Beyond the Blend: A Ground-Truth Analysis of Bitcoin Mixer User Patterns

Employing machine learning to unravel the relationship between pre- and post-mixing transactions of Bitcoin mixer users.

Delft University of Technology August 21, 2025

P. H. M. de Haan

Delft University of Technology
Faculty of Technology, Policy & Management

Beyond the Blend: A Ground-Truth Analysis of Bitcoin Mixer User Patterns

Employing machine learning to unravel the relationship between pre- and post-mixing transactions of Bitcoin mixer users.

P.H.M. de Haan

Study programme: Engineering & Policy Analysis

Master's Thesis

First Supervisor: **dr. Rolf van Wegberg**
Second Supervisor: **dr. ir. Floortje d'Hont**
External Advisor FIOD: **Kevin Lubbertsen MSc.**

Delft, 2025

Abstract

Bitcoin mixers break the visible trail between incoming and outgoing transactions. By severing the link between pre-mixing and post-mixing addresses, they provide anonymity that is attractive for laundering illicit funds. For investigators, this creates two obstacles: the vast number of outputs that overwhelm capacity, and the lack of knowledge of internal mixer mechanics that forces reliance on external transaction signals.

This thesis investigates whether transaction patterns before and after mixing can reduce the pool of possible post-mixing addresses linked to a pre-mixing address. The aim is not to prove exact one-to-one links but to narrow the search space so investigators can focus on the most likely outcomes.

We use a unique dataset seized from Bestmixer.io, a centralised mixer dismantled in 2019, containing thousands of verified pre- and post-mixing addresses. The analysis proceeds in two stages. First, we cluster wallets on address-level attributes using HDBSCAN, which yields only coarse profiles. Second, we build transaction graphs capturing how funds move through the mixer, learn graph embeddings with a Graph Autoencoder, and cluster them with k-means. This graph-based view reveals clearer transaction patterns. Pre-mixing, we identify consolidators pooling funds, straightforward depositors from exchanges, aggregator funnels combining smaller inputs, and higher-risk users via unregulated services. Post-mixing, we find splitters dispersing funds, large distributors sending bigger amounts to fewer addresses, and straightforward users with minimal redistribution.

We then test whether pre-mixing patterns can predict post-mixing outcomes. Using tree-based ensemble models (Random Forest and Gradient Boosting) with graph embeddings and the original deposit amount, the best model achieves 48 percent accuracy across five classes, more than double the 20 percent baseline. This demonstrates that transaction graph signals can probabilistically reduce the investigative search space.

The study provides the first ground-truth typology of mixer transaction patterns and shows that probabilistic "de-mixing" is feasible. Rather than pinpointing a single post-mixing address, the method highlights a smaller set of likely candidates, offering law enforcement a way to prioritise leads without access to a mixer's internal mechanics.

Preface

Dear reader,

It feels surreal to be nearing the end of a project that, at times, felt endless. Over the past six months I have had the pleasure of exploring a world that was previously unknown to me. I underestimated both how much I would grow interested in the subject and how challenging it would be to remain stress-free during the writing process. This journey has taught me a great deal about where my interests lie and what I need to find satisfaction in my work. I am proud of the final result and of taking on the challenge of applying methods that were entirely new to me. I am equally grateful for the support I received along the way.

My time at the FIOD has been an incredible learning experience, opening my eyes to a side of investigative work I had never seen before. I owe a big thank you to Rolf for recognising my interest in this field during your class last year and giving me this opportunity. I appreciated having a supervisor with whom I could be my amicable self, sharing laughter alongside serious feedback (and quick shout-out to Joyce for always laughing at my jokes!). My gratitude also goes to Floortje for your willingness to delve into a topic outside your usual domain, your curiosity in understanding the technical details of my work, and your encouragement to make my findings more understandable. Thank you to Kelvin, who I have only seen stressed when home renovations were possibly delayed, but never in any other situation; I greatly enjoyed learning about cybercrime investigations from you. Finally, special thanks to Mennolt for giving me regular feedback and keeping tabs on whether my thesis was actually finished or not, and of course Teun for supporting me through the process.

As I'm writing this I feel slightly melancholic to be nearing the end of 7 years of university, though I also feel relieved that I finally finished this project which showed me the wide range of motivation a person can have. Having said that, enjoy the read!

P.H.M. de Haan

Delft, August 2025

Contents

1 Introduction

This opening chapter follows the path from Bitcoin's limited, quasi-anonymous privacy to the rise of mixers that deepen that privacy and, in doing so, enable large-scale illicit finance. It shows how existing regulation struggles to curb this development, then distils four gaps in research and practice that emerge from non-regulatory methods to tackle illicit mixer use. These gaps shape the research objective and help frame one central research question supported by three sub-questions. The chapter concludes with a brief roadmap of how the remainder of the research addresses each element.

1.1 Bitcoin's Anonymity and the Rise of Mixers

Bitcoin is a decentralised digital currency that enables secure, peer-to-peer transactions without intermediaries like banks (Nakamoto, 2008). Its global adoption has steadily grown (Sergio & Wedemeier, 2025), driven by various uses. While many adopt Bitcoin as an investment vehicle (Mattke et al., 2020), it also serves as a store of value in economically unstable regions (Sergio & Wedemeier, 2025), and potentially as a hedge against inflation (Blau et al., 2021). Bitcoin has a number of unique properties that make it attractive to adopt. Firstly, the currency gives some degree of anonymity, but it is not fully anonymous. We call it *quasi*-anonymous, because transactions are visible publicly but linking them directly to individual identities is non-trivial (Campbell-Verduyn, 2018). Secondly, Bitcoin is decentralised. This means that there is no central authority governing the blockchain it is on. Third, Bitcoin transactions are quick and not bound by national borders.

Bitcoin's quasi-anonymity incentivises its users that desire more anonymity to search for ways to obfuscate their transactions. One way that Bitcoin owners can obfuscate their transaction path is to use Bitcoin mixers. Bitcoin mixing involves aggregating Bitcoin transactions from various sources and redistributing them to obscure their original origins and destinations[1] (Arbabi et al., 2023; Crawford & Guan, 2020). Thus, mixing severs direct links and severely complicates Bitcoin tracing. This could for example be used to protect individuals' assets from theft by obfuscating large coin trails (Silva Ramalho & Igreja Matos, 2021). However, it is mostly used in the criminal circuit.

[1]This process can be thought of as putting multiple coloured marbles into a black box, shaking it, and drawing out random marbles to return to users. It is now unclear which marble originally belonged to whom.

1.2 The Illicit Appeal of Mixers and the Limits of Regulation

Blockchain analysts estimate that in 2023, $22.2 billion was laundered through the use of cryptocurrencies, with most sophisticated criminals using mixers (Chainalysis, 2024). The practice of mixing has led to the rise of centralised mixing services. Users send Bitcoin to these services, which pool and shuffle the funds before returning mixed coins (minus a fee) to specified wallets (Holt et al., 2023; Pakki et al., 2021). While many of these services are scams (van Wegberg et al., 2018), reputable ones can expand rapidly once trust is established. (Crawford & Guan, 2020). In addition to obfuscating illicit funds, there have also been cases where operators of mixing services partnered with darknet markets, promoting the markets to their users and channelling large volumes of funds to and from them (*United States v. Larry Dean Harmon*, 2019; *United States v. Sterlingov*, 2021). Given the often illicit nature of funds passing through mixers, it is crucial to trace them so law enforcement can link suspect wallets to real-world identities and apprehend offenders. We look to regulation as a potential solution.

Anti-money laundering (AML) regulation on cryptocurrency aims to curb this illicit use but is difficult to implement. Regulatory efforts such as the European Union's 5th AML Directive have introduced Know Your Customer (KYC) requirements for custodial service providers[2], a category that includes exchanges and, technically, also centralised mixing services. However, a centralised mixing service meets the EU definition of a custodial service provider solely because it can control users' private keys, not because of its frequent association with illicit activity. Thus, the regulations only cover mixers "by accident", not as a targeted effort to curb illicit use. This omission could create a regulatory gap, leaving room for interpretation and inconsistent enforcement (Silva Ramalho & Igreja Matos, 2021). Compounding this regulatory difficulty is the lack of a consistent global approach: countries classify Bitcoin differently and impose varying regulations (Kethineni & Cao, 2020; Liu & Dong, 2025). Criminals exploit these discrepancies by relocating to jurisdictions with minimal or no enforcement (Rysin & Rysin, 2020).

Given the often illicit nature of funds passing through mixers, the key challenge is to trace these flows so that law enforcement can link suspect wallets to real-world identities. Rather than seeking to eliminate mixers altogether, our focus lies in developing methods that allow investigators to follow illicit funds despite the obfuscation they create. Since

[2]Defined in the directive as entities that safeguard private cryptographic keys (akin to bank PIN numbers) on behalf of customers, to hold, store, and transfer virtual currencies (AMLD5, 2018)

regulatory measures alone have been insufficient to aid with this, we explore technical solutions.

1.3 Research and Societal Gaps in Mixer Analysis

This section identifies one key societal gap and three academic gaps that shape the specific focus of our research. The societal gap stems from the realities of investigating mixer activity, while the academic gaps highlight underexplored areas in the literature. Each is discussed in greater detail in Section 2.2.

1.3.1 Limited Law Enforcement Capacity

Investigating each mixer individually is impractical for law enforcement agencies due to limited resources and the likelihood of investigative dead ends. It is more efficient for investigators to focus on promising leads rather than analysing each transaction in isolation (Goldsmith et al., 2020). Using a method like taint analysis (see e.g. Tironsakkul et al., 2020) can help in reducing the number of potential output addresses, but these methods are plagued by the inherent uncertainty of heuristics. Therefore, developing other ways to link pre- and post-mixing addresses can help law enforcement in focusing their limited resources on the addresses that are most likely to be linked to an illicit input address.

1.3.2 Absence of Direct Input-Output Correlation

Despite the potential benefits of linking pre- and post-mixing addresses (which we also call de-mixing) for law enforcement, research on this topic is limited. Existing methods predominantly focus on classifying internal mixer addresses and transactions (Shojaeinasab et al., 2023; Sun et al., 2022; Wu et al., 2021; Ye et al., 2024). The limited research that is available on de-mixing typically depends on outdated mixers and algorithms (de Balthasar & Hernandez-Castro, 2017; Hong et al., 2018), or heuristic assumptions (Tironsakkul et al., 2020). Thus, a clear research gap exists for methods capable of correlating pre- and post-mixing transactions. Addressing this would enable tracing illicit activities even when mixer configurations remain unknown or rapidly evolve.

1.3.3 Scarcity of Ground-Truth Data

A possible explanation for the limited research on input-output correlation is the general scarcity of reliable ground-truth data on mixer transactions, as most studies mentioned in the previous paragraph use self-labelled data. Since mixers intentionally obscure transaction trails, reliably labelling mixer-related addresses or matching pre-mixing addresses to corresponding post-mixing addresses is very difficult. To mitigate this difficulty, researchers commonly rely on heuristics (Tironsakkul et al., 2020; Wu et al., 2021), which have limited accuracy. Alternative approaches involve creating self-labelled datasets (Shojaeinasab et al., 2023; Sun et al., 2022; Wu et al., 2021; Ye et al., 2024). Even studies employing externally verified data, such as Du et al. (2024), highlight the limited dataset size as a critical constraint. Therefore, lack of ground-truth often leads to less reliable and smaller datasets, presenting another research gap.

1.3.4 Limited General Analysis of User Transaction Patterns

Given the difficulty of directly linking pre- and post-mixing addresses, analysing user transaction patterns offers a promising alternative. Current analyses of mixer transactions often remain narrowly focused, either examining singular hacking events without focusing on mixers (D. Y. Huang et al., 2018) or specific groups (Goldsmith et al., 2020), thereby limiting their general applicability. Conversely, studies that do consider the broader context, do this on an abstraction level too high for a focus on mixers (Rosenquist et al., 2024; Vlahavas et al., 2024), limiting their usefulness when studying mixers. Hence, comprehensive analyses examining general transaction patterns across multiple users remain scarce, yet are essential for better understanding how mixers are used.

In sum, practice and literature point to four intertwined shortcomings that any effective, non-regulatory response must overcome: (1) investigators lack the capacity to follow the overwhelming volume of mixer outputs; (2) reliable, generalisable techniques for correlating deposits with withdrawals remain elusive; (3) progress is hamstrung by the scarcity of verified, ground-truth data on mixer flows; and (4) broad, cross-user analyses of transaction behaviour (an avenue that could bypass some of these constraints) are still rare. Addressing these gaps is essential if technical approaches are to complement regulation and give law enforcement agencies practical methods against illicit mixer use.

1.4 Research Objective and Questions

Reflecting on the use of mixers, regulation, and the societal and research gaps we identified, we summarise our findings in a problem statement and use that to formulate a research objective. After this, we posit research questions to guide our research.

Problem Statement Investigators must trace large volumes of mixer outputs with limited resources while having no access to mixers' internal mechanics. Regulatory remedies (e.g., KYC obligations for custodial providers) are uneven across jurisdictions and do not resolve the technical tracing challenge. Existing academic work on mixer output prediction requires knowledge on internal mixer mechanics, relies on heuristics that do not generalise, or requires ground truth at scales that are unavailable. Consequently, the more practical direction of research is not to prove one-to-one links, but to prioritise likely post-mixing candidates from observable, external transaction data, so that investigators can focus effort where it is most productive. This means that, rather than identifying the single correct address, we aim to select a subset of post-mixing addresses that is much smaller than the total set of possible outputs and quantify the likelihood that the correct address is within it[3]. This approach reduces the search space investigators must examine while still capturing most of the likely candidates. We pursue this by relying on externally observable transaction patterns rather than knowledge of a mixer's internal mechanics, supporting law enforcement in a way that has the potential to generalise across mixers.

Research Objective This framing leads to the following objective:

Research Objective

To identify externally observable pre- and post-mixing transaction patterns and use these insights to develop a method that highlights the most probable post-mixing patterns based on pre-mixing patterns. This probabilistic approach should reduce the candidate set of post-mixing addresses, enabling investigators to focus their efforts more effectively.

Reaching this objective addresses the four gaps identified in Section 1.3. We reduce investigative workload by narrowing the candidate pool of post-mixing addresses (Gap 1). We replace difficult one-to-one input–output linking with a pattern-based probabilistic

[3]For example, if there are 100 possible post-mixing addresses, our method might narrow this down to 20 and estimate that there is a 60% probability that the correct address is among them.

approach (Gap 2). We use the ground-truth data of the now-defunct centralised Bitcoin mixer called Bestmixer, further explained in Section 3.2 (Gap 3), and we focus on patterns that can generalise across users rather than isolated cases (Gap 4).

