



**TÜRKİYE CUMHURİYETİ
ADANA ALPARSLAN TURKEŞ SCIENCE AND TECHNOLOGY
UNIVERSITY**

**GRADUATE SCHOOL
COMPUTER ENGINEERING DEPARTMENT**

**REVEALING UNEXPLORED DEANONYMIZATION RISKS OVER
THE ETHEREUM BLOCKCHAIN**

Yasir KILIÇ

Ph.D.



**TÜRKİYE CUMHURİYETİ
ADANA ALPARSLAN TÜRKES SCIENCE AND TECHNOLOGY
UNIVERSITY**

**GRADUATE SCHOOL
COMPUTER ENGINEERING DEPARTMENT**

**REVEALING UNEXPLORED DEANONYMIZATION RISKS OVER
THE ETHEREUM BLOCKCHAIN**

Yasir KILIÇ

Ph.D.

**THESIS ADVISOR
Assoc. Prof. Dr. Ali İNAN**

ADANA, 2025

CERTIFICATION OF APPROVAL

(REVEALING UNEXPLORED DEANONYMIZATION RISKS OVER THE ETHEREUM BLOCKCHAIN)

This **Ph.D.** thesis, completed under the conditions determined by the relevant regulations, by Yasir KILIÇ with student number 21801303003 in Adana Alparslan Türkeş Science and Technology University Institute of Graduate School Computer Engineering Department, has been evaluated by the following academic committee, and approved in accordance with the relevant articles of the "Adana Alparslan Türkeş Science and Technology University Graduate Education Regulations".

Prof. Dr. Mustafa AKYOL
Institute of Graduate School Manager

Assoc. Prof. Dr. Mümine KAYA KELEŞ
Head of the Department

Assoc. Prof. Dr. Ali İNAN
Academic Advisor

Thesis Advisory Committee Members

Assoc. Prof. Dr. Ali İNAN
Adana Alparslan Türkeş Science and Technology University (ATÜ)

Assoc. Prof. Dr. Fatih ABUT
Çukurova University (ÇÜ)

Assoc. Prof. Dr. Fatih KILIÇ
ATÜ

Asst. Prof. Dr. Haydar TUNA
Osmaniye Korkut Ata University (OKÜ)

Assoc. Prof. Dr. Yasin KAYA
ATÜ

Thesis Defence Date

18/12/25

DECLARATION OF CONFORMITY

In this thesis study, which was prepared following the thesis writing rules of Adana Alparslan Türkeş Science and Technology University Institute of Graduate School, I declare that I provide all the information, documents, evaluations and results in accordance with scientific ethics and moral codes without resorting to any means or assistance that would be contrary to scientific ethics and traditions. I also declare that I refer to all of the articles I used in this study with appropriate references and accept all moral and legal consequences if a situation is found contrary to my statement regarding my work.

19/12/2025

[Signature]

Yasir KILIÇ

ÖZET

ETHEREUM BLOKZİNCİR ÜZERİNDEN KEŞFEDİLMEMİŞ ANONİMSİZLEŞTİRME RİSKLERİNİN ORTAYA ÇIKARILMASI

Yasir KILIÇ

Doktora, Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ali İNAN

Aralık 2025, 88 sayfa

Blokzincir teknolojisi, özellikle Ethereum, DeFi, dijital kimlik ve yönetim gibi alanlarda önemli ilerlemeleri mümkün kılmıştır; zira sunduğu sözde anonimlik gizliliği artırırken aynı zamanda atıf yapılabilirliği ve hesap verebilirliği zorlaştırmaktadır. Tornado Cash'in yasa dışı fonların aklanmasındaki rolü nedeniyle yaptırıma uğraması gibi yüksek profilli vakalar, blokzinciri ortamında kötü amaçlı fon hareketlerinin izlenebilmesi için etkili mekanizmalara duyulan acil ihtiyacı gözler önüne sermektedir. Bu tezde, Ethereum üzerindeki kötü amaçlı fon hareketlerinin izlenebilirliğini artırmak ve hesap sahibi ilişkilendirmesini gerçekleştirmek amacıyla tasarlanmış, davranış farkına dayalı bir çerçeve olan *EDL* adlı yeni bir çözüm öneriyoruz. Önerilen yaklaşım, geleneksel uzaklık metriklerinin kısıtlamalarını aşarak model tabanlı bir ilişkilendirme stratejisi sunmaktadır. Bu çerçevede, açıklanabilir makine öğrenimi tekniklerinin entegrasyonu ile karar süreçlerinin şeffaflığı ve yorumlanabilirliği garanti altına alınmıştır. Özellikle, önce hesapları davranışsal karakteristikler üzerinden profilleyerek, önemli davranışsal farklılıklar bulunsa bile aynı kişi tarafından kontrol edilen hesapların belirlenmesini mümkün kılıyoruz. Karar alma sürecinde şeffaflığı sağlamak için doğası gereği açıklanabilir bir yapay zekâ (AYZ) modeli kullanarak blokzinciri araştırmacılarına anonimsizleştirme mekanizmalarına ilişkin sağlam ve yorumlanabilir içgörüler sunuyoruz. Gerçek dünya veriseti üzerinde gerçekleştirdiğimiz kapsamlı deneyler, *EDL*'nin çeşitli ampirik koşullarda güncel yöntemlerin ötesinde bir performans sergilediğini ve Ethereum blokzincirindeki kötü amaçlı hesapların deanonymizasyonuna ilişkin temel zorlukları etkili biçimde ele aldığını göstermektedir.

Anahtar Kelimeler: Blokzincir güvenliği, Ethereum, hesap sahibi ilişkilendirilmesi, anonimsizleştirme, açıklanabilir makine öğrenimi.

ABSTRACT

REVEALING UNEXPLORED DEANONYMIZATION RISKS OVER THE ETHEREUM BLOCKCHAIN

Yasir KILIÇ

Ph.D. Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Ali İNAN

December 2025, 88 pages

Blockchain technology, especially Ethereum, has enabled major advances in areas like DeFi, digital identity, and governance, as its pseudonymity boosts privacy while hindering attribution and accountability. High-profile cases, such as the sanctioning of Tornado Cash for its role in laundering illicit funds, highlight the urgent need for effective mechanisms to trace malicious fund activity on the blockchain environment. In this dissertation, we propose a novel solution called EtherDiffLinkage (*EDL*), a behavioral-difference-based framework designed to implement account owner linkage and improve the traceability of malicious fund movements on Ethereum. Technically, we propose a model-based supervised linkage strategy that addresses the limitations of traditional distance metrics. In this framework, the integration of explainable machine learning techniques guarantees transparency and interpretability of the decision. Specifically, we first profile accounts based on behavioral characteristics; enabling the identification of accounts controlled by the same owner, even in cases where significant behavioral variations are present. To ensure transparency in its decision-making process, we leverage an inherently explainable AI (xAI)-based model, providing blockchain investigators with robust and interpretable insights into the mechanisms of deanonymization. Extensive experiments on a real-world dataset demonstrate that *EDL* outperforms state-of-the-art methods across diverse empirical conditions, effectively addressing key challenges in the deanonymization of malicious accounts on the Ethereum blockchain.

Keywords: Blockchain security, Ethereum, account owner linkage, deanonymization, explainable machine learning.

Dedication

This dissertation is dedicated to my beloved sons, Selim Alp & Ömer Efe KILIÇ, whose unwavering love, patience, and joy have been my constant source of strength. Their smiles have been my greatest motivation, and I hope this work inspires you to always pursue your dreams with passion and perseverance. I also extend this dedication to my wife, Vahide Nida KILIÇ, whose continuous support and understanding have greatly contributed to the completion of this work.

ACKNOWLEDGEMENTS

Academic graduation studies are never an easy journey, yet the strong support I received from many people made this period a truly rewarding and memorable experience. I feel deeply grateful to everyone who has contributed to my academic and personal growth throughout this process.

First and foremost, I would like to express my sincere appreciation to my advisor, Assoc. Prof. Dr. Ali İNAN, for his constant guidance, patience, and encouragement. His insightful advice and extensive knowledge have greatly shaped my research and academic development. His dedication to science and his approachable nature inspired me not only as a researcher but also as a person. I am truly thankful for his support and mentorship during every stage of my graduate studies.

I would also like to extend my gratitude to my committee members, Assoc. Prof. Dr. Fatih ABUT and Assoc. Prof. Dr. Fatih KILIÇ, for their valuable comments, constructive suggestions, and continuous encouragement throughout the preparation of this dissertation. Their feedback has been instrumental in improving the quality of my work.

My deepest appreciation goes to my dear wife and colleague, Asst. Prof. Dr. Vahide Nida KILIÇ, whose encouragement and insightful ideas have broadened my perspective and helped me approach my research from new angles. Her endless patience and belief in me have been a constant source of motivation.

I am also profoundly thankful to my beloved parents (Ömer & Yeter KILIÇ) and brothers (Muratcan & İbrahim KILIÇ) for their unconditional support, and trust. Their presence has always given me strength through both challenging and joyful times.

Finally, I would like to thank all my teachers, colleagues, and friends who have accompanied and supported me during my academic journey. Each of them, in their own way, has made this experience both meaningful and unforgettable.

TABLE OF CONTENTS

CERTIFICATION OF APPROVAL	i
ÖZET	iii
ABSTRACT	iv
<i>Dedication</i>	v
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
LIST OF TABLES	x
LIST OF ABBREVIATIONS	xi
LIST OF SYMBOLS	xii
1. INTRODUCTION	1
2. LITERATURE REVIEW	10
2.1 Deanonymization Solutions over Bitcoin (BTC)	10
2.2 Deanonymization Solutions over Ethereum (ETH)	12
2.2.1 Account Categorization (AC)	14
2.2.2 Account Owner Linkage (AOL)	16
3. BACKGROUND	19
3.1 Ethereum Blockchain as a Distributed Ledger	19
3.2 Formal Definitions	21
3.3 Explainable Artificial Intelligence(xAI) Methods	24
3.4 Ciphertext Indistinguishability Concept	26
4. PROBLEM DEFINITION	28
4.1 Problem Description	28
4.2 Problem Formulation	29
5. SOLUTION OVERVIEW	32
5.1. Motivational Background	32
5.2. Methodological Foundation	33
6. PROPOSED SOLUTION	39
6.1. Training Phase	41
6.1.1. Behavioral Profiling Module	41
6.1.2. Behavioral Difference-based Model Construction Module	47

6.1.3. Evaluation Module	57
6.2. Inference Phase	60
7. EXPERIMENTAL RESULTS	62
7.1. Experimental Setup	62
7.2. Baseline Experiments	65
7.3. Performance Analysis of <i>EtherDiffLinkage</i>	68
7.4. Comparative Analysis	72
7.5. Case Study: A Real-World Blockchain Investigation with Human-AI Collaboration using <i>EDL</i>	75
8. DISCUSSION	79
9. CONCLUSION AND FUTURE WORK	81
REFERENCES	82
CURRICULUM VITAE	

LIST OF FIGURES

Figure 3.1 ER diagram of Ethereum data.....	23
Figure 4.1 Illustration of the research problem	28
Figure 6.1 Workflow of EDL	40
Figure 7.1 Baseline Experiments.....	65
Figure 7.2 Behavioral Grouping Analysis.....	67
Figure 7.3 Train-Test Ratio Effect	68
Figure 7.4 <i>K</i> -Fold Model Performance Analysis	69
Figure 7.5 Ablation Study	70
Figure 7.6 Overall Feature Importance Scores.....	72
Figure 7.7 Performance Comparison under Varying Anonymity Set Size	74
Figure 7.8 Inference result at (a) rank 1 (b) rank 2	77



LIST OF TABLES

Table 2.1 An Overview of Deanonymization Studies for Ethereum accounts	15
Table 3.1 Information about raw Ethereum data	22
Table 6.1 Overview of Extracted Features	42
Table 6.2 Hypothetical Toy Dataset	52
Table 6.3 Learning Target Aware Constructed Dataset	53
Table 6.4 Bin-wise updates for $\Delta F2$ after the first feature update.	54
Table 7.1 Baseline Experiment Settings	66
Table 7.2 Performance Comparison over benchmark dataset	73



LIST OF ABBREVIATIONS

AI	: Artificial Intelligence
ML	: Machine Learning
xAI	: Explainable Artificial Intelligence
DeFi	: Decentralized Finance
ETH	: Ethereum
BTC	: Bitcoin
dApps	: Decentralized Applications
EBM	: Explainable Boosting Machines
AUC	: Area Under the Curve
LR	: Logistic Regression
XGB	: XgBoost Model
DT	: Decision Tree Model
LGB	: Light Gradient Boosting Model

LIST OF SYMBOLS

$T_i. \tau$	: Timestamp of the transaction
$T_i. S$	: Sender address
$T_i. R$	: Recipient address
$T_i. V$	: Transferred amount
$T_i. P$	: Price of gas per unit
$T_i. E$	: Error flag for the transaction
$T_i. C$	: Contract address involved
$T_i. G$	: Gas consumed for the transaction

For detailed information, please refer to Section 3.2.

1. INTRODUCTION

Traditional financial systems operate under centralized control, where trusted intermediaries such as banks, clearing houses, or payment processors (e.g., SWIFT or Visa networks) are responsible for validating, recording, and authorizing transactions. In such systems, the correctness of account balances/credit limits and the legitimacy of transfers depend on institutional oversight, regulatory compliance, and the integrity of the organizations maintaining private ledgers. As detailed in Section 3.1, blockchain technology¹, however, represents a fundamental departure from this centralized model. Instead of relying on intermediaries, it establishes consensus among a distributed set of nodes/participants who collectively maintain a shared and tamper-resistant ledger. Through consensus protocols (e.g., Proof of Work in Bitcoin or Proof of Stake in Ethereum since 2022), transactions are verified and appended to the blockchain by independent nodes, ensuring both transparency and security without the need for mutual trust. This decentralized validation process removes single points of failure, minimizes reliance on institutional authorities, and enables direct peer-to-peer value exchange within a trustless environment.

From a privacy perspective, the blockchain system fundamentally differs from traditional financial systems in which users are required to provide personal information such as name, national identification number, or bank-issued credentials to establish and operate an account. In conventional infrastructures, account ownership and identity verification are maintained through institutional control. Blockchain systems, in contrast, achieve transparency without directly exposing account identities. A single individual/user of blockchain systems may create and manage multiple accounts. Each account is identified by a unique cryptographic address such as an Ethereum account address rather than by personally identifiable information. This design allows individuals to perform transactions anonymously since their activities are associated with cryptographic keys of corresponding accounts instead of verifiable identity attributes of the account owner. Although every transaction is permanently recorded on a public

¹ Blockchain is fundamentally a distributed ledger system that records and verifies data across multiple network nodes.

ledger to ensure transparency and verifiability, the information about identities of the actual account owners remain hidden. This represents a significant departure from the identity-bound operation model of traditional financial institutions.

Due to the intrinsic anonymity provided by blockchain systems, inference of the hidden identities behind malicious activities (such as money laundering, ransomware cash-outs, phishing-related fund diversion, and fraud) has become critically important. Therefore, developing analytical methods capable of deanonymizing suspicious accounts without compromising the integrity of the overall system has emerged as a fundamental research focus in blockchain security. Towards this end, investigators, regulators, and researchers constantly seek to methodologically identify accounts that exhibit similar transactional characteristics (e.g., transfer frequency, interaction timing, or value distribution) in order to trace fund movements or uncover coordinated behavior across related accounts. Since blockchain systems are inherently pseudonymous, linking accounts to real-world entities through personal names or organizational records is not possible. The objective, therefore, is to extract identity-relevant insights such as behavioral profiles, or co-ownership of accounts linkage from a transparent yet pseudonymous ledger environment. This problem is commonly referred to as *deanonymization of blockchain accounts* (Béres et al., 2020; Du et al., 2023; Gao et al., 2021; Zhou et al., 2022).

Among existing blockchain systems, Ethereum has attracted particular research attention due to its distinctive features that enable complex and autonomous interactions through smart contracts and blockchain protocols (*The Rise of the Ethereum Economy*, n.d.). Unlike the other blockchain systems such as Bitcoin and Solana, Ethereum's programmable infrastructure allows users to deploy and execute contract-based operations involving tokens, liquidity pools, and automated exchanges. Although this flexibility has accelerated innovation in the blockchain ecosystem, it has also made Ethereum an appealing environment for malicious actors seeking to exploit its transparent and programmable nature.

Recent incidents involving illicit fund transfers (*North Korean Hackers Return to Tornado Cash despite Sanctions*, n.d.; *The \$477 Million FTX Hack: A New Blockchain Trail*, n.d.; *Tracing the Trail of the Upbit Hack*, n.d.) have shown that mixing services/contracts are one

of the most commonly used tool for malicious activity on Ethereum. Tornado Cash is one of the most prominent examples of mixing services. This smart-contract-based mixing service allows a user to deposit funds and later withdraw them to a desired destination account by presenting a cryptographic note, enabling the withdrawal to occur at any subsequent time. Under ideal conditions, if the mixer contract receives n deposits, then at any given time t it can generate at most n withdrawals, since each withdrawal must correspond to one previously issued deposit note. Although all incoming and outgoing transactions are fully transparent on-chain, the linkage between individual deposits and withdrawals remains concealed, making it impossible to determine which withdrawal corresponds to which deposit. In practice, users may control multiple accounts and interact with the mixer at different times to further obscure ownership linkage or to transfer funds to accounts to which they do not wish to send assets directly.

As exemplified in Figure 1.1, a set of accounts $\{0x1, 0x2, 0x3, \dots\}$ send deposits into the mixer, and at later times another set of accounts $\{0x4, 0x5, 0x6, \dots\}$ withdraw funds. For example, once $0x1$ deposits at time t_1 , any subsequent withdrawal (e.g., t_4 from $0x4$, t_5 from $0x5$, t_6 from $0x6$) becomes part of its anonymity set—meaning that any of these accounts could plausibly be linked to $0x1$. Because the contract does not preserve or reveal the correspondence between the input and output flows, establishing an exact ownership linkage is fundamentally infeasible. This form of obfuscation is conceptually similar to anonymization mechanisms studied in spatio-temporal privacy research (AbuElHassan et al., 2025; Cicek et al., 2014; Shaham et al., 2019; Zacharouli et al., 2007), where unlinkability is achieved by disrupting the direct correspondence between observed inputs and outputs (e.g., trajectory permutation, or time-shuffling). In mixer contracts, the temporal and structural mixing of deposits and withdrawals serves an analogous role, complicating traceability despite full public transparency of the underlying transactions.

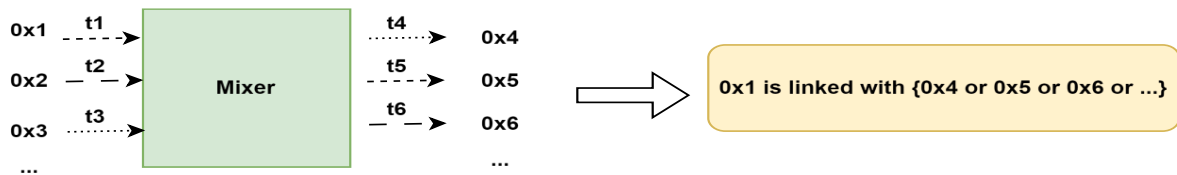


Figure 1.1. Tracing Challenge under Mixing Service Usage

A notable example is the 2022 *Tornado Cash* case, where a cryptocurrency mixing service built on Ethereum was sanctioned by the U.S. Treasury after being linked to the laundering of approximately \$455 million in stolen funds attributed to the North Korean cybercrime group *Lazarus*. Although partial fund tracing was achieved, this incident revealed the challenges of establishing accountability in decentralized systems. Various similar, yet independent cases of malicious funds transfer have been reported against the Ethereum infrastructure (*The \$477 Million FTX Hack: A New Blockchain Trail*, n.d.; *Tracing the Trail of the Upbit Hack*, n.d.) . Therefore, it is important to devise effective deanonymization techniques for the Ethereum as addressed in this thesis.

As a research problem, two main approaches have been proposed for deanonymization in the Ethereum domain. These are account categorization (AC) methods, which classify accounts into behavioral groups such as scams, hacks, or exchanges based on transactional patterns (Payette et al., 2017; Pragasam et al., 2023; Zhou et al., 2022), and account owner linkage (AOL) methods, which link distinct accounts of the same user/owner (Béres et al., 2020; Du et al., 2023; Hu et al., 2023; Kılıç & İnan, 2024; Tang et al., 2021; Victor, 2020a).

AC methods generally achieve high accuracy, since behavioral categorization involves assigning accounts to a limited number of distinguishable categories (Zhou et al., 2022) and the overall behavioral patterns are separable across these groups (Payette et al., 2017). However, they do not yield an inference regarding ownership (e.g. account owner linkage), since it is common for an individual to operate several accounts that exhibit distinct transactional behaviors or reflect different operational motivations (Li et al., 2021; *The \$477 Million FTX Hack: A New Blockchain Trail*, n.d.).

Contrary to AC methods, AOL methods require analyzing/modelling transactional behavior of an account at the owner level and therefore facilitate the tracing of fund movements across accounts. Under reasonable assumptions, where domain knowledge allows for the accurate formation of anonymity set containing candidate related accounts, linkage can be performed either through heuristic transaction-pattern analysis (Hu et al., 2023; Tang et al., 2021) or through learning-based models (Du et al., 2023) trained on public transaction data.

As a research problem, Account Owner Linkage (AOL)–based deanonymization solutions are required to effectively trace fund transfers involved in such illicit activities. In this context, deanonymization task is as the challenging task of linking source and destination accounts that are controlled by the same account owner, enabling investigators to trace transactional relations back to a common owner. Please refer the Chapter 4 for a technical definition of our research problem.

Although numerous solutions have addressed the AOL problem (Du et al., 2023; Fu et al., 2022; Gao et al., 2021; Hu et al., 2023; Payette et al., 2017; Victor, 2020a; Zhou et al., 2022), they have three limitations:

- (i) They often operate under the assumption that an account owner behaves similarly across distinct accounts of hers implying *homophilic* behavior. The converse of this overly rigid assumption has been shown repeatedly in cases such as using hot and cold accounts to complicate tracking, or engaging in distinct financial motivations (e.g. bridge accounts (*North Korean Hackers Return to Tornado Cash despite Sanctions*, n.d.; *The \$477 Million FTX Hack: A New Blockchain Trail*, n.d.; *Tracing the Trail of the Upbit Hack*, n.d.)). Such variability in behavior could severely hinder the effectiveness of deanonymization methods that link accounts based on similarity of behavior.
- (ii) Existing learning-based AOL approaches rely on deep-learning-driven black-box feature extraction, which provides limited explanatory insight into the resulting deanonymization decisions, and although post-hoc explainable-AI (xAI) techniques can offer partial interpretability, their outputs remain inherently unreliable because they merely approximate the underlying black-box model (Slack et al., 2020). Therefore, there is a critical need for inherently xAI methods (for technical description, please refer to Section 3.3).
- (iii) They perform naïve distance-based similarity matching after featurization, which again relies on a *homophilic* behavior assumption and fails to capture heterogeneous usage motivations.

