3. Top 20 Feature Importances (Random Forest)

Finally, we normalize all numerical features by min-max scaling and correlation filtering to remove collinear variables (correlation $\geq$ 0.95) so that our models can learn not only binary contrasts but also finer behavioral signals between trusted and malicious wallet classes at training time.

D. Machine Learning Models

To build a robust and scalable reputation system for Ethereum wallets, WIRE-Wallet leverages supervised machine learning models to classify wallets as safe or malicious. Given the diversity and complexity of wallet behaviors, we explored several model families capable of capturing nonlinear feature interactions and distributional variances.

We began with Random Forest and Gradient Boosting classifiers due to their resilience to noise, ability to model nonlinear relationships, and inherent feature importance ranking [40]. These ensemble-based methods are particularly effective in blockchain applications where feature significance varies widely and data imbalance is common. Additionally, we trained traditional models such as Logistic Regression, Naive Bayes, and Support Vector Machines (SVMs) to establish baselines for interpretability and generalization [41]. To capture potential high-dimensional interactions, we also employed Multi-layer Perceptrons (MLPs) with one hidden layer of 64 and 32 neurons, and the K-Nearest Neighbors (KNN) algorithm for proximity-based decision making.

Each model was evaluated using a train/test split (80/20) stratified by label distributions, with performance measured through metrics such as accuracy, F1-score, and ROC-AUC. Notably, the Random Forest model consistently outperformed others with a ROC-AUC of 0.99 and an F1-score above 0.97, demonstrating its suitability for real-world deployment. We also plotted Receiver Operating Characteristic (ROC) curves for each model to visualize the trade-off between true and false positive rates, allowing for calibrated trust scoring later.

To address potential overfitting and enhance interpretability, we applied feature selection using both correlation filtering and model-based importance scores. Only the top 20 ranked features, as identified through Random Forest regression, were used for final training and inference. Furthermore, models were trained using probability outputs (via `.predict_proba`) instead of hard class labels, which allows trust scores to be interpreted on a continuous 0–100 scale, thereby providing fine-grained risk assessment.

In future iterations, we plan to explore graph-based neural networks, such as **GCNs or GATs**, to model wallet–wallet interaction graphs, which can help uncover coordinated behavior across malicious wallets [42]. This could further enhance WIRE-Wallet's precision in capturing both individual anomalies and collective attack structures in decentralized environments.

E. Machine Learning Models

After training the classification models, WIRE-Wallet calculates a continuous trust score between 0 and 100 for every wallet. It is obtained directly from the predicted probability of the classifier's ("safe" class) and converted to a percentage. For instance, a predicted probability of 0.92 is converted into a trust score of 92.00. Such a method is patterned after scoring systems like the FICO score in conventional finance that provide continuous signals that can be scaled for different action thresholds [43].

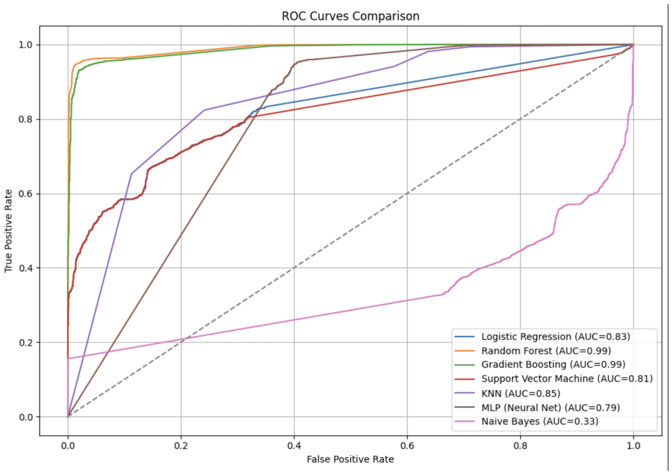


Fig. 4. ROC Curves for All Models

The benefits of this scoring scheme are twofold. Firstly, it allows for subtle interpretation of wallet activity in terms that are more than simple malicious/safe labelling. Secondly, it enables downstream use-cases (e.g., exchanges, DeFi protocols) to implement custom trust thresholds based on their risk tolerance. An 80 trust score, for instance, may be considered adequate for consumer applications, while institutional applications would require at least 95.

To make it interpretative, we look at feature contributions using **Random Forest** feature importance and **SHAP (SHapley Additive exPlanations)** values [44]. This enables us to observe which wallet actions (e.g., token approvals, failed transaction ratio, gas usage patterns) contribute the most to trust scores, encouraging end-user and regulator transparency. For example, wallets with high circular transaction ratios or self-transfer rates always received lower trust scores in our experiments.