Research Questions Given our research objective, we have established the following main research question:

Main Research Question

To what extent do pre- and post-mixing Bitcoin transaction networks display patterns that can be leveraged to narrow the pool of plausible post-mixing addresses linked to a given pre-mixing address?

This study is guided by the following sub-questions:

Sub-question 1: How effectively can pre- and post-mixing wallets be clustered based on aggregated address-level attributes?

Sub-question 2: How effectively can pre- and post-mixing addresses be clustered using features drawn from their transaction network graphs?

Sub-question 3: How reliably do clusters formed from pre-mixing transaction patterns predict the corresponding clusters in post-mixing transactions?

The first sub-question focuses on the most basic level: can we meaningfully group (i.e. cluster) pre- and post-mixing wallets using only features of the individual addresses that deposit funds into the mixer and receive funds from the mixer (such as activity duration or balance)? This provides a baseline understanding of whether address-level attributes alone carry useful patterns. How we define a wallet is further explained in Section 2.1.

The second sub-question adds more context by including the structure of transactions surrounding a pre- and post-mixing address. By analysing patterns in the broader transaction graph (such as how connected an address is or how fast funds move through the network) we assess whether this information improves clustering. This analysis therefore includes multiple wallets and transactions as compared to the singular wallets analysed in sub-question 1.

The third sub-question evaluates the predictive value of the clusters made for sub-

question 2. Specifically, it asks whether knowing what pattern a user exhibits before mixing helps us predict the kind of cluster they will fall into after mixing. If so, this would allow us to reduce the pool of potential post-mixing addresses by focusing only on those that match the pattern of the pre-mixing cluster.

1.5 Thesis Structure

This thesis is structured as follows. Chapter 2 provides the necessary background on Bitcoin transactions and address clustering, followed by a review of existing literature on mixer usage, illicit transaction patterns, detection techniques, and de-mixing approaches. Chapter 3 outlines the methodology, including the conceptual model, data sources, and analytical procedures used to address the three sub-questions. Chapter 4 presents the results of the exploratory data analysis and of each sub-question in sequence. Chapter 5 discusses the findings, their scientific and practical implications, and outlines key limitations and directions for future research. Finally, Chapter 6 concludes the thesis by summarising the main insights and contributions.

2 Bitcoin Fundamentals and Research Landscape

Chapter 2 establishes the essential context for this thesis by explaining Bitcoin transaction mechanics, introducing address clustering, Bitcoin services, and reviewing existing literature on mixing services. This provides the technical foundation for understanding both the workings of Bitcoin and the research design, and clarifies how the gaps identified in Chapter 1 were derived.

2.1 Understanding Bitcoin: Concepts and Mechanics

This section explains the core mechanics of the Bitcoin system, emphasising how users engage with it through wallets and addresses, how value is transferred using the UTXO model, and providing a brief overview of Bitcoin services. Understanding these concepts is essential for this thesis, as we rely on wallet and address clustering to identify individual users and analyse their transaction patterns. Differentiating between individual users and Bitcoin services is also crucial, which is why this distinction is discussed.

2.1.1 Bitcoin Addresses, Wallets and the UTXO Model

Bitcoin transactions transfer value between users and are verified through digital signatures using private keys (comparable to a bank PIN). Each recipient is identified by a Bitcoin address, which is derived from their public key (similar to a bank account number). A Bitcoin wallet is software or hardware that manages a user's private keys and generates new addresses for receiving and sending funds. It allows users to initiate transactions and track their balances. A wallet therefore contains multiple addresses belonging to one user; this is a very important concept to understand. When we say that we "cluster" addresses, we often mean that we group addresses as belonging to a single user, i.e. that those addresses are most likely in the same wallet. In order to prevent confusion between the concept of clustering in machine learning and Bitcoin address clustering, we will refer to address clusters as wallets. When mentioning pre- and post-mixing addresses, keep in mind that those addresses also belong to a wallet.

Bitcoin's transaction system is based on the Unspent Transaction Outputs (UTXO) model, where funds are represented as individual outputs from previous transactions rather than stored in account-like balances. You can think of a UTXO as a cheque with a certain amount of Bitcoin on it; if you receive three cheques, you can't just merge

them together into one big one.

If we look at the example in Figure 1 below, we see on the left that Alice has an address A with 3 UTXO's of 0.3, 0.4, and 0.6 BTC respectively. She received these UTXO's as transactions, and the address stores these transactions exactly like they were received. They are therefore not aggregated like in a normal bank account. If Alice wants to send 0.5 Bitcoin (BTC) to Bob, she will have to combine UTXO's because she does not own a UTXO with 0.5 BTC. The address therefore chooses UTXO's 1 and 2 as inputs to the transaction, totalling 0.7 BTC. The transaction takes the two inputs and uses them to create two new UTXO's. Because Alice wants to transfer 0.5 BTC to Bob, one UTXO of 0.5 BTC is created. The amount that is left (0.2 BTC) is transferred to a new address B belonging to Alice, minus a "transaction fee" that she has to pay. This new address is called a *change address*, and belongs to Alice's wallet.

In sum, Alice spends UTXO 1 and 2 to create UTXO 4 and 5. UTXO 4 is transferred to Bob, while UTXO 5 is transferred back to Alice's wallet but into a different address. This model is essential to understanding the "Change Address Heuristic" discussed in the next section.

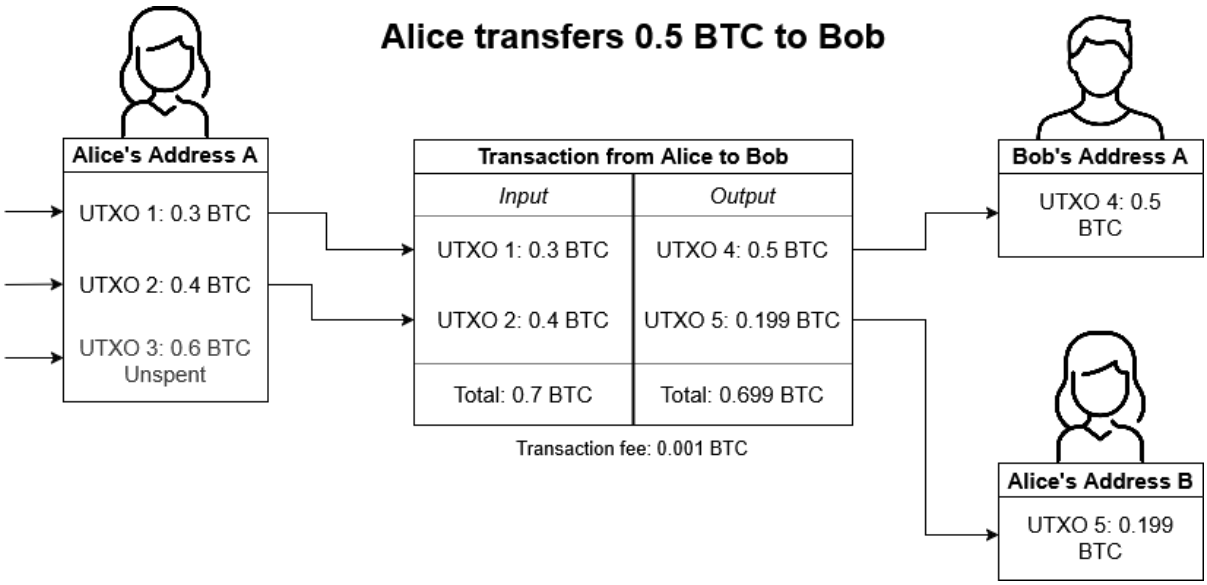

Figure 1: Diagram of the UTXO model

2.1.2 Address Clustering

Although each address is publicly visible on the blockchain, it does not reveal its owner or whether it is linked to other addresses. This is where address clustering becomes essential: it allows analysts to infer which addresses likely belong to the same wallet

or, in some cases, to the same user controlling multiple wallets. Address clustering uses transaction-level heuristics to group addresses based on patterns that suggest common control (Meiklejohn et al., 2013). Two commonly used heuristics are:

1. **Common Input Heuristic**: If multiple addresses are used as inputs in a single transaction, they are likely controlled by the same user, since spending from them requires access to all corresponding private keys.

2. **Change Address Heuristic**: When a transaction returns unspent funds to the sender (see the UTXO model), this is typically done via a newly generated "change address." Identifying these change addresses helps link them to the sender's wallet.

Once addresses are clustered, transaction analysis can shift from the address level to a higher abstraction (such as wallets or users) enabling clearer insights into fund flows, transactional patterns, and potential links to illicit activity. However, these clustering methods are not foolproof. Users may adopt privacy-enhancing strategies like avoiding address reuse or deliberately obfuscating transaction patterns to avoid detection. As a result, clustering remains a process with uncertainty, making validation against ground-truth data critical.

2.1.3 Bitcoin Services

Whereas an individual Bitcoin user generally controls a handful of addresses that are activated only when the owner wishes to send or receive funds, Bitcoin services (see Appendix A.1 for an extensive list) operate as always-on infrastructure for thousands or even millions of customers. Their wallets aggregate deposits, combine outputs, and forward transactions at a scale that dwarfs typical personal activity. Because they intermediate other people's money, services usually fall under AML and KYC rules, making them attractive choke points for regulation, but also prime targets for circumvention by illicit actors who seek out no-KYC or lightly regulated providers. From an analytical perspective, this distinction is crucial: a single service wallet can represent the behaviour of thousands of end-users, so clustering heuristics must first identify service nodes before meaningful patterns of individual user behaviour can emerge.

2.2 Understanding Mixer Use: Insights from Prior Studies

In Section 1.3 we highlighted one societal and three academic gaps that steer the direction of this thesis. In this section, we explain how we found those three academic gaps by reviewing prior research on Bitcoin mixers across four themes. First, we zoom into mixer users by exploring user security behaviour and perceptions of mixing services. Second, we examine research pertaining to user transaction patterns associated with illicit activities. Third, we look at internal mixer mechanics by discussing approaches to distinguish mixer from non-mixer addresses. Fourth, we assess work related to linking addresses used to deposit into a mixer (pre-mixing addresses) with those used to withdraw from it (post-mixing addresses). Finally, we aggregate the findings of this literature review to distil the three academic gaps that steer the direction of this thesis.

2.2.1 Users of Mixers

Crawford and Guan (2020) investigated the features, public perception, and success rates of 69 Bitcoin mixing services by analysing public discussions on forums and the operational characteristics of the mixers. The study revealed that although Bitcoin mixer users prioritised privacy, mixing services faced significant challenges in establishing user trust due to widespread fraud, with 28% of mixers identified as scams. Users frequently discussed mixer reliability on forums, and trust was often built through community endorsements. The authors noted that users often sought features like random delays, randomised fees, and non-logging policies to ensure anonymity. However, the limited trust in mixers and the prevalence of scams deterred widespread adoption despite strong privacy demands.

Miedema et al. (2023) analysed ground-truth transaction data from the centralised mixing service BestMixer to understand user behaviours. The study explored how users attempted to mitigate risks such as attribution or scams. The findings revealed limited adoption of security measures like IP obfuscation and the use of multiple output addresses. Despite this, users trusted the service with substantial funds. The study also highlighted user reliance on mixers despite significant information asymmetry and potential regulatory risks. This is an interesting result when compared to Crawford and Guan (2020), and might be explained by the fact that BestMixer was perceived as a more established and trusted service.

The limited research on mixer users highlights the central role of trust, yet also suggests

that users often invest minimal effort into securing their anonymity. The findings indicate that, despite the emphasis on privacy, many users do not adopt comprehensive strategies to protect it, relying instead on the perceived trustworthiness of the mixer itself. Notably, existing studies do not consider users' full transaction histories when assessing behaviour. This oversight may obscure important patterns, such as careless routing of Bitcoin into and out of mixers, which could compromise anonymity.

2.2.2 Transaction Behaviour Involving Illicit Activities

Vlahavas et al. (2024) took a broad view of Bitcoin transactions using an unsupervised machine learning method, which resulted in multiple distinct user clusters. They showed that while the largest cluster consisted primarily of low-input, low-output transactions akin to regular users, other clusters were more indicative of high-volume services, mining pools, or potentially illicit operations. One smaller subset exhibited comparatively higher fees and frequent reliance on CoinJoin[4]-like transactions, underscoring patterns more aligned with anonymisation or mixing attempts. Although their study did not focus on mixers specifically, it highlighted how user activity in Bitcoin was far from uniform: some participants followed ordinary payment habits, while others employed more sophisticated methods, including features consistent with laundering techniques.

D. Y. Huang et al. (2018) performed a two-year measurement study of ransomware-related transactions. They followed ransomware payments as they moved across the blockchain. The authors noted that mixers played an important role in helping ransomware operators obscure the destination of ransom payments. Specifically, once victims' funds reached the ransomware's wallet cluster, the authors observed that attackers often moved those Bitcoins to a mixer before cashing out at an exchange, thereby complicating efforts to link the funds to real-world identities. However, the authors did not describe specific patterns associated with mixing, i.e. *how* the attackers moved funds to mixers.

Rosenquist et al. (2024) analysed money flows to and from Bitcoin addresses associated with a variety of illicit activities and provided a characterisation of transaction patterns. They found that a small elite of addresses collected the majority of criminal funds and that mixers acted as the main hub connecting different crime types. Mixing services were connected to far more counterparties and more funds flowed through fewer wallets than

[4]CoinJoin is a transaction method that combines multiple senders' inputs and outputs into one transaction, obscuring which inputs paid which outputs. It is a decentralised mixing protocol.

any other illicit-use category. This pattern implied that either a small number of operators controlled most mixers or that large numbers of offenders channelled their proceeds through the same few services. When the authors traced outgoing flows from reported abuse addresses, they found that mixers functioned as "bottlenecks" for laundering criminal funds. However, the patterns towards mixers that the authors observed are relatively basic and do not describe in detail how funds flowed towards the mixers.

Goldsmith et al. (2020) analysed six real-world hack transaction networks to examine how two distinct hacking groups laundered Bitcoin after compromising exchanges, including their use of mixing services. The temporal dynamics of the subnetworks were most informative: how quickly hackers offloaded their coins and whether they used mixers consistently or only at the final stage. One group ('alpha') gradually dispersed stolen funds, often routing them through intermediary addresses and potentially into mixers over time, while the other ('beta') held large sums and then sent them in bursts to mixers or exchanges within a short window. This emphasis on the timing of mixer interactions, not just their presence, allowed investigators to distinguish between laundering strategies and clarified when criminals engaged mixing services and how they accelerated the cash-out process.

Collectively, these studies illustrate that mixers play a key role in obscuring illicit funds and have spurred initial attempts to characterise how criminals leverage these services. Nonetheless, existing work either adopts a high-level perspective on mixer usage or examines only narrow case studies of specific hacking or ransomware groups. So while research on transaction patterns surrounding mixers does exist, explorations of transaction patterns of mixer users specifically remain sparse.