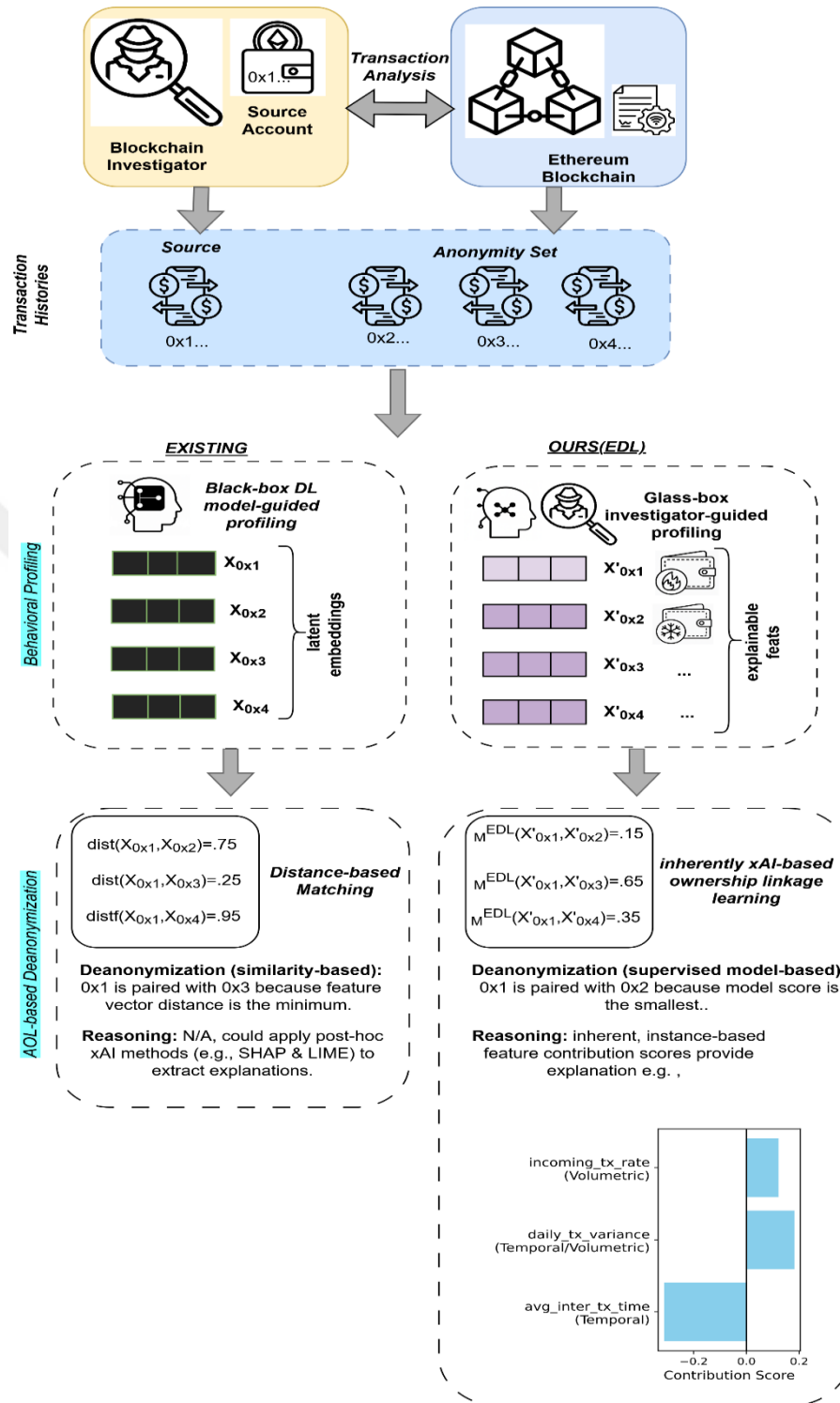


Figure 1.2 Overview of the Proposed Solution

To address these limitations, in this dissertation, we propose *EtherDiffLinkage (EDL)*, a novel solution for the AOL-based deanonymization task that leaves room for on-purpose *heterophilic* behavior, which is the case of a user exhibiting varying behavior over distinct accounts of hers. *EDL*, therefore, is more flexible and as we empirically show in Section 7.4 more successful.

As reviewed in Section 2.2.2, another major difference between our proposed solution and existing work pertains to how user-behavior is represented: *EDL* employs transparent features that are fully interpretable to investigators, whereas prior work creates feature sets embedding mixer-specific patterns or deep-learning-based representations (please refer to Sec. 6.1.1 for details).

Additionally, from an AOL-solution perspective, contrary to existing approaches that perform naive distance-based similarity matching over their learned feature representations to infer the closest account in the embedding space, *EDL* employs a supervised learning model M^{EDL} like characterized in Section 3.4 that learns ownership linkages based on behavioral-difference patterns (please refer to Chapter 6 for methodological details). The model M^{EDL} focuses on learning the specific attributes and degrees to which malicious users alter their behavior to differentiate one account from another. This enables adaptive matching based not only on similarities but also on these deliberate behavioral deviations, which existing similarity-based solutions fail to model.

All points on which our approach diverges from existing solutions are visually summarized in Fig. 1.2. Consider hot–cold account configurations—as exemplified by the wallet icons in the right-hand behavioral-profiling block of the figure where a single owner deliberately uses two addresses that fall into strongly divergent behavioral characteristics. In this setup, hot accounts typically generate bursty, high-frequency transactional sequences, whereas cold accounts remain sparse and inactive for extended periods. Although hot–cold setups are only one illustrative example, similar divergent behavioral regimes frequently arise in real multi-account operational settings (Li et al., 2021; *The \$477 Million FTX Hack: A New Blockchain Trail*, n.d.; *Tracing the Trail of the Upbit Hack*, n.d.). Existing similarity-based approaches embed such accounts far apart in the feature space (mirroring the left-side “Distance-based Matching” block

of Figure 1.2) and consequently produce an incorrect ownership linkage. In contrast, *EDL*, as visualized in the right-side “Inherently xAI-based Ownership Linkage Learning” block of the figure, leverages its supervised learning mechanism to explicitly account for these behavioral disparities and produce appropriately small model scores, enabling correct ownership linkage even under such *heterophilic* behavior. According to a set of comprehensive experiments given in Section 7.4, *EDL* achieves an F1-score of 0.87, outperforming the previous state-of-the-art method with a relative improvement of 2.7%. It also maintains a performance advantage across varying anonymity set sizes, achieving up to a 32% improvement over existing solutions. These results support effectiveness of the proposed solution.

On the explainability side, as outlined bottom of Figure 1.2, existing solutions typically rely on deep-learning-based black-box models, which cannot provide inherent explanations and therefore require post-hoc frameworks such as LIME (Ribeiro et al., 2016) or SHAP (Lundberg & Lee, 2017)—methods that often yield approximate and potentially unreliable explanations (for the taxonomic description, please refer to Section 3.3). Contrary to the existing, *EDL* incorporates an inherently xAI-driven modeling approach that, as illustrated in right bottom of the figure, directly outputs instance-level contribution scores for the decision, enabling transparent and trustworthy reasoning for deanonymization. As exemplified right bottom of the figure, in a hot-cold account configuration, temporal indicators such as *avg_inter_tx_time* receive strong negative contributions for the bursty, high-frequency hot account while volumetric features like *incoming_tx_rate* contribute differently for the sparse and dormant cold account, collectively revealing why two behaviorally divergent accounts may still be linked to the same owner. Section 7.5 further demonstrates this capability through a real in-the-wild case study, where *EDL* achieves both more accurate and more explainable deanonymization than existing approaches. Due to such reliable explainability properties, to the best of our knowledge, *EDL* is the first *AOL* solution that enables practical human-AI collaborative blockchain investigation.

In summary, the primary contributions of this dissertation are as follows:

- To the best of our knowledge, we introduce a novel deanonymization solution named *EDL*, which uniquely addresses the AOL problem in Ethereum by explicitly leveraging behavioral divergences among accounts.
- We conduct a systematic comparison of *EDL* with state-of-the-art approaches and demonstrate its superior performance as well as its practical advantages in effectively resolving the AOL problem.
- We perform an extensive and robust analysis of *EDL* under varying anonymity set sizes, showing that the proposed solution maintains a consistently high performance even in settings with large anonymity set size (please see Section 7.4).
- We further evaluate *EDL* in a real-world blockchain investigation case study (please refer to Section 7.5), illustrating its practical benefits and its enhanced level of explainability, which clearly differentiates it from existing solutions.

The remainder of this dissertation is structured as follows. Chapter 2 reviews the relevant literature on blockchain deanonymization, highlighting key studies and existing limitations. Chapter 3 presents the technical background required to understand the concepts and methods employed throughout the dissertation. Chapter 4 formally defines the research problem addressed in this work, while Chapter 5 introduces key points of the proposed solution. Chapter 6 provides a detailed methodological description of *EDL*, outlining its architecture and underlying mechanisms. Chapter 7 reports the experimental evaluation and analyzes the results. Chapter 8 discusses practical considerations related to the proposed solution. Finally, Chapter 9 concludes the dissertation by summarizing the main findings and outlining potential directions for future research.

2. LITERATURE REVIEW

2.1 Deanonymization Solutions over Bitcoin (BTC)

Although the concept of distributed ledgers predates Bitcoin, blockchain systems gained widespread scholarly and practical attention following the introduction of Bitcoin (BTC) blockchain system in 2008, which—through its proof-of-work consensus mechanism and elegant solution to the double-spending problem—established the first fully decentralized and trustless digital currency platform in the form of blockchain system (Nakamoto & Bitcoin, 2008). The BTC introduced the concept of a publicly verifiable and immutable distributed ledger, enabling trustless financial transactions among anonymous accounts. This innovation not only laid the foundations for decentralized finance (DeFi) an ecosystem enabling peer-to-peer financial services without traditional intermediaries—but also demonstrated how public blockchain architectures can support pseudonymous interactions, even though privacy is not an inherent design property of all blockchain systems, particularly permissioned or private variants.

Since blockchain systems enable users to perform transactions through cryptographic account addresses that are not associated with any descriptive or personally identifiable information, the system naturally facilitates anonymous transactional activity. This design, while originally intended to ensure user privacy, has also paved the way for misuse, as evidenced by numerous investigative reports revealing that malicious actors exploit blockchain systems for illicit purposes such as money laundering, ransomware payments, and unauthorized cross-border fund transfers (please refer recent crypto crime reports (Chainalysis Team, 2023, 2024; Elliptic Enterprises Limited, 2025)). Therefore, accurately detecting and tracing malicious fund transfers necessitates the development of effective deanonymization techniques for blockchain systems and motivates further research efforts in this direction.

Deanonymization over cryptocurrency transactions have been formulated in various different ways/stages in the literature, including account categorization, transaction analysis, and funds. These studies generally study to infer associations between blockchain addresses and their underlying real-world owners, enabling the identification of malicious account owners and the

accurately tracing their funds. In what follows, we highlight the most relevant strands of this research; for a comprehensive survey of the broader deanonymization landscape, see (Ziegler et al., 2025).

As an initial solution towards to the deanonymization problem for blockchain systems, Reid and Harrigan analyzed the BTC transaction graph to demonstrate how heuristic rules can linkage account owners (Reid & Harrigan, 2013). Their dual-layer graph model captured both transaction-level and user-level relationships, enabling the grouping of multiple account addresses under the same owner through *common-input* and *change-address* heuristics. By combining graph structures with off-network information, they identified several high-profile entities, including mining pools and hacker groups. However, the solution’s dependence on manually crafted heuristics constrains its generalizability, especially under adversarial or rapidly evolving transaction patterns.

In 2013, Meiklejohn et al. then advanced transaction-based deanonymization by conducting an extensive empirical study that correlated BTC addresses with real-world services such as Mt. Gox and Silk Road (Meiklejohn et al., 2013). Their methodology applied heuristic clustering to the complete transaction ledger and validated its results using external exchange data. This deanonymization process confirmed that pseudonymous accounts involved in illicit marketplaces could be reliably linked through transaction behavior and auxiliary metadata. However, their approach remains limited for the funds tracing as deanonymization problem, as heuristic clustering primarily supports fund-flow attribution rather than precise ownership linkage across diverse account behaviors.

Ron and Shamir implemented a quantitative analysis of the full BTC transaction graph, formalizing address-linking heuristics and statistically evaluating the connectivity of major exchange services such as Mt. Gox and Deepbit (Ron & Shamir, 2013). Their work demonstrated that the majority of BTC volume could be traced to a small number of centralized actors, emphasizing that anonymity on the blockchain is largely dependent on the concentration of transactional hubs. Yet, from an ownership-linkage focused deanonymization standpoint,

their analysis is constrained by its reliance on coarse transaction-hub structures, which reveal dominant actors but cannot resolve fine-grained address-owner relationships.

Building upon prior transaction-graph analyses, Baumann et al. applied large-scale data aggregation from blockchain.info to track the operational footprints of exchange services such as Mt. Gox (Baumann et al., 2014). Their approach utilized public API data to correlate high-frequency transactional clusters with identifiable commercial entities. However, because the method depends heavily on service-level aggregation signals, it offers limited resolution for inferring fine-grained ownership linkage across individual user accounts.

Spagnuolo et al. introduced a tool as an integrated deanonymization framework combining blockchain parsing, address clustering, and owner attribution modules (Spagnuolo et al., 2014). The tool automated the extraction of behavioral patterns and transactional relationships to infer ownership of Bitcoin addresses. Their results included the identification of an address linked to a contract ordered by the operator of Silk Road, underscoring the forensic potential of systematic deanonymization systems.

Fleder et al. expanded the deanonymization frontier by integrating social-layer information with blockchain data (Fleder et al., 2015). They cross-referenced Bitcoin forum user activity with on-chain transaction histories to infer direct and indirect links between pseudonymous forum accounts and known services, including Silk Road, WikiLeaks, and Satoshi Dice. This fusion of off-network behavioral traces and on-chain evidence illustrated the power of hybrid deanonymization approaches.

Similarly with prior transaction analysis research, Lischke and Fabian examined the BTC transaction network over its first four years, identifying ownership patterns of major actors and constructing a longitudinal map of address activity that revealed the gradual centralization of monetary flows (Lischke & Fabian, 2016).

2.2 Deanonymization Solutions over Ethereum (ETH)

Following extensive research on deanonymization within BTC and other early blockchain systems, scholarly attention has gradually shifted toward more complex, account-based

architectures with the emergence of Ethereum (Buterin, 2014). Unlike earlier cryptocurrencies that adopted a transaction-based model where each input and output could be analyzed independently, Ethereum introduced a fundamentally different paradigm by maintaining persistent accounts—stateful entities whose balances, nonces, and (for contract accounts) storage persist across blocks—and by enabling programmable smart contracts. This architectural shift substantially broadened the spectrum of user interactions and financial activities recorded on-chain, thereby introducing new layers of transparency alongside novel privacy vulnerabilities. Due to availability of autonomous contract-based operations, Ethereum blockchain systems have become target for malicious actors operating within its blockchain ecosystem. Its unique properties have been repeatedly exploited in large-scale illicit cases (*North Korean Hackers Return to Tornado Cash despite Sanctions*, n.d.; *The \$477 Million FTX Hack: A New Blockchain Trail*, n.d.; *Tracing the Trail of the Upbit Hack*, n.d.), demonstrating how Ethereum creates criminal opportunities. As a result, the necessity of developing robust deanonymization techniques for Ethereum has become increasingly evident, prompting a growing body of research aimed at uncovering behavioral and transactional patterns that may reveal hidden associations between accounts. Therefore, in this dissertation, we focus on the deanonymization problem over Ethereum blockchain. Specifically, in the rest of this chapter, we review deanonymization methods over Ethereum blockchain.

Prior research towards deanonymizing Ethereum accounts can be categorized in two major problem domains that have often been examined separately: Account Categorization (AC) and Account Owner Linkage (AOL). The AC-based deanonymization problem is concerned with the task of classifying accounts into predefined functional categories (e.g. miners, scam, hack, exchange) based on their behavioral and transactional characteristics. AOL, on the other hand, focuses on identifying and linking multiple accounts that are likely controlled by the same real-world owner. The primary motivation for AC is selectively applying sanctions onto (or to the extreme, completely ban) a subset of illicit accounts, while the motivation behind AOL is typically to trace coordinated or illicit on-chain activities.

As listed in Table 2.1, we provide a categorized chronological summary of existing studies about deanonymization of Ethereum accounts. In the rest of this chapter, we review each study in more detail.

2.2.1 Account Categorization (AC)

The first notable study addressing the AC problem in Ethereum was conducted by Payette et al., who performed an early behavioral analysis of the Ethereum address space (Payette et al., 2017). Their work aims to classify blockchain addresses according to their functional roles such as users, miners, exchangers, and smart contracts based on transactional behavior and interaction frequency. The authors employed clustering algorithms over extracted on-chain features, including transaction counts, gas usage, and temporal activity patterns, in order to identify structural similarities among accounts. The results provided one of the first empirical insights into the heterogeneity of Ethereum participants, demonstrating that behavioral profiling could serve as a foundation for subsequent deanonymization studies.

Building upon the above approach, Zhou et al. proposed a behavior-aware model for account deanonymization on Ethereum (Zhou et al., 2022). Their study introduced an *Account Interaction Graph (AIG)* that captures both direct transactional links and higher-order behavioral relationships among accounts. To effectively extract latent structural and temporal patterns, the authors used a hierarchical graph attention encoder capable of representing interactions across multiple layers of the network.

In a complementary study, Pragasam et al. adopted a machine-learning-driven approach to classify accounts according to their transactional behavior (Pragasam et al., 2023). Using a labeled dataset comprising over four thousand Ethereum addresses, the authors engineered 14 behavioral features (e.g. including balance dynamics, transaction frequency, and inter-transaction intervals) to train multiple ensemble classifiers such as Random Forest, Gradient Boosting, and XGBoost. Among these, XGBoost achieved the highest classification performance, correctly distinguishing between benign and malicious account categories, including exchange, scam, and phishing entities. Their results highlighted that behavioral regularities in transaction data can serve as implicit identifiers, facilitating behavioral characteristics of pseudonymous accounts.

While these AC-based deanonymization solutions provide valuable insights into the behavioral characteristics of blockchain accounts, their effectiveness in deanonymization remains limited when considering that malicious actors often exhibit heterogeneous behavioral patterns across multiple accounts during fund transfers. In such adversarial scenarios, relying solely on categorical behavior analysis may fail to uncover coordinated operations that span numerous pseudonymous addresses. Therefore, establishing inter-account relationships through AOL-based deanonymization approaches becomes indispensable for effective forensic investigation, particularly for accurately tracing illicit fund movements and linking malicious activities to their underlying controlling entities.

Table 2.1 An Overview of Deanonymization Studies for Ethereum accounts

Study	Year	Task	Approach
Payette et al. (2017)	2017	AC	Clustering
Victor and Friedhelm (2020)	2020	AOL	Heuristics
Beres et al. (2020)	2020	AOL	Profiling by quasi-identifiers
Tang et al. (2021)	2021	AOL	Mixer heuristics
Zhou et al. (2022)	2022	AC	Graph learning
Wang et al. (2023)	2023	AOL	Mixer heuristics
Hu et al. (2023)	2023	AOL	Tx sequence learning
Du et al. (2023)	2023	AOL	Graph feature learning
Pragasam et al. (2023)	2023	AC	Ensemble feature learning
Kilic and Inan (2024)	2024	AOL	Profiling
Our study (2025)	2025	AOL	Behavioral difference learning

2.2.2 Account Owner Linkage (AOL)

An early deanonymization method to address the AOL problem was presented in (Victor, 2020b), who introduced a set of heuristics for entity identification on Ethereum’s account-based architecture. Acknowledging that ownership inference is fundamental for analyzing network properties such as wealth distribution, user activity, and illicit fund flows, the study proposed clustering techniques based on observable transactional behaviors, including deposit address reuse, repeated participation in airdrops, and token transfer authorization patterns. These heuristics aimed to uncover groups of accounts likely controlled by the same entity, providing early quantitative evidence that a considerable share of Ethereum accounts exhibit coordinated behavior. While this rule-based approach effectively demonstrated the feasibility of entity-level linkage and remains a valuable methodological baseline, its dependence on deterministic transactional patterns limits its adaptability to evolving user strategies and obfuscation mechanisms within the blockchain ecosystem.

As an improved version of above work, Beres et al. addressed AOL problem by analyzing transactional correlations within Tornado Cash, a popular Ethereum-based mixer named Tornado Cash (Béres et al., 2020). Their study focused on establishing linkages between deposit and withdrawal transactions to uncover potential relationships among participating accounts. In addition to employing heuristic linkage strategies grounded in observable transactional patterns, the authors introduced the use of quasi-identifiers such as time-of-day activity and gas price values to profile user behavior. This work represents one of the first systematic attempts to incorporate quasi-identifiers into Ethereum deanonymization, demonstrating that even partial contextual features can reveal meaningful associations between accounts engaging in anonymous mixer transactions.

As a separate study, Tang et al., extended the preceding study by critically examining the gas price-based quasi-identifier proposed by Beres et al., observing that transactions with identical gas price values were, in most cases, initiated by the same account (Tang et al., 2021). This finding suggested that the gas price heuristic contributed little to linking distinct accounts based on same ownership and thus had limited practical value for effective account linkage. In addition, the authors performed a comprehensive statistical analysis of Tornado Cash

transactions and introduced several novel linkage patterns derived from recurring deposit–withdrawal behaviors, thereby proposing refined heuristics to enhance the accuracy of AOL-based deanonymization.

To generalize mixer usage analysis studies, Wang et al. conducted some empirical analysis involving both Tornado Cash and another Ethereum-based mixer, Typhoon, discussing cross-chain usage patterns and the impact of OFAC sanctions on mixer activity (Wang et al., 2023). Following the heuristic linkage strategies proposed in prior studies, the authors estimated the actual anonymity-set sizes to evaluate the effectiveness of existing heuristics under real-world conditions. Their findings revealed that, while these methods can successfully deanonymize transactions originating from careless users, the overall reduction in anonymity sets remains limited under ideal operational scenarios (e.g. real-world blockchain investigation). This outcome indicates that such heuristic approaches lack methodological robustness and adaptability, highlighting their restricted generalizability when confronted with evolving transaction behaviors (e.g. heuristic aware mixer usage).

To overcome the limitations of heuristic-based AOL deanonymization approaches, several machine learning–driven methods have been proposed in recent years. Hu et al. were among the first to address this direction by introducing *BERT4ETH*, a transformer-based deep learning framework that models Ethereum transactions as temporal sequences. By learning contextual dependencies within transaction orders, their model was able to capture behavioral regularities of account interactions, demonstrating experimentally that sequential behavioral profiling can play a critical role in solving the AOL problem. In contrast to this sequence-oriented paradigm, Du et al. proposed a solution named *MixBroker*, a graph-topology–based model that represents transactions as interconnected nodes and edges to infer ownership relationships among accounts. While both approaches achieved promising results, they ignored two critical aspects: the potential unreliability of homophilic behavioral assumptions underlying blockchain transactions, and the lack of transparency and explainability which is essential for robust blockchain investigation.

Overall, notice that almost all *AOL* studies assume that accounts controlled by the same owner exhibit similar behavior as the behavioral pattern of the owner. Therefore, these studies try to characterize the behavior of each account and match them to link account owners and solve AOL. A very limited number of studies (Béres et al., 2020; Kılıç & İnan, 2024) indicate that the pretense of *homophilic behavior* might be false: the same owner may behave differently on distinct accounts of her own. Our solution builds upon the findings of this second line of research. We too rely on behavioral patterns of accounts, but our solution is flexible enough to handle both homophilic and diverse behavior. The experimental results in Section 7.4 demonstrate that the proposed method *EDL* improves deanonymization performance by 15% to 32%, depending on the anonymity set size, outperforming existing state-of-the-art approaches. These findings confirm the superiority of the proposed method in comparison to established benchmarks.

Moreover, since learning-based state-of-the-art AOL solutions (Du et al., 2023; Hu et al., 2023) rely on deep-learning based black-box models that cannot provide intrinsic explainability, they require post-hoc xAI methods such as SHAP or LIME—approaches that often produce approximate and potentially unreliable explanations (Slack et al., 2020). This limitation is particularly critical in blockchain investigation, where trustworthy reasoning behind each linkage decision is essential. In contrast, our solution is able to handle this limitation by employing an inherently xAI-based machine learning model that directly generates feature-level importance scores (please refer to Section 6.2 for methodological details), enabling transparent and traceable justification of model decisions. To the best of our knowledge, this is the first AOL solution that enables practical human–AI collaborative blockchain investigation. The real-world case study presented in Section 7.5 further demonstrates the practical effectiveness of this explainability capability.

3. BACKGROUND

In this chapter, preliminary concepts related to this dissertation are described.

3.1 Ethereum Blockchain as a Distributed Ledger

Blockchain technology relies on a distributed ledger architecture, wherein transaction records are replicated across a decentralized network. Unlike traditional systems that depend on a single organization, distributed ledgers address the limitations of central authorities by guaranteeing that every node independently verifies the state of the system. Consequently, this replication enables the ledger to remain consistent even in the presence of adversarial participants, as data correctness is enforced through a consensus mechanism (Buterin & Griffith, 2017; Nakamoto & Bitcoin, 2008).

While distributed ledgers are not a novel concept, blockchain technology relies on cryptographically linked data structures and append-only update semantics. Specifically, each block aggregates validated transactions and metadata necessary for consensus agreement. As discussed in prior work (e.g., (Nakamoto & Bitcoin, 2008; Yaga et al., 2018)), this chained structure addresses the limitations of traditional data integrity mechanisms; under reasonable assumptions regarding the computational power of the adversary, the protocol guarantees the immutability of historical data.