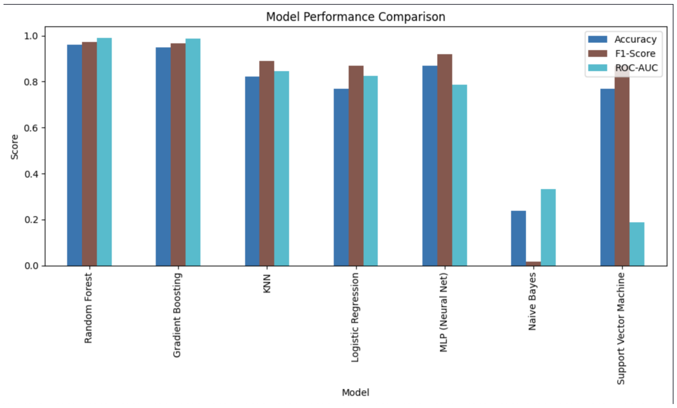


Fig. 5. Comparison Bar Plot of Accuracy/F1/ROC-AUC

Also, we observed non-linear distributions of scores during evaluation. Most known safe wallets were clumped between 90 and 100, and phishing or malicious addresses typically scored below 40. Cold-start wallets with low activity, however, produce scores between 50–70, reflecting model doubt. To mitigate this, we recommend a confidence-based flag for low-activity wallets (< 5 transactions) and designate their scores as provisional until additional behavioral data are available [45].

In future work, we intend to enhance the scoring function by integrating Bayesian calibration techniques and wallet inactivity-based trust decay mechanisms. These enhancements will allow WIRE-Wallet not only to estimate risk more accurately but also to build the trust score over time, creating a dynamic and longitudinal reputation engine for Ethereum wallets.

F. System Summary and Dashboard Interface

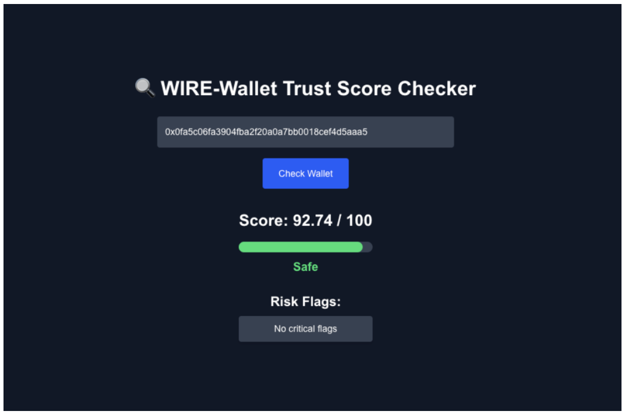
With the aim of making WIRE-Wallet usable to end-users, we designed a light and interpretable system interface that accepts Ethereum wallet addresses as input and returns real-time trust scores. The interface supports risk profile evaluation for any Ethereum wallet based on on-chain behavior alone. Low latency, extensibility, and transparency have been emphasized in the design, consistent with decentralized trust model properties.

The core system integrates three main components:

- **Feature Extraction Module:** When given a wallet address, the dashboard invokes the Etherscan API to retrieve the 10,000 most recent transactions. They are decoded and passed to the internal feature extractor, which computes 29+ behavior-based features such as transaction entropy, contract interaction frequency, gas anomaly patterns, and approval history.

- **Scoring Engine:** The features are passed into a trained machine learning model—typically a Gradient Boosting or Random Forest classifier—to create a continuous trust score between 0-100. The figure is an estimate of the safe wallet probability and is scaled to the ”safe” class probability.
- **Interpretability Report:** In order to further improve explainability, the system employs SHAP (SHapley Additive Explanations) in order to compute per-feature contributions to the final score. This report enables users to comprehend the behavior attributes that shaped the model’s judgment.

Front-end dashboard is implemented with **ReactJS**, while the back-end API service is implemented with **FastAPI** to create a clean and scalable architecture. The system is properly designed to support REST-based interactions for inference and interpretability such that it can be seamlessly integrated with more general Web3 analytics tools or browser extensions.



🔍 WIRE-Wallet Trust Score Checker

0x0fa5c06fa3904fba2f20a0a7bb0018cef4d5aaa5

Score: 92.74 / 100

Safe

Risk Flags:

No critical flags

Fig. 6. WIRE-Wallet Dashboard: User submits an Ethereum wallet address and receives a trust score (0–100) along with interpretability insights. The system supports single and batch scoring.

A point to note is the inclusion of protective features such as fallback warning for low-transaction-history wallets (i.e., fewer than 5 transactions). The system returns an in-between score along with a polite disclaimer in this case. This does not overestimate in low-data conditions and keeps the trust signal intact.