2.2.3 Mixer Detection

Shojaeinasab et al. (2023) analysed transaction patterns of three different mixers after creating their own dataset by using said mixers. They uncovered global patterns that allowed them to construct an algorithm that could label transactions as belonging to a mixer. They explicitly mentioned the difficulties of acquiring a large dataset, so they settled for a smaller dataset. Because of this, the validation of their algorithm was difficult.

Similarly, Wu et al. (2021) proposed both a general abstraction model for mixing services, and a heuristic algorithm to identify mixing transactions. They used the algorithm

to examine real-world mixing services to reveal their mechanisms and role in criminal activities, shedding light on one part of their three-part abstraction model. They were able to construct a larger dataset of mixer input and output transactions by performing experiments and using public APIs. They were therefore able to validate their model, which had a high accuracy of correctly labelling transactions.

Sun et al. (2022) developed a classifier model based on deep learning that could classify transactions as either normal or mixing transactions. As input for the model, they use the entire transaction tree linked to a certain transaction, which is an interesting approach since it doesn't just consider local features but places a transaction in a larger context. They used a larger dataset constructed from a number of heuristics to determine involvement with a mixing service, which is a limitation as they do not have access to ground-truth or human-labelled data.

Ye et al. (2024) advanced mixing detection by first clustering mixer addresses into distinct roles based on their transaction patterns, such as those primarily collecting, redistributing, or aggregating funds. These clusters then informed an ensemble of classification machine-learning models which evaluate both transaction-specific features (e.g. amounts and frequencies) and topological properties (like centrality). By mapping each address to one of the identified roles and then applying this dual-layer classification, they were able to flag mixing addresses with great accuracy. The authors acknowledged that a limitation was that they used a relatively small dataset, which could have biased their model.

The research above shows that methods to cluster and classify addresses belonging to mixers are plentiful. However, a limitation for all of these studies is that there is a lack of verified and complete ground-truth data to rigorously test the models. To validate their models, they often use datasets that were labelled based on heuristics, which are less accurate and robust than ground-truth data.

2.2.4 De-mixing Research

We define de-mixing as the process of analysing and deconstructing a mixer's operations to reliably link input transactions with their corresponding output transactions. Achieving this undermines a mixer's ability to conceal fund flows, thereby negating its primary purpose.

de Balthasar and Hernandez-Castro (2017) examined how mixers like Darklaunder, He-

lix, and Alphabay actually functioned, showing that their reliance on central addresses, incomplete obfuscation algorithms, and narrow traffic patterns opened them up to practical de-mixing. By making a series of test transactions and tracing the flows, the authors revealed how easy it was for an external observer to correlate inputs and outputs, especially given certain typical mixer behaviours and features. Their findings emphasised how simple missteps such as storing users' transaction histories ultimately allowed both criminals and law enforcement to 'unwind' the mixing process. However, mixers have become more sophisticated since then, so the ease with which the authors were able to de-mix the mixers has diminished.

Similarly, Hong et al. (2018) proposed a de-mixing algorithm for Helix with a very high accuracy rate. However, the same limitations applied to this study as they did to de Balthasar and Hernandez-Castro (2017), as Helix was an old mixing service and technology has advanced since then.

Tironsakkul et al. (2020) proposed a tracking method called address taint analysis, which shifted the focus from traditional transaction-level taint analysis to an approach that tracked Bitcoin at the address level. By applying this technique to different mixing transaction samples, the study demonstrated that the technique held promise for reconnecting deposited Bitcoins with their withdrawn counterparts. However, the authors acknowledged limitations such as the relatively high number of potential outputs, challenges in performing the technique effectively without internal knowledge of the mixer, and potential inaccuracies or false positives due to improper use of filtering criteria.

Du et al. (2024) set out to unmask users of Tornado Cash, the best-known mixer on Ethereum (the second-largest cryptocurrency after Bitcoin) that criminals frequently used to launder stolen crypto. The authors first converted every Tornado deposit and withdrawal into a graph with user addresses as nodes and interactions with the mixer as edges. They also attached basic facts to each node: how many times it interacted with the mixer, when, how much fee it paid, etc. The authors then trained a Graph Neural Network link-prediction model to link pre-mixing to post-mixing nodes. To train and test the model, they obtained 103 examples of known pre- and post-mixing address pairs, along with larger artificial sets. Their model correctly linked hidden pairs far better than earlier methods (up to 64% higher accuracy) and ran efficiently even on millions of transactions. The work showed that simple timing and usage patterns around Tornado

Cash provided promising leads for de-mixing, though the authors conceded that their pool of "ground-truth" examples was still small and that future mixers might adapt.

In sum, there has been limited (recent) research on de-mixing mixers, though de-mixing research of Ethereum-based mixer Tornado Cash comes very close. While traditional taint analysis could hold promise for Bitcoin-based mixers, other methods rely on old and error-prone mixing algorithms to de-mix. Besides that, ever-evolving mixing algorithms make it difficult to establish a reliable, generalisable approach to de-mixing that remains effective against newer, more sophisticated mixing techniques. Finally, even for successful de-mixing models, obtaining ground-truth data for training and testing remains difficult.

2.2.5 Academic Gaps

As discussed in Section 1.3, this chapter identifies three key academic gaps based on the analysis of the four thematic areas.

Absence of Direct Input-Output Correlation The first gap arises from the discussion on mixer detection and de-mixing in the final paragraphs of this chapter. While a substantial body of research focuses on identifying addresses associated with mixing services, far less attention has been paid to the direct correlation between pre-mixing and post-mixing addresses. Existing studies on this topic are either limited in scope, outdated, or focus on blockchains other than Bitcoin.

Scarcity of Ground-Truth Data The second gap concerns the consistent reliance on internal or proprietary knowledge of mixers in studies on detection and de-mixing. Many of these studies explicitly cite the lack of ground-truth data as a major limitation, highlighting its scarcity. Incorporating ground-truth data could significantly enhance the validity and robustness of such analyses.

Limited General Analysis of User Transaction Patterns The third gap emerges from the broader literature on mixer usage and illicit transaction behaviour. Although extensive research exists on transaction patterns related to illicit activity, fewer studies specifically examine the transaction behaviour of mixer users. Moreover, there is a general lack of research on how users interact with mixing services, both in terms of individual behaviour and broader usage patterns.

3 Research Design

This chapter presents the methodological approach used to investigate transaction patterns of mixer users. The chapter begins by outlining the conceptual framework, followed by a detailed description of the data sources, preprocessing steps, and analytical methods used to cluster and compare pre- and post-mixing data.

3.1 Conceptual Framework of Clustering Mixer Users

The conceptual framework provides a structured lens to systematically analyse the pre- and post-mixing patterns of users. It breaks down the mixing process into distinct phases and clarifies how these phases may relate to each other. This is essential to this research, as it ensures that our analysis is grounded in a clear, theory-informed understanding of mixer use.

Hypotheses and Conceptual Model Wu et al. (2021) abstract mixing as a three-phase sequence: (1) receiving inputs, (2) mixing, and (3) producing outputs. While Wu et al. (2021) focus on Phase 2, this research concentrates on Phases 1 and 3 and broadens their scope.

We posit two hypotheses that are coupled to our research questions:

H1 *Contextual patterns:* users exhibit stable pre- and post-mixing transaction patterns beyond the singular deposit/withdrawal.

H2 *Phase linkage:* characteristics observed in Phase 1 are informative about those in Phase 3 (i.e., what Phase 1 looks like says something about what Phase 3 looks like).

In Wu et al. (2021), Phase 1 refers narrowly to transferring funds *into* the mixer and Phase 3 to the mixer *sending* funds to new addresses. We hypothesise that meaningful transactional patterns exist in the wider context around these actions. Put differently, how funds are wired towards a mixer and how users manage them afterwards may reveal characteristic patterns. This is embedded within broader behavioural routines rather than being reducible to a single deposit and withdrawal and shapes **H1**.

To reflect this broader scope, we redefine Phase 1 as "Input preparation and transfer" (the build-up and the deposit), and Phase 3 as "Output handling and redistribution" (the

withdrawal and subsequent distribution). Each phase is analysed in a 60-day window before the deposit and after the withdrawal as described in Section 3.5.1. With these newly defined phases we shape **H2**. The plausibility of **H2** is consistent with evidence that many criminal cash-out procedures follow relatively simple, repeatable routines (e.g., Nazzari (2023) on Conti's laundering patterns).

Figure 3.1 visualises our conceptual model. For each mixing event, Phases 1 and 3 comprise transaction networks[5] that respectively precede and follow the core mixing operation in Phase 2. Phase 2 is taken as a 72-hour interval (the maximum retention time reported for Bestmixer). Bestmixer is our principal data source; see Section 3.2.

The transaction networks of Phases 1 and 3 are grouped into clusters using three feature families: address-level, transaction-level, and graph-level characteristics. Address and transaction attributes describe individual wallets and their histories; graph attributes capture the structural properties of the local network. Table 1 summarises the features; details follow in Sections 3.2.3 and 3.5.2.

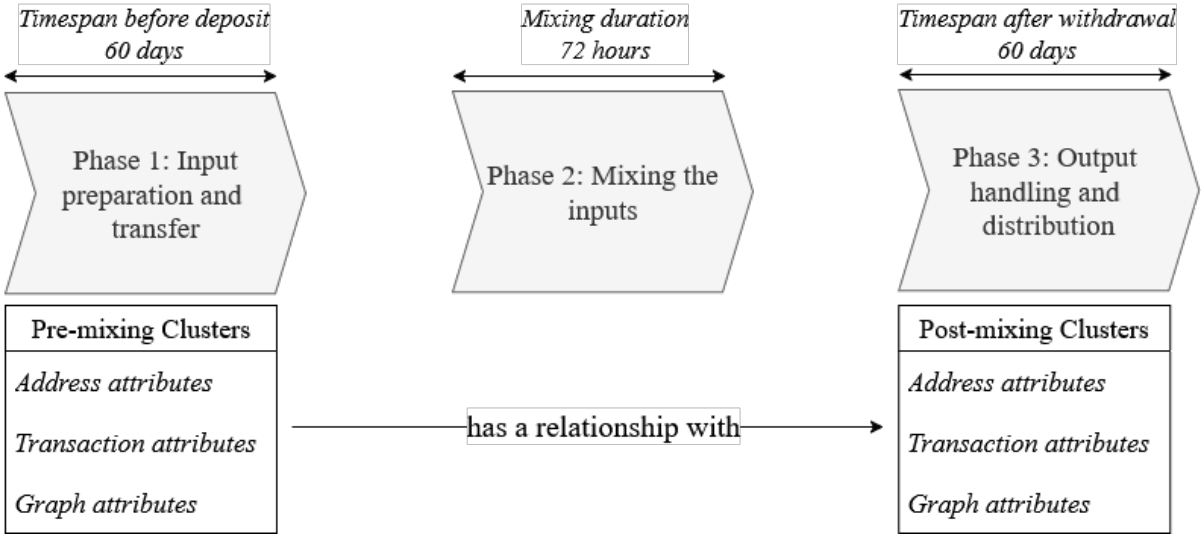

Figure 2: Conceptual Model

Defining Clustering *Clustering* is an important concept for answering the research questions because it lets us group mixer-users that exhibit similar patterns. Fung (2001) defines clustering as "the grouping together of similar data items into clusters". Witten et al. (2017) extends this idea: "These [obtained] clusters should reflect some mechanism at work in the domain from which instances or data points are drawn, a mechanism that

[5] A transaction network in Bitcoin is a directed graph: nodes represent wallets (or addresses, depending on the aggregation level) and edges represent transfers, capturing the flow of funds.

Table 1: Overview of variables by attribute type

Address	Graph	Transaction
balance	n_nodes	value
totalSent	n_edges	timeDelta
totalReceived	density	
totalTxIn	diameter	
totalTxOut	average_degree	
totalAddresses	max_degree	
average_received	average_betweenness	
average_spent	average_closeness	
activity_duration	degree_assortativity	
transaction_frequency	num_output_addresses	
spent_txo_ratio		

causes some instances to bear a stronger resemblance to one another than they do to the remaining instances".

Following these definitions, we are looking for Bitcoin wallets (for sub-question 1) and transaction graphs (sub-question 2) that bear similar attributes and can therefore be clustered. The goal is to see whether a user belonging to a certain pre-mixing cluster has a higher likelihood of belonging to a certain post-mixing cluster. If so, this would unveil a mechanism where the way users route funds into a mixer says something about the way they redistribute their mixed funds. We therefore use clusters as a concrete representation of the transactional patterns we aim to analyse.

3.2 Data Sources and Pre-processing

This study relies on two primary sources:

1. **Bestmixer Data** Bestmixer was a centralised mixing service that was taken down in 2019. The dataset is provided by law enforcement, and gives us a large amount of ground-truth data to work with. The dataset contains

   - an *orders* file (customer requests sent to Bestmixer),
   - a list of Bestmixer *wallet addresses*, and
   - the *transaction history* of those wallets.

2. **Chainalysis Reactor** Reactor is a blockchain investigation platform developed by Chainalysis that provides address and wallet data. We distinguish between *non-*

exposure and *exposure* data. A list of non-exposure data used is shown later in Table 4. Exposure data shows *direct* and *indirect* exposure of addresses to certain categories. A full list of these categories can be found in Appendix A.1. *Direct* exposure indicates that an address directly transacts with another address with a category label. *Indirect* exposure identifies the services or entities that ultimately transact with an address; even when the funds first pass through one or more ordinary addresses on the way.

The goal of the pre-processing in this chapter is to construct a list of valid Bestmixer orders that link pre-mixing addresses to corresponding post-mixing addresses, each converted into a wallet and enriched with relevant attributes from the Chainalysis API. The following subsections explain this process in three steps: (1) describing the raw data, (2) filtering the Bestmixer orders to retain only valid transactions, and (3) aggregating the pre-mixing and post-mixing addresses into one wallet using address clustering as described in Section 2.1.2. The resulting dataset contains one pre-mixing wallet and one post-mixing wallet per order, both belonging to the same user.

3.2.1 Bestmixer Data

The Bestmixer dataset provides ground-truth information that enables us to analyse real-world mixer usage. It consists of three files.

Orders A court-authorised wiretap captured network traffic for order placement to `Bestmixer.io` during four intervals:

2018-07-18 – 2018-08-13	clear-web only
2018-11-12 – 2019-01-06	clear-web only
2019-02-07 – 2019-03-06	clear-web only
2019-03-21 – 2019-05-22	clear-web only
2019-04-15 – 2019-05-22	Tor only

Each captured request yielded the variables listed in Table 2. "Clear-web only" refers to orders placed with the clear-web website that people were able to access through the "regular" internet. For the last time period mentioned, network traffic was observed for orders placed from the TOR browser. The TOR browser can be used to access the dark web anonymously, which is an unregulated form of the internet.