Another defining property of blockchain relies on the transparency of its state, wherein all transactions are globally visible. Specifically, this visibility guarantees that any participant can audit the evolution of account balances and the sequence of events that produced the current ledger state.

Among the earliest practical implementations of this paradigm is Bitcoin, introduced in 2008, which demonstrated the feasibility of such a transparent, append-only ledger maintained without central authority. However, while Bitcoin established the foundational model for peer-to-peer value transfer, subsequent platforms have been proposed to address the limitations of simple monetary transactions. In this regard, Ethereum distinguishes itself by providing a general-purpose, programmable execution environment through smart contracts. Therefore,

this capability enables automated interactions and application logic to run directly on the blockchain.

Each entity on Ethereum is represented by an account, which can hold Ether (ETH), interact with other accounts, and maintain state. Accounts are divided into two categories: *Externally Owned Accounts (EOAs)* and *Contract Accounts (CAs)*. EOAs are controlled by users who hold private keys and can initiate transactions. CAs, in contrast, are governed entirely by smart contract code: they cannot create transactions independently but execute predefined logic whenever invoked by an external trigger, typically from an EOA.

Transactions on Ethereum occur either directly between EOAs or between an EOA and a CA acting as an intermediary. When a transaction is executed, the Ethereum Virtual Machine (EVM) processes it deterministically, updating the global state. Unlike Bitcoin's limited scripting language, Ethereum's Turing-complete execution environment allows CAs to embed complex business logic, enabling the development of decentralized finance (DeFi) protocols, non-fungible tokens (NFTs), and decentralized autonomous organizations (DAOs).

From the conceptual perspective, transactions are grouped into blocks validated through the Proof-of-Stake consensus mechanism. This chained structure guarantees the integrity and immutability of the ledger by maintaining a cryptographic reference to the previous block.

From the anonymity perspective, cryptographic primitives such as elliptic-curve digital signatures, hash functions, and Merkle Patricia trees underpin Ethereum's transaction and state management (Buterin, 2014; Wood, 2014). Each Externally Owned Account (EOA) is secured by a private key under the secp256k1 elliptic curve. From this key, a public key is derived and hashed with Keccak-256, the final 20 bytes of which form the user's address (Gilad & Herzberg, 2017). This mechanism ensures that accounts can be generated in an unbounded manner and remain unlinkable to real-world identities at the protocol level. The cryptographic one-wayness of Keccak-256 makes it computationally infeasible to reverse the process from an address back to the private key, thereby providing strong key ownership guarantees. Such Ethereum's pseudonymity arises from this address-generation scheme: users interact through hexadecimal

identifiers rather than explicit identities. However, unlike privacy-focused protocols that incorporate zero-knowledge proofs or confidential transactions, Ethereum does not conceal the transaction graph itself. Every transfer, smart contract call, and balance update is permanently visible on-chain, timestamped, and ordered within blocks by the consensus mechanism. The Ethereum Virtual Machine (EVM) processes each transaction deterministically, updating the global state, while Merkle Patricia trees maintain verifiable proof of inclusion for every account and contract entry.

3.2 Formal Definitions

Raw block data for the Ethereum blockchain is publicly accessible and can be downloaded locally by setting up a full node. Alternatively, tools such as the Etherscan API² provide efficient and user-friendly access to blockchain data without requiring a local setup. Descriptions of these publicly available Ethereum content, along with the notations used in subsequent discussions, are detailed in Table 3.1.

Figure 3.1 presents the ER diagram illustrating key entities—Wallet Owner, Account, Transaction, and Block—along with their properties. Specifically, the Wallet Owner entity highlights the primary objective of this study: demonstrating the mechanism wherein users create multiple accounts under a single wallet to enhance anonymity. As indicated in the diagram, each transaction is initiated by a single account, but an account can engage in multiple transactions as either a sender or recipient. Please notice that the wallet owner entity is used instead of account owner to emphasize user-level control over multiple accounts managed under a single wallet interface; hereafter, both terms are used **interchangeably** for simplicity. This one-to-many relationship is essential for understanding user behavior and the mechanisms supporting anonymity within the Ethereum blockchain.

Definition 1 (Ethereum Transaction). An Ethereum transaction T_i represents a state transition recorded on the Ethereum blockchain and is modeled as a structured tuple of observable attributes drawn from the raw block data.

² <https://docs.etherscan.io/>

Formally, $T_i = (T_i.\tau, T_i.S, T_i.R, T_i.V, T_i.P, T_i.C, T_i.G, T_i.E, \dots)$, where $T_i.\tau$ denotes the timestamp of the transaction, $T_i.S$ the sender address, $T_i.R$ the recipient address, $T_i.V$ the transferred amount of Ether, $T_i.P$ the gas price, $T_i.C$ the contract address if applicable, $T_i.G$ the gas consumed, and $T_i.E$ an error flag indicating the success or failure of execution. Additional attributes such as transaction hash, block number, nonce, input data, and function name are also associated with T_i as specified in Table 3.1.

Table 3.1 Information about raw Ethereum data

Field	Description
Block Number	Block number of the transaction
Timestamp	Timestamp of the transaction
TxHash	Transaction hash value
Nonce	Transaction nonce
Block Hash	Block hash value
Transaction Index	Index of the transaction in the block
From	Sender address
To	Recipient address
Value	Transferred amount
Gas	Gas for the transaction
Gas Price	Price of gas per unit
isError	Error flag for the transaction
TxReceipt Status	Transaction receipt status
Input	Input data of the transaction
Contract Address	Contract address involved
Cumulative Gas Used	Cumulative gas used up to this transaction
Gas Used	Gas consumed for this transaction
Confirmations	Number of confirmations for the transaction
MethodID	Method ID of the transaction
Function Name	Smart contract function name

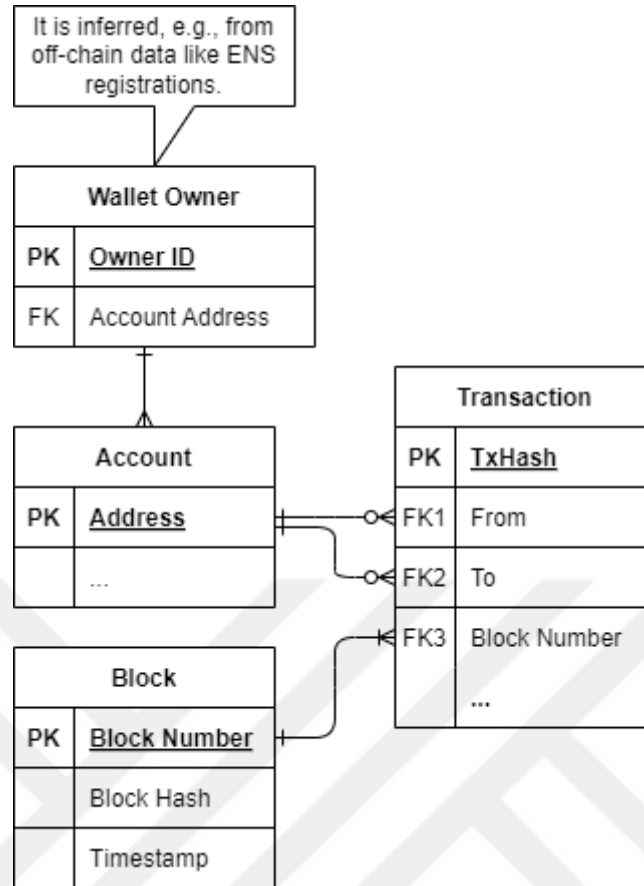


Figure 3.1 ER diagram of Ethereum data

Each transaction T_i is cryptographically linked to a unique block via its block number and block hash and is ordered within the block by its transaction index. As shown in Figure 3.1, every transaction originates from exactly one account $T_i.S$, and may either transfer value to another externally owned account ($T_i.R$) or invoke a contract at address $T_i.C$. Transactions collectively form the full transaction set $\mathcal{T}$, which captures both account-to-account transfers and contract interactions.

At the user level, a wallet owner may control multiple accounts, thereby initiating or receiving multiple transactions across different addresses. Specifically, this one-to-many mapping between wallet owners and accounts is critical for the analysis of anonymity that relies on ownership unlinkability in Ethereum.

Please notice that the notion of anonymity considered here specifically corresponds to ownership linkage anonymity, also referred to as *ownership unlinkability property*, which captures the inability of an attacker to establish linkages between different addresses controlled by the same owner. This is distinct from other forms of anonymity such as transaction or network-level anonymity.

3.3 Explainable Artificial Intelligence(xAI) Methods

Explainable Artificial Intelligence (xAI) methods aim to make machine learning (ML) models more interpretable by providing human-understandable explanations for their predictions. The increasing use of opaque models such as deep neural networks in sensitive domains (e.g., finance, healthcare, and blockchain forensics) has intensified the demand for transparency and accountability. From a methodological perspective, xAI techniques can be broadly categorized into two groups: post-hoc and inherent methods (see the recent survey in (Schwalbe & Finzel, 2024)). Although both aim to clarify the reasoning behind model decisions, they differ fundamentally in how explanations are produced and how faithfully they reflect the model's internal logic.

Post-hoc approaches (e.g., SHAP (Lundberg & Lee, 2017), LIME (Ribeiro et al., 2016), Integrated Gradients (Sundararajan et al., 2017)) generate explanations after model training. These techniques typically approximate the decision behavior of black-box models by perturbing inputs and observing changes in model outputs. While such approximations can provide useful insights, they do not necessarily align with the model's true reasoning process. Recent studies (Amparore et al., 2023; Bove et al., 2024; Slack et al., 2020) highlight this limitation and argue that many post-hoc explanations may be unreliable, since they operate independently of the model's internal mechanisms. As a result, post-hoc xAI often enhances interpretability at the surface level but falls short in delivering epistemic transparency.

Contrary to post-hoc approaches, inherently xAI-based learning models are designed such that interpretability is embedded in their structure. These models allow the influence of each input feature to be directly observed and quantified. Therefore, inherent xAI approaches provide model-intrinsic transparency, offering a more reliable and reproducible understanding of why a given decision was made. Specifically, in domains where auditability and accountability are

critical for many applications, inherently interpretable models rely on transparent architecture. This mechanism guarantees that the produced explanations are both trustworthy and faithful to the model's internal mechanics. Consequently, such models are favored as they address the limitations of opaque decision-making systems.

Technically, an inherently xAI-based ML model learns a function $F: \mathbb{R}^d \rightarrow \mathbb{R}$ that maps a vector of input features $\vec{x}$ to a target variable y , given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$. Among various regression models, some approaches incorporate built-in mechanisms for explainability, enabling direct interpretation of their decision-making processes. These *inherently explainable AI models* provide insights into how input features influence predictions without requiring separate post hoc analysis.

Two common categories of inherently explainable models include *feature importance-based methods* and *rule-based decision structures*. Feature importance methods assess the contribution of each input feature to the model's output. A gradient-based approach quantifies the importance of feature x_j as:

$$I(x_j) = \sum_{i=1}^n \left| \frac{\partial F(\vec{x}_i)}{\partial x_j} \right| \quad (3.1)$$

which captures the sensitivity of the model's output to changes in x_j . This allows for direct assessment of how variations in individual features impact predictions.

Another category includes *rule-based models*, such as decision lists and rule ensembles, which express $F(\vec{x})$ using interpretable conditions:

$$F(\vec{x}_i) = \sum_{k=1}^K w_k \cdot I(\vec{x}_i \in R_k) \quad (3.2)$$

, where $I(\vec{x}_i \in R_k)$ is an indicator function specifying whether $\vec{x}_i$ satisfies rule R_k and w_k represents the corresponding weight. These models provide structured and interpretable decision logic, making them useful in applications where understanding predictions is essential.

3.4 Ciphertext Indistinguishability Concept

The notion of ciphertext indistinguishability originates from the seminal work of Goldwasser and Micali (Goldwasser & Micali, 1982, 1984), which laid the foundation of modern computational cryptography. In their influential paper, they introduced the idea that encryption should conceal all partial information about the plaintext, providing the first rigorous definition of *semantic security*. This principle was later reformulated into a family of game-based indistinguishability definitions, each differing by the capabilities granted to the adversary.

One of the most popular formalizations is indistinguishability under chosen-plaintext attack (IND-CPA), which captures the requirement that an adversary, even when allowed to obtain ciphertexts for plaintexts of its choice, cannot distinguish between the encryptions of two equal-length messages. Formally, an encryption scheme is IND-CPA secure if any probabilistic polynomial-time adversary has only a negligible advantage in correctly guessing which of the two chosen plaintexts corresponds to a given challenge ciphertext.

Technically, the general attack steps can be formulated a game between attacker and challenger as follows:

1. The attacker selects n plaintexts $m_1, \dots, m_n$.
2. The attacker submits all these n plaintexts to the encryption oracle $\mathcal{O}_{\text{enc}}(\cdot)$.
3. The encryption oracle returns the corresponding ciphertexts $c_i \leftarrow \text{Enc}_k(m_i)$ for each submitted plaintext.
4. The attacker collects all (m_i, c_i) pairs and thus can learn the plaintext–ciphertext relations for those queries.
5. Based on the obtained plaintext–ciphertext pairs, the attacker attempts to recover the private key used by $\mathcal{O}_{\text{enc}}(\cdot)$ (or otherwise exploit the information to break confidentiality).

After these steps, as the final stage of the game, the attacker submits two challenge plaintexts m_0 and m_1 to the challenger. The challenger selects a uniformly random bit $b \leftarrow \{0,1\}$, computes the challenge ciphertext $c^* \leftarrow \text{Enc}_k(m_b)$, and returns c^* to the attacker. The attacker then attempts to guess which message was encrypted by outputting a bit $\hat{b}$.

The challenge ciphertexts are said to be *indistinguishable* in the chosen-plaintext attack if the attacker cannot guess b significantly better than random. Formally, indistinguishability requires that the attacker's advantage $\text{Adv}^{\text{IND-CPA}}(\text{Attacker}) = |\Pr[\hat{b} = b] - 0.5|$ is negligible (i.e., asymptotically smaller than any inverse polynomial) with respect to the security parameter. Equivalently, the event $\hat{b} = b$ must not be known to the attacker with a probability that is non-negligibly greater than 0.5. In this dissertation, we design a novel game that adapts this paradigm to the context of AOL problem to formulate our research problem. For details, please refer to Section 4.2 for technical formulation of our research problem.



4. PROBLEM DEFINITION

This chapter defines the research problem of this dissertation. As reviewed in Chapter 2, existing studies about deanonymization of Ethereum accounts through two distinct approaches: account categorization (AC) and account-owner linkage (AOL). Since this study focuses on linking accounts controlled by the same owner, it falls into the AOL category.

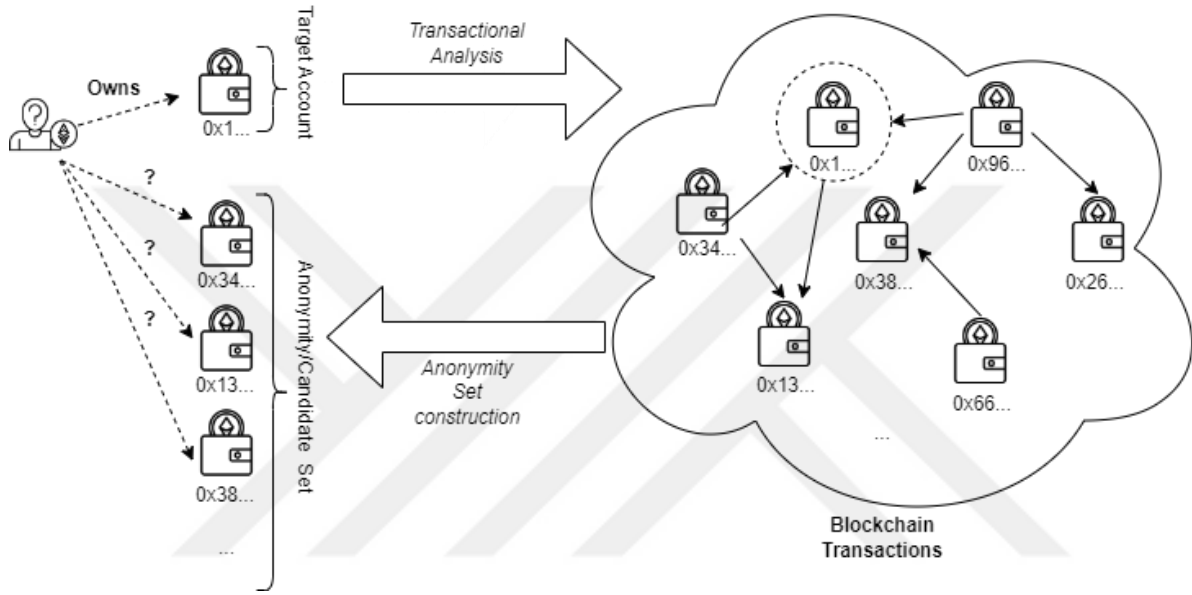


Figure 4.1 Illustration of the research problem

4.1 Problem Description

In this study, we address the AOL problem from the perspective of a real-world blockchain investigator. As illustrated in Fig. 4.1, the investigation starts from a target account flagged for scrutiny and analyzes its transactional behavior using data extracted from the public Ethereum blockchain, as summarized in Table 3.1. Based on these observations, the investigator leverages domain expertise or established heuristics to construct an anonymity set consisting of accounts that are likely controlled by the same owner.

Please note that the core challenge of the problem lies in accurately identifying the specific account, among an anonymity set of K candidate accounts, that is controlled by the same owner. Additionally, making the identified account explainable is essential for enabling the for

enabling blockchain investigators to conduct more reliable analyses within the framework of human–AI collaboration in blockchain investigations. Accordingly, the decision-making mechanism should be transparent and verifiable from the perspective of investigators.

4.2 Problem Formulation

To formulize our research problem, we design a novel game by inspiring from IND-CPA concept (please refer to Section 3.4). Please recall from IND-CPA, an adversary adaptively selects plaintexts and receives their corresponding cipher texts, attempting to determine which plaintext was encrypted in the challenge phase. Similarly, in our formulation, the attacker selects observable entities (source addresses) and receives the corresponding publicly available transactions from the ledger. The challenge then requires the attacker to determine which address, among a set of K candidate addresses, exactly one shares the same ownership as the chosen source. So, traditional IND-CPA setup extended to quantify how well an adversary can correctly distinguish between possible ownership relations in the anonymity set. Overall, the key idea is to capture how effectively an attacker can correlate Ethereum account ownership based on observable transactional behavior, analogous to how IND-CPA measures an attacker’s ability to distinguish ciphertexts under chosen plaintexts.

Similar to the IND-CPA game, our formulated game named *K-Ownership Unlinkability Game* models the interaction between a challenger and an adversary in the context of AOL-based deanonymization problem. In this formulated game, the structure of the game mirrors the IND-CPA game, where ciphertexts are replaced by observable ledger traces, and plaintexts correspond to blockchain addresses within an anonymity set.

Technically, the game proceeds as follows: The attacker selects a target account address a_0 and an anonymity set $S = \{a_1, a_2, \dots, a_K\}$ from the global account space $\mathcal{A}$. These accounts are analogous to the plaintext candidates (m_0, m_1) in the IND-CPA setting, where exactly one element in S shares ownership with a_0 . The challenger, acting as the oracle, assigns ownership relations and determines the true same-owner index $j^* \leftarrow \{1, \dots, K\}$, ensuring that $\mathcal{O}_L(a_0, a_{j^*}) = 1$, while for all $i \neq j^*, \mathcal{O}_L(a_0, a_i) = 0$. Next, the challenger returns to the attacker the corresponding public transactions $\mathcal{T}(a_0), \mathcal{T}(a_1), \dots, \mathcal{T}(a_K)$, where $\mathcal{T}(\cdot)$ denotes the public

transformation mapping an account to its on-chain behavioral. This step directly parallels the IND-CPA challenge phase, where the ciphertext $c^* = \text{Enc}_k(m_b)$ hides the identity of the underlying plaintext.

Finally, the attacker outputs a guessed index $\hat{j} \in \{1, \dots, K\}$, attempting to identify which address in S belongs to the same owner as a_0 . The adversary's success probability is defined as $\Pr[\hat{j} = j^*]$, and the advantage over random guessing is given by $\text{Adv}^{\text{Unlink}}(K) = \left| \Pr[\hat{j} = j^*] - \frac{1}{K} \right| \leq \text{negl}(K)$.

Definition 2 (K- Ownership Unlinkability Game). Ethereum satisfies the K-ownership unlinkability property if, for every polynomial attacker $\mathcal{Att}$, there exists a negligible advantage over random guessing $1/K$ such that $\text{Adv}_{\text{A,L}}^{\text{Unlink}}(K) = \left| \Pr[\hat{j} = j^*] - \frac{1}{K} \right| \leq \text{negl}(K)$.

Notice that the main challenge is to check whether an attacker can break this anonymity with non-negligible advantage. Building on this foundation, the problem addressed in this work is to develop a comprehensive and powerful model that is capable of successfully breaking such anonymity, thereby instantiating the AOL problem. Notationally, consider a target account with a_0 and the anonymity set with as $S = \{a_1, \dots, a_K\}$ where $K > 1$.

Under this setup, we formulate the AOL problem as a ownership linkage learning problem. Specifically, the goal is to learn a model as follows:

$$M: X_0^F \times \{X_1^F, X_2^F, \dots, X_K^F\} \rightarrow j^* \in \{1, 2, \dots, K\} \quad (3.2)$$

such that the predicted candidate account index j^* correctly identifies that a_j and a_{j^*} belong to the same owner, i.e., $(E(a_j, a_{j^*}) = 1)$.

Moreover, to facilitate collaboration with blockchain investigators, the constructed model M should inherently provide explanations, ensuring transparency in its decision process. Since the investigator validates and interprets predictions, M should also highlight the most influential

features $F' \subseteq F$ in linking addresses. These features, represented as elements of the profile vector X_j^F , may include behavioral characteristics such as temporal and volumetric behavioral features.

Overall, intuitively, in theoretical perspective, an attacker cannot determine whether the addresses in anonymity set belong to the same owner with probability significantly better than random guessing. Thus, *ownership unlinkability* represents a fundamental anonymity guarantee of the Ethereum system: while transactions are transparent and publicly verifiable, the mapping between owners and their addresses should remain hidden. However, in practice, this guarantee can be broken by behavioral analysis as reviewed in Chapter 2. Moreover, our proposed model named *EDL* can break this type of linkage anonymity by leveraging behavioral patterns across multiple accounts as theoretically demonstrated in Section 5.2.

5. SOLUTION OVERVIEW

This section presents the foundation of the proposed solution for deanonymization of malicious Ethereum users named *EtherDiffLinkage* (EDL) as attacker model and theoretically demonstrates how its corresponding attack model $\mathcal{Att}_{EDL}$ can break ownership unlinkability anonymity as described in Definition 2 of Section 4.2.

5.1. Motivational Background

As detailed Section 3.1, the anonymity of Ethereum users is not solely determined by cryptographic primitives but also depends on behavioral properties. Prior research in blockchain forensics has emphasized that observable user activity can be analyzed along multiple behavioral dimensions, including transaction timing (Anoica & Levard, 2018; Lee et al., 2020; Zanelatto Gavião Mascarenhas et al., 2020), transaction volumes (Ferretti & D’Angelo, 2020; Guo et al., 2019), and infrastructural pathways (Béres et al., 2020; Sun et al., 2019; Tang et al., 2021; Victor, 2020a). Although such categorization is not exhaustive, various studies show that it can provides a useful lens through which to examine how malicious actors employ obfuscation to frustrate ownership attribution (e.g Agarwal et al., 2022; Oliveira et al., 2022; Zhou et al., 2022).

Although malicious Ethereum addresses can in principle be profiled by their on-chain behavior, adversaries commonly deploy obfuscation tactics that target observable behavioral axes. For example, on the temporal axis, attackers disturb timing signals by inserting irregular delays, randomizing inter-transaction intervals, splitting bursty sequences into time-spread transfers, or using relayers/private-publication channels to introduce unpredictable propagation hops, all of which have been shown to reduce the effectiveness of timing-based linkage and classifier features (Dorri et al., 2019). Besides, volumetric strategies fragment large payments into many small outputs (peel-chain / splitting) and vary amounts to defeat amount-fingerprinting; these patterns and their deanonymization/traceability implications are studied in several empirical works (Gong et al., 2023; Kappos et al., 2022). Overall, such tactics are frequently combined in practice, single-axis heuristics quickly lose discriminative power; robust forensic methods therefore need to explicitly consider such obfuscation strategies.