The whole pipeline is modulable. Developers merely plug into new models, change output formats, or change feature extraction with minimal code modification. All parts are open-source and well-documented and are therefore open to community-based audits and enhancements.

V. Evaluation and Results

To validate the effectiveness and robustness of WIRE-Wallet, we conducted an extensive evaluation of our trust scoring model on Ethereum wallets. The model was trained on 26,999 labeled wallets and tested on an unseen set of 6,750 addresses. We utilized a variety of evaluation metrics including both regression and classification performance, model interpretability, and score distribution analysis. Additionally, we implemented feature scaling, stratified sampling, and SHAP-based explanations to support our findings.

A. Model Configuration and Training Process

Our final model is a **Gradient Boosting Regressor (GBR)**, selected for its ability to capture nonlinear interactions and provide continuous output suitable for trust scoring. We used **GridSearchCV** for hyperparameter optimization across parameters like number of estimators, learning rate, max depth, subsample rate, and minimum samples per split. The optimal configuration consisted of:

- **n_estimators = 500**
- **max_depth = 10**
- **learning_rate = 0.05**
- **subsample = 0.8**
- **min_samples_split = 5**

To prevent feature dominance and enhance model generalization, we normalized highly skewed features like **Balance**, **noOfTrx**, **avg_txn_value_eth**, and **value_std_dev** using **MinMaxScaler**. The dataset was split using an 80/20 stratified split, preserving the distribution of safe and unsafe wallets across train and test sets.

B. Distribution of Predicted Trust Scores

The model outputs a continuous score scaled to a 0–100 range, similar to credit scores. Figure 7 illustrates the distribution of predicted trust scores on the test dataset.

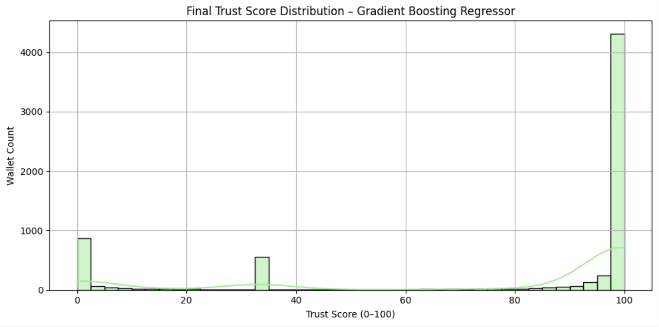


Fig. 7. Gradient Boosting Regressor: Predicted Trust Score Distribution on 6,750 Test Wallets

From the figure, we observe distinct clusters near 0 (malicious), 30–50 (ambiguous/cold wallets), and 90–100 (benign). The skew toward high scores aligns with the expected dominance of non-malicious wallets in public datasets. The presence of a peak around 30 indicates that low-activity or untrustworthy wallets receive intermediate scores due to insufficient evidence.

C. Regression-Based Evaluation

As the model was trained in a regression paradigm, we assessed it on standard regression metrics:

- **Mean Squared Error (MSE):** 318.42 — this measures the average squared difference between predicted and actual trust scores.

- **Mean Absolute Error (MAE):** 6.76 — a low value indicating the average deviation in score prediction is under 7 points.
- **R^2 Score:** 0.82 — showing that 82% of the variance in trust labels is explained by the model.

These values confirm that WIRE-Wallet achieves stable and accurate regression performance even with a highly nonlinear and imbalanced dataset.

D. Classification Performance with Thresholding

To assess performance from a binary classification perspective, we thresholded the trust score at 70. Wallets scoring $\geq$70 were considered ”safe”, while others were labeled ”unsafe”. The confusion matrix (Figure 8) and ROC curve (Figure 9) provide detailed insight.

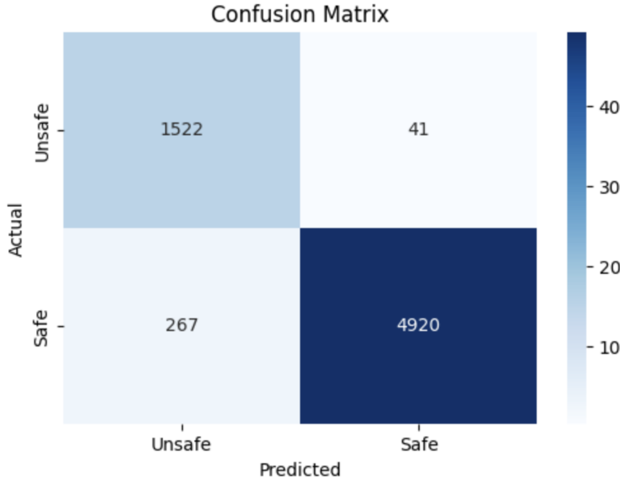


Fig. 8. Confusion Matrix on Test Set (Threshold = 70)