Table 2: Order-file fields

Field	Description
`deposit_address`	Unique address generated by Bestmixer for the customer's deposit
`datetime`	Timestamp in Central European Time when the order page was served
`ip_address`	Source IP of the HTTP request
`status`	Last status displayed to the customer
`deposit_amount`	Amount received at `deposit_address`
`coin`	Cryptocurrency used (BTC, BCH, or LTC)
`language`	UI language chosen
`application`	Browser cookie identifying the session
`uid`	Customer identifier (supplied or auto-generated)
`user_agent`	Full HTTP `User-Agent` string
`order_id`	Mixer-side numeric order identifier
`out_address[`n`]`	One or more payout addresses specified by the customer

Wallet Addresses A list of roughly 200,000 Bitcoin, Bitcoin Cash, and Litecoin addresses belonging to Bestmixer.

Transaction History The Bestmixer transaction log comprises roughly 250,000 rows (Table 3). It shows the internal transactions of Bestmixer. We use this log together with the orders file to confirm which orders were actually received by Bestmixer.

Table 3: Wallet-log fields

Field	Description
`confirmed`	Boolean flag; `true` if the transaction settled
`date`	Block time of the transaction
`type`	Type of transaction: `sent to`, `received with`, or `payment to yourself`
`address`	Bestmixer wallet involved in the transaction
`amount`	Value transferred in the transaction
`id`	Mixer-side transaction identifier

3.2.2 Obtaining Valid Orders

Because not all orders captured by the wiretap were actually paid for and executed, we don't know which orders are actually valid. To determine this we use both the orders file and the transaction log. We first check the transaction log for validity to be used later on in the process. We then use the orders file to determine which orders are valid. Detailed pre-processing steps including how many data points were removed in each step are found in Appendix A.2.

Transaction Log We start with 241,713 unverified data points. After removing unconfirmed datapoints and internal or irrelevant payments, we are left with 241,270 valid data points detailing internal Bestmixer transactions.

Orders We start with 36,083 orders, and only keep Bitcoin orders, remove duplicates and cancelled orders, and cross-verify orders with the previously cleaned transaction log. This leaves us with 26,092 valid orders. However, in Section 3.2.3 we find that we can't obtain information on a number of post-mixing addresses associated with an order and therefore have to remove 186 orders. Thus, our final dataset contains 25,680 unique orders. Note that this is not a dataset on unique users, as some orders are placed by the same user at different times.

3.2.3 Creating Pre- and Post-Mixing Wallets

We aggregate deposit and withdrawal addresses into *wallets* (address clusters; see Section 2.1.2) to analyse patterns at the user level.

Pre-mixing Wallets For each validated order we start from the `deposit_address`, which is unique per order. Using the Chainalysis Reactor API, we retrieve all funding transactions into this address and aggregate the funding addresses into a single pre-mixing wallet using the Common-Input Heuristic. Because each deposit address is order-unique, these funding addresses belong to the ordering user. We then enrich this wallet with non-exposure and exposure attributes used later in the analyses. We extract exposure data directly from the Chainalysis API. For non-exposure data, we use both Chainalysis API output and derive several variables ourselves. Table 4 lists all non-exposure features we use.

Post-mixing Wallets When placing an order, users specified the address(es) to receive their mixed funds. Because Chainalysis does not cluster these post-mixing addresses (lacking access to the Bestmixer dataset thereby not knowing that these addresses likely belong to the same user), we aggregate individual post-mixing addresses ourselves to form custom post-mixing wallets.

However, it is not guaranteed these addresses all belong to the same person or entity as the original order. There are five possibilities:

Feature	Description
balance	Total balance of the wallet (# BTC)
totalSent	Total sent amount (# BTC)
totalReceived	Total received amount (# BTC)
totalTxIn	Number of incoming transactions
totalTxOut	Number of outgoing transactions
totalAddresses	Number of addresses in the wallet
average_received	Average received per transaction (# BTC / tx)
average_spent	Average sent per transaction (# BTC / tx)
activity_duration	Active duration (days)
transaction_frequency	Transactions per day
spent_txo_ratio	Output-to-input transaction ratio

Table 4: Overview of features of wallets.

1. All post-mixing addresses belong to the original ordering entity.

2. They belong to a different entity.

3. They belong to a mix of the original entity and another non-service entity.

4. They belong to a service (e.g., NGO, darknet market, etc.).

5. They are a combination of service and non-service entities.

For our research, differentiating between the first three cases is unnecessary. Our focus is on identifying pre- and post-mixing patterns and assessing whether pre-mixing clusters can predict post-mixing clusters, regardless of who owns the post-mixing addresses.

The fourth case (payments to a service) does not affect our analysis but does prevent the creation of complete transaction graphs. This is because services have too high a transaction volume to model into a usable graph. The fifth case is problematic, as service wallets (due to high transaction volume) would distort our variables. Excluding only service wallets would misrepresent the data, so we choose to remove these mixed cases entirely. This affects 186 data points (0.7% of our data).

3.3 Exploratory Data Analysis

Prior to clustering, we conduct an Exploratory Data Analysis (EDA) to better understand the structure and distribution of the variables included in the wallet datasets. The primary aims of the EDA are: (1) to assess the presence of outliers and data sparsity, (2) to evaluate differences between service and non-service wallets, and (3) to inform appropriate feature

engineering[6] choices for subsequent analyses. We perform the same analyses for both pre- and post-mixing wallets.

We begin by distinguishing service wallets (e.g., exchanges, darknet markets) from non-service wallets, as service wallets tend to significantly differ from non-service wallets. For wallets without a Chainalysis-provided category label, we introduce a separate `missing` category to retain these data points for further analysis. We then visualise the category distribution on a log scale and examine the empirical cumulative distribution functions (ECDFs) for both non-exposure and exposure attributes. The non-exposure attributes are normalised prior to ECDF plotting, due to their widely differing scales. In contrast, exposure attributes are plotted without normalisation, as they are already expressed as percentages. The results of the EDA are found in Section 4.1.

3.4 Clustering Wallet Attributes

The first sub-question is the following:

SQ 1

How effectively can pre- and post-mixing wallets be clustered based on aggregated address-level attributes?

After pre-processing, we retain 25,680 pre-mixing wallets and 18,487 post-mixing wallets (we only look at post-mixing wallets linked to an order without a service pre-mixing wallet). To address this sub-question we use the unsupervised algorithm HDBSCAN, which groups wallets with similar attributes while labelling unassigned cases as noise. Clustering of wallets on the blockchain is a well-established approach as demonstrated by for example Vlahavas et al. (2024), Chordia and Shinde (2024), or Kehinde et al. (2024).

3.4.1 Clustering Method

Algorithm: HDBSCAN Although several clustering methods are available, we choose the HDBSCAN algorithm to answer this sub-question (McInnes et al., 2017). This is a clustering method that identifies clusters of data points based on density. Put simply, HDBSCAN looks for places in the data where many points are packed closely together

[6]Feature engineering refers to the process of creating, transforming, or selecting variables to improve the performance of machine learning models. It helps ensure that the input data is in a useful form for analysis.

and defines these as clusters, while points that are scattered too far from any cluster are labelled as noise and left unassigned.

There are several reasons why HDBSCAN is well suited for the dataset we are using:

1. *Classifying noise*: the HDBSCAN algorithm does not include noisy points into clusters, but classifies them as 'noise'. This is useful for our data since we know that it is quite sparse and has a lot of noise because of this.

2. *Size of clusters*: HDBSCAN allows for the creation of clusters of differing sizes. This is well suited to our dataset since we know that the dataset is quite skewed, so the formation of clusters of equal size is unlikely.

3. *Intuitive hyperparameters*: the inputs for HDBSCAN are the minimum number of points a cluster should include (`min_cluster_size`) and the minimal amount of neighbours a core point needs (`min_samples`). These hyperparameters are generally intuitive to use because they simply allow us to directly influence the size and the specificity of the clusters (Hunt & Reffert, 2021).

3.4.2 Feature Selection and Data Preparation

Before clustering data, it must first be carefully prepared. This section explains how relevant features were selected, how the data was grouped for analysis, and which other preparation steps were applied to make the variables suitable for clustering.

Variables and Clustering Scope We use variables that are externally observable (see Table 4), deliberately excluding those unique to Bestmixer. This ensures generalisability to other mixers for which only open-source data is available.

Because service entities (e.g. exchanges, gambling sites) dominate many attribute values, their high values can overshadow that of individual users. To counter this, we run the clustering algorithm both on the full dataset and on a subset restricted to Chainalysis' `missing` category, which typically captures smaller, less visible wallets likely to belong to individuals.

Scaling The dataset also has a large number of features with very different scales and highly skewed distributions, which is further highlighted in Section 4.1. If untreated, this can distort clustering: large-valued features dominate distance calculations, and heavy

tails cause outliers to overwhelm clusters. We therefore apply the Yeo–Johnson transformation (Yeo & Johnson, 2000) using `scikit-learn`'s `PowerTransformer`. This method reshapes skewed distributions to be more symmetric, bringing all features closer to a comparable, bell-shaped form. This makes it easier for the clustering algorithm to compare features fairly.

Dimensionality Finally, we split the variables into two groups: *non-exposure* (11 variables) and *exposure* (63 variables). Analysing all 74 variables at once would lead to the "curse of dimensionality" (Aggarwal et al., 2001): as the number of dimensions grows, data points become increasingly spread out, making distances less meaningful. This means that the clustering algorithm can group the data points less effectively. By separating the feature sets, we reduce this effect and allow for more stable clustering.

3.4.3 Parameters and Model Evaluation

Parameter selection To select the optimal HDBSCAN parameters, we perform a grid search[7] over both `min_cluster_size` and `min_samples`, varying each from 2 to 1002 in steps of 50. We choose a grid search because it is a straightforward method to test a large number of parameters. We evaluate the parameters on clustering performance metrics (described in the next section). Since the metric scores level off before reaching 1002, we do not expand the range beyond that point. Full grid search results are covered in Section 4.2.

For the full dataset, we use both metric scores and visual inspection of the results to choose the most suitable parameter combinations. In particular, we look at whether the resulting clusters align with the known Chainalysis service categories. In some cases, a slightly lower-scoring model is preferred if it produces clearer and more meaningful groupings; for instance, if exchanges are placed in their own cluster rather than being merged with unrelated services. This balance between statistical fit and interpretability allows us to select clustering results that are both technically sound and analytically useful.

We repeat this process separately for the subset of `missing` wallets. Since all of these wallets fall under the same category and lack service labels, we cannot assess cluster

[7] A grid search is a systematic way of finding the best parameters for a model by trying out all possible combinations within a predefined set, and comparing their performance.

quality based on known types. Instead, we focus on whether the clustering output shows enough variation to reveal different patterns. For example, we avoid results where almost all wallets are grouped into just two or three large clusters with one dominating cluster, as this would offer little insight into the diversity of wallet attributes.

Model evaluation With unsupervised machine learning such as clustering, there are no true labels given in the data. As a result, you cannot directly measure how "correct" the model's classifications are. Instead, clustering metrics must be used. However, as Brun et al. (2007) point out, it can be challenging to find metrics that effectively evaluate unsupervised models. We look towards the literature to find three metrics that are used often: the Silhouette Score, Davies-Bouldin Index, and Calisnki-Harabasz Index.

The Silhouette Score has been used before in clustering users in Bitcoin transaction networks (Vlahavas et al., 2024). The other metrics have been used alongside the Silhouette index before to evaluate clustering algorithms (Ahmed Al-Kerboly & Al-Kerboly, 2024; Ashari et al., 2023; Pecuchova & Drlik, 2022). The metrics describe the following:

1. **Silhouette Score**: Measures how well each data point fits within its assigned cluster compared to other clusters. A high score (closer to 1) means that points are close to others in their own cluster and far from points in other clusters, indicating that clusters are well-separated (Rousseeuw, 1987).

2. **Davies-Bouldin Index**: Compares how spread-out each cluster is and how far apart clusters are from one another. A lower score indicates that clusters are compact and clearly separated (Davies & Bouldin, 1979).

3. **Calinski-Harabasz Index**: Measures how distinct and dense the clusters are by comparing the variation within clusters to the variation between clusters. A higher score indicates better-defined, well-separated clusters (Caliński & Harabasz, 1974).

Finally, we compute a ranking of parameter-combinations which result in the most desirable combination of all three evaluation metrics. This means that we consider all three metrics evenly: we sum the ranking of all three metrics and then pick the highest ranking to determine the best parameters.

3.5 Creating and Clustering Transaction Graphs

The second sub-question is the following:

> **SQ 2**
>
> How effectively can pre- and post-mixing addresses be clustered using features drawn from their transaction graphs?

To answer this question, we convert each raw Bestmixer order into a three-step pipeline:

1. Build a directed Bitcoin wallet graph that captures all transactions preceding the deposit or following withdrawal;
2. Compress that graph into a compact numerical representation (an *embedding*) using a Graph Autoencoder (GAE);
3. Group the resulting embeddings into clusters with k-means clustering algorithm.

This methodology is well established in prior work on Bitcoin and other networks. Graph topological features are frequently used for clustering (Gaihre et al., 2019), and graph embeddings provide richer structural information. Several studies apply nearly the same pipeline: Shah et al. (2021) use a similar workflow with a different embedding model; Nan and Tao (2018) combine a GAE with k-means to detect mixing services; and Emane et al. (2024) cluster Bitcoin embeddings from a graph neural network. Together, these examples show that our approach is firmly grounded in existing research.

This section explains how and why we execute the three methodological steps of this sub-question.

3.5.1 Graph Construction

Our objective is to link pre-mixing patterns to post-mixing patterns at the *user* level. Wallet attributes alone (balances, counts, frequencies) miss *how* funds are routed. By constructing transaction graphs we capture connectivity and flow context around each mixer interaction. This structural signal is exactly what our hypotheses require: if Phase 1 behaviour says something about Phase 3, that information will be reflected in the graph patterns rather than in isolated attributes. This is why we construct transaction graphs.

We want to cluster these graphs to uncover patterns to analyse. Clustering algorithms

need the data to be in a certain format to work, but graphs are irregular and not numerical so clustering algorithms don't work in it. A GAE turns each graph into a compact embedding (list of numbers) that preserves the structure of the graph and with that the information that it carries. How exactly that works is explained later in this section.

Data Preparation We exclude every order whose pre-mixing or post-mixing wallet is tagged by Chainalysis as an exchange, darknet market, or other high-volume service. Such wallets send and receive hundreds of transactions per hour, so any graph rooted in them would explode in size, thereby introducing far too much noise for meaningful analysis. After this filter the working set shrinks from 25,680 wallets to 18,487 pre-mixing graphs and 13,237 post-mixing graphs.