Another important dimension concerns the behavioral heterogeneity of addresses controlled by the same owner. In practice, a single individual or organization may operate multiple Ethereum accounts for distinct purposes such as trading, staking, custody, or fee management (Oliveira et al., 2022; *The Best Crypto Wallets of March 2023 – Forbes Advisor*, n.d.; *The Complete Guide to Web3 Wallets. Thirdweb, Jan. 13, 2023*, n.d.). These role-specific usages produce consistent yet distinct behavioral patterns, meaning that addresses under common ownership can intentionally appear dissimilar (Hu et al., 2023). Such diversity has been observed in both legitimate and malicious contexts, where the same actor manages separate wallets for operational separation, liquidity control, or security isolation (*The \$477 Million FTX Hack: A New Blockchain Trail*, n.d.; *Tracing the Trail of the Upbit Hack*, n.d.). Hence, effective deanonymization should not focus solely on detecting similarity but also on recognizing structured behavioral differences that reflect functional roles. Modeling such inter-account variation enables more accurate linkage, allowing analyses to remain robust against obfuscation while capturing implicit signals of coordinated multi-account activity (Oliveira et al., 2022; *The Best Crypto Wallets of March 2023 – Forbes Advisor*, n.d.).

5.2. Methodological Foundation

As noted in Section 3.1, in addition to the formal security guarantees, the effective anonymity of Ethereum users is primarily maintained through operational and behavioral measures — for example, avoiding address reuse, employing transaction-mixing services or privacy-preserving Layer-2 solutions, splitting funds across multiple pseudonymous addresses, and obscuring network-level metadata (e.g., via Tor or VPN). This reliance on user practices means that protocol-level pseudonymity does not automatically equate to *ownership unlinkability* (refer to Definition 2 of Section 4); rather, privacy outcomes depend heavily on how users manage keys and construct transactions, and on whether they adopt supplementary privacy tools or follow disciplined operational security. However, this anonymity might be vulnerable to a novel deanonymization attack named *EDL*, which can be launched by an attacker model $\mathcal{Att}_{EDL}$. In this attack scenario, the attacker collects all Ethereum transactions along with a set of reliably linked account pairs and leverages corresponding behavioral information to associate, within the anonymity set, accounts that are controlled by same owner.

Specifically, $\mathcal{Att}_{EDL}$ leverages behavioral differences that are derived from behavioral attributes to construct a set of ground-truth labels representing distinct account ownership patterns to model learning target accurately. Thus, the AOL problem introduced in Section 4.2 is transformed into an ownership linkage learning task based-on possible linked account pairs. To capture these inter-account relations as accurately as possible, a well-trained model is employed to learn from behavioral difference features and then subsequently perform the inference. In the remainder of this subsection, we discuss the design choices as argument (See Argument 1 and 2), construct the model by formalizing behavioral-difference representations and the corresponding learning target, and then demonstrate the theoretical potential of the attacker model $\mathcal{Att}_{EDL}$ (see Conjecture1).

Argument 1 (Behavioral Difference Representation). Behavioral differences between Ethereum accounts can be represented in various ways. A natural approach would be to encode proportional deviations across features, thereby reflecting relative variation in behavioral patterns such as transaction frequency, transfer size, or gas usage. However, such an approach immediately faces difficulties in stability and explainability when applied to real-world data.

Consider the following scenario. Account a_1 executes a large number of very small transfers (e.g., 100 transactions of \$0.01\$ ETH), whereas account a_2 performs only a handful of significantly larger transfers (e.g., 5 transactions of \$0.20\$ ETH). In principle, one might attempt to capture their behavioral dissimilarity by computing relative differences across features. The proportional formulation, expressed as $\overrightarrow{z}_{\text{rel}} = \left| \frac{\overrightarrow{x}_i - \overrightarrow{x}_j}{\overrightarrow{x}_i} \right|$, seems attractive since it emphasizes how much larger or smaller one account's behavior is relative to another. However, the method introduces several complications. It becomes undefined whenever a denominator coordinate is zero, unstable whenever values approach zero, and biased by sign dependence. For instance, if account a_1 performed a few negligible transfers close to 0 ETH, the proportional difference for transaction value would artificially inflate to extremely large values, even though such microtransactions are investigatively irrelevant.

An alternative representation might be the absolute difference as $\overrightarrow{z}_{\text{abs}} = |\overrightarrow{x}_i - \overrightarrow{x}_j|$. Unlike the relative representation, this measure is finite for all normalized inputs, symmetric in its

arguments, and invariant under sign reversal. Returning to the previous example, the absolute gap between 0.20 ETH and 0.01 ETH is recorded as 0.19 ETH—transparent, finite, and interpretable. From an explainability perspective, investigators can report that account a_2 conducts on average 0.19 ETH larger transfers than account a_1 , which is both stable and faithful to observed behavior. In contrast, describing the same relation as "1900% larger transfers" exaggerates the difference due to scale effects and hinders interpretability.

Notice that alternative measurements such as norm-based distances, cosine similarity, or information-theoretic divergences could in principle be employed to represent behavioral differences; however, we exclude these methods since they hinder feature-wise explainability in the context of Ethereum investigations. Therefore, we adopt the absolute difference formulation as the behavioral difference representation.

Argument 2 (Dissimilarity-based learning target). A suitable learning model over behavioral difference vectors between Ethereum accounts can be formulated in two distinct ways. On the one hand, one may adopt a similarity-based objective that rewards pairs with small behavioral gaps, or alternatively a dissimilarity-based objective that emphasizes and preserves behavioral divergence. The question, however, is which formulation better reflects the reality of the ownership linkage problem. As motivated in Section 5.1, users on Ethereum often manage multiple addresses, each dedicated to a specific function such as interacting with decentralized exchanges, engaging with DeFi lending protocols, collecting airdrops, or transferring funds to custodial wallets. Prior work has shown that such separation of roles is intentional, serving both organizational and privacy purposes, and that it leads to heterogeneous behavioral signatures across accounts owned by the same individual. Consequently, a similarity-based model that attempts to minimize behavioral differences across pairs may not align with the actual structure of user behavior. Two addresses under the same owner may legitimately differ in transfer frequency, token diversity, or contract interaction patterns, not because they belong to different users, but because they serve different roles within that user’s activity portfolio. Under these conditions, a similarity-oriented objective tends to compress informative variation, projecting accounts onto a narrow space of behavioral resemblance. Consider, for example, two addresses that both interact with a popular exchange contract or

engage in routine ERC-20 transfers. A similarity-driven model would overvalue such superficial overlap, even though it provides little evidence of shared control. The discriminative power of behavioral features is thereby weakened, as the model learns to ignore subtle but meaningful deviations that actually encode ownership distinctions.

On the other hand, a dissimilarity-based learning formulation offers a more contrastive and realistic perspective. Rather than amplifying resemblance, it seeks to expose and preserve differences that are stable, structural, or difficult to forge. Infrastructural traces such as the provenance of initial funds, persistent contract dependencies, or long-range transaction topologies cannot be easily obfuscated or retroactively modified. When dissimilarities arise in these features, they provide reliable evidence that two accounts are not linked, even if their superficial usage patterns appear aligned. In this sense, learning dissimilarity acts as a normalization over behavioral space: it constrains the model to attend to invariant, high-signal deviations while preventing common transactional motifs from overshadowing critical distinctions.

We therefore argue that, within the Ethereum ownership linkage domain, a dissimilarity-based learning objective constitutes a more favorable and practical design choice. Besides, by focusing on why two accounts are not behaviorally identical, the model produces explanations that are inherently contrastive and feature-level, making its reasoning both interpretable and forensically meaningful. Consequently, treating behavioral differences as the principal signal and learning to maximize their explanatory contribution offers a pragmatic balance between realism, robustness, and explainability.

Conjecture 1: Let $\mathcal{Att}_{EDL}$ denote an attacker model with access to the full transaction ledger $\mathcal{L}$.

Then $\mathcal{Att}_{EDL}$ can have theoretical feasibility to break the ownership unlinkability property of Ethereum users (refer to Definition 2) at non-negligible advantage.

Proof sketch: Define a set of Ethereum accounts $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ where each account a_i is associated with a vector of behavioral features extracted from $\mathcal{L}$. Let $\mathcal{P} = \{(a_i, a_j, y_{ij})\}$ denote the set of candidate account pairs with corresponding ownership label.

Assume that $\mathcal{Att}_{EDL}$ obtains a labeled subset $\mathcal{P}_{train} \subset \mathcal{P}$, with labels $y_{i,j} \in \{0,1\}$ indicating whether (a_i, a_j) represents the same owner ($y_{i,j} = 0$) or distinct owners ($y_{i,j} = 1$). Based on $\mathcal{P}_{train}$, the attacker $\mathcal{Att}_{EDL}$ may learn a dissimilarity scoring function from behavioral feature vectors as $F: R^d \rightarrow R$ and $F(\Delta \vec{x}_{i,j}) \mapsto s_{i,j}$, where $s_{i,j}$ is the dissimilarity score of the pair.

Given a target account a_i and a candidate anonymity set $\mathcal{C} = \{a_j\}$ of possible linked accounts, the attacker predicts the true counterpart by selecting the pair with the minimal score as

$$a_j^* = \arg \min_{a_j \in \mathcal{C}} \mathcal{Att}_{EDL}(\Delta \vec{x}_{i,j}).$$

Let $a_j^\dagger$ denote the actual linked account to a_i .

The prediction is correct if and only if $\forall a_j \in \mathcal{C} \setminus \{a_j^\dagger\}, F(\Delta \vec{x}_{i,j}^\dagger) < F(\Delta \vec{x}_{i,j})$.

Let the random variables $S^* = F(\Delta \vec{x}_{i,j}^\dagger)$ and $S = F(\Delta \vec{x}_{i,j})$ (for $a_j \neq a_j^\dagger$), representing the score distributions for true links and non-links, respectively. Moreover, suppose that the trained dissimilarity-based learning model of $\mathcal{Att}_{EDL}$ achieves stochastic dominance in the lower direction, meaning that the score distribution of true links S^* is stochastically smaller than that of non-links S .

Formally, for every threshold $t \in [0,1]$, $\Pr[S^* < t] > \Pr[S < t]$, indicating that true link pairs are more likely to yield lower dissimilarity scores than random, unrelated pairs. Equivalently, this relationship can be expressed in terms of cumulative distribution functions (CDFs) as $F_{S^*}(t) = \Pr[S^* \leq t] \geq F_S(t) = \Pr[S \leq t], \forall t \in [0,1]$. The inequality $F_{S^*}(t) \geq F_S(t)$ reflects the left-shifted nature of the dissimilarity distribution for true links, consistent with the model's objective of minimizing distance between same-owner accounts. Consequently, the expected dissimilarity of true links is strictly smaller as $E[S^*] = \int_0^1 F_{S^*}(t) dt < \int_0^1 F_S(t) dt = E[S]$.

To analyze the success probability, let define $S_{min} = \min_{a_j \neq a_j^\dagger} S_j$, where each $S_j \sim S$ represents the dissimilarity score of a non-link candidate. Assuming independence and identical distribution among non-link scores, the minimum-score distribution is given by $F_{min}(t) = \Pr[S_{min} \leq t] = [F_S(t)]^{|\mathcal{C}|-1}$, $f_{min}(t) = \frac{d}{dt} F_{min}(t)$. Therefore, the success probability of the attacker model $\mathcal{Att}_{EDL}$ can be expressed as $Succ_{EDL} = \int_0^1 \Pr[S^* < t] f_{min}(t) dt$.

In contrast, a random guess within the anonymity set of size $|\mathcal{C}|$ achieves $Succ_{RG} = \frac{1}{|\mathcal{C}|}$. Accordingly, the advantage margin of the attacker becomes as follows:

$$\Delta = Succ_{EDL} - Succ_{RG} = \int_0^1 (\Pr[S^* < t] - \Pr[S < t]) f_{min}(t) dt.$$

Under consideration for a well-trained model with discriminative behavioral features, since $\Pr[S^* < t] > \Pr[S < t]$, the integrand remains strictly positive which follows that $\Delta > 0$, and moreover, Δ increases with higher model accuracy, richer feature representations, and greater behavioral separability. Therefore, under full transaction visibility and the existence of learnable behavioral difference vectors, the attacker $\mathcal{Att}_{EDL}$ achieves non-negligible success in linking pseudonymous Ethereum accounts, thereby breaking ownership unlinkability anonymity of Ethereum users as defined in Definition 2 of Section 4.2. Please refer to Section 7 for experimental results that support this theoretical claim, demonstrating that the proposed attacker model achieves non-negligible success in real-world scenarios. In the next section, we present a detailed methodology of the proposed attack model $\mathcal{Att}_{EDL}$ for *EDL*.

6. PROPOSED SOLUTION

As motivated in Section 5.1, one of the major challenges in deanonymization task as research problem described in Section 4.2 is the variability in user behavior on Ethereum, shaped by differing usage contexts such as hot, cold, and bridge accounts (Oliveira et al., 2022; *The Best Crypto Wallets of March 2023 – Forbes Advisor*, n.d.; *The Complete Guide to Web3 Wallets. Thirdweb, Jan. 13, 2023*, n.d.) (Thirdweb, 2023; Forbes, 2023; Oliveira, 2022). For example, one may rely on a hot account for routine transactions, prioritizing convenience and speed, while using a cold account, generally kept offline, for storing larger amounts of cryptocurrency, either for long-term storage or potentially illicit activities (Hu, 2023; FTX Hack, n.d.; Upbit Hack, n.d.). Additionally, bridge accounts, facilitating cross-chain asset transfers, further complicate behavioral patterns by introducing another layer of transactional diversity (*The \$477 Million FTX Hack: A New Blockchain Trail*, n.d.; *Tracing the Trail of the Upbit Hack*, n.d.).

As discussed in Section 3.1, the pseudonymous nature of the blockchain, allowing users to generate unlimited accounts, amplifies these behavioral discrepancies, making it more difficult to link accounts belonging to the same owner. Given the impracticality of preventing such variations in account usage, the most effective approach is to develop models that incorporate these behavioral differences, improving the accuracy of the deanonymization.

In this study, we propose a model named *EtherDiffLinkage(EDL)* as a solution that considers possible behavioral differences and formulates the AOL problem as a regression-based ownership linkage learning task. Moreover, this approach is built upon xAI models so that collaboration between the blockchain investigator and the AI model enhances the robustness of such deanonymization process.

In the rest of the chapter, we present the technical details of *EDL*, with the proposed xAI-driven framework for the task described in Section 4.2.

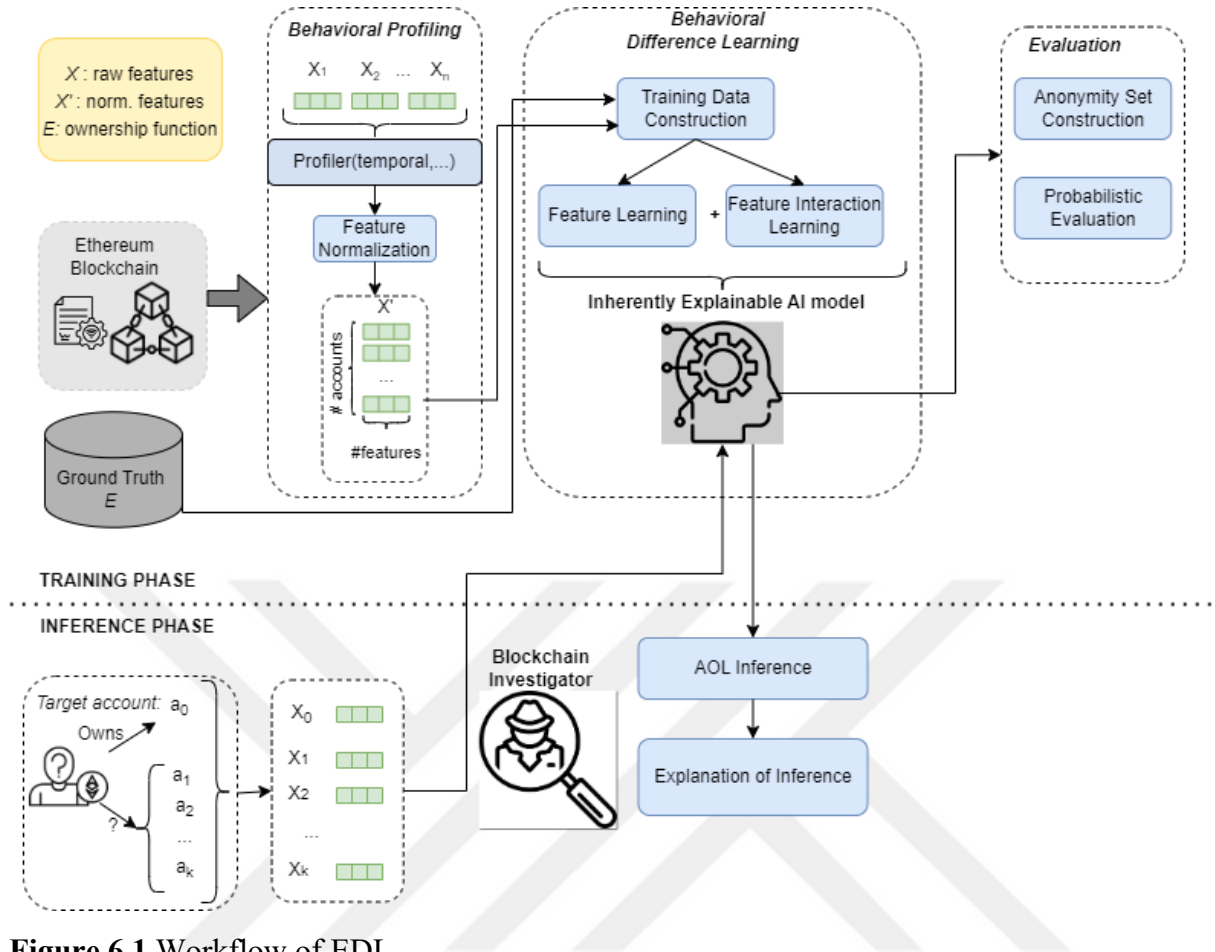


Figure 6.1 Workflow of EDL

As shown in Figure 6.1, *EDL* operates in two phases: training and deanonymization. The training phase is structured based on the input of raw blockchain data $\mathcal{D}$ and the ownership function E , obtained from external sources as ground truth. This phase consists of three key modules:

- A behavioral profiling module that extracts and normalizes behavioral features from raw transactions to construct meaningful behavioral representations.
- A behavioral difference learning module that processes these features to train an xAI model, leveraging behavioral difference-based feature learning and interaction analysis to capture ownership linkages.
- An evaluation module that utilizes the trained model to assess deanonymization performance in a probabilistic manner.

The inference phase of EDL consists of three main steps:

- *Anonymity set construction*: Identifying K candidate accounts for the target account a_0 .
- *AOL inference*: Evaluating behavioral similarities using the trained model to determine the most likely linked account.
- *Explanation of inference*: Providing feature-level and interaction-level insights to help investigators interpret and validate the results.

The input to this phase includes a target account and a set of potential accounts assumed to share the same ownership, forming an anonymity set. The output is the most likely linked account, identified using the model trained in the previous phase. Because this model is inherently explainable, the inference process highlights the significance of different features in the final decision, ensuring transparency for blockchain investigators. The following sections present a detailed technical description of each phase in *EDL*.

6.1. Training Phase

6.1.1. Behavioral Profiling Module

Profiler: For each account $a_j \in \mathcal{A}$, the full transaction history $\mathcal{D}$ is extracted from the blockchain, capturing both incoming and outgoing transactions. This dual perspective enables the profiling of each a_j as both a sender and a recipient, considering the behavior of other interacting accounts. The transaction data includes raw data fields described in Section 3.1. Using this transaction data $\mathcal{D}$, behavior profile vectors X are generated based on the specified attribute set F , as listed in Table 6.1, profiling each account based on its transactional behavior.

From the perspective of a blockchain investigator, we systematically compile the behavioral characteristics of Ethereum users by integrating both disclosed attributes and those identified in the literature (Bonifazi et al., 2022a; Pan et al., 2024; Said et al., 2021; Tan et al., 2023). These features are categorized into three distinct groups: temporal, volumetric, and infrastructural, as presented in Table 6.1, with a focus on ensuring clarity and explainability for human investigators. Notably, this profiling approach is designed to support investigative analyses and can be extended with additional human-understandable features to enhance its applicability.

Formally, the Profiler function P processes the transaction attributes of each set of Ethereum transactions $\mathcal{D} = \{T_1, \dots, T_n\}$, where each transaction T_i is characterized by attributes summarized in Table 3.1. The primary objective of P is to generate an embedding vector X_j for each account $a_j \in \mathcal{A}$, based on its transaction history $\mathcal{D}^j$ and a predefined feature set F .

Table 6.1 Overview of Extracted Features

Category	Feature ID (#)	Description
Temporal	F1–F5 (5)	Average, maximum, minimum time between transactions; transaction frequency; time of day profile
Volumetric	F6–F10 (5)	Total, average, maximum, and minimum value of transactions; standard deviation of transaction values in ETH unit.
Infrastructural	F11–F16 (6)	Ratio of contract interactions; average, maximum, minimum gas price; ratio of failed transactions; average gas used per transaction.

P maps an account’s transaction history and a feature set to an embedding vector as $P: (\mathcal{D}^j, F) \rightarrow X_j^F$, where X_j^F denotes the behavioral characteristics of account (a_j) based on its transaction set $\mathcal{D}^j$ and the feature set F .

Temporal Profiling. One of the most common approaches to analyzing transaction behavior is to examine it from a temporal perspective (Ao et al., 2021; Bai et al., 2021; Farrugia et al., 2020; Pan et al., 2024). Several experimental studies have demonstrated that Ethereum account owners exhibit similar temporal behavioral characteristics even when using different accounts (Béres et al., 2020; Kılıç & İnan, 2024). On the other hand, the critical role of timing in illicit activities underscores the necessity of temporal analysis. For example, some accounts display a tendency to initiate immediate transfers, suggesting that the account is primarily used for laundering funds or quick trades (Payette et al., 2017). This behavior strongly indicates specific transactional purposes, such as obfuscating the origin of assets. Temporal profiling, therefore,

offers crucial insights into user behavior, making it a valuable tool in deanonymization efforts, particularly for identifying account owners involved in illicit activities.

Another key factor is the time of day at which transactions occur, which serves as a valuable attribute for profiling the temporal behavior of accounts, exhibiting quasi-identifier capabilities as demonstrated in prior analyses (Béres et al., 2020). To enhance the profiling process, we incorporate additional statistical measures, specifically focusing on transaction gaps and frequencies. The complete set of temporal features is derived from the i th transaction timestamp, (T_i, τ) , as follows:

Average Time Between Transactions (F_1) measures the average time interval between consecutive transactions, characterizing the user's transaction frequency and activity consistency over time. It can be computed as follows:

$$F_1 = \frac{1}{n-1} \sum_{i=2}^n (T_i \cdot \tau - T_{i-1} \cdot \tau), \quad (6.1)$$

where n represents the total number of transactions.

Maximum Time Between Transactions (F_2) captures the longest duration between two consecutive transactions, highlighting extended inactivity periods that may indicate sporadic usage patterns as follows:

$$F_2 = \max_{i=2}^n (T_i \cdot \tau - T_{i-1} \cdot \tau) \quad (6.2)$$

Minimum Time Between Transactions (F_3) represents the shortest time gap between consecutive transactions, capturing the user's highest observed transaction rate:

$$F_3 = \min_{i=2}^n (T_i \cdot \tau - T_{i-1} \cdot \tau) \quad (6.3)$$

Transaction Frequency F_4 , measured in transactions per unit time, is computed as follows:

$$F_4 = \frac{n}{T_n \cdot \tau - T_1 \cdot \tau} \quad (6.4)$$

where $T_n \cdot \tau$ and $T_1 \cdot \tau$ denote the timestamps of the first and last transactions, respectively.