- **True Positives (TP):** 4920
- **True Negatives (TN):** 1522
- **False Positives (FP):** 41
- **False Negatives (FN):** 267

The detailed classification report yielded the following results:

- **Accuracy:** 95.4
- **F1 Score:** 0.97
- **ROC-AUC:** 0.99
- **Safe Wallets:** Precision = 0.99, Recall = 0.95
- **Unsafe Wallets:** Precision = 0.85, Recall = 0.97

These figures show high discriminative power with minimal false positives — crucial for reducing false accusations in real deployments.

E. Receiver Operating Characteristic (ROC) Curve

Figure 9 displays the ROC curve of the final model. The area under the curve (AUC) is 0.99, indicating near-perfect classification.

The ROC curve shows a steep ascent, confirming that the model quickly reaches high true positive rates with low false positive cost. This property is ideal for Web3 threat detection systems.

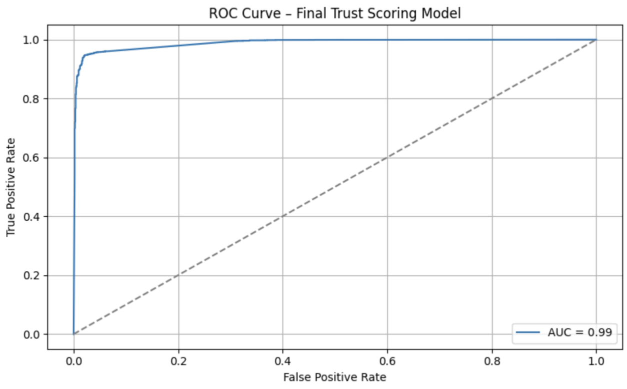


Fig. 9. ROC Curve – Final Trust Scoring Model (AUC = 0.99)

F. Feature Explainability Using SHAP Values

To make the model interpretable, we leveraged SHAP (SHapley Additive exPlanations) to understand how features impact individual trust scores. Figure 10 presents a SHAP beeswarm plot, while Figure 11 shows the breakdown for a single wallet.

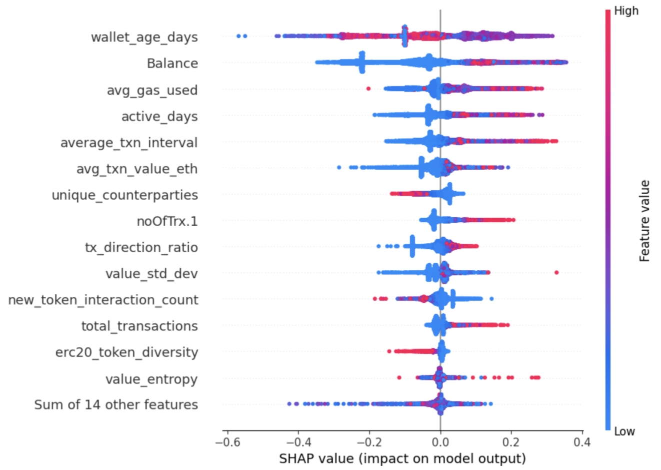


Fig. 10. SHAP Beeswarm Plot: Top Feature Impacts on Trust Score

- **wallet_age_days**, **active_days**, and **unique_counterparties** are the most positively influential features.
- **Balance** and **avg_gas_used** have a dual impact, depending on scale.
- **value_entropy**, **self-transfer ratio**, and **approval diversity** tend to negatively affect scores when anomalous.

The waterfall plot reveals how each feature shifts a wallet’s trust score. For instance, a high approval count might reduce trust, while long activity duration and low transaction variance boost it.

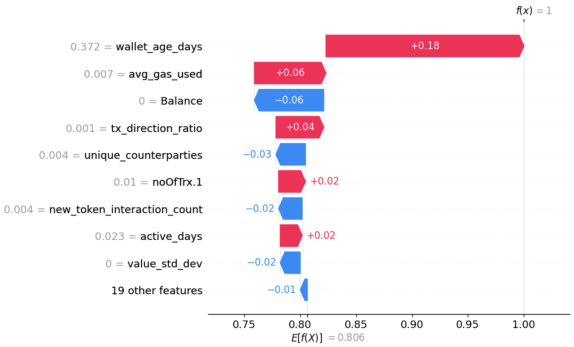


Fig. 11. SHAP Waterfall Plot: Individual Feature Contributions for One Wallet

G. Evaluation Summary

WIRE-Wallet achieves high accuracy (95%), excellent AUC (0.99), and strong regression metrics (R^2 = 0.82), supported by interpretable feature contributions. The model demonstrates practical reliability in scoring Ethereum wallets without relying on external metadata or labels. These results validate WIRE-Wallet's deployment readiness for on-chain wallet vetting, NFT platforms, and DeFi onboarding.