The filter runs in two passes. Starting with 25,680 orders, we first drop any whose pre-mixing wallet is a service, leaving 18,487 usable pre-mixing graphs. From that subset we then drop orders whose post-mixing wallet is a service, ending with 13,237 post-mixing graphs. This two-stage pruning gives us a clean set where each order has both a complete pre-mixing graph and a complete post-mixing graph ready for comparison.

Creating the Graph For each order we build a *directed multigraph* in `networkx`, a Python library for graph analysis. A directed multigraph is a network in which every edge has a direction and any two nodes can share multiple parallel edges. We model the graph as follows:

- *Nodes* represent Bitcoin wallets and carry the attributes mentioned previously in Table 4.
- *Edges* are arrows between the nodes that represent transaction. They carry two attributes: the amount of bitcoin transferred (`value_per_tx`) and the time before the Bestmixer deposit or withdrawal from Bestmixer (`timeDelta`).

We model wallets (addresses grouped by Chainalysis address clustering heuristics) as nodes rather than individual transactions. We do this because it keeps one node per object we ultimately want to cluster and lets us attach address-level features such as active duration, balance, and transaction frequency. If every transaction were a node, the same wallet would explode into hundreds of nodes, introducing more noise.

Graph Constraints Even after removing obvious service wallets, some wallets still act as busy hubs. For example, there might be multiple service wallets present deeper in the transaction graph. We need to find a balance between having enough information to find patterns and preventing adding too much noise. To stop any single graph from blowing up we apply three pragmatic limits.

Table 5: Graph construction constraints and rationale

Constraint	**Value**	**Why**
Transaction cap	189 incoming transactions (95^{th} percentile)	Removes extreme heavy-tailed wallets while retaining the vast majority (95%) of cases within the observation window.
Time window	60 days (pre-deposit and post-withdrawal)	Limits including a wallet's entire transaciton history and thereby graph blow-up. Trade-off between graph size and relevant information (see Appendix A.3).
Depth limit	5 steps (hops) from starting wallet	Balances reach with computational tractability and aligns with prior work (Rosenquist et al., 2024).

3.5.2 Graph Autoencoder

Clustering methods cannot directly analyse graph data. Therefore, we use *graph embeddings*. These embeddings simplify complex graphs into compact vectors (lists of numbers) that preserve essential information about the graph's structure and its attributes.

Graph Autoencoder Operation We generate graph embeddings using a Graph Autoencoder (GAE), specifically the implementation from the `Pytorch Geometric` library (Kipf & Welling, 2016).

First, the GAE transforms the graph data into a mathematical representation called an *adjacency matrix*. In this matrix, each row and column represent a node (a Bitcoin wallet), and each entry indicates whether two nodes are connected by a transaction (an edge). This provides a clear overview of the graph's structure.

Next, the GAE incorporates additional *features*, such as transaction amounts and timing information. It does so by assigning these features to the nodes (for example, the balance of an address) and edges (the transaction amounts or timing). Together, the adjacency matrix and these features form the complete input to the GAE.

The GAE then learns embeddings through two stages:

1. The *encoder* compresses the information from the adjacency matrix and node features into compact embeddings for each node. Initially, the embeddings are randomly assigned, but the GAE continuously adjusts them during training to better reflect the structure and attributes of the graph.

2. The *decoder* takes these embeddings and attempts to reconstruct the original adjacency matrix as accurately as possible. After reconstruction, the decoder compares its output with the true adjacency matrix. The difference between the reconstructed and true matrices is called the *loss*. The GAE then adjusts the embeddings to reduce this loss, thereby improving the quality of the embeddings.

The goal is simple: if the embeddings allow the decoder to accurately rebuild the original graph, they must contain valuable information about the graph's structure and node characteristics. We stop training the GAE when its performance on a separate, unseen part of the data no longer improves, meaning it has stabilised. This ensures the embeddings capture general patterns rather than memorising specific training examples, which is called overfitting.

Feature Selection and Graph Metrics To create useful embeddings, each node and transaction in our graph is associated with certain features. Nodes have the same characteristics as described in sub-question 1 for the same reasons mentioned there. Transactions (edges) include two additional features: the transferred amount (`value_per_tx`) and the time difference relative to when the deposit to or withdrawal from Bestmixer occurred (`timeDelta`). Additionally, we calculate several metrics that describe the overall structure of each graph, as summarised in Table 6.

Table 6: Graph-level metrics derived from transaction networks

Metric	**Description**
`n_nodes`	Number of wallets (nodes) in the graph
`n_edges`	Number of directed transactions (edges)
`density`	Ratio of actual to possible edges; network interconnectedness
`diameter`	Longest shortest path between any two nodes
`mean_degree`	Average number of connections per node (incoming + outgoing)
`max_degree`	Maximum number of connections for a single node
`avg_betweenness`	Average number of shortest paths passing through each node
`avg_closeness`	Average closeness of each node to all other nodes
`degree_assortativity`	Indicates if nodes tend to connect to nodes with similar degrees (positive number), or nodes with different degrees (negative number)

Parameter Selection Three main choices affect the quality of the embeddings produced by the GAE:

- *Number of embeddings*: how many numbers each embedding should have. More numbers can capture more details, but too many can cause the model to overfit (memorise training data instead of generalise).
- *Number of hidden dimensions*: the complexity of patterns the GAE can learn internally. More layers help the model capture detailed patterns, but too many layers also risk overfitting.
- *Learning rate*: how fast the model adjusts itself during training. A high learning rate can speed up training but might cause unstable results; a low learning rate leads to more stable but slower learning.

We determine the best settings through a grid search, testing different combinations as shown in Table 7. The quality of each combination is judged by how accurately the GAE can reconstruct graphs. This accuracy is measured by the *total loss*, which represents the difference between the original graph and its reconstruction. A lower loss indicates that the embeddings effectively capture the graph's essential information.

Table 7: Hyperparameter Configuration

Parameter	**Values**
Number of embeddings	2, 3, 4, 5, 6, 7, 8
Number of hidden dimensions	32, 64, 128
Learning rate	0.01, 0.005, 0.001

3.5.3 Clustering Method

Embeddings generated by GAEs typically do not form well-separated clusters. Instead, they often occupy a single dense region (Kipf & Welling, 2016; Pan et al., 2018). Therefore, the HDBSCAN clustering algorithm we used in sub-question 1 is not suited for this analysis. HDBSCAN is a density-based clustering method that is designed to find clusters of varying densities and shapes, and it tends to group uniformly dense data into a single large cluster; exactly what we want to avoid.

Algorithm: k-means The k-means clustering method (MacQueen, 1967) is better suited for this type of distribution, as it assumes that clusters are spherical and evenly

sized. Since our embedded data is compact, continuous, and lacks clear density-based separations, k-means can more effectively partition it into a predefined number of clusters.

Parameter Selection To choose the number of clusters (`k`) used in k-means clustering, we employ the *elbow method*. The elbow method involves plotting the total variance within clusters, called the within-cluster sum of squares (WCSS), against different values of `k`. As `k` increases, WCSS naturally decreases. However, at some point, the improvement from adding more clusters diminishes significantly. This point, where the curve sharply changes direction (an "elbow"), indicates a suitable balance between complexity and explanatory power. We select the number of clusters corresponding to this elbow point as the optimal `k`.

Model Evaluation The embeddings created by the GAE tend to be evenly spread out in space, without forming tight, separate groups (Kipf & Welling, 2016). This happens because the GAE is designed to capture general patterns in the graph, not to create clearly separated clusters. As a result, standard clustering evaluation scores like the Silhouette Score do not work well here, because they rely on there being clear gaps between clusters.

To check whether the resulting clusters are meaningful, we use a statistical test called ANOVA (Analysis of Variance). This test tells us whether the variables we are analysing are significantly different between the clusters. If they are, that suggests that the clusters reflect real differences. Finally, we inspect the clusters manually to see if we can describe clear profiles for each one based on their features.

3.6 Post-Mixing Cluster Prediction

The final sub-question is:

SQ 3

How reliably do clusters formed from pre-mixing transaction patterns predict the corresponding clusters in post-mixing transactions?

To answer this, we use the cluster labels previously assigned to each pre- and post-mixing transaction graphs (from sub-question 2). By linking each pre-mixing graph to its corresponding post-mixing cluster, we can directly evaluate how closely the pre- and post-mixing clusters align.

The use of graph embeddings as input to prediction models is well established in the literature. For instance, Z. Huang et al. (2023) generated embeddings with a graph neural network to predict the service category of a given address. Likewise, Lo et al. (2023) employed graph embeddings in a classification model to distinguish between licit and illicit wallets and Koronaios and Koloniari (2025) used embeddings as features in a classifier to label addresses as malicious or non-malicious. Overall, numerous studies demonstrate that graph embeddings can serve as effective features for supervised machine learning models.

Only some post-mixing addresses (those labelled as `missing`) are clustered in sub-question 2. Post-mixing addresses belonging to known services (e.g., exchanges or darknet markets) were previously excluded from clustering. To account for these addresses, we group all known service addresses into a separate third cluster. Additionally, some post-mixing graphs contained just a single node (due to no further transactions within our 60-day observation window), which could not be embedded by the GAE. We assign these 1,522 single-node addresses to a distinct fourth cluster. This ensures each pre-mixing graph is linked to exactly one post-mixing cluster.

In this section, we first statistically evaluate the strength of association between pre- and post-mixing clusters. Then, we build predictive models to classify post-mixing clusters based on pre-mixing information. Finally, we discuss the evaluation methods and metrics used to assess the performance of these predictive models.

Statistical Testing for Cluster Association We first perform a statistical test, known as the *Chi-squared test of independence*, to evaluate if there's a meaningful relationship between pre- and post-mixing cluster assignments (Pearson, 1900). To measure the strength of any observed relationship, we use *Cramér's V*, a metric ranging from 0 (no association) to 1 (perfect association), which helps interpret the degree of association intuitively (Cramér, 1946).

Prediction Supervised machine learning models used for prediction require input variables that are used as *predictors*. Since we already combined structural and attribute information into embeddings (from sub-question 2), we directly use these embeddings as predictors because they directly represent our transaction graphs. Additionally, we include the original `deposit_amount`, as we expect it to strongly predict post-mixing

activity (since post-mixing transaction values are closely tied to pre-mixing amounts). Lastly, we include pre-mixing cluster labels to test whether the original cluster assignments themselves provide predictive value.

Tree-based ensemble methods such as Random Forests and Gradient Boosting are well-suited for our classification task. These models handle mixed-type data effectively and do not require a lot of feature engineering. The models use a collection of simple decision-making rules to classify data points into different groups. They decide on these rules by trial-and-error. Random Forests offer strong baseline performance, robustness to overfitting, and interpretable metrics for feature importance (i.e. which features drive the prediction) (Breiman, 2001). We also use a Gradient Boosting prediction model. This model iteratively corrects the errors of previous trees and performs well on structured data (Chen & Guestrin, 2016).

Parameter Selection To choose the right parameters for the models, we ran the grid search seen in Table 8. Given the large number of hyperparameters and the wide range of possible values, an exhaustive grid search would be computationally infeasible. To overcome this, we use a method called *Randomized Search Cross-Validation* (`RandomizedSearchCV`). This method randomly samples a manageable number of possible settings and then evaluates each combination systematically.

Specifically, we perform 500 random trials, each time testing a unique combination of settings. To make sure the results are reliable, we use a process called *5-fold cross-validation*. This means we first split our training data into five smaller parts (folds). We then train our model five times; each time using four parts for training and the remaining fifth part for validation (testing how well it performs). By averaging the results across these five trials, we gain a reliable measure of how well each setting combination works. This helps avoid overfitting (James et al., 2013). The best-performing combination is then chosen to make our final predictions.

Model Evaluation To assess how well the prediction models perform, we divide the dataset into two parts: one for training the model (80%) and one for testing how well it performs on unseen data (20%). This ratio is often used for supervised machine learning models (Satrya et al., 2022).

We use several standard metrics to assess prediction quality:

Table 8: Hyperparameter Configuration for Random Forest and Gradient Boosting

Parameter	**Value Range / Options**
Random Forest (RF)	
Number of trees	100–200 (uniform integers)
Maximum tree depth	5–20 (uniform integers)
Minimum samples to split a node	2–4 (uniform integers)
Minimum samples at a leaf node	1–2 (uniform integers)
Maximum number of features	`sqrt`, `log2`
Bootstrap sampling	`True`, `False`
Gradient Boosting (GB)	
Number of trees	100–200 (uniform integers)
Learning rate	0.005–0.2 (log-uniform)
Maximum tree depth	3–15 (uniform integers)
Minimum samples to split a node	2–4 (uniform integers)
Minimum samples at a leaf node	1–2 (uniform integers)
Subsample ratio	0.6–1.0 (uniform)

- *Accuracy* measures the proportion of correct predictions. Although easy to interpret, accuracy can be misleading if clusters differ substantially in size (the model could achieve high accuracy by simply predicting the largest cluster) (Powers, 2011).

- *Precision* measures the fraction of correct predictions within each predicted cluster. Precision matters when the cost of incorrectly classifying transactions into the wrong post-mixing clusters is high.

- *Recall* measures how completely the model identifies all true members of a cluster. Recall is essential if missing true cluster members (false negatives) carries significant consequences.

- The *F1-score* balances precision and recall, giving a unified metric that equally weighs both measures.

Since our clusters differ significantly in size, we report *macro-averaged* precision, recall, and F1-score metrics, ensuring each cluster equally influences the overall evaluation, irrespective of their relative frequency in the dataset.

Finally, we assess *feature importance*, identifying which features (embeddings, deposit amounts, or pre-mixing cluster labels) contribute most to the prediction. Understanding feature importance helps clarify what aspects of pre-mixing patterns are most useful for predicting subsequent post-mixing patterns (Breiman, 2001).

4 Results

This chapter presents the results of our analyses. It begins with the Exploratory Data Analysis (EDA), followed by the results for sub-questions 1 and 2, which are organised into separate sections focusing on pre-mixing and post-mixing wallets, respectively. This helps us determine pre- and post-mixing transaction patterns and quantifies them. Sub-question 3 brings these two perspectives together and considers patterns across both wallet sets, resulting in a method that can predict (with a degree of probability) which post-mixing pattern will be exhibited given a pre-mixing pattern.

4.1 Exploratory Data Analysis

This section describes the distributional characteristics of the pre- and post-mixing wallets based on category labels, non-exposure variables, and exposure data. Please refer to Section 3.2 for an explanation on exposure and non-exposure data. All accompanying figures, including category distributions and Empirical Cumulative Distribution Function (ECDF) plots, are provided in Appendix B.1. This chapter is important because investigating the skew and sparsity of the data is necessary to inform our methodological decisions.