Frequent Time-of-Day Activity (F_5): To capture the user's dominant transaction time, a periodic transformation of timestamps is applied based on the hour component as follows:

$$F_5 = \frac{1}{n} \sum_{i=1}^n \left(\sin \left(\frac{2\pi H(T_i, \tau)}{24} \right), \cos \left(\frac{2\pi H(T_i, \tau)}{24} \right) \right) \quad (6.5)$$

, where $H(T_i, \tau)$ extracts the hour time from the transaction timestamp. This feature provides a two-dimensional representation of dominant transaction times, ensuring a continuous and periodic encoding of user activity within the 24-hour cycle.

Volumetric Profiling. To characterize an account's financial activity, a set of hand-crafted volumetric features derived from transaction values $T_i \cdot V$ are utilized. These features serve as key indicators of user behavior, capturing patterns in transaction magnitude and frequency that are instrumental in profiling Ethereum users. Since financial behavior on blockchain networks often correlates with user intent and strategy—such as investment, trading, or scam-related activity—volumetric metrics enable the differentiation of distinct behavioral archetypes.

Total Transaction Value (F_6) represents the cumulative value of all transactions associated with the account, providing an estimate of the total volume of assets transferred over the observed period as follows:

$$F_6 = \sum_{i=1}^n T_i \cdot V \quad (6.6)$$

Average Transaction Value (F_7) measures the mean transaction value, offering insight into the typical magnitude of transactions as follows:

$$F_7 = \frac{1}{n} \sum_{i=1}^n T_i \cdot V \quad (6.7)$$

Maximum Transaction Value (F_8) captures the highest observed transaction value, which may indicate occasional large transfers, commonly associated with high-net-worth accounts or specific laundering strategies as follows:

$$F_8 = \max_{i=1}^n T_i \cdot V \quad (6.8)$$

Minimum Transaction Value (F_9) represents the smallest transaction value, which can reflect micro-transactions or minimal asset transfers as follows:

$$F_9 = \min_{i=1}^n T_i \cdot V \quad (6.9)$$

Standard Deviation of Transaction Values (F_{10}) quantifies the variability in transaction values, indicating whether transactions are relatively uniform or highly dispersed as follows.

$$F_{10} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (T_i \cdot V - F_7)^2} \quad (6.10)$$

To illustrate the overall features, a high F_6 or F_8 indicates significant asset movement, potentially reflecting high-value transfers or periodic large transactions. A low F_7 combined with a high F_{10} may suggest irregular spending behavior, possibly associated with dynamic trading or laundering activity. Conversely, consistently low F_7 and F_9 suggest routine micro-transactions, often characteristic of automated accounts or contract interactions.

Infrastructural Profiling. In Ethereum, infrastructural features capture crucial aspects of a transaction's operational and structural behavior, distinguishing it from traditional financial systems. These features include gas price, transaction success rates, and smart contract interactions, providing insights into user behavior within the Ethereum network.

For instance, a high contract interaction ratio suggests frequent engagement with decentralized applications (dApps) or automated systems, indicating sophisticated usage patterns. The following infrastructural features are defined for transactions $T_i \in \mathcal{D}$:

- *Ratio of Contract Interactions* (F_{11}) represents the proportion of transactions that interact with smart contracts, reflecting the extent of engagement with decentralized services as follows:

$$F_{11} = \frac{\sum_{i=1}^n \text{OI}(T_i.C \neq \emptyset)}{n} \quad (6.11)$$

- *Average Gas Price* (F_{12}) measures the mean gas price paid per transaction, indicating the user's cost sensitivity as follows:

$$F_{12} = \frac{1}{n} \sum_{i=1}^n T_i \cdot P \quad (6.12)$$

- *Maximum Gas Price* (F_{13}) captures the highest gas price used in transactions, which may suggest urgency or priority in transaction processing.

$$F_{13} = \max_{i=1}^n T_i \cdot P \quad (6.13)$$

-*Minimum Gas Price* (F_{14}) represents the lowest gas price used, indicating the user's willingness to wait for transaction confirmation at minimal cost.

$$F_{14} = \min_{i=1}^n T_i \cdot P \quad (6.14)$$

-*Ratio of Failed Transactions* (F_{15}) quantifies the proportion of failed transactions, shedding light on execution reliability and potential gas inefficiencies as follows:

$$F_{15} = \frac{\sum_{i=1}^n T_i \cdot E}{n} \quad (6.15)$$

-*Average Gas Used per Transaction* (F_{16}) measures the mean gas consumption, providing insights into the complexity of transactions executed by the account.

$$F_{16} = \frac{1}{n} \sum_{i=1}^n T_i \cdot G \quad (6.16)$$

To illustrate, a high F_{11} may indicate frequent smart contract interactions, while an elevated F_{15} suggests inefficiencies or erroneous transactions. Additionally, variations in gas price metrics ($F_{12} - F_{14}, F_{16}$) provide insights into transaction prioritization strategies. By analyzing these attributes, it can be distinguished between different usage behaviors, such as cost-conscious users versus those prioritizing transaction speed. These infrastructural features, alongside temporal and volumetric characteristics, can be aggregated into profile vectors X_j^F for each account $a_j \in \mathcal{A}$, representing key behavioral patterns.

Overall, upon extracting these transaction attributes, 16 distinct features comprising 6 infrastructural, 5 volumetric, and 5 temporal attributes are obtained as summarized in Table 6.1. This comprehensive feature set facilitates a detailed analysis of account owner behaviors, capturing multiple dimensions of activity.

Moreover, previous studies and blockchain forensic analyses (Hu et al., 2023; Wu et al., 2023) have demonstrated that incoming transactions, alongside outgoing ones, play a critical role in shaping behavioral patterns. For instance, in Ethereum account profiling, a DeFi trader may frequently send transactions to multiple decentralized exchanges (DEXs) while receiving fewer

but larger incoming transfers from profits or liquidity staking. In contrast, a cold wallet used primarily for storage may exhibit infrequent outgoing transactions, with incoming transfers occurring sporadically for long-term asset accumulation. These distinct patterns highlight the necessity of analyzing incoming and outgoing transactions separately for a precise behavioral assessment. Therefore, behavioral profiling considers both incoming and outgoing transactions to capture asymmetric activity patterns. In this context, the function P computes the features listed in Table 6.1 separately for incoming and outgoing transactions to capture asymmetric behavioral patterns.

Feature Normalization. After extracting all features, the feature matrix $X \in R^{|A| \times |F|}$ is obtained from the profiler P . As shown in Table 6.1, the extracted features span different ranges, making direct comparisons with raw values challenging. To ensure uniform scaling and facilitate the processing for an AI model, the normalization *z-score* is applied to all feature values separately, transforming X as follows:

$$X' = \frac{X - \bar{X}}{S} \quad (6.17)$$

where X represents the original feature matrix, $\bar{X}$ and S denote the mean and standard deviation computed across each feature dimension, respectively, and X' is the normalized feature matrix. In the experiments conducted in Chapter 7, this normalized feature vector serves as the primary representation, with explainability derived from its contribution to the model output.

6.1.2. Behavioral Difference-based Model Construction Module

From the perspective of AI model collaboration with blockchain investigators, a key property of *EtherDiffLinkage(EDL)* is the *Behavioral Difference-based Explainable Model Construction* module, which trains a model to learn usage patterns based on behavioral differences. This module also leverages an inherently xAI model like described in Section 3.3, providing explainable outputs to meet the explainability requirements of the research problem.

As illustrated in Figure 6.1, *EDL* is designed to learn complex relationships while constructing an explainable learning model. It achieves this by leveraging human-understandable features and adopting an inherently xAI model. From an explainability perspective, the primary function

of this model is to learn the ownership distance between account pairs in an intrinsic manner, where the decision-making process is influenced by key explanatory factors.

Technically, *EDL* framework integrates an xAI model construction mechanism based on behavioral differences using a three-part design. The first part, **input formulation**, defines the learning task as a regression-based dissimilarity prediction problem. Specifically, as introduced in Section 5.2, behavioral differences between account pairs are quantified and used as supervision targets. These dissimilarity scores serve as a proxy for linkage prediction, allowing the system to infer whether a given account pair should be considered linked.

The second part, **xAI model construction**, employs the Explainable Boosting Machines(EBM) model (Lou et al., 2013), a boosting-based learning model designed for interpretable analysis over tabular data. EBM operates in two parallel learning tracks: the first focuses on individual feature contributions, while the second captures interactions between feature pairs. This dual-path structure enables the model to estimate behavioral dissimilarity with fine-grained interpretability, making it particularly suitable for forensic or compliance-related applications.

The third part, **prediction and explanation**, utilizes the learned dissimilarity scores to make final linkage decisions. In this phase, the model outputs both the predicted dissimilarity value and an explanation of the contributing features or interactions. From the standpoint of explainable AI, this ensures that the decision process remains transparent and auditable.

In the subsequent section, these parts of the model are detailed, with a particular focus on how it integrates both feature-level and interaction-level information during the learning process.

Input Formulation: Please recall that for each account $a_j \in \mathcal{A}$, the corresponding profile vector $X_j \in R^d$ is computed as $X_j = \mathcal{P}(T_j)$, where $T_j \subset \mathcal{T}$ is the transaction set associated with account a_j . To generate training data for the model, we consider all unordered account pairs $(a_j, a_{j'}) \in \mathcal{A} \times \mathcal{A}$ such that $j \neq j'$. For each pair, we evaluate an ownership function $E: \mathcal{A} \times \mathcal{A} \rightarrow \{0, 1, \emptyset\}$, where $E(a_j, a_{j'}) = 1$ indicates that the two accounts belong to the same owner, $E(a_j, a_{j'}) = 0$ denotes different owners, and $E(a_j, a_{j'}) = \emptyset$ signifies unknown ownership information.

For each valid pair $(a_j, a_{j'})$ where $E(a_j, a_{j'}) \neq \emptyset$, we compute the absolute difference (as justified in Argument 1) between their profile vectors $X_{j,j'} = |X_j - X_{j'}| = \Delta \overrightarrow{x_{j,j'}}$. This vector serves as the input for training stage. The corresponding dissimilarity label $y_{j,j'}$ is set to 0 when $E(a_j, a_{j'}) = 1$, indicating that both elements belong to the same owner, and to 100 when $E(a_j, a_{j'}) = 0$, indicating different ownership.

Please note that the problem could be formulated as a classification task. However, the presence of extremely large anonymity sets in blockchain environments favors a regression-based approach. This formulation allows for modeling continuous behavioral differences between account pairs, which is more suitable for capturing subtle variations that may not be adequately represented by discrete class labels.

xAI Model Construction: Please recall our research problem described in Section 4.2. The EBM (Lou et al., 2013) model is perfectly fit the research problem—not only due to its strong predictive performance, but also because it facilitates explainability through the proposed features engineered in Section 6.1.1, thereby enabling human blockchain investigators to validate and understand the model's decisions. To this end, the EBM model is reformulated and re-implemented so that it conforms to the specific requirements of the research problem. In the empirical analysis, results are presented to demonstrate its applicability within the given context.

Technically, the model M_θ , parameterized by θ , is trained using gradient boosting over shallow regression trees. The model learns an additive function of the form as follows:

$$M_\theta(X_{j,j'}) = \underbrace{\sum_k f_k(X_{j,j'}^{(k)})}_{\text{Individual Contribution}} + \underbrace{\sum_{(k,l) \in \mathcal{I}} f_{k,l}(X_{j,j'}^{(k)}, X_{j,j'}^{(l)})}_{\text{Interaction Contribution}} + \beta \quad (6.18)$$

where $f_k: R \rightarrow R$ is a univariate shape function learned for feature k , $f_{k,l}: R^2 \rightarrow R$ represents pairwise interaction functions between selected feature pairs $(k,l) \in \mathcal{I}$, $\beta \in R$ is a scalar bias term. Notice that the function structure under consideration consists of an additive composition

of sub-functions $\{f_k\}_{k=1}^K$, where each f_k operates on a selected subset of the d behavioral features and their pairwise interactions. These features are specifically designed to be interpretable and actionable from the perspective of a blockchain investigator, as introduced in Section 6.1.1. In operational terms, this perspective prioritizes features that reflect semantically meaningful behavioral traits—such as interaction regularity, counterparty diversity, and contract usage patterns—that are commonly used in real-world blockchain forensic analysis (Elliptic, 2022; Elliptic Enterprises Limited, 2025).

The function class learns to model dissimilarity between account behaviors based on both individual feature responses and a selected set of interaction terms. In principle, the number of possible pairwise interactions among d features are $\frac{d(d-1)}{2}$, which implies a quadratic growth in feature space complexity. Exhaustively modeling all such interactions would impose a significant computational load and increase the risk of overfitting, especially when the number of training samples is limited relative to the number of modeled terms.

To mitigate this, a domain-guided pruning strategy is used. In this strategy, the interaction space is restricted to pairs of features that belong to distinct semantic categories. The key idea is to avoid modeling redundant within-category dependencies and instead capture cross-category interactions that yield more meaningful logical insights, for example, the co-variation between transaction timing and counterparty diversity. By imposing this principled restriction, computational complexity is reduced from $\mathcal{O}(d^2)$ to $\mathcal{O}(m \times n)$, where m and n denote the number of features in disjoint groups. At the same time, variance is lowered by suppressing spurious correlations. Consistent with findings in the literature on generalized additive models with interactions (e.g., (Lou et al., 2013)), this selective incorporation of interpretable and semantically grounded interactions improves generalization, preserves transparency, and enhances the robustness of the resulting dissimilarity function.

Consider a hypothetical toy dataset as shown in Table 6.2, which is constructed to model behavioral patterns across Ethereum accounts under ownership constraints. While the dataset is synthetic, its structure and attribute distributions are inspired by typical behavioral patterns observed in real-world blockchain usage (Bonifazi et al., 2022b; Cavaleiro et al., 2024),

making it suitable for illustrating the core mechanisms of the proposed modeling framework. Each account in the dataset is represented by six typical behavioral features: two drawn from temporal dimensions (average transaction interval (refer F1 in Section 6.1.1), transaction frequency (refer F4 in Section 6.1.1)), two from volumetric characteristics (average transaction value (refer to F7 in Section 6.1.1), variance (refer to F10 in Section 6.1.1)), and two infrastructural indicators (contract interaction ratio (refer F11 in Section 6.1.1), average gas used per transaction (refer to F16 in Section 6.1.1)). All features have been standardized using Equation 6.17 to ensure comparability. Furthermore, accounts are grouped under a shared ownership label, indicating that they are managed by the same entity. The motivation for constructing this dataset lies in examining how multiple accounts under a common owner may exhibit either convergent or intentionally divergent transactional behavior. For instance, accounts $\{0x1, 0x2\}$ and $\{0x3, 0x4\}$, which are associated with $O1$ and $O2$ respectively, are deliberately configured to display distinct behavioral profiles---presumably as an evasion strategy. This is demonstrated by their strongly negative similarity scores, suggesting a lack of alignment in behavioral space. In contrast, accounts $\{0x5, 0x6\}$ under $O3$ exhibit highly similar behavior, as reflected by a near-perfect similarity score. Such alignment may correspond to coordinated activity or functional duplication across accounts. Using this structure, we aim to enable a principled evaluation of account linkage, providing an analytically tractable environment for testing ownership linkage strategies in blockchain forensic settings.

Moreover, by constructing such a dataset, the objective is to demonstrate that the proposed solution named *EDL*, can effectively learn both accounts exhibiting behavioral similarity and those deliberately engineered to display behavioral divergence as a means of obfuscation.

Technically, given a table as shown in Table 6.2, dissimilarity scores can then be assigned to each account pair using Input Formulation module of *EDL*. Formally, we consider all unordered account pairs $(a_j, a_{j'}) \in \mathcal{A} \times \mathcal{A}$, $j \neq j'$. For each valid pair, the absolute difference between their feature vectors is computed, i.e., $X_{j,j'} = |X_j - X_{j'}| = \Delta \overrightarrow{x_{j,j'}}$. The corresponding dissimilarity label $y_{j,j'}$ is assigned as 0 for same owner, 100 for different owner.

In this setup, the equi-frequency binning approach (as assigned in Table 6.3) is applied to enhance explainability by introducing granularity into the constructed feature space.

Table 6.2 Hypothetical Toy Dataset

Address	F1	F2	F3	F4	F5	F6	Owner ID
0x1	-0.731	1.234	-0.661	-0.486	-0.634	-0.264	O1
0x2	1.707	-1.032	-0.844	-0.5	-1.202	-1.321	O1
0x3	-0.779	1.385	2.146	2.234	1.637	1.849	O2
0x4	1.069	-1.184	-0.478	-0.481	-0.918	-0.792	O2
0x5	-0.63	-0.277	-0.112	-0.39	0.502	0.264	O3
0x6	-0.636	-0.126	-0.051	-0.376	0.615	0.264	O3

The objective of learning approach is to learn both behavioral similarity and dissimilarity patterns using an xAI model. Formally, consider a dataset of entities $E = \{e_1, \dots, e_n\}$, each associated with a behavior vector $x_i \in R^d$ and an owner label $o_i \in O$. The goal is to derive per-feature 1-dimensional additive functions $f_k: R \rightarrow R$ such that pairwise dissimilarity between behavioral profiles can be captured and decomposed in an explainable manner. Following the paradigm of an xAI model as described in Section 3.3, the residuals are initialized as $r_i^{(0)} := y_i$, where y_i corresponds to the dissimilarity label for a given sample pair. In the dichotomous regime, it is assigned that $y_{ij} = 0$ if $o_i = o_j$ (implies *same owner* $\Rightarrow$ *min. dissimilarity*), and $y_{ij} = 100$ if $o_i \neq o_j$ (implies *different owners* $\Rightarrow$ *max. dissimilarity*). Such dichotomous supervision has capabilities that facilitate training in both similar and contrastive behaviors simultaneously.

Consider the constructed toy dataset in Table 6.3, where each sample is represented by six behavioral differences $(\Delta F_1, \dots, \Delta F_6)$ together with a binary dissimilarity label $y \in \{0, 100\}$. Each feature is discretized into at most three bins (corresponding to the unique observed values). To highlight both additive and interactive effects, we allow one interaction term by setting. In this setting, the model can include not only univariate additive functions f_k but also one bivariate function $f_{k,l}$ over a selected feature pair (k, l) .

Table 6.3 Learning Target Aware Constructed Dataset

<i>ID Pair</i>	$\Delta F1$	$\Delta F2$	$\Delta F3$	$\Delta F4$	$\Delta F5$	$\Delta F6$	<i>Dissimilarity Target</i>
0x1, 0x2	0	4	2	2	3	4	0
0x1, 0x3	2	2	0	0	0	0	100
0x1, 0x4	0	4	2	2	3	3	100
0x1, 0x5	2	4	1	2	0	1	100
0x1, 0x6	2	4	0	2	0	1	100
0x2, 0x3	4	0	0	0	0	0	100
0x2, 0x4	4	2	1	2	1	1	100
0x2, 0x5	4	0	0	2	0	0	100
0x2, 0x6	4	0	0	2	0	0	100
0x3, 0x4	0	4	4	4	4	4	0
0x3, 0x5	2	4	4	4	4	4	100
0x3, 0x6	2	4	4	4	4	4	100
0x4, 0x5	4	0	1	2	0	0	100
0x4, 0x6	4	0	1	2	0	0	100
0x5, 0x6	2	2	2	2	2	2	0

The training begins as in the purely additive case. At iteration $t = 0$, residuals are initialized as $r_i^{(0)} = y_i$. Consider ΔF_1 , which has bins $\{0,2,4\}$. The bin-wise residual means are $\overline{r_{\Delta F_1=0}} = 0$, $\overline{r_{\Delta F_1=2}} = 100$, $\overline{r_{\Delta F_1=4}} = 100$.

With a learning rate $\eta = 0.1$, the updates are: $f_{\Delta F_1}^{(1)}(0) = 0$, $f_{\Delta F_1}^{(1)}(2) = 10$, $f_{\Delta F_1}^{(1)}(4) = 10$.

Residuals are recomputed as $r_i^{(1)} = y_i - \widehat{y}_i^{(1)}$ and subsequently used to update ΔF_2 , yielding the contributions summarized in Table 6.4.

Selection of the most informative pair. Once univariate updates are in place, the model evaluates candidate pairs for potential interactions. Training all $\binom{6}{2}$ pairs is unrealistic, so the FAST algorithm (Lou et al., 2013) scores each pair by residual variance reduction. So, we just consider the pair $(\Delta F_1, \Delta F_3)$ in the toy dataset.

From the calculation perspective, the overall residual variance is $\text{Var}(r) = 1600$. Residuals are then grouped into the 3×3 joint histogram $H_{1,3}$. For example, in the bin $(\Delta F_1 = 0, \Delta F_3 = 2)$ two samples fall with mean residual 50 and variance 2500, whereas in several other bins (e.g., $(\Delta F_1 = 2, \Delta F_3 = 0)$ or $(\Delta F_1 = 2, \Delta F_3 = 1)$) the within-bin variance is 0 since all residuals are identical. Weighing each bin variance by its relative frequency $n_{b_1, b_3}/N$ and summing across bins yields the average conditional variance $E[\text{Var}(r | H_{1,3})] = 333.33$. The FAST score is calculated as follows: $\text{Score}(\Delta F_1, \Delta F_3) = 1600 - 333.33 \approx 1266.67$. This high score indicates that conditioning on $(\Delta F_1, \Delta F_3)$ substantially reduces residual variance, i.e. the between-bin variation of residual means is large. Hence the pair $(\Delta F_1, \Delta F_3)$ is selected as informative interaction, capturing dependencies that univariate updates alone cannot explain.

Table 6.4 Bin-wise updates for ΔF_2 after the first feature update.

ΔF_2 value	Residual mean after ΔF_1 value	Contribution μ_b
0	90	9.0
2	57.78	5.78
4	65.24	6.53

Learning the 2D histogram function. The joint histogram $H_{1,3}$ partitions the space into a 3×3 grid. For each cell (b_1, b_3) , the mean residual after the latest univariate updates is calculated as follows:

$$\overline{r_{b_1, b_3}} = \frac{1}{|H_{1,3}(b_1, b_3)|} \sum_{x_i \in H_{1,3}(b_1, b_3)} r_i \quad (6.19)$$

These averages are scaled by η and added as contributions as follows:

$$f_{\Delta F_1, \Delta F_3}^{(1)}(b_1, b_3) = \eta \cdot \overline{r_{b_1, b_3}} \quad (6.20)$$

Updated prediction rule. After including the interaction, the prediction rule becomes as:

$$\hat{y}_i = \sum_{k=1}^6 f_k(\rho(H_k, x_i)) + f_{\Delta F_1, \Delta F_3}(\rho(H_1, x_i), \rho(H_3, x_i)) \quad (6.21)$$

Therefore, the model captures both marginal (additive) effects and one selected bivariate effect. For instance, a particular cell ($\Delta F_1 = 2, \Delta F_3 = 0$) might yield a high residual mean, leading to a stronger correction that purely additive updates could not explain. Notice that such example illustrates that while additive terms explain much of the variance, a carefully chosen interaction enhances the model’s ability to capture structured dependencies. Notice that the selection of $(\Delta F_1, \Delta F_3)$ is motivated not only by residual variance reduction but also by the interpretability gained when interactions come from distinct feature categories. Although only one interaction is permitted here, the mechanism generalizes to more pairs. From a methodological perspective, this balances model complexity with interpretability, aligning with the parsimony–transparency trade-off discussed in the literature (Caruana et al., 2015; Lou et al., 2013).

From the overall algorithmic perspective, to summarize the entire training procedure (as outlined in Algorithm 1), the process begins with the initialization of the xAI model as M_θ . The input dataset is then constructed by generating candidate account pairs from the set of Ethereum accounts A ensuring that no account is paired with itself (lines 5-6). Each candidate pair (a_j, a'_j) is evaluated based on the ownership function $E(a_j, a'_j)$ which serves as ground truth labeling: a value of 1 indicates that the accounts belong to the same owner, while 0 signifies distinct ownership (line 7). Account pairs for which ownership information is unavailable are excluded from training (lines 8-9).