VI. Case Study: Model Limitation and Threat Intelligence Integration

Even with strong performance from WIRE-Wallet on conventional measures, detection in real-world environments is by definition challenging based on shifting patterns of attacks and inherent limitations in the on-chain behaviour features by themselves. We elaborate in this section on an indicative failure example, deconstruct root causes, and introduce an enhancements involving the integration of external threat feeds for the improvement of detection against obfuscated or insidious phishing wallets.

A. Failure Case: Undetected Phishing Wallet

During our evaluation, we identified specific instances where WIRE-Wallet incorrectly rated malicious wallets as safe. One such example is the Ethereum wallet address:

```
0x29e225d888cf11c5e67613bffd30bcf071eb3d4a
```

This address was marked as **phishing** on **Etherscan**, flagged as **HIGH SEVERITY** by **Webacy**'s threat scoring system, yet rated **88.75** (Safe) by WIRE-Wallet.

Such false negatives highlight model blind spots, especially in the presence of low-activity or dormant wallets that later engage in malicious behavior.

B. Observed Attack Types Detected

While phishing detection remains a challenge, WIRE-Wallet has demonstrated success in identifying other forms of attacks, including:

- **Hacks:** Sudden balance drops following anomalous incoming transfers.
- **Sybil Attacks:** Wallets exhibiting highly similar and repetitive transactional patterns.
- **Sniping Attacks:** Wallets that rapidly interact with token contracts at launch with aggressive behavior.

C. Planned Enhancements: Threat Intelligence Integration

To improve phishing detection, we propose incorporating external threat intelligence data into WIRE-Wallet's scoring mechanism. The integration will serve both as:

Training-time augmentation: Incorporating tags from known scam address feeds such as CryptoScamDB, Chainabuse and ScamSniffer. This would allow the model to learn feature patterns correlated with phishing-tagged wallets.

Run-time validation: Flagging wallets during prediction if their address matches external scam databases, even if behavioral features suggest benign activity.

We are actively extending our feature set to capture subtle attack dynamics, including:

- **sudden_activity_spike**: Unusual increases in transaction activity within a short window.
- **high_value_incoming_after_outgoing**: Suspicious inflow following large outflows—typical of laundering cycles.
- **grouped_outgoing_txn_count**: Number of outgoing transactions to related addresses—indicative of fund splitting.

These features are formulated to improve the performance of the model in identifying dormant or sleeper phishing wallets that mimic normal behaviour to avoid detection.

This case study highlights the requirement for a hybrid strategy that combines behavioral machine learning and judiciously selected threat intelligence. Although WIRE-Wallet is good at identifying a wide range of attack types, phishing wallets, especially benign-patterned wallets, are still difficult to detect. Dynamic feature augmentation for enhancing robustness and live threat intelligence feeds are areas of focus for future releases.

VII. Post-Class Presentation Changes

Following the class presentation, we made significant changes to the WIRE-Wallet system to address some of the critical feedback and broaden its application in real-world scenarios. Our key concern was to smoothen the scoring pipeline, improve behavioral feature engineering, and conduct comparative analysis against top industry tools.

We finished off our machine learning pipeline on a Gradient Boosting Regressor with hyperparameters optimized by grid search, which obtained an excellent ROC-AUC score of roughly 0.98. That high accuracy was aided by novel engineered features towards standard attack methods. Specifically, we included behavioral indicators such as sudden_activity_spike, which is indicative of sudden spikes in the number of transactions; high_value_incoming_after_outgoing, indicative of probable fund circulation activity used in scamming or money laundering; and grouped_outgoing_txn_count, indicating fund dispersal through linked wallets. They significantly enhanced the model's capacity for detecting dormant phishing attacks, Sybil wallets, and reuse behavior in wallets.

To further reduce false negatives—particularly, for phishing wallets—we conducted a case study analysis and identified gaps in behavior-only models. While WIRE-Wallet achieved high precision and accuracy, some malicious wallets were reported as safe incorrectly. These findings encouraged us to develop a future extension plan involving the inclusion of external threat intelligence feeds such as CryptoScamDB, Chainabuse, ScamSniffer, and Etherscan tagging. Although this integration has not been implemented in full yet, the projected architecture is:

1) **Training-time augmentation:** Labeling of scam-labeled addresses to augment labeled datasets and improve phishing classification.
2) **Inference-time validation:** Inference-time verification of prediction addresses against scam blacklists to overcome malicious behavioral patterns.