Chainalysis Categories The category distribution for both pre- and post-mixing wallets follows a long-tailed pattern. In the pre-mixing set, wallets labelled as `missing` and various types of exchanges are the most common. In the post-mixing set, a similar shape is observed, though with a higher share of wallets linked to `darknet market` services.

Non-Exposure Variables Most wallets have very low values for non-exposure features such as balance, transaction volume, and activity duration, with a small number of high-value outliers. This sparsity is even stronger among post-mixing wallets, where values are more concentrated near zero. These variables are therefore highly skewed.

Exposure Variables Exposure data shows similar skew. Direct exposure is concentrated in the `exchange` and `Unknown` categories, so most wallets directly received funds from an exchange or a non-service wallet. Indirect exposure is dominated by various exchange types. In the post-mixing set, there is a slight increase in exposure to illicit services.

A comparison of descriptive patterns is shown in Table 9. We use this to conclude that the initial data we are working with is very skewed and sparse.

Table 9: Summary of descriptive patterns in pre- and post-mixing wallets

Characteristic	**Pre-Mixing Wallets**	**Post-Mixing Wallets**
Category distribution	Dominated by `missing` and exchange-related labels; long-tail shape	Similar distribution; higher prevalence of `darknet market`
Non-exposure variables	Skewed distributions with most values near zero; some high outliers	More skewed; values further concentrated at the lower end
Direct exposure	High share of `exchange` and `Unknown` categories	Similar pattern; slightly more exposure to illicit services
Indirect exposure	Mostly exchange-related; few wallets exposed to other services	Comparable to pre-mixing; more frequent `darknet market` exposure

4.2 User Profiles from Address Attribute Clusters

The first sub-question reads as follows:

SQ 1

How effectively can pre- and post-mixing wallets be clustered based on aggregated address-level attributes?

By answering this question we gain a baseline understanding of the relationship between pre- and post-mixing wallets and how users interact with the mixer. It forms a starting point for the rest of this research and gives us more insight into the data.

We previously made a distinction between exposure and non-exposure features. We conclude that exposure data is not well suited for clustering pre- and post-mixing bitcoin wallets, so we exclude it from this chapter. The full results of this analysis can be found in Appendix B.2. Additionally, the full results of the grid search we ran to determine the best HDBSCAN algorithm parameters can be found in Appendix B.3.

4.2.1 Pre-Mixing Wallets

We apply the HDBSCAN algorithm to the pre-mixing dataset with both `min_cluster_size` and `min_samples` set to 602. Although this is not the top-ranked configuration (ranking score: 95.17 vs. 73.33), it is the highest-ranked setting that yields six clusters instead of

three. This is preferred because it allows us to differentiate more between clusters and identify more detailed profiles. The resulting clusters are shown in Table 10, with Cluster –1 representing outliers. It shows the total count of wallets in a certain cluster, and which percentage of those wallets fall into a certain Chainalysis category (see Appendix A.1 for the full list). Evaluation scores are as follows:

- Silhouette Score: 0.610
- Davies-Bouldin Score: 0.361
- Calinski-Harabasz Score: 7,324.745

These scores fall in a moderate range for unsupervised clustering (e.g. Silhouette $>$ 0.6 is often seen as acceptable (Rousseeuw, 1987)).

Table 10: Category distribution per pre-mixing cluster (in %)

Cluster	`missing`	`exchange`	`p2p exchange`	`sanct. entity`	`other`	**Total Count**
Cluster -1	22	60	0	0	18	3,488
Cluster 0	0	0	100	0	0	1,280
Cluster 1	100	0	0	0	0	17,726
Cluster 2	0	0	0	78	12	700
Cluster 3	0	100	0	0	0	662
Cluster 4	0	100	0	0	0	641
Cluster 5	0	100	0	0	0	1,183

The first thing we can notice is that Cluster 1 is by far the largest and contains only addresses of the `missing` category, so most of these addresses are very similar and have no discernible attributes compared to the entire dataset. Besides that, we see that clusters are mostly differentiated by category, so the clusters do not give us a lot more extra information besides the fact that wallets belonging to different categories differ from each other. To gain more insight into individual users, we separately analyse the `missing` wallets.

For the subset of `missing` addresses, we set `min_cluster_size` to 652 and `min_samples` to 402. These parameters rank second-best. The top-ranked parameters produce 696 more outliers, so we favour fewer outliers over marginally higher cluster quality. The resulting scores are:

- Silhouette Score: 0.260
- Davies-Bouldin Score: 1.300
- Calinski-Harabasz Score: 4,797.243

These scores are substantially lower, indicating weak clustering due to limited differences in variable values. The resulting distribution is shown in Table 11. It shows the total number of wallets per cluster and the percentage of wallets that fall into a cluster.

Table 11: Pre-mixing clusters in the `missing` category

Cluster	**Count**	**Percentage**
Cluster -1 (outliers)	10,935	59.2%
Cluster 0	813	4.4%
Cluster 1	2,483	13.4%
Cluster 2	4,256	23.0%

Although the algorithm labels more than half of wallets as outliers, we compare the remaining clusters in Figure 3. This figure shows a strip-plot, indicating the normalised values of the non-exposure variables. This means that the lowest value of each variable corresponds to 0 and the highest to 1, and all other data points are divided onto that new scale relatively. Each dot on the plot represents one wallet with a certain value. This allows us to compare the three clusters, even though their variables are on different scales. While the three clusters do not look entirely well-defined (resulting in low evaluation metric scores), they do show some clear differences. Using these differences we can construct three profiles:

- **Cluster 0** — *"Dormant Whales"*: Wallets with a long activity duration but infrequent activity. They have handled large total transaction volumes over time, yet individual transactions are relatively small on average.
- **Cluster 1** — *"Casual Users"*: Contains wallets with few total transactions, short active duration, and low averages. It seems like the wallet was created for the specific purpose of depositing funds into the mixer, which could explain why this cluster has a specific `spent_txo_ratio`.
- **Cluster 2** — *"High-Volume Bursts"*: The final cluster distinguishes itself by containing high-volume addresses with a short duration of activity. The addresses in this cluster make few, but large deposits to the mixer.

Creating these profiles is valuable because, even with only weak separation between groups, we already see distinct patterns emerging before mixing. This suggests that a broader analysis could help anticipate post-mixing behaviour.

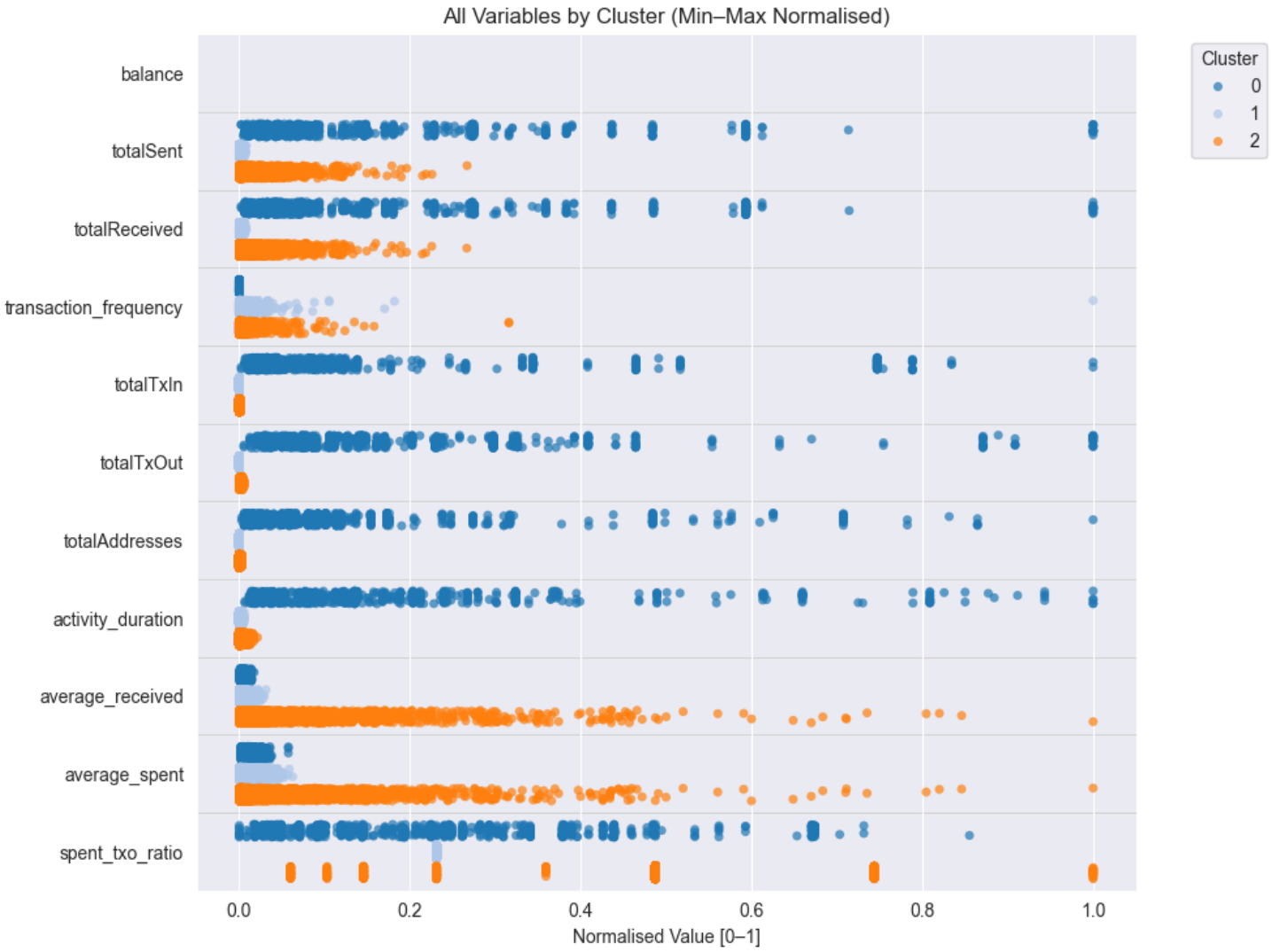

Figure 3: Strip plot of pre-mixing features by cluster

4.2.2 Post-Mixing Addresses

To analyse the post-mixing addresses, we apply the HDBSCAN clustering algorithm using `min_cluster_size` and `min_samples` set to 402. This parameter combination yields the highest evaluation metrics:

- Silhouette Score: 0.360
- Davies-Bouldin Score: 0.883
- Calinski-Harabasz Score: 8,367.212

The evaluation metrics are worse than for the pre-mixing clusters, indicating that post-mixing wallets are more difficult to cluster (i.e. have very similar values). The resulting clusters are shown in Table 12, with Cluster –1 denoting outliers.

The algorithm identifies relatively few outliers (10.4%), suggesting that most post-

Table 12: Category distribution per post-mixing cluster (in %)

Cluster	missing	exchange	p2p exchange	darknet market	other	**Total Count**
Cluster -1	64	1	4	6	25	1,961
Cluster 0	0	100	0	0	0	1,906
Cluster 1	100	0	0	0	0	2,705
Cluster 2	100	0	0	0	0	10,592
Cluster 3	0	0	0	67	33	863
Cluster 4	0	0	99	0	1	765

mixing addresses share similar attributes. Cluster 2 contains the majority of `missing` addresses, again indicating that this category lacks distinguishing characteristics. In contrast, Cluster 3 combines multiple categories, suggesting that some post-mixing addresses from different known services have similar attribute values. However, the clustering algorithm again splits wallets mostly based on their categories.

Therefore, we also analyse the `missing` addresses separately. The best-scoring parameter combination in this subset uses `min_cluster_size` = 2 and `min_samples` = 802. The resulting distribution is shown in Table 13.

Table 13: Post-mixing clusters in the `missing` category

Cluster	**Count**	**Percentage**
Cluster -1 (outliers)	807	5.5%
Cluster 0	538	3.7%
Cluster 1	2,699	18.5%
Cluster 2	10,509	72.2%

Figure 4 visualises the clusters using a strip plot of the normalised feature values. Compared to the pre-mixing clustering, the post-mixing clusters appear significantly noisier. As a result, it becomes difficult to construct clear behavioural profiles. Attempting to label these clusters risks oversimplifying or misrepresenting the underlying variation in the data. We therefore opt not to do that, and refrain from creating user profiles. This is important because it raises the question of whether distinct profiles will also emerge when analysing full transaction graphs rather than individual wallets.

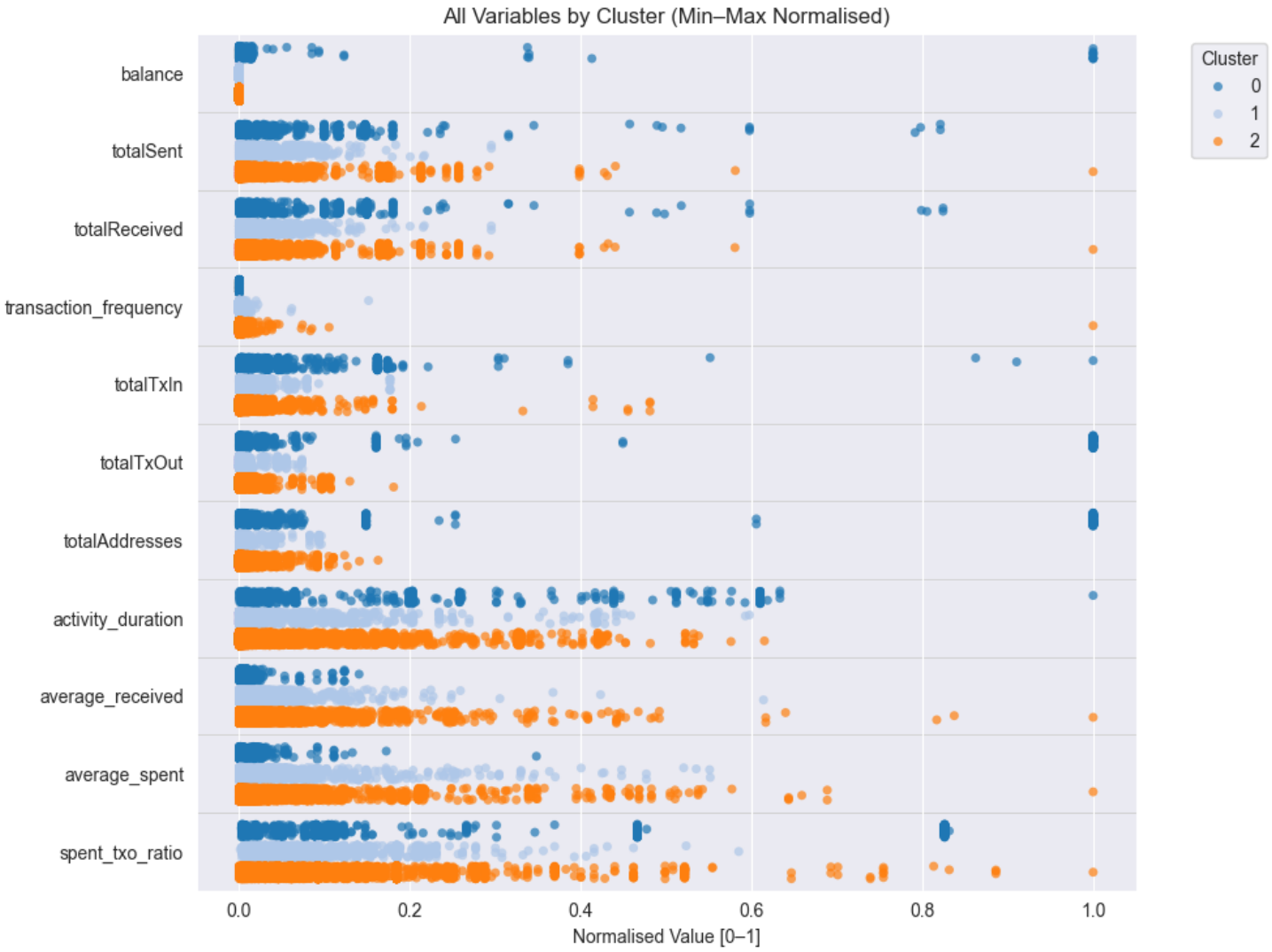

Figure 4: Strip plot of post-mixing features by cluster

4.3 User Profiles from Transaction Graph Clusters

Now that we know that at least pre-mixing wallets have some differentiating patterns, we look to see whether we can differentiate them even more. We have learned that post-mixing wallets themselves have less differentiating properties, but perhaps that changes when expanding the scope of the analysis. Therefore, the wallets we created for sub-question 1 form the starting point for the following analysis. The second sub-question reads as follows:

SQ 2

How effectively can pre- and post-mixing addresses be clustered using features drawn from their transaction graphs?