Algorithm 1 Behavioral Difference-Based Training

Require: Set of accounts $\mathcal{A}$, feature matrix X , ownership function $E : \mathcal{A} \times \mathcal{A} \rightarrow \{0, 1, \emptyset\}$

Ensure: Trained model M_θ

```
1: procedure INITIALIZE MODEL
2:   Initialize the model  $M_\theta$ 
3: end procedure
4: procedure GENERATE TRAINING DATA
5:   Construct candidate account pairs:
6:    $\mathbb{A} = \{(a_j, a'_j) \mid a_j, a'_j \in \mathcal{A}, j \neq j'\}$ 
7:   for each  $(a_j, a'_j) \in \mathbb{A}$  do
8:     if  $E(a_j, a'_j) = \emptyset$  then
9:       continue // Skip pairs with undefined ownership
10:    end if
11:    Compute element-wise absolute difference:
12:     $X_{j,j'} = |X_j - X_{j'}|$ 
13:    if  $E(a_j, a'_j) = 1$  then
14:      Assign label:  $y_{j,j'} = 0$  // Same owner
15:    else if  $E(a_j, a'_j) = 0$  then
16:      Assign label:  $y_{j,j'} = 100$  // Different owner
17:    end if
18:  end for
19: end procedure
20: procedure COMPUTE SAMPLE WEIGHTS
21:   Compute frequency-based weighting:
22:    $w_{j,j'} = \frac{1}{\text{freq}(y_{j,j'})}$ 
23:   where  $\text{freq}(y_{j,j'})$  represents the occurrence count of  $y_{j,j'}$ .
24: end procedure
25: procedure TRAIN MODEL
26:   Minimize the weighted loss function:
27:    $\mathcal{L} = \sum_{j,j'} w_{j,j'} \cdot L(y_{j,j'}, M_\theta(X_{j,j'}))$ 
28:   where  $L$  is a loss function (e.g., Mean Squared Error).
29: end procedure
30: return Trained model  $M_\theta$ 
```

For each valid account pair, a behavioral difference vector $X_{j,j'}$ is computed by taking the element-wise absolute difference of their feature representations (line 12). This transformation captures behavioral difference patterns between account pairs. Based on the ownership function E , the corresponding target labels are assigned: a distance of 0 for linked accounts and 100 for non-linked accounts (lines 13-16). These assignments establish a clear separation between linked and non-linked pairs, facilitating effective training.

Due to the inherent class imbalance between linked and non-linked pairs, a cost-sensitive weighting mechanism is employed to prevent bias toward the majority class. The weight

assigned to each sample, $w_{j,j'}$, is computed inversely proportional to its frequency in the dataset, ensuring that the model does not become biased toward majority-class instances (lines 21-23). This weighting strategy enhances the model's ability to generalize across diverse account behaviors.

With the training dataset fully constructed, the AI model M_θ is trained using a regression-based loss function (lines 26-27). The objective is to minimize $\mathcal{L}$, where $\mathcal{L}$ is a standard loss function such as Mean Squared Error (MSE), ensuring that the model effectively learns to distinguish ownership linkages based on behavioral differences.

The training process continues until convergence, optimizing M_θ to predict ownership linkages between Ethereum accounts based on their behavioral distance vectors, as described in Equation 5.20 and 5.21. Once trained, the model is deployed in the inference phase, where it infers potential ownership candidates within an anonymity set.

As outlined in Algorithm 1, this training phase aims to learn behavioral variations between Ethereum accounts and optimize the model to distinguish linked and non-linked account pairs.

6.1.3. Evaluation Module

Anonymity Set Construction: To assess the inference performance of the pre-trained model M_θ as described in Section 6.1.2, an anonymity set is constructed. This set consists of accounts likely managed by the same owner, as exemplified in Figure 1.1.

Consider a specific target account a_0 . A set of K candidate accounts, $\{a_1, a_2, \dots, a_K\}$, is constructed, including one positive account a_{j^*} , where the index j^* correctly identifies the candidate account belonging to the same owner as a_0 , i.e., $E(a_0, a_{j^*}) = 1$.

Probabilistic Evaluation: In an ideal scenario, the inference process would always identify the positive account a_{j^*} as the closest account, ensuring it consistently ranks first among all candidate accounts. However, challenges such as transactional noise and the dynamic nature of Ethereum addresses make achieving exact matches in all cases unrealistic. This underscores the

need for a probabilistic evaluation framework that assesses model performance beyond strict *top-1* accuracy. Instead, the framework should measure the frequency with which the correct account appears within a meaningful rank threshold.

Inspired by the traditional Area Under the Curve (AUC) metric (Hanley & McNeil, 1982), the concept of a hit ratio is introduced to quantify the probability that the positive account a_{j^*} appears within the *top* – τ ranked candidates. Let $\text{rank}(a_{j^*})$ denote the position of a_{j^*} in the sorted list of candidate accounts. Rather than focusing solely on whether a_{j^*} ranks first, we define the *hit ratio* H_τ as the proportion of test cases in which a_{j^*} appears within the *top* – τ positions:

$$H_\tau = \frac{1}{N} \sum_{i=1}^N I(\text{rank}(a_{j^*}) \leq \tau) \quad (6.22)$$

where N is the total number of test cases, and $I(\cdot)$ is the indicator function, which returns 1 if the condition is true and 0 otherwise. This formulation enables the model to probabilistically evaluate ranking effectiveness across different levels of precision.

To comprehensively assess ranking quality across all possible rank thresholds, the AUC score is reformulated based on H_τ . Given a range of rank thresholds ($\tau = 1, 2, \dots, \tau_{max}$). The AUC is defined as:

$$\text{AUC} = \frac{1}{\tau_{max}} \sum_{i=1}^{\tau_{max}} H_i \quad (6.23)$$

This approach evaluates model performance even when an exact match is not achieved. By aggregating performance across multiple ranking thresholds, it provides a holistic view of inference effectiveness, making it particularly suitable for large anonymity sets where exact linkage is infeasible due to the natural challenges of large-scale blockchain environments.

To illustrate, consider an example where a target account a_0 is compared against a set of five candidate accounts, $S = \{a_1, a_2, a_3, a_4, a_5\}$, where a_3 is the positive account (*i.e.*, $E(a_0, a_3) =$

1). Suppose the distances computed by the pre-trained model M_θ yield the following ranking as follows: $\text{dist}(a_0, a_2) < \text{dist}(a_0, a_3) < \text{dist}(a_0, a_1) < \text{dist}(a_0, a_5) < \text{dist}(a_0, a_4)$.

Consider an example where a target account a_0 is compared against a set of five candidate accounts $S = \{a_1, a_2, a_3, a_4, a_5\}$, with a_3 being the positive account (i.e., $E(a_0, a_3) = 1$). Suppose the distances computed by the pre-trained model M_θ result in the following ranking as follows: $\text{dist}(a_0, a_2) < \text{dist}(a_0, a_3) < \text{dist}(a_0, a_1) < \text{dist}(a_0, a_5) < \text{dist}(a_0, a_4)$.

In this scenario, a_3 is ranked second among the candidate accounts, making the inference unsuccessful under the exact matching criterion, as it does not achieve *top-1* ranking.

However, using a probabilistic ranking approach, we assess the model effectiveness at different ranking thresholds using the hit ratio, as described in Equation 6.22. The hit ratio H_τ measures the likelihood that the positive account appears within the *top* τ ranked positions, as defined in Equation 6.23. Evaluating H_τ at different thresholds:

- $H_1 = 0$, since a_3 is ranked second and does not appear within the top 1.
- $H_2 = 1$, as a_3 is ranked within the top 2.
- $H_3 = 1$, as a_3 is ranked within the top 3.
- $H_4 = 1$, as a_3 is ranked within the top 4.
- $H_5 = 1$, as a_3 is ranked within the top 5.

To obtain an aggregate ranking performance measure, AUC score is computed as follows:

$$\text{AUC} = \frac{1}{\tau_{\max}} \sum_{i=1}^{\tau_{\max}} H_i \quad (6.24)$$

For this example, with $\tau_{\max} = 5$, we obtain the following result as $\text{AUC} = \frac{1}{5}(0 + 1 + 1 + 1 + 1) = 0.8$.

This metric is employed in Chapter 7 to evaluate the proposed methods under varying anonymity set conditions, facilitating a clearer and more evidence-based comparison of their effectiveness.

6.2. Inference Phase

Algorithm 2 describes the inference phase of *EDL*. In this phase, the behavioral feature vector of the target address a_j and the feature vectors of the candidate addresses $\{a_1, a_2, \dots, a_K\}$ are fed into the model (lines 1–3). After receiving these inputs, the model computes the behavioral difference vector $X_{j,k}$ between the target address a_j and each candidate a_k (lines 4–6).

Algorithm 2 Deanonymization of Wallet Ownership

Require: a_0 : Target , Anonymity Set: $AS = \{a_1, a_2, \dots, a_K\}$, M_Θ : From Algorithm 1

Ensure: Identified candidate address $a_{j^*} \in AS$ and Explanation

```

1: procedure COMPUTE BEHAVIORAL DIFFERENCES
2:   for each candidate  $a_j \in \{a_1, a_2, \dots, a_K\}$  do
3:      $X_{0,j} = |X_0 - X_j|$ 
4:   end for
5: end procedure
6: procedure PREDICT OWNERSHIP DISSIMILARITIES/DISTANCES
7:   Compute ownership distances through  $M_\Theta$ :
8:    $dist(y_{0,j}) \leftarrow M_\Theta(X_{0,j}), \quad \forall j \in \{1, 2, \dots, K\}$ 
9: end procedure
10: procedure IDENTIFY MOST PROBABLE CANDIDATE
11:   Select candidate with minimum distance:
12:    $j^* = \arg \min_j dist(y_{0,j})$ 
13: end procedure
14: procedure RETRIEVE EXPLAINABILITY REPORT
15:   Extract feature importances:
16:    $Explanation \leftarrow M_\Theta.importance(X_{0,j^*})$ 
17: end procedure
18: return  $a_{j^*}$ , Explanation

```

The computed difference vectors are then passed to the trained M_Θ , which outputs a distance score indicating the ownership similarity between the target address and each candidate (lines 7–8). A lower distance score suggests a higher likelihood that the two addresses belong to the same account owner. Finally, the inference identifies the most probable owner j^* by selecting the candidate address with the minimum predicted distance score (line 9). If j^* is controlled by the same owner, the inference has successfully broken the anonymous account ownership.

Algorithm 2 also details the explainable inference phase of EDL, which aims to deanonymize account ownership while maintaining explainability. During this phase, the behavioral feature vector of a target address a_0 and those of candidate addresses $a_1, a_2, \dots, a_K$ from the anonymity

set are input into the pre-trained model M_θ . For each candidate a_j , the algorithm computes the behavioral difference vector $X_{0,j}$ by taking the absolute difference between the feature vectors of a_0 and a_j (lines 2-3).

These difference vectors are then passed to the trained model M_θ , which predicts ownership distances, quantifying the similarity between the target address and each candidate (lines 7–8). A lower distance score indicates a higher likelihood of shared ownership. The algorithm then determines the most probable owner a_{j^*} by selecting the candidate with the minimum predicted distance score (lines 11-12). Finally, it retrieves an explainability report by extracting the importance of the characteristics of X_{0,j^*} , providing information on the rationale for the model decision (lines 10-11). If the selected candidate address a_{j^*} belongs to the same owner as a_0 (that is, $E(a_0, a_{j^*}) = 1$), the inference is considered successful in deanonymizing account ownership.

To summarize, *EDL* employs an xAI model like described in Section 3.3 to enhance transparency in identifying the most likely associated account. By embedding interpretability into its inference pipeline, the system enables investigators to trace and justify predictions using comprehensible model outputs, thereby improving auditability and fostering trust in blockchain deanonymization.

Furthermore, investigators can validate *EDL*'s predictions by cross-referencing external blockchain data sources (e.g., Etherscan, Chainalysis), thereby reducing the risk of false positives. This xAI-driven approach enhances accountability and scalability while preserving interpretability throughout the investigation process.

From an empirical perspective, the effectiveness of *EDL* is supported by the comprehensive experimental analyses presented in Section 7, which demonstrate that it outperforms competing methods in both internal design and overall performance. Furthermore, for a real-world demonstration of how *EDL* effectively supports blockchain investigators, see the case study presented in Section 7.5.

7. EXPERIMENTAL RESULTS

In this chapter, we present a comprehensive evaluation of the proposed solution by addressing four key research questions aimed at assessing its effectiveness as follows:

RQ1: To what extent can *EDL* accurately deanonymize account ownership when analyzing naïve transactional behaviors, and how does its performance compare to a baseline method?

RQ2: How sensitive is *EDL* to variations in critical hyperparameters, including training set size and the choice of inherently xAI backbone model?

RQ3: How does *EDL* perform relative to existing state-of-the-art methods in terms of deanonymization accuracy and robustness?

RQ4: How well does *EDL* support human-AI collaboration in practical blockchain investigation scenarios?

To systematically address these questions, the dataset characteristics and experimental configuration are first presented in Section 7.1. Baseline performance is established in Section 7.2, based on the transactional behaviors introduced in Section 6.1.1. The impact of key hyperparameters on model performance is analyzed in Section 7.3, followed by a comparative evaluation against state-of-the-art methods in Section 7.4. Finally, Section 7.5 presents a real-world case study demonstrating the applicability of *EDL* to forensic investigations involving human-AI collaboration.

7.1. Experimental Setup

Dataset Description. Since the literature provides only one publicly available real-world benchmark dataset containing 103 account pairs for the AOL problem, we strictly adhere to the original evaluation protocol and dataset configuration of (Du et al., 2023). We also replicate its reported results to ensure fairness and reproducibility before conducting our own comparisons.

Unfortunately, this benchmark frames AOL as a binary classification task with an anonymity-set size of only two—an unrealistic setup compared to real-world investigations, where anonymity sets commonly range between 100 and 200 accounts (see, e.g., Fig. 20 of (Béres et al., 2020) for an analysis of anonymity-set sizes).

To better reflect operational conditions, we additionally construct a dynamic evaluation dataset in which anonymity-set sizes are systematically varied (e.g., 2, 5, 100, 200). In parallel, we expanded a ground truth set of 168 real-world account pairs, each belonging to the same owner, compiled from the literature (Béres et al., 2020; Du et al., 2023) and expanded through various incidents, including notable disclosures such as the Upbit tracing case (*Tracing the Trail of the Upbit Hack*, n.d.), where account ownership was publicly revealed by the blockchain community. To gain comprehensive insights into the behavior of these accounts, we extracted their transaction histories using the *Etherscan API*³, focusing on the raw data listed in Table 3.1.

In addition to the distinct accounts in the ground truth set, the transaction histories of 10,000 unique accounts were extracted to expand the anonymity set. These accounts were then used through 10 random replications for experimental evaluation.

After obtaining the transactional data for the accounts, behavioral profiling vectors are extracted in three categories as defined in Section 6.1.1: temporal, volumetric, and infrastructural. These vectors are then used to construct the dataset required for training of the learning model. This dataset serves as the training dataset for the model, which learns dissimilarity or distance based on the behavioral differences of the accounts, as detailed in Section 6.1.2.

Settings. The overall implementation of *EDL* was developed using our custom Python code, integrating scikit-learn for model training and evaluation, while employing interpret⁴ to extract feature importance and explain model predictions. To construct behavioral representations for each distinct wallet, we retrieved the most recent 10,000 transactions per account via the Etherscan API. Feature vectors were then computed using our custom feature engineering pipeline, following the methodologies outlined in Section 6.1.1. These features capture temporal, volumetric, and infrastructural characteristics, as listed in Table 6.1.

³ <https://docs.etherscan.io/etherscan-v2>

⁴ <https://interpret.ml/docs/ebm.html>

We compared five widely used inherently xAI models as the backbone of *EDL*: EBM (Lou et al., 2013), LR (Hosmer Jr et al., 2013), DT (Breiman et al., 2017), XGB (Chen & Guestrin, 2016), and LGB (Ke et al., 2017). Each model was trained using its default configuration to ensure consistency in evaluation.

To enhance the robustness and reliability of our results, we applied cross-validation for model evaluation, minimizing the risk of overfitting and ensuring a fair comparison across models.

Evaluation Metrics. As mentioned in Section 4.2, solutions to the AOL problem aim to predict the correct account from a set of candidate accounts by establishing ownership linkage as a deanonymization task. To evaluate these types of solutions, standard machine learning metrics, including precision, recall, and F1-score, are utilized earlier in the literature (Du et al., 2023). In these metrics, precision measures the proportion of correctly linked account pairs among those predicted as linked (true positives over the sum of true positives and false positives). Meanwhile, recall quantifies the proportion of actual linked account pairs that were correctly predicted (true positives over the sum of true positives and false negatives). The F1-score represents the harmonic means of precision and recall, providing a balanced assessment of model performance.

Furthermore, as discussed in Section 6.1.3, exact linkage may not always be feasible in large-scale blockchains like Ethereum due to the large size of anonymity sets. Therefore, we also incorporate the AUC metric, as defined in Equation 6.21 and 6.22, into our experiments to evaluate linkage performance under partial linkage conditions.

For a robust evaluation, we utilize 10-fold cross-validation, where the dataset is randomly divided into 10 equal subsets. In each iteration, nine subsets are allocated for training, while the remaining subset is used for testing. The overall performance is determined by averaging the results across all folds.

7.2. Baseline Experiments

To address *RQ1*, we simulate naïve profiling attacks as a baseline for the AOL task using only profile vectors. These attacks are categorized along the three behavioral dimensions described in Section 6.1.1: temporal, volumetric, and infrastructural.

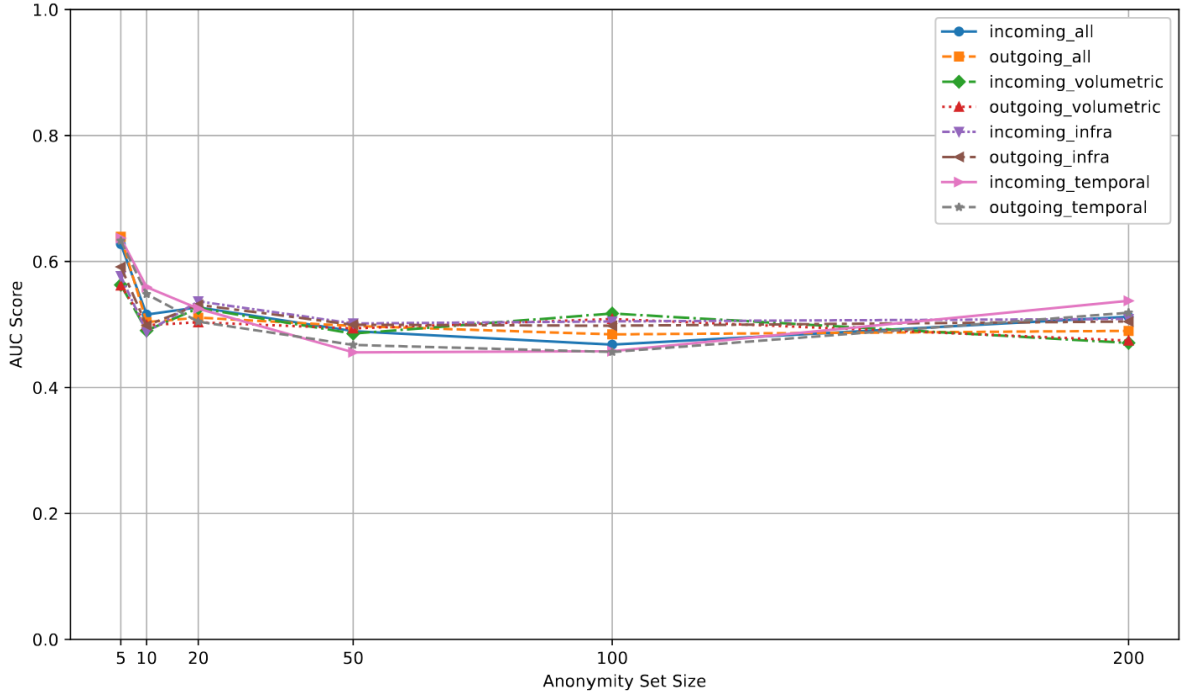


Figure 7.1 Baseline Experiments

We construct eight distinct baseline scenarios based on these behavioral categories and transaction directions (incoming, outgoing). A detailed description of these scenarios is provided in Table 6.1, where each scenario specifies the transaction data and feature categories used in the baseline profiling attack. By default, profile vectors are generated from outgoing transactions; however, when the incoming setting is applied, profiling includes both outgoing and incoming transactions. Using this transaction configuration, baseline profiling attacks are conducted across temporal, volumetric, infrastructural, and combined feature categories (all).

Table 7.1 Baseline Experiment Settings

Setting	Description
<i>outgoing_all</i>	Outgoing transactions with features from all categories: temporal, volumetric, and infrastructural.
<i>outgoing_temporal</i>	Outgoing transactions considering only temporal features.
<i>outgoing_volumetric</i>	Outgoing transactions considering only volumetric features.
<i>outgoing_infrastructural</i>	Outgoing transactions considering only infrastructural features.
<i>incoming_all</i>	Combined outgoing and incoming transactions with features from all categories: temporal, volumetric, and infrastructural.
<i>incoming_temporal</i>	Combined outgoing and incoming transactions with only temporal features.
<i>incoming_volumetric</i>	Combined outgoing and incoming transactions with only volumetric features.
<i>incoming_infrastructural</i>	Combined outgoing and incoming transactions with only infrastructural features.

As shown in Figure 7.1, the performance of these baseline attacks under varying settings reveals significant limitations, as evidenced by the low AUC scores. Notably, an AUC score close to 0.5 indicates performance nearing random chance, highlighting the unsuccessful nature of the attack in these cases. This implies two possible reasons: first, the features selected for profiling account behavior may be inadequate, leading to suboptimal outcomes. Alternatively, it may reflect the likelihood that account owners exhibit varying behaviors when managing multiple accounts, a factor that cannot be disregarded.

Given that the features employed are widely accepted and commonly used in the literature (Baci et al., 2022; Béres et al., 2020; Farrugia et al., 2020; Kılıç & İnan, 2024; Li et al., 2021; Pan et al., 2024) the latter explanation appears more plausible. Account owners may, in fact, exhibit varying behavioral patterns when interacting with different accounts, even if they control them.

To investigate this further, we conduct a separate experiment where the behavioral differences of account owners are examined in more detail. In this experiment, we analyze account owner

behaviors based on the behavioral categories described in Section 6.1.1. For this analysis, we use a naïve clustering approach, specifically K-means (Lloyd, 1982), chosen for its simplicity and efficiency in handling large volumes of transaction data.

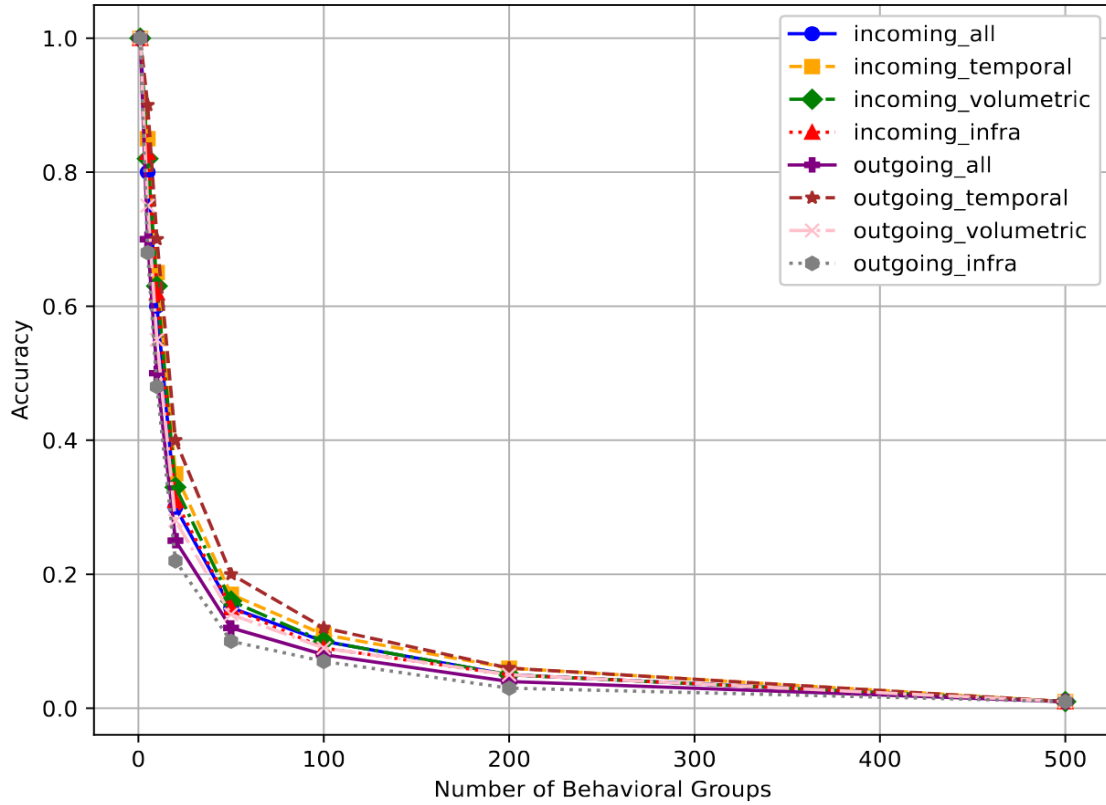


Figure 7.2 Behavioral Grouping Analysis

We also examine the frequency with which accounts controlled by the same account cluster into the same behavioral group, as shown in Figure 7.2. The accuracy outcomes of this behavioral categorization indicate that the optimal scenario occurs when accounts controlled by the same owner are consistently assigned to the same groups, regardless of the number of groupings employed. However, the results indicate that accuracy generally decreases as the number of behavioral groups increases. This decline implies that account owners exhibit distinct behavioral patterns across their accounts.