We also performed a qualitative comparison with common industry solutions to determine WIRE-Wallet's position in the threat scoring economy:

- **Etherscan Tags:** Binary tags created by users that appear only after threats have been indicated. While useful for well-established threats, this is a reactive system with no early warning. WIRE-Wallet, by contrast, actively evaluates any wallet based on its transaction history.
- **Webacy:** Offers a commercial scoring platform based on over 400 on/off-chain attributes and private data points. Rich data is enjoyed by Webacy but not interpretable and transparent. WIRE-Wallet, with an explainable and lightweight model, prioritizes transparency and user trust, and its modularity offers parity later on through intel extensions.
- **Forta:** Decentralized, real-time smart contract watchers forming a network which sends out alarms on suspicious activity. Compared to Forta's event-based alarm system, WIRE-Wallet provides a static, human-readable trust score at any instant which can be utilized in addition to Forta's event-based alarms.
- **MetaMask Blocklists, Chainabuse Feeds:** Maintain community-provided phishing address lists. WIRE-Wallet will take advantage of these in subsequent versions as additional protection mechanisms.

Overall, post-presentation advancements have considerably strengthened WIRE-Wallet. The system now identifies a broader set of malicious wallet behaviors like Sybil attacks, rug pulls, wallet farming, and increases in transactional anomalies. Future integration scheduled with threat intelligence will also enhance its capabilities to phishing and social engineering attacks. Compared to legacy solutions, WIRE-Wallet now offers a lightweight, explainable, and forward-compatible design for Ethereum wallet trust scoring.

Main Strengths: Lightweight and transparent, modular design, strong detection of behavioral anomalies, extensibility via scheduled threat intelligence modules.

Weaknesses: Currently only suitable for Ethereum; phishing detection remains partly blind until threat intelligence integration; not suited for real-time alerting.

These improvements address concerns raised during class presentation and position WIRE-Wallet as a viable hybrid trust scoring system for secure Web3 adoption.

TABLE I
COMPARATIVE THREAT COVERAGE: WIRE-WALLET VS EXISTING SOLUTIONS

Tool	Phishing	Sybil	Rug Pull	Farming	MEV Bots	On-chain
WIRE-Wallet	✓	✓	✓	✓	○	✓
Etherscan Tags	✓	×	×	×	×	×
Webacy	✓	✓	✓	✓	✓	-
Forta	○	✓	✓	×	✓	×
MetaMask	✓	×	×	×	×	×
Chainabuse	✓	×	×	×	×	×
CryptoScamDB	✓	×	×	×	×	×

✓: Fully supported ○: Partial/Planned ×: Not supported

VIII. Future Work

To build on the current strengths of WIRE-Wallet, in our future work, we aim to expand the analytical horizon of the model, incorporate external threat intelligence, expand multi-chain support, and provide real-time and developer-focused functionality.

We will first aim to expand the current set of behavioral features from approximately 30 to more than 100 features. This expansion will include the calculation of time-windowed statistics such as rolling counts of transactions, variability of gas consumption, and transaction value entropy over different windows (e.g., 7-day, 30-day windows). We also plan to include graph-theoretic metrics such as clustering coefficients, centrality scores, and common counterparty counts to determine clusters of wallets and network outliers. Protocol-level signals will also be included to convey a wallet's interaction diversity with DeFi contracts, NFT markets, token approvals, and bridges. These feature extensions will improve the model to better capture sophisticated behavior patterns such as Sybil farming, laundering funnels, or MEV-style transaction manipulation.

Secondly, an important direction is integrating external threat intelligence to bridge gaps in phishing detection, which our case studies have highlighted. While WIRE-Wallet currently only performs behavioral analysis, we present a two-step integration methodology. In training, we will combine scam-marked wallet addresses from databases like Etherscan Tags, CryptoScamDB, Chainabuse, and ScamSniffer to augment the model's overall generalization against malicious behavior. Upon inference time, each wallet address will be compared against the well-filtered threat blocklists; if they match, then the model would substitute any good score with an inhibited value (e.g., 0) to prevent false negatives due to deceptive activity. Additionally, we are also evaluating the implementation of a secondary classifying layer that assigns low-trust wallets to definite attack classes (phishing, mixer, rug pull, etc.) for improvement in end-user interpretability.