By extracting features from the pre- and post-mixing transaction graph with the GAE,

we look to quantify the transaction graph of different users. This allows us to cluster those quantified graphs and examine transaction patterns, which we can then use for prediction in sub-question 3.

4.3.1 Pre-Mixing Transaction Graphs

Quantifying and clustering the pre-mixing transaction graph provides us with insights into the users of this mixing service, and forms the *input* for the prediction model we develop in sub-question 3.

Embeddings and Clustering Following the grid search found in Appendix B.4, we choose to use 8 embedding dimensions, 128 hidden dimensions, and a learning rate of 0.0005 to create the embeddings for pre-mixing graphs. The result of the GAE is that we reduced every graph to a list of 8 numbers (*embeddings*) which we can then cluster.

For k-means clustering of the embeddings, we choose 4 clusters as the optimal number following the elbow-method graph in Appendix B.5. Running the algorithm results in clusters of the following sizes:

Table 14: Cluster sizes for pre-mixing transaction graphs

Cluster	**Count**	**Percentage**
Cluster 0	6,181	33.5%
Cluster 1	2,968	16.1%
Cluster 2	6,071	32.9%
Cluster 3	3,242	17.6%

Variables An ANOVA test confirms that all variables discussed below vary significantly between clusters ($p < 0.05$). Table 15 summarises the topological graph metrics for each cluster, while Table 16 shows the wallet- and transaction-level metrics. Metrics are expressed in relative terms to enable straightforward comparison across variables before developing user profiles. Definitions of all variables were previously provided in Tables 4 and 6.

Table 15: Cluster Characteristics Graphs

Metric	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Nodes	Moderate	Low	High	Low
Edges	Moderate	Low	High	Low
Density	Low-Moderate	Moderate–High	Low	High
Diameter	Moderate	Low	High	Low
Avg. Degree	Moderate	Low	Moderate	Low
Max Degree	Low-Moderate	Low	High	Low
Avg. Betweenness	Moderate	High	Moderate	High
Avg. Closeness	Moderate-Low	Moderate–High	Low	High
Assortativity	Moderate	Low	Moderate	Low

Table 16: Cluster Characteristics Wallets and Transactions

Metric	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Balance	Low	Very High	Low	Low
Total Sent & Received	Low	High	Low	High
Total Tx.	Low	Moderate	Low	High
Total Tx. In	Low	Moderate	Low	High
Total Tx. Out	Low	Moderate	Low	High
Total Addresses	Low	Moderate	Low	Very High
Avg. Received	High	Moderate	High	Low
Avg. Spent	High	Moderate	High	Low
Activity Duration	Low	High	Moderate	High
Tx. Frequency	Low	Moderate	Low	Very High
Spent TxO. Ratio	Moderate	Moderate	Moderate	Moderate
Value per Tx	Moderate-High	Moderate	High	Low
Time Delta	Moderate-High	Moderate	High	Moderate

Services Composition In addition to wallet and graph metrics, the presence and types of services in each graph help explain variations in metrics, thereby supporting cluster interpretation. Table 17 shows the number of graphs per cluster that have one or more service. A high percentage means that many of the graphs in a certain cluster contain at least one service. This indicates that the money deposited into a mixer likely (partly) originated from that service. Figure 5 shows the number and kind of services in the graphs per cluster. Each colour represents a different cluster. The y-axis shows the kinds of services that are present in graphs of a certain cluster, with the x-axis denoting how often those services show up in the graphs. This is important because it gives us an idea of how many and what kinds of services often show up in the graphs of a different cluster, which helps interpret them.

Table 17: Number and percentage of graphs with at least one service per cluster

Cluster	Count with Service	Percentage
Cluster 0	2,933	47.45%
Cluster 1	2,954	99.53%
Cluster 2	4,308	70.96%
Cluster 3	3,122	96.30%

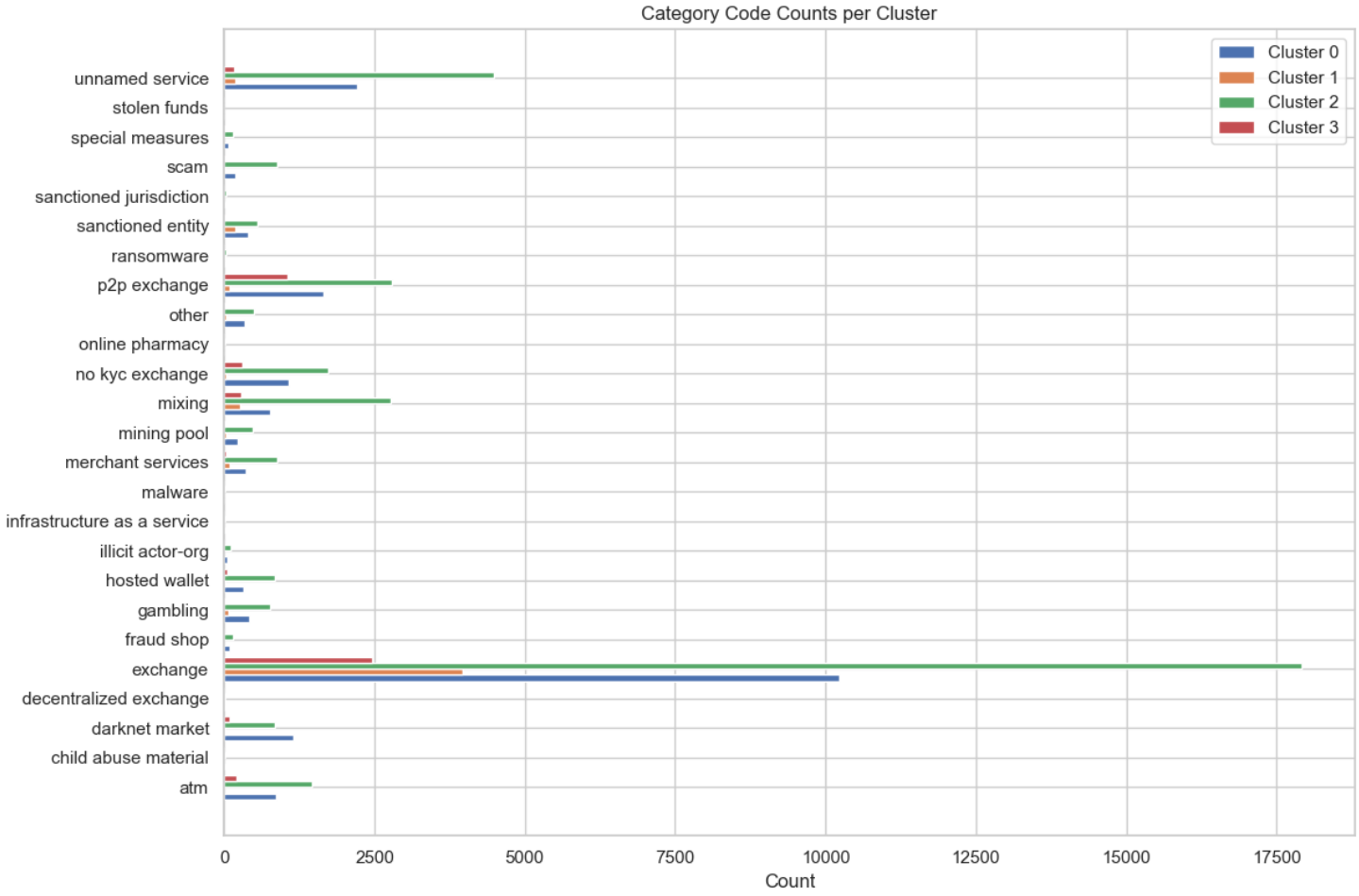

Figure 5: Horizontal bar chart showing service count per pre-mixing cluster

Cluster profiles The structural metrics (Table 15), wallet- and transaction-level metrics (Table 16), and service counts (Table 17 and Figure 5) combine to reveal four distinct user types:

- **Cluster 0: "Consolidator".**
  - *Graph structure.* Graphs have moderate node and edge counts, low–moderate density, and a moderate diameter. This indicates lightly connected, average-sized structures.
  - *Wallets and transactions.* Wallets contain few addresses and low total volume, but high average transaction values. Funds trickle slowly to the mixer in occasional

high-value bursts.

– *Service usage.* 2,933 of 6,181 graphs (47.45%) include at least one service, usually a single exchange.

We see that not a lot of services are used, but the low-moderate degree values indicate there must be some hub-like wallets in the graphs. The high values therefore most likely come from both service and non-service wallets. From this, we conclude that these users consolidate larger sums (shown from the high value per transaction) from one or two sources before mixing.

- **Cluster 1: "Straightforward Depositor".**

  – *Graph structure.* Graphs are small, with the lowest node and edge counts and moderate–high density. Centrality scores are the highest of all clusters.

  – *Wallets and transactions.* These wallets have the highest balances, long activity durations, and high volume. However, individual transactions are moderate in size and spread out over time.

  – *Service usage.* 2,954 of 2,968 graphs (99.53%) contain a service, typically one or two exchanges.

  Users seem to fund the mixer directly from an exchange in few hops, as we see that almost all graphs have at least one service, likely an exchange. The compact graph shape and strong centrality suggest minimal routing beyond the deposit. Therefore, we characterise this pattern as straightforward.

- **Cluster 2: "Aggregator Funnels".**

  – *Graph structure.* The largest graphs, with high node/edge counts and low density. Several nodes act as high-degree hubs funnelling funds.

  – *Wallets and transactions.* Address-level metrics are low, but average transaction size is high, with a long time-delta to the mixer. Funds move slowly in sizeable chunks.

  – *Service usage.* 4,308 of 6,071 graphs (70.96%) include a service; mostly exchanges.

  The pattern suggests that many small wallets forward service withdrawals into central hubs, reflected in the high degree values. These then take a long time, shown by the high diameter and time delta, before depositing large amounts into the mixer, as evidenced by the high value per transaction. Users aggregate funds and then slowly funnel them

to the mixer.

- **Cluster 3: "Higher-Risk User".**
  - *Graph structure.* Similar to Cluster 1 but with many more addresses and much higher transaction frequency. Density is the highest of all clusters.
  - *Wallets and transactions.* Wallets have low balances and make frequent low-value transfers. Activity duration is high, likely due to service involvement.
  - *Service usage.* 3,122 of 3,242 graphs (96.30%) contain a service, relatively more peer-to-peer or no-KYC exchanges.

  These users resemble Cluster 1 because of their compact graph shape and presence of services. They rely more heavily on higher-risk (no-KYC and peer-to-peer) exchanges to fund their deposits into the mixer, which is why we denote them as having a higher risk than Cluster 1.

These profiles and their quantified graph characteristics form one half of the research objective of identifying pre-mixing patterns that can help predict post-mixing patterns.

4.3.2 Post-Mixing Transaction Graphs

The results of this part of the sub-question provide the *output* of the prediction model we develop in sub-question 3. Instead of all the user profiles we construct in this section, our prediction model will narrow it down to one of the profiles constructed in this section.

Embeddings and Clustering We choose to use 8 embedding dimensions, 128 hidden dimensions, and a learning rate of 0.001 to create the embeddings for pre-mixing graphs. The elbow plot in Appendix B.5 points to either three or four natural groupings. Visual inspection of both options shows that a three-cluster solution provides clearer separation, so we proceed with $k = 3$. Cluster sizes and their corresponding percentages relative to the total are given in Table 18.

Table 18: Cluster sizes for post-mixing transaction graphs

Cluster	Count	Percentage
Cluster 0	4,909	40.9%
Cluster 1	4,706	39.2%
Cluster 2	3,622	19.9%

Variables The results of the clusters are shown in Tables 19 and 20. An ANOVA test confirms that every variable in both tables varies significantly across clusters ($p < 0.05$).

Table 19: Cluster Characteristics Graphs

Metric	**Cluster 0**	**Cluster 1**	**Cluster 2**
Nodes	High	Moderate	Moderate–Low
Edges	High	Moderate	Moderate–Low
Density	Low	High	High
Diameter	Moderate	Moderate	Moderate
Avg. Degree	Moderate	Moderate	Moderate
Max Degree	High	Moderate	Low
Avg. Betweenness	Moderate	Moderate-Low	Moderate
Avg. Closeness	Moderate	Moderate	Moderate
Assortativity	Moderate	Moderate	Moderate

Table 20: Cluster Characteristics Addresses

Metric	**Cluster 0**	**Cluster 1**	**Cluster 2**
Balance	Moderate	High	Moderate
Total Sent & Received	Moderate–Low	Moderate	Moderate
Total Tx.	Moderate	Moderate	Moderate
Total Tx. In	Moderate	Moderate	Moderate
Total Tx. Out	Moderate	Moderate	Moderate
Total Addresses	Moderate	Moderate	Moderate
Avg. Received	Low	High	Low
Avg. Spent	Low	High	Low
Activity Duration	Moderate	Moderate	Moderate
Tx. Frequency	Moderate	Moderate	Moderate
Spent TxO. Ratio	Moderate	Moderate	Moderate
Value per Tx	Low	High	Low
Time Delta	Moderate	Moderate	Moderate

Service Composition As with the pre-mixing analysis, service composition is an additional lens for interpretation. Table 21 lists how many graphs in each cluster contain at least one service node, while Figure 6 breaks down the mix of service types.