7.3. Performance Analysis of *EtherDiffLinkage*

To answer RQ2, we first analyze the performance of *the EDL* across two key components: the xAI model, as characterized in Section 3.3 and the behavioral profiling approach outlined in Section 6.1.1. Specifically, we aim to see how these components contribute to the overall performance of *EDL*.

Model Evaluation: A crucial step in training *EDL* is the selection and tuning of its key hyperparameters. In this part, we focus on two fundamental hyperparameters: the backbone AI model and the training-to-testing ratio. To systematically assess their impact, we evaluate model performance across various hyperparameter configurations. Based on the experimental results, we determine the optimal configuration and designate the best-performing model as the backbone for *EDL*. A detailed analysis of these evaluations is presented in the following discussion.

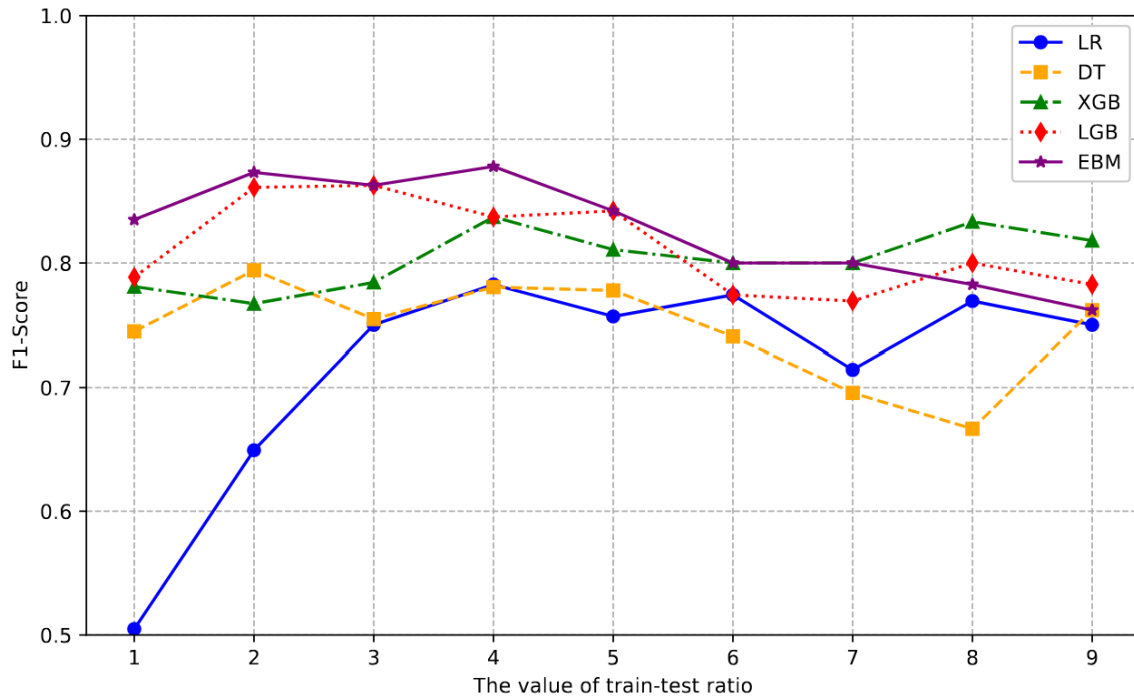


Figure 7.3 Train-Test Ratio Effect

As illustrated in Figure 7.3, we first examine the impact of varying the training-to-testing ratio on model performance by evaluating randomly selected data splits with ratios ranging from 1

to 9. The results indicate that the EBM achieves the highest F1-score, reaching 0.87 when the training-to-testing ratio is set to 4. Furthermore, to implement robust validation of these results, we conduct additional experiments using varying K-fold cross-validation, as depicted in Figure 6.4. In this setup, we vary the value of K by increasing the training proportion as K increases. The results demonstrate that model performance generally improves with larger training data availability. Among the models compared, *EBM*-based models as formulated in Section 6.1.2 achieve the best performance, obtaining an F1-score of 0.873 under 10-fold cross-validation.

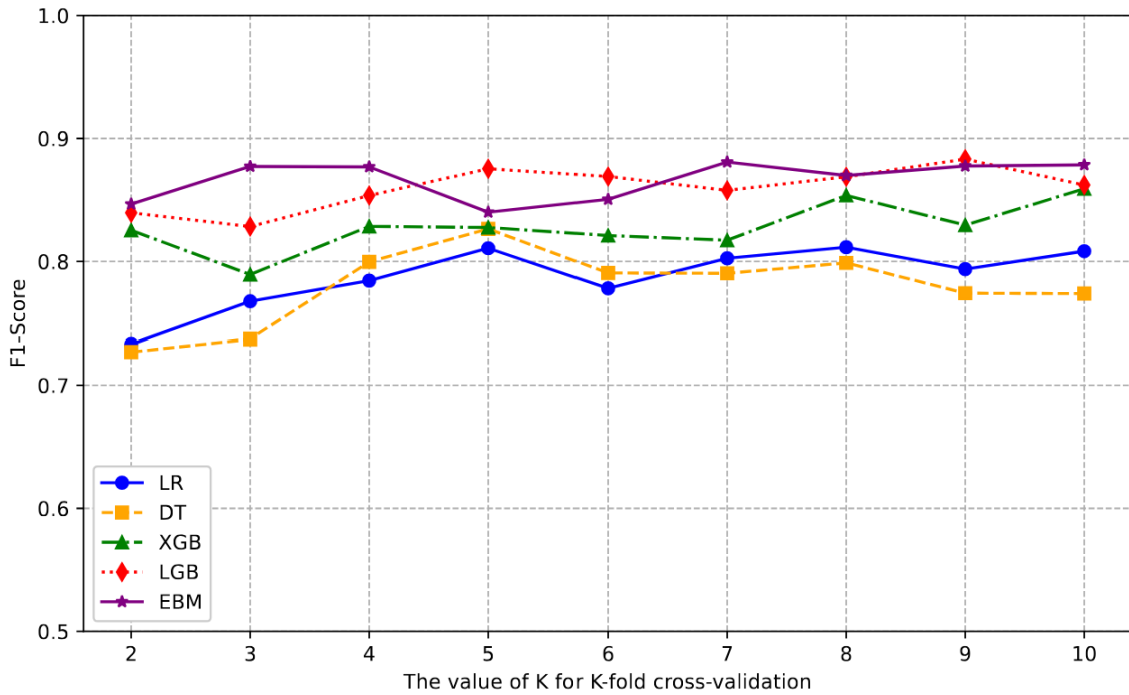


Figure 7.4 K-Fold Model Performance Analysis

Moreover, the other key observations are as follows:

- In general, as the value of K increases, meaning a larger proportion of data is used for training, model performance either improves or remains stable. This trend suggests that as models are exposed to more data, they better capture behavioral patterns between accounts managed by the same owner, leading to improved performance.
- Among the compared models, boosting-based methods such as *XGB*, *LGB*, and *EBM* consistently outperform traditional models including *DT* and *LR*. The relatively lower performance of *DT* can be attributed to its rule-based structure, which is prone to

overfitting, while LR 's linear nature limits its ability to capture complex patterns in addressing AOL tasks.

- The EBM model demonstrates superior performance across most settings, achieving the highest F1-scores compared to other models. This result highlights the effectiveness of EBM's explainable structure and suggests its potential as a promising backbone model of *EDL* for solving the AOL task described in Section 4.2.

Ablation Study. To assess the contribution of each behavioral feature category in *EDL*, we constructed three ablated variants of our full model by removing temporal, volumetric, and infrastructural features, respectively. The performance of these variants is presented in Figure 7.5. Removing any feature group results in a measurable decline in precision, recall, and F1, indicating that each category plays a meaningful role in the deanonymization process. Overall, removing any of the temporal, volumetric, or infrastructural feature groups leads to a similar reduction in performance, indicating that each behavioral dimension provides complementary and essential signals for accurate deanonymization. These results confirm that all three behavioral dimensions contribute complementary evidence and that effectiveness of *EDL* relies on their combined use.

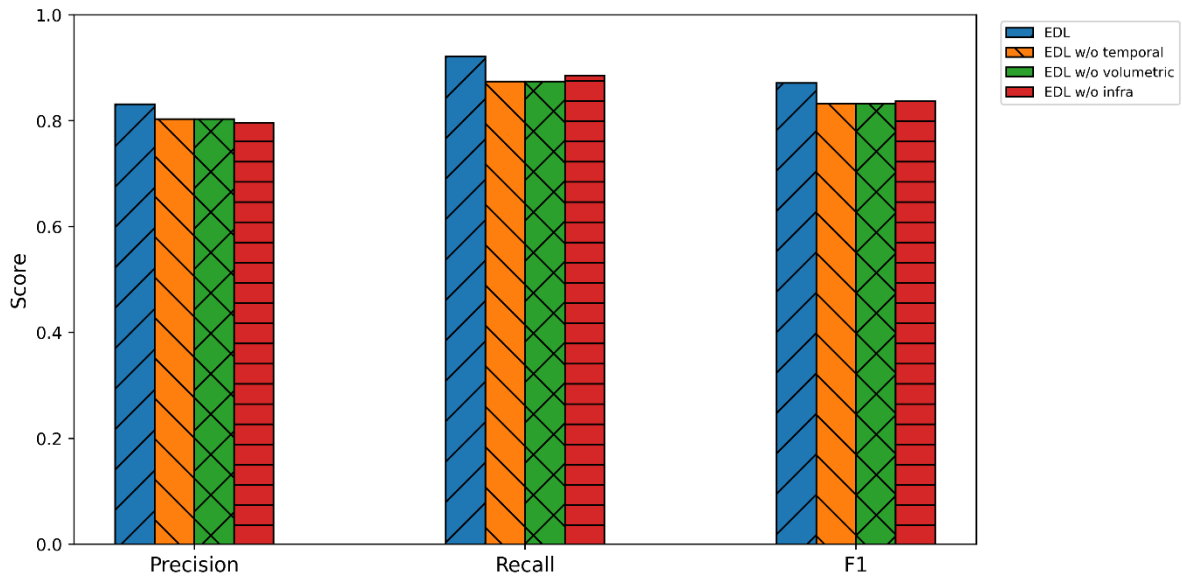


Figure 7.5 Ablation Study

Evaluation on Features: We analyze the impact of the features on the backbone model. The goal is to extract the globally influential features directly from the model, which is feasible due to the inherently explainable nature of the model.

At the global level as an average case, the most important features, sorted in descending order, are shown in Figure 7.6. The key observations from these results are as follows:

- Notably, in addition to the default outgoing transaction features, incoming transaction-related features (in `_` prefixed attributes) also exhibit high importance scores. This indicates that transactions received by an account might serve as significant indicators for deanonymizing account ownership. The high ranking of these features suggests that behavioral patterns related to incoming funds can be as revealing as outgoing transaction activities.
- Among the top three most important features, two are related to gas consumption per transaction. This observation implies that account owners tend to exhibit consistent gas consumption patterns across different accounts, likely reflecting the computational workload associated with their transactions. Such behavioral consistency indicates that transaction gas usage might serve as a distinguishing factor in the AOL task.
- The experimental results presented in Figure 7.6 demonstrate that infrastructural, temporal, and volumetric transaction features contribute significantly to the AOL task. The leading features highlight the importance of understanding account behavior across different dimensions. The strong influence of temporal features suggests that activity frequency and transaction inter-arrival times are crucial in distinguishing account ownership.
- Furthermore, the distribution of feature importance underscores the fact that no single feature category dominates the prediction model. Instead, a combination of infrastructural, temporal, and volumetric factors collectively improves deanonymization performance. The relatively lower importance of contract interaction ratios implies that while smart contract engagement is relevant, it is less distinctive compared to the other core transaction characteristics.

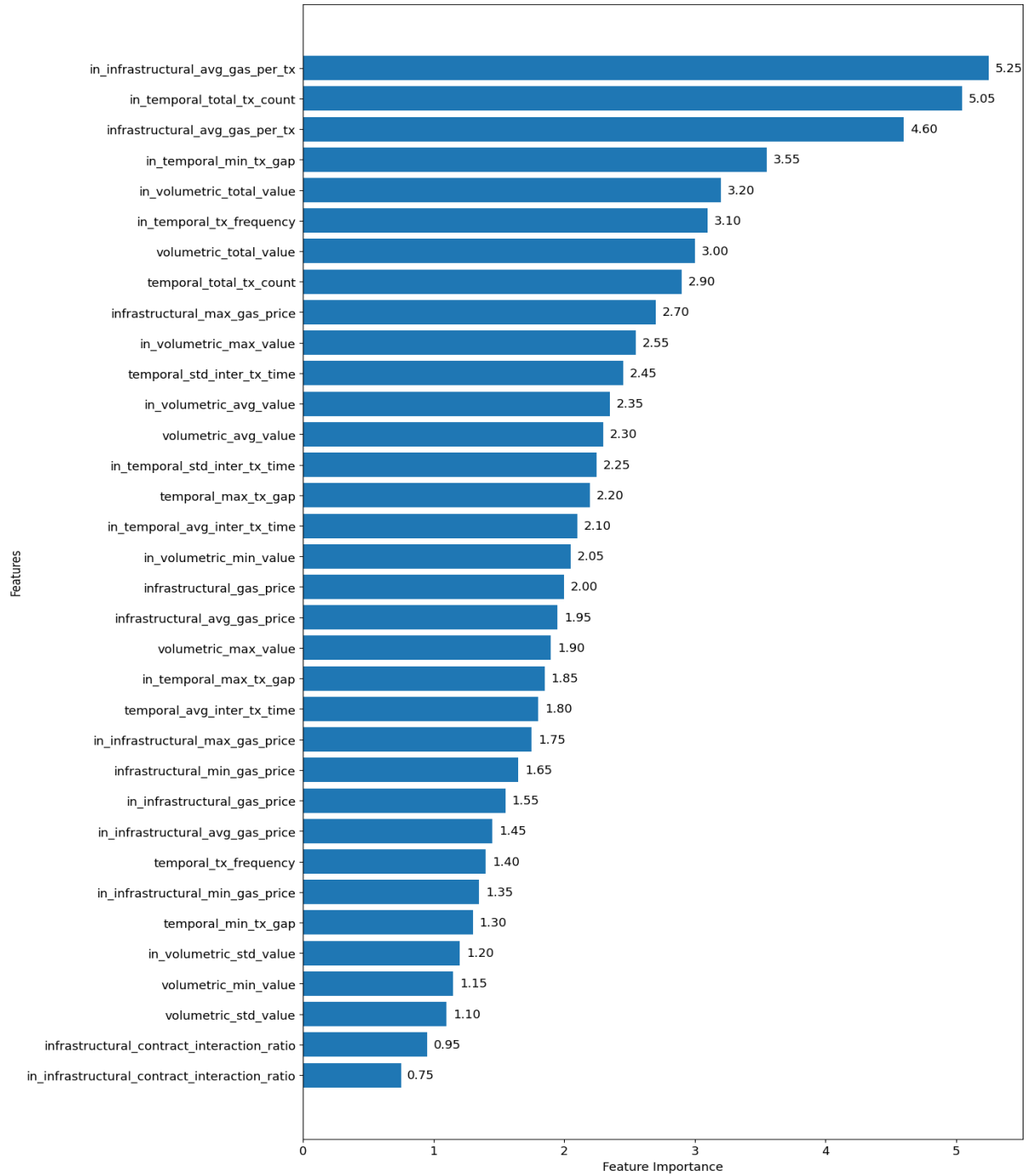


Figure 7.6 Overall Feature Importance Scores

7.4. Comparative Analysis

To comprehensively evaluate the performance of our solution as an answer to RQ3, we conduct a comparative analysis against two recent state-of-the-art methods:

- *BERT4ETH* (Hu et al., 2023), as a transaction sequence behavior-based account representation learning approach proposed by Hu et al. This method leverages BERT-based sequence modeling to capture transaction patterns and infer account ownership.
- *MixBroker* (Du et al., 2023), as a graph-based feature learning approach introduced by Du et al., designed to correlate mixer users by analyzing their structural and interaction properties within the Ethereum network.

To ensure a fair comparison, we fine-tune all methods to achieve their optimal performance on the evaluation datasets. The results are analyzed based on key performance metrics, as detailed in Section 7.3. By benchmarking *EDL* against recent methods, we aim to evaluate its effectiveness in a comparative manner.

Table 7.2 Performance Comparison over benchmark dataset

<i>Metric</i>	<i>MixBroker</i>	<i>BERT4ETH</i>	<i>EDL(Ours)</i>
Precision	<u>0.878</u>	0.775	0.893
Recall	<u>0.847</u>	0.742	0.867
F1-Score	<u>0.854</u>	0.760	0.878

After standardizing the experimental conditions for each solution, we evaluated the effectiveness of our solution by comparing its classification performance with the other methods based on precision, recall, and F1-score. According to the results presented in Table 7.2, we obtain the following key observations:

- *EDL* and *MixBroker* achieve comparable performance, while *BERT4ETH* exhibits relatively lower scores across all evaluation metrics. This discrepancy suggests that *BERT4ETH*'s underlying assumption that accounts controlled by the same owner exhibit similar transaction sequence behaviors fails to effectively capture the complexities of the AOL problem. Unlike *EtherDiffLinkage* and *MixBroker*, which leverage diverse behavioral and structural features to enhance address linkage, *BERT4ETH* primarily relies on sequential transaction patterns, which may not generalize well across different types of Ethereum users. Consequently, its effectiveness

is limited when applied to the broader AOL problem, where user behavior varies significantly.

- *EtherDiffLinkage* achieves performance metrics comparable to those of *MixBroker* while exhibiting a relative improvement of 1.7% in precision, 2.3% in recall, and 2.7% in F1-score. This demonstrates that *EDL* not only matches but slightly surpasses *MixBroker* in classification performance. Unlike *MixBroker*, which is specifically designed for mixer users, *EDL* can be applied to any pair of Ethereum accounts, making it a more versatile solution for the AOL problem. The combination of its broader applicability and superior performance highlights our solution as a more effective and generalizable approach compared to *MixBroker*.

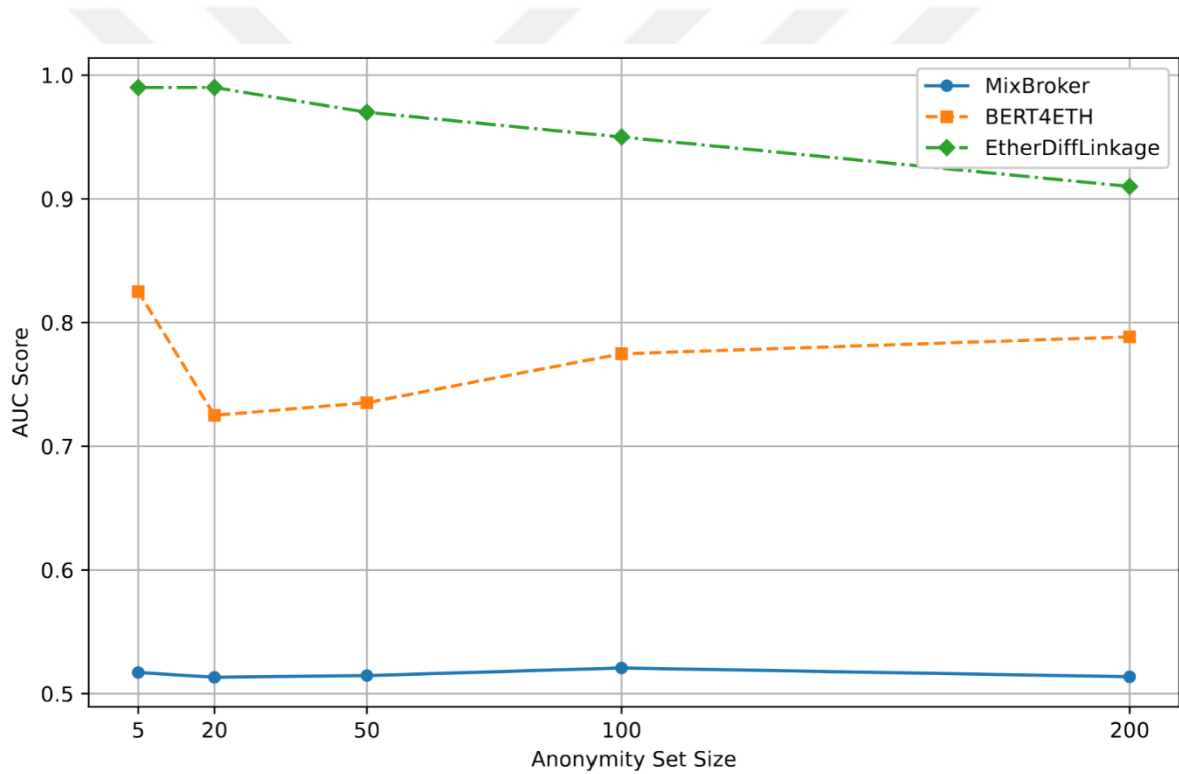


Figure 7.7 Performance Comparison under Varying Anonymity Set Size

In a separate experiment, we utilize AUC metric to conduct a comparative analysis of *EDL*, as shown in Figure 7.7. Based on the results presented, we make the following key observations:

- *MixBroker* exhibits the worst performance across all anonymity set sizes, which can be attributed to its design being specifically tailored for mixer users. While it performs well in low-anonymity settings due to its classification-based approach, its performance degrades significantly as the anonymity set size increases, approaching near-random chance levels (0.5) in larger anonymity sets.
- Our solution consistently outperforms the other two methods in probabilistic linkage performance across all anonymity set sizes. It achieves a relative improvement of approximately 15% over its closest competitor, *BERT4ETH*, even at large anonymity set sizes, with a maximum improvement of 32%. This indicates that the behavioral difference patterns captured by *EDL* are more effective for the AOL task than transaction sequence-based behavior modeling. Even when the anonymity set size increases, *EDL* maintains its advantage by successfully ranking candidate accounts, demonstrating its robustness in addressing the AOL problem.

These results demonstrate that our solution is effective not only in small anonymity sets but also in larger ones, more successfully deanonymizing account ownership even under highly anonymized conditions. Additionally, by leveraging its inherent xAI model, it provides interpretable explanations for each prediction. Please refer to next section for a real-world case study demonstrating *EDL* in practice.