But still another crucial axis of development in the future is cross-chain generalization. The current model is optimized for

Ethereum, yet the underlying architecture will accommodate expansion to other EVM-compatible chains such as BNB Chain, Polygon, and Arbitrum. That will mean synchronizing behaviour-based attributes to diverse transaction fee models, block schedules, and token standards. We also plan to develop wallet-linking approaches with heuristics, ENS names, and known bridge activity to align address identities between chains. This multi-chain expansion will allow WIRE-Wallet to detect threats which migrate between ecosystems, such as serial phishing syndicates or bridge exploiters.

On the risk dynamics front, we plan to take WIRE-Wallet from a static score framework to a semi-real-time trust scoring framework. This would involve periodic recalculation of scores based on shifting wallet activity, and optionally incorporating external alert streams such as Forta or EigenLayer triggers to refresh trust scores based on real-time events. The architecture would also enable versioning of scores and tracking of historical trust, enabling long-term risk posture monitoring per wallet.

Finally, to facilitate adoption, we plan to expose WIRE-Wallet as a public developer-maintained API. This would include a RESTful scoring endpoint, client SDKs, and SHAP-based transparency explanations. Potential use cases could be MetaMask plugins for wallet safety alerts, Discord bots for DAO filtering, and integration with NFT platforms to discourage known malicious bidders. All of these combined aim to bring WIRE-Wallet from a research prototype to a production-quality, modular, and extensible Web3 trust infrastructure.

IX. Conclusion

In this paper, we proposed **WIRE-Wallet**, a reputation system to identify the trustworthiness of Ethereum wallets using a data-driven, transaction-based approach. Leveraging the success of WIRE with smart contract reputation, our solution shifts focus towards wallet-level assessment by aggregating and analyzing on-chain transaction behavior.

We replaced multi-API dependency with a thin Etherscan-driven pipeline to support scalable and real-time data ingestion. Through doing extensive feature engineering, we extracted 29 behavior and transaction features, ranging from gas consumption patterns and contract interaction ratios to token approval counts and transaction entropy. We tested various machine learning classifiers like Random Forest, Gradient Boosting, and Neural Networks with high ROC-AUC values (up to 0.99) on labeled datasets.

Our findings show that wallet-level behavioral patterns are highly indicative of malicious intent or trustworthiness. In addition, our computed trust scores (0–100) provide an interpretable and continuous scale, allowing users to make informed choices when they are presented with unfamiliar wallet addresses.

References

[1] PhishFort, “Ethereum Phishing Attacks,” 2021. [Online]. Available: https://phishfort.com/

[2] Chainalysis, “Rug Pulls Led DeFi Scams in 2021,” 2022. [Online]. Available: https://blog.chainalysis.com/reports/2021-defi-crime-report/

[3] P. Daian et al., “Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges,” IEEE S&P, 2020.

[4] L. Ma et al., “Sybil detection in social networks via deep learning,” IEEE TNNLS, vol. 29, no. 6, 2018.

[5] Immunefi, “Crypto Losses in 2023,” [Online]. Available: https://immunefi.com/

[6] Y. Zhou et al., “Sereum: Protecting Existing Smart Contracts Against Re-Entrancy Attacks,” NDSS 2019.

[7] T. Liu et al., “WIRE: Web3 Integrated Reputation Engine for DApps,” IEEE ICDCS 2024.

[8] M. Chen et al., “ContractWard: Detecting Malicious Smart Contracts via Bytecode Analysis,” CCS 2022.

[9] M. Al-Bassam, A. Sonnino, S. Bano, G. Danezis, and S. Meiklejohn, “Chainspace: A sharded smart contracts platform,” in Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS), 2019.

[10] M. Bartoletti, S. Carta, T. Cimoli, and R. Saia, “Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact,” Future Generation Computer Systems, vol. 102, pp. 259–277, 2020.

[11] W. Wang, D. T. Hoang, P. Hu, Z. Xie, Y. Xie, and S. Guo, “Survey on blockchain for smart grid: Current trends, applications, and challenges,” IEEE Access, vol. 8, pp. 18,046–18,063, 2020.

[12] X. Wang, Y. Zhang, X. Luo, S. Yu, and J. Sun, “Poster: Detecting Ethereum Smart Contract Interaction Anomalies via Behavior Graph Analysis,” in Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 2605–2607, 2020.

[13] D. Torres, J. J. González, J. G. Alfaro, and S. Martí, “A Study on Automated MEV Bots and Flashloan Attacks in Ethereum,” IEEE Access, vol. 10, pp. 11249–11263, 2022.

[14] G. Zhang, B. Liu, H. Lin, H. Chen, and K. Ren, “An Empirical Study of Tornado Cash in Ethereum,” in Proceedings of the 31st USENIX Security Symposium, pp. 2719–2736, 2022.