Table 21: Number and percentage of graphs containing ≥1 service

Cluster	Count with Service	Percentage
Cluster 0	4,662	94.97%
Cluster 1	3,197	88.27%
Cluster 2	4,284	91.03%

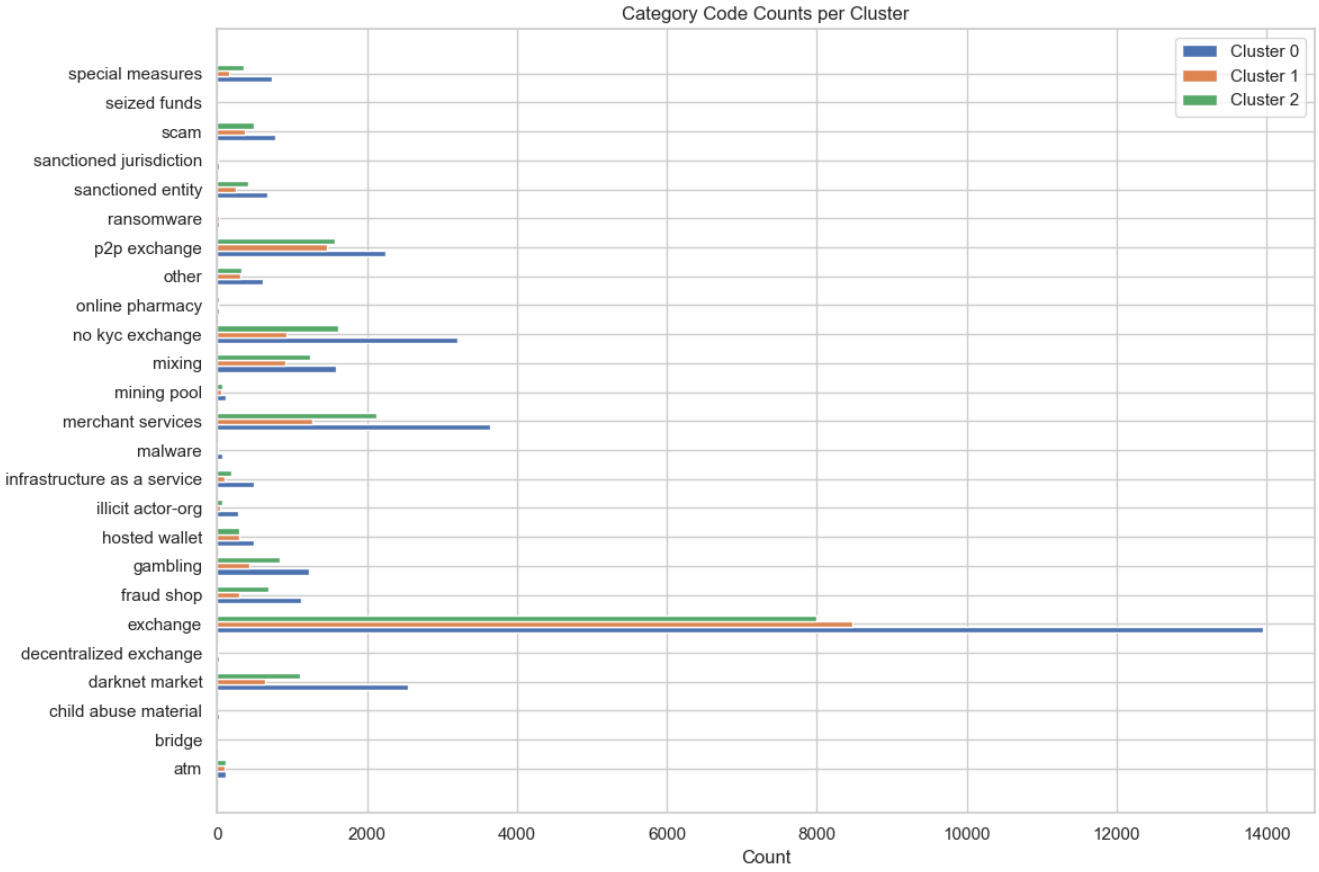

Figure 6: Horizontal bar chart showing service count per post-mixing cluster

Cluster profiles The graph-level metrics, address-level metrics, and service compositions combine to reveal three distinct post-mixing user types:

- **Cluster 0: "Splitter".**
  - *Graph structure.* Graphs are large and sparse: high node/edge counts, low density, moderate diameter, and the highest max-degree. A few hub nodes split funds to many others.
  - *Wallets and transactions.* Wallet metrics are moderate, but average transaction size is low. Funds leave the mixer in many small payments fanning out from hubs.
  - *Service usage.* Nearly all graphs (95%) touch at least one service; this cluster has the highest total service count.

These patterns suggest users who split funds into many small wallets. The high max degree indicates money is funneled to one or more hubs, while the high node count and low value per transaction values suggest funds are split toward many wallets and services in small amounts. The user most likely cashes out at multiple different services.

- **Cluster 1: "Big-Time Distributor".**
  - *Graph structure.* Mid-sized graphs with moderate nodes/edges, high density, and moderate max-degree. Fewer hubs, shorter, more interconnected paths.
  - *Wallets and transactions.* Highest average transaction values of all clusters; other metrics are moderate. Funds move in large hops.
  - *Service usage.* About 88% of graphs include a service; slightly lower than others.

  This pattern reflects a user operating a compact, tightly linked wallet network (indicated by high density), funnelling large amounts to downstream services.

- **Cluster 2: "Straightforward User".**
  - *Graph structure.* Smallest, densest graphs: low node/edge counts, high density, lowest max-degree. No hubs, short paths.
  - *Wallets and transactions.* Moderate–low values across all metrics; similar to Cluster 0.
  - *Service usage.* Over 91% include a service, with slightly more illicit categories than Cluster 1.

  These users appear to withdraw and cash out without complex routing, indicated by the small dense graphs with lower node counts. The absence of hubs and the tight, small graphs point to direct usage with minimal redistribution.

Although the cluster profiles are less distinct than the pre-mixing user profiles, particularly regarding wallet values, the differences remain sufficient to define meaningful profiles. These profiles represent the patterns we aim to predict in line with our research objective: given a pre-mixing transaction graph, determine which post-mixing profile is most likely to occur.

4.4 Post-Mixing Cluster Prediction

The final sub-question is as follows:

SQ 3

How reliably do clusters formed from pre-mixing transaction patterns predict the corresponding clusters in post-mixing transactions?

The results in this section introduce the probabilistic dimension of our approach: given a pre-mixing transaction graph, we can estimate the likelihood that its corresponding post-mixing graph belongs to a particular user profile. While this does not yield a definitive match, it significantly narrows the search space; from all possible post-mixing outcomes to a smaller, more targeted set of likely profiles.

Model Parameters Presenting the full set of grid search results is impractical due to the scale of the search space; the resulting output spans over 2,500 unique parameter-value pairs. It is therefore impractical to show the results of the full grid search in a graph or table. The best scoring parameters can be found in Table 22.

Table 22: Best Parameter Configuration (RandomizedSearchCV)

Parameter	**Selected Value**
Gradient Boosting (GB)	
Learning rate	0.026
Maximum depth	10
Minimum samples at a leaf	1
Minimum samples to split a node	2
Number of trees	156
Subsample ratio	0.667
Random Forest (RF)	
Bootstrap sampling	False
Maximum depth	16
Maximum number of features	`sqrt`
Minimum samples at a leaf	1
Minimum samples to split a node	2
Number of trees	146

Independence Test The chi-squared independence test and Cramér's V association resulted in the following values:

- Chi-Squared Test Statistic: 1096,9410
- p-value: 0.000
- Cramér's V: 0.1407

We can see that there is a significant ($p < 0.05$) relationship between pre- and post-mixing cluster labels. However, the Cramér's V value is relatively low, as a value of 0 indicates no association and a value of 1 a perfect association.

Prediction We have five post-mixing cluster labels: Clusters 0-2 are outlined in Section 4.3.2. We denote Cluster 3 as a service address, and Cluster 4 is an address that has no activity 60 days from the day the mixer had paid out the deposited funds (see Section 3.6).

Table 23 shows the weighted average scores per metric and the accuracy score for the Random Forest, Gradient Boosting, and the Ensemble model combining the two. The ensemble model scores best on all metrics, though just slightly. Table 24 shows the results of the Ensemble model. The metrics have been explained in Section 3.6, and the "Support" column indicates how big the sample was that the model tested its predictions on.

Model	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Random Forest	0.48	0.47	0.45	0.47
Gradient Boosting	0.47	0.47	0.45	0.47
Ensemble	0.50	0.48	0.46	0.48

Table 23: Classification report for post-mixing clusters

Post-Mixing Cluster	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.48	0.64	0.55	981
1	0.49	0.41	0.45	724
2	0.45	0.63	0.53	941
3	0.48	0.28	0.35	744
4	0.72	0.14	0.23	303
Accuracy			**0.48**	3693
Macro Avg	0.53	0.42	0.42	3693
Weighted Avg	0.50	0.48	0.46	3693

Table 24: Ensemble model classification report for post-mixing clusters

Besides the overall accuracy of 0.48, some patterns in the results are worth pointing out. For Clusters 0 and 2, the model performs quite well: it correctly finds most of the real cases (recall of 0.64 and 0.63), and when it does make a prediction, it's fairly often right (precision of 0.48 and 0.45). Because of this, both clusters get an F1-score above 0.50, which shows a good balance between finding cases and being correct. Cluster 4 is different: it has high precision (0.72), meaning its predictions are usually right, but very

low recall (0.14), meaning it misses most of the true cases. So the model is cautious with Cluster 4; it only predicts it when it's very sure, but it often overlooks it.

Finally, Figure 7 shows the feature importances per feature included in the prediction model. We see that `deposit_amount` has the highest predictive power, followed by the embeddings produced in sub-question 2. The figure also shows that the four pre-mixing clusters that we identified barely contribute to the prediction. This is important because it allows us to refine our models in the future to reduce noise and focus on the variables that are actually contributing to the prediction.

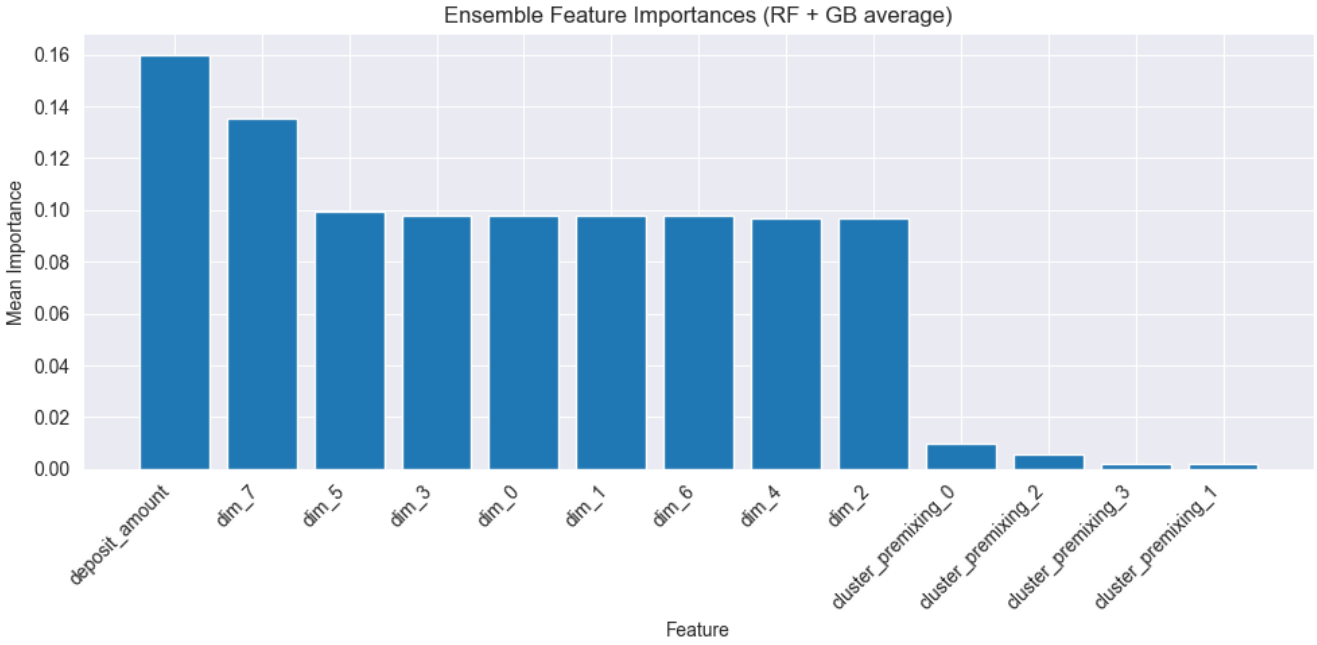

Figure 7: Feature importances for ensemble model

5 Discussion

The goal of this thesis was to identify pre-and post-mixing transaction patterns and use those patterns to develop a method that highlights the most probable post-mixing patterns based on pre-mixing pattern. This approach should be able to aid law enforcement efforts in combating illicit cryptocurrency usage. In this section, we reflect on the study's results, examine their scientific and operational implications, outline the study's limitations, and identify directions for future research.

5.1 Reflection on the Results

The main research question we set out to answer is the following:

Main Research Question

To what extent do pre- and post-mixing Bitcoin transaction networks display patterns that can be leveraged to narrow the pool of plausible post-mixing addresses linked to a given pre-mixing address?

We did this by first looking at pre- and post-mixing wallets, then analysing transaction graphs, before developing a prediction model. We will reflect on these sub-questions in this section.

Clustering Based on Address Attributes The results for sub-question 1 show that clustering based on pre- and post-mixing wallet attributes has limited utility. A high proportion of outliers and the dominance of a single large cluster indicate little variation in the selected features.

Nevertheless, three pre-mixing profiles, "Dormant Whales," "Casual Users," and "High-Volume Bursts," provide an initial, albeit noisy, view of possible usage patterns. However, with more than half of wallets left unclustered, these profiles cannot serve as a reliable general typology. The short activity duration observed in many pre-mixing wallets suggests they are created solely for depositing funds, likely to minimise address reuse and hinder traceability when handling potentially illicit transactions. This behaviour limits the available transaction history per wallet, making meaningful clustering challenging due to a lack of data.

Post-mixing clustering proved even less informative. The data was more homogeneous,

clear profiles did not emerge, and many addresses were linked to illicit services, a pattern consistent with the role of mixers in obscuring criminal