7.5. Case Study: A Real-World Blockchain Investigation with Human-AI Collaboration using *EDL*

To demonstrate the applicability of *EDL* in real-world blockchain investigations in response to RQ4, we analyze its effectiveness in a blockchain investigation scenario as follows:

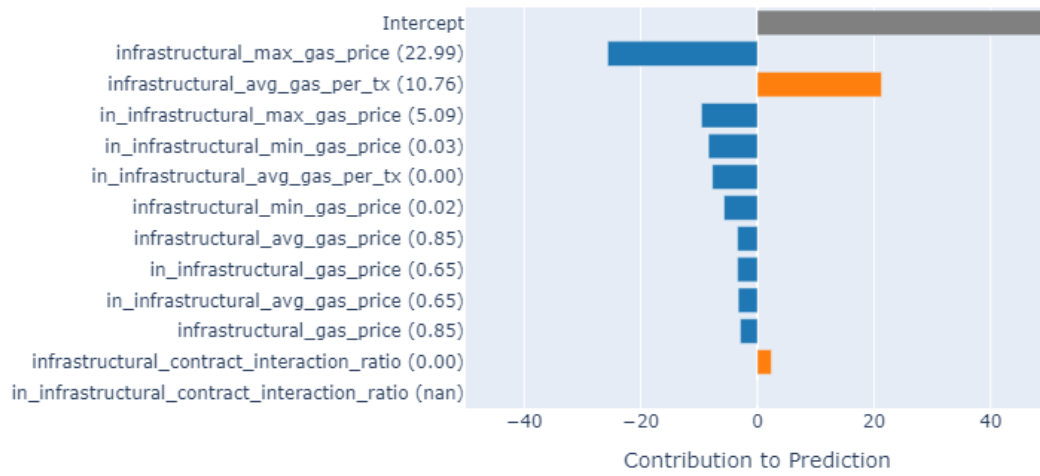
In December 2019, an account, denoted as A1 (0x01598e22f...) interacted with Tornado Cash, making a deposit transaction on Dec-27-2019 at 06:20:44 AM UTC. Based on domain knowledge and prior blockchain analysis, 134 potential withdrawal addresses were identified, forming an anonymity set. The objective is to determine the most likely withdrawal address linked to A1.

To achieve this, we employ *EDL*, which constructs behavioral profile vectors for each address and computes behavioral difference vectors for all possible account pairs as model inputs. The *EDL* then assigns an ownership distance score between 0 and 100 to each pair, with lower scores indicating a stronger likelihood of ownership. By ranking these distances, the most probable withdrawal address is inferred. To validate the results, we verify the ground truth using well known heuristics, confirming that the actual withdrawal address is A2 (0x85894273...).

When the *EDL* is applied to this case, A2 ranks 2nd in the inferred results. In contrast, *MixBroker* ranks A2 as 23rd, while BERT4ETH ranks it 27th. In detail, regarding explainability, *EDL*'s inherently explainable-by-design properties facilitate feature-level interpretability of model outputs, enabling investigators to analyze the underlying factors influencing predictions. As illustrated in Figure 7.7, the feature importance of each inference result is provided, supporting a systematic investigation of predictions and allowing blockchain investigators to identify and discard misleading inference results.

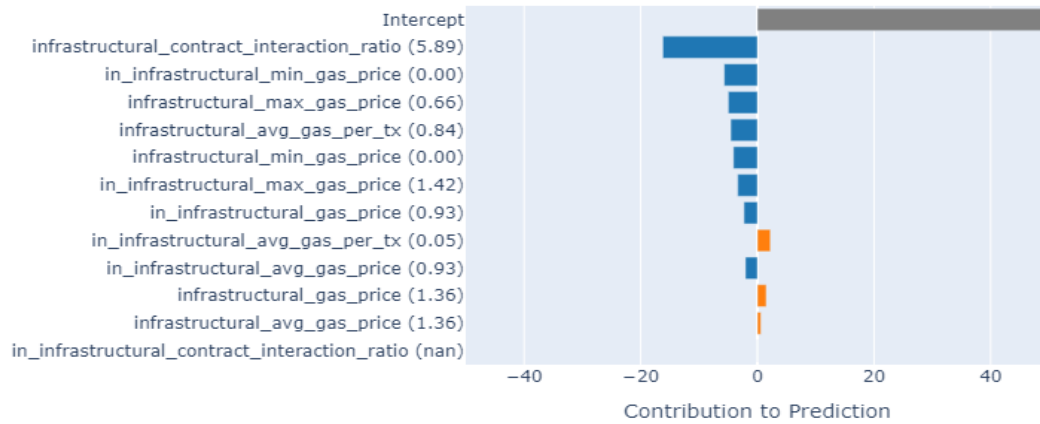
In this case, the account that should ideally be ranked first is observed to be ranked second. A detailed examination of Figure 7.8 reveals that infrastructural features play a dominant role in the decision-making process. Specifically, in the rank-1 analysis (Figure 7.8 (a)), the difference in the user's average gas consumption per transaction has a significant impact on the model's decision. The substantial contribution of *infrastructural_avg_gas_per_tx* and *infrastructural_max_gas_price* indicates that gas consumption difference patterns are highly influential in this investigation.

Local Explanation (Actual: 0 | Predicted: 3.63)



(a)

Local Explanation (Actual: 0 | Predicted: 11)



(b)

Figure 7.8 Inference result at (a) rank 1 (b) rank 2

However, in the result at rank-2 (Figure 7.7(b)), the impact of these gas-related features diminishes. Instead, *infrastructural_contract_interaction_ratio* appears as a dominant feature, indicating that contract interaction difference patterns influence the model's decision differently in this ranking. The reduction in the significance of gas consumption features, along

with the shift towards contract interaction-based features, suggests these accounts have similar contract interaction patterns. This discrepancy highlights the importance of granular analysis in blockchain investigations, enabling investigators to identify cases where dominant features may introduce ranking inconsistencies. By leveraging *EDL*'s capabilities, investigators can refine their assessments and improve the accuracy of ownership attribution, reducing the likelihood of incorrect linkages. This underscores its practical applicability and robustness in real-world blockchain investigations.



8. DISCUSSION

In this chapter, we discuss two practical concerns regarding the real-world applicability of the proposed solution.

Generalization. Our solution is designed to model adaptive behavioral relationships rather than fixed mixer-specific patterns, enabling natural extensibility to unseen or evolving obfuscation strategies. The system’s profiling and linkage architecture is not tailored to any mixing protocol; instead, it captures generalizable transactional behaviors observable across the Ethereum ecosystem. Specifically, the extracted behavioral features in EDL (please refer to Section 6.1.1) are built entirely around the fundamental transaction dynamics of the Ethereum infrastructure, rather than any application-level structure such as mixers, tumblers, or specific privacy protocols. This design choice ensures that the profiling mechanism remains agnostic to service types and instead captures intrinsic, quantifiable patterns of account interaction that can emerge under any transaction scenario. So, even when novel or evolving mixing strategies appear, their operational footprints—expressed through measurable temporal, volumetric, or infrastructural variations—continue to map naturally into the same feature space.

This general, interpretable feature design allows both the model and human investigators to reason about unseen behavioral shifts without requiring mixer-dependent heuristics, task-specific retraining, or handcrafted rules. Furthermore, extending EDL to other smart-contract-based mixing mechanisms (e.g., TyphoonCash, Railgun, cross-chain mixers) only requires introducing chain- or asset-specific preprocessing modules, mirroring the adaptations described in recent work on multi-chain deanonymization. Therefore, these properties imply that the underlying methodology is well suited to handling emerging and evolving obfuscation behaviors.

Ethical Issues. *EDL* is developed with explicit attention to forensic integrity and ethical constraints. All data used in this study comes from publicly accessible blockchain transactions and publicly disclosed labels, ensuring that no personally identifiable information is processed. The framework is intended strictly as an investigative aid in analyzing suspicious or illicit fund flows, and all outputs are probabilistic linkage scores, not deterministic accusations. These

inferences require verification by human analysts and are not intended for automated attribution. Moreover, to further mitigate risks such as false positives or misuse, we recommend the use of transparent models (e.g., inherently XAI models as described in Section 3.3) together with confidence-score reporting, adjustable decision thresholds, and auditable model outputs. These safeguards help maintain the system’s suitability for investigative contexts while upholding ethical standards and minimizing potential harm.



9. CONCLUSION AND FUTURE WORK

In this dissertation, we investigate the problem of deanonymizing wallets associated with illicit activities by identifying behavioral differences. We systematically analyze account behaviors from an investigator’s perspective through behavioral profiling and propose *EtherDiffLinkage (EDL)*, a framework designed to link accounts controlled by the same owner based on transactional behavior. *EDL* constructs account profiles by analyzing blockchain transaction histories, focusing on temporal, volumetric and infrastructural features, which serve as behavioral signatures for ownership linkage. We frame the *AOL* problem as a machine learning task based on ownership dissimilarity and evaluate our approach using a real-world dataset of account pairs confirmed by multiple sources to be illicitly controlled by the same entity. Experimental results demonstrate that *EDL* achieves up to a 32% improvement in AUC and 2.7% improvement in F1-score compared to state-of-the-art methods, even across varying anonymity set sizes. These findings highlight the effectiveness of the proposed solution in blockchain investigations.

In future work, we plan to enhance *EDL* by incorporating the evolving patterns of malicious activities. Rather than building a model from the ground up, we anticipate that leveraging state-of-the-art large language models (LLMs) augmented with domain-specific knowledge will introduce innovative improvements in addressing this challenge. Additionally, we plan to focus on increasing the transparency and interpretability of the solution, addressing the inherent need for explainability in this domain. This will not only make the model more understandable to investigators but also foster greater trust in its outcomes.

REFERENCES

- AbuElHassan, S., Abo Alian, A., AbdelKader, T., & Badr, N. (2025). A Review on Privacy-Preserving Techniques for Spatiotemporal Data. *International Journal of Data Science and Analytics*, 1–14.
- Agarwal, R., Thapliyal, T., & Shukla, S. (2022). Analyzing malicious activities and detecting adversarial behavior in cryptocurrency-based permissionless blockchains: An Ethereum use case. *Distributed Ledger Technologies: Research and Practice*, 1(2), 1–21.
- Amparore, E., Marseguerra, M., & Galassi, A. (2023). LEAF: Local Explanations via Aggregated Feature attribution. *arXiv Preprint arXiv:2311.09341*. <https://arxiv.org/abs/2311.09341>.
- Anoica, A., & Levard, H. (2018). Quantitative description of internal activity on the Ethereum public blockchain. 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 1–5. <https://doi.org/10.1109/NTMS.2018.8328741>.
- Ao, X., Liu, Y., Qin, Z., Sun, Y., & He, Q. (2021). Temporal high-order proximity aware behavior analysis on Ethereum. *World Wide Web*, 24(5), 1565–1585.
- Baci, A. B., Brousmiche, K., Amal, I., Abdelhédi, F., & Rigaud, L. (2022). Detecting illicit Ethereum accounts based on their transaction history and properties and using machine learning. *The International Conference on Deep Learning, Big Data and Blockchain*, 97–108.
- Bai, Q., Zhang, C., Liu, N., Chen, X., Xu, Y., & Wang, X. (2021). Evolution of transaction pattern in Ethereum: A temporal graph perspective. *IEEE Transactions on Computational Social Systems*, 9(3), 851–866.
- Baumann, A., Fabian, F., & Lischke, M. (2014). Insights into the Bitcoin Network: A View from the Inside. *Proceedings of the 14th IFIP TC 6.11 Conference on e-Business, e-Services and e-Society (I3E)*, 441, 1–15. https://doi.org/10.1007/978-3-662-44760-3_1.
- Béres, F., Seres, I. A., Benczúr, A. A., & Quinyne-Collins, M. (2020). Blockchain is watching you: Profiling and deanonymizing Ethereum users.
- Bonifazi, G., Corradini, E., Ursino, D., & Virgili, L. (2022a). Defining user spectra to classify Ethereum users based on their behavior. *Journal of Big Data*, 9(1), 37.
- Bonifazi, G., Corradini, E., Ursino, D., & Virgili, L. (2022b). Defining user spectra to classify Ethereum users based on their behavior. *Journal of Big Data*, 9(1), 37. <https://doi.org/10.1186/s40537-022-00586-3>.
- Bove, A., Bertossi, L., & others. (2024). Fail-Aware Explanations: On the Robustness of Post-Hoc Explanation Techniques. *arXiv Preprint arXiv:2401.01945*. <https://arxiv.org/abs/2401.01945>.

Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). *Classification and Regression Trees*. Routledge.

Buterin, V. (2014). *Ethereum Whitepaper: A next-generation smart contract and decentralized application platform*. <https://ethereum.org/en/whitepaper/>.

Buterin, V., & Griffith, V. (2017). *Casper the Friendly Finality Gadget*. <https://arxiv.org/abs/1710.09437>.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). *Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission*. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1721–1730. <https://doi.org/10.1145/2783258.2788613>.

Cavalheiro, E. A., Vieira, K. M., & Thue, P. S. (2024). *The impact of investor greed and fear on cryptocurrency returns: A Granger causality analysis of Bitcoin and Ethereum*. *Review of Behavioral Finance*, 16(5), 819–835. <https://doi.org/10.1108/RBF-08-2023-0224>.

Chainalysis Team. (2023). *2024 Crypto Crime Report*. Chainalysis. <https://www.chainalysis.com/crypto-crime-report-2024/>.

Chainalysis Team. (2024). *2025 Crypto Crime Report*. Chainalysis. <https://www.chainalysis.com/blog/cryptohacking-stolen-funds-2025/>.

Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Cicek, A. E., Nergiz, M. E., & Saygin, Y. (2014). *Ensuring location diversity in privacy-preserving spatio-temporal data publishing*. *The VLDB Journal*, 23(4), 609–625.

Dorri, A., Roulin, C., Jurdak, R., & Kanhere, S. S. (2019). *On the activity privacy of blockchain for IoT*. *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, 258–261.

Du, H., Che, Z., Shen, M., Zhu, L., & Hu, J. (2023). *Breaking the anonymity of Ethereum mixing services using graph feature learning*. *IEEE Transactions on Information Forensics and Security*.

Elliptic. (2022). *Obfuscation on the blockchain: How to detect and mitigate the risks*. <https://www.elliptic.co/blog/obfuscation-on-the-blockchain-how-to-detect-and-mitigate-the-risks>.

Elliptic Enterprises Limited. (2025). *Typologies Report 2025: How Criminals Evade Detection Across Blockchains*. <https://www.elliptic.co/research/typologies-report-2025>.

- Farrugia, S., Ellul, J., & Azzopardi, G. (2020). Detection of illicit accounts over the Ethereum blockchain. *Expert Systems with Applications*, 150, 113318.
- Ferretti, S., & D'Angelo, G. (2020). On the Ethereum blockchain structure: A complex networks theory perspective. *Concurrency and Computation: Practice and Experience*, 32(12). <https://doi.org/10.1002/cpe.5493>.
- Fleder, M., Kester, M. S., & Pillai, S. (2015). Bitcoin transaction graph analysis. *arXiv Preprint arXiv:1502.01657*.
- Fu, B., Yu, X., & Feng, T. (2022). CT-GCN: A phishing identification model for blockchain cryptocurrency transactions. *International Journal of Information Security*, 21(6), 1223–1232.
- Gao, Y., Shi, J., Wang, X., Shi, R., Yin, Z., & Yang, Y. (2021). Practical deanonymization attack in Ethereum based on P2P network analysis. *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications*, 1402–1409. <https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00191>.
- Gilad, Y., & Herzberg, A. (2017). Elliptic curve cryptography in practice. *Computers & Security*, 67, 65–86. <https://doi.org/10.1016/j.cose.2016.09.001>.
- Goldwasser, S., & Micali, S. (1982). Probabilistic Encryption. *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC)*, 365–377. <https://doi.org/10.1145/800070.802212>.
- Goldwasser, S., & Micali, S. (1984). Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2), 270–299. [https://doi.org/10.1016/0022-0000\(84\)90070-9](https://doi.org/10.1016/0022-0000(84)90070-9).
- Gong, Y., Chow, K. P., Yiu, S. M., & Ting, H. F. (2023). Analyzing the peeling chain patterns on the Bitcoin blockchain. *Forensic Science International: Digital Investigation*, 46, 301614.
- Guo, D., Dong, J., & Wang, K. (2019). Graph structure and statistical properties of Ethereum transaction relationships. *Information Sciences*, 492, 58–71. <https://doi.org/10.1016/j.ins.2019.04.013>.
- Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1), 29–36.
- Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression*. John Wiley & Sons.
- Hu, S., Zhang, Z., Luo, B., Lu, S., He, B., & Liu, L. (2023). BERT4ETH: A pre-trained transformer for Ethereum fraud detection. *Proceedings of the ACM Web Conference 2023*, 2189–2197.

Kappos, G., Yousaf, H., Stütz, R., Rollet, S., Haslhofer, B., & Meiklejohn, S. (2022). How to Peel a Million: Validating and Expanding Bitcoin Clusters. Proceedings of the 31st USENIX Security Symposium (USENIX Security '22). <https://doi.org/10.48550/arXiv.2205.13882>.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30.

Kılıç, Y., & İnan, A. (2024). Explainable profiling attacks on Ethereum blockchain users based on volumetric and temporal behavior. 2024 32nd Signal Processing and Communications Applications Conference (SIU), 1–4.

Lee, X. T., Khan, A., Sen Gupta, S., Ong, Y. H., & Liu, X. (2020). Measurements, analyses, and insights on the entire Ethereum blockchain network. *Proceedings of The Web Conference 2020*, 155–166. <https://doi.org/10.1145/3366423.3380103>.

Li, J., Baldimtsi, F., Brandao, J. P., Kugler, M., Hulays, R., Showers, E., Ali, Z., & Chang, J. (2021). Measuring illicit activity in DeFi: The case of Ethereum. *Financial Cryptography and Data Security. FC 2021 International Workshops: CoDecFin, DeFi, VOTING, and WTSC*, 197–203.

Lischke, M., & Fabian, B. (2016). Analyzing the Bitcoin Network: The First Four Years. *Future Internet*, 8(1), 7. <https://doi.org/10.3390/fi8010007>.

Lou, Y., Caruana, R., Gehrke, J., & Hooker, G. (2013). Accurate intelligible models with pairwise interactions. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 623–631.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., & Savage, S. (2013). A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. *Proceedings of the 2013 Internet Measurement Conference*, 127–140. <https://doi.org/10.1145/2504730.2504747>.

Nakamoto, S., & Bitcoin, A. (2008). A peer-to-peer electronic cash system. *Bitcoin*.–URL: <https://Bitcoin.Org/Bitcoin.Pdf>, 4(2), 15.

North Korean hackers return to Tornado Cash despite sanctions. (n.d.). <https://www.elliptic.co/blog/north-korean-hackers-return-to-tornado-cash-despite-sanctions>.

Oliveira, P. H. F. S., Rezende, D. M., Bernardino, H. S., Villela, S. M., & Vieira, A. B. (2022). Analysis of account behaviors in Ethereum during an economic impact event.

Pan, B., Stakhanova, N., & Zhu, Z. (2024). EtherShield: Time-interval analysis for detection of malicious behavior on Ethereum. *ACM Transactions on Internet Technology*, 21(1), 1–30.

Payette, J., Schwager, S., & Murphy, J. (2017). Characterizing the Ethereum address space. <https://cs229.stanford.edu/proj2017/final-reports/5244232.pdf>.

Pragasam, T. T. N., Thomas, J. V. J., Vensuslaus, M. A., & Radhakrishnan, S. (2023). CEAT: Categorising Ethereum addresses' transaction behaviour with ensemble machine learning algorithms. *Computation*, 11(8), 156.

Reid, F., & Harrigan, M. (2013). An Analysis of Anonymity in the Bitcoin System. In *Security and Privacy in Social Networks* (pp. 197–223). Springer. https://doi.org/10.1007/978-1-4614-4139-7_10.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “ Why should i trust you?” Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

Ron, D., & Shamir, A. (2013). Quantitative Analysis of the Full Bitcoin Transaction Graph. *Proceedings of the 17th International Conference on Financial Cryptography and Data Security*, 7859, 6–24. https://doi.org/10.1007/978-3-642-39884-1_2.

Said, A., Janjua, M. U., Hassan, S.-U., Muzammal, Z., Saleem, T., Thaipsisutikul, T., Tuarob, S., & Nawaz, R. (2021). Detailed analysis of Ethereum network on transaction behavior, community structure, and link prediction. *PeerJ Computer Science*, 7, e815.

Schwalbe, G., & Finzel, B. (2024). A comprehensive taxonomy for explainable artificial intelligence: A systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, 38(5), 3043–3101.

Shaham, S., Ding, M., Liu, B., Lin, Z., & Li, J. (2019). Machine learning aided anonymization of spatiotemporal trajectory datasets. *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 1–6.

Slack, D., Hilgard, S., Jia, H. L., & Singh, S. (2020). Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. *arXiv Preprint arXiv:2003.00567*. <https://arxiv.org/abs/2003.00567>.

Spagnuolo, M., Maggi, F., & Zanero, S. (2014). BitIodine: Extracting Intelligence from the Bitcoin Network. *Financial Cryptography and Data Security*, 8438, 457–468. https://doi.org/10.1007/978-3-662-45472-7_29.

Sun, H., Ruan, N., & Liu, H. (2019). Ethereum analysis via node clustering. *Lecture Notes in Computer Science, Network and System Security*, 11928, 114–129. https://doi.org/10.1007/978-3-030-36938-5_7.

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks. *arXiv Preprint arXiv:1703.01365*. <https://arxiv.org/abs/1703.01365>.

Tan, R., Tan, Q., Zhang, Q., Zhang, P., Xie, Y., & Li, Z. (2023). Ethereum fraud behavior detection based on graph neural networks. *Computing*, 105(10), 2143–2170.

Tang, Y., Xu, C., Zhang, C., Wu, Y., & Zhu, L. (2021). Analysis of address linkability in Tornado Cash on Ethereum. *China Cyber Security Annual Conference*, 39–50.

The \$477 million FTX hack: A new blockchain trail. (n.d.). <https://www.elliptic.co/blog/the-477-million-ftx-hack-following-the-blockchain-trail>.

The Best Crypto Wallets of March 2023 – Forbes Advisor. (n.d.). <https://www.forbes.com/advisor/investing/cryptocurrency/best-crypto-wallets/>.

The Complete Guide to Web3 Wallets. Thirdweb, Jan. 13, 2023. (n.d.). <https://blog.thirdweb.com/web3-wallet/>.

The rise of the Ethereum economy. (n.d.). <https://www.coinbase.com/tr/learn/market-updates/deep-dive-ethereum-economy>.

Tracing the Trail of the Upbit Hack. (n.d.). <https://www.cylinx.io/blog/tracing-the-trail-of-the-upbit-hack/>.

Victor, F. (2020a). Address clustering heuristics for Ethereum. *Financial Cryptography and Data Security*, 617–633.

Victor, F. (2020b). Address clustering heuristics for Ethereum. *Lecture Notes in Computer Science. Financial Cryptography and Data Security*, 12059, 617–633. https://doi.org/10.1007/978-3-030-51280-4_33.

Wang, Z., Chaliasos, S., Qin, K., Zhou, L., Gao, L., Berrang, P., Livshits, B., & Gervais, A. (2023). On how zero-knowledge proof blockchain mixers improve, and worsen user privacy. *Proceedings of the ACM Web Conference 2023*, 2022–2032.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger (Yellow Paper). Ethereum Project. <https://ethereum.github.io/yellowpaper/paper.pdf>.

Wu, J., Huang, B., Liu, J., Li, Q., & Zheng, Z. (2023). Understanding the dynamic and microscopic traits of typical Ethereum accounts. *Information Processing & Management*, 60(4), 103384.

Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2018). Blockchain Technology Overview (No. NIST IR 8202). National Institute of Standards and Technology (NIST). <https://doi.org/10.6028/NIST.IR.8202>.

Zacharouli, P., Gkoulalas-Divanis, A., & Verykios, V. S. (2007). A k-anonymity model for spatio-temporal data. *2007 IEEE 23rd International Conference on Data Engineering Workshop*, 555–564.

Zanelatto Gavião Mascarenhas, J., Ziviani, A., Wehmuth, K., & Vieira, A. B. (2020). On the transaction dynamics of the Ethereum-based cryptocurrency. *Journal of Complex Networks*, 8(4), cnaa042. <https://doi.org/10.1093/comnet/cnaa042>.

Zhou, J., Hu, C., Chi, J., Wu, J., Shen, M., & Xuan, Q. (2022). Behavior-aware account de-anonymization on Ethereum interaction graph. *IEEE Transactions on Information Forensics and Security*, 17, 3433–3448. <https://doi.org/10.1109/TIFS.2022.3208471>.

Ziegler, M. H., Nowostawski, M., & Katt, B. (2025). A Systematic Literature Review of Information Privacy in Blockchain Systems. *Journal of Cybersecurity and Privacy*, 5(3), 65.