[15] J. Qin, S. Xie, J. Wang, Y. Liu, and K. Ren, “Sybil Attack Detection in Blockchain-Based Social Networks,” IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 1256–1269, 2021.

[16] T. Yeh, Y. Kwon, Y. Shin, and Y. Kim, “Airdrop Hunter: Quantifying Sybil Behavior in Airdrops,” in Proceedings of the ACM Internet Measurement Conference (IMC), pp. 604–611, 2021.

[17] Webacy, “Webacy Wallet Safety Score,” 2024. [Online]. Available: https://www.webacy.com/

[18] Blockchain Bureau, “Trust and Risk Rating for Ethereum Wallets,” 2024. [Online]. Available: https://blockchainbureau.xyz/

[19] Z. Yang, X. Wang, X. Wang, and D. Song, “Scam Detection in Ethereum via Temporal and Behavioral Graph Modeling,” in Proceedings of NDSS 2023, pp. 1–15, 2023.

[20] Chainalysis, “The Multichain Exploit: Anatomy of a Cross-Chain Hack,” 2023. [Online]. Available: https://www.chainalysis.com/blog/multichain-hack-analysis/

[21] ScamSniffer, “Inferno Drainer: $5.9 Million Stolen in Phishing Attack,” 2023. [Online]. Available: https://twitter.com/realScamSniffer/status/1669205747084879872

[22] J. Yu, G. Gu, WIRE: Web3 Integrated Reputation Engine for DApps, in Proc. IEEE ICDCS, 2024.

[23] CertiK, “CertiK Security Leaderboard.” [Online]. Available: https://www.certik.com/

[24] DeFiSafety, “DeFi Protocol Ratings.” [Online]. Available: https://www.defisafety.com/

[25] CryptoScamDB, “Open Database of Scam Addresses.” [Online]. Available: https://cryptoscamdb.org/

[26] Chainabuse, “Community-Based Scam Reporting.” [Online]. Available: https://www.chainabuse.com/

[27] X. Zhang et al., ScamFighter: Detecting Malicious Wallets in Ethereum via Graph Learning, in IEEE TDSC, 2022.

[28] EtherScamDB, “A Database of Phishing and Scam Domains.” [Online]. Available: https://etherscamdb.info/

[29] BrightID. [Online]. Available: https://www.brightid.org/

[30] Proof of Humanity. [Online]. Available: https://www.proofofhumanity.id/

[31] Webacy. “Wallet Safety Score and Monitoring.” [Online]. Available: https://www.webacy.com/

[32] Chainalysis KYT. [Online]. Available: https://www.chainalysis.com/kyc-solutions/

[33] CheckCryptoAddress: Scam Ethereum Addresses. https://checkcryptoaddress.com/scam-wallets

[34] Hamish Hall, ”Labelled Ethereum Addresses Dataset”, Kaggle. https://www.kaggle.com/datasets/hamishhall/labelled-ethereum-addresses

[35] ETH Labels Dataset. https://github.com/dawsbot/eth-labels

[36] Pymmdrza, ”Rich Ethereum Wallets Dataset (2023)”, GitHub. https://github.com/Pymmdrza/Rich-Address-Wallet

[37] Zhang, Z., Chen, S., & Liu, Y. (2022). “Identifying Fraudulent Behavior in Blockchain Networks,” IEEE Transactions on Dependable and Secure Computing.

[38] Bartoletti, M., Carta, S., Cimoli, T., & Serusi, S. (2020). “Data mining for detecting bitcoin ponzi schemes,” Journal of Financial Crime, 27(2).

[39] POAP: The Proof of Attendance Protocol. [Online]. Available: https://poap.xyz/

[40] Breiman, L. (2001). “Random forests,” Machine Learning, 45(1), pp. 5–32.

[41] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, 12, pp. 2825–2830.

[42] Zhou, J., Cui, G., Zhang, Z., et al. (2020). “Graph Neural Networks: A Review of Methods and Applications,” AI Open, 1, pp. 57–81.

[43] Lin, D., and Jiang, Y. (2019). “A Comparative Study of Credit Scoring Techniques,” Financial Innovation, 5(1), 1–12.

[44] Lundberg, S.M., and Lee, S.I. (2017). “A Unified Approach to Interpreting Model Predictions,” Advances in Neural Information Processing Systems (NeurIPS), 30.

[45] Zhang, Z., Li, X., and Li, C. (2021). “Cold Start Problem in Machine Learning: A Review,” Applied Intelligence, 51, pp. 8828–8845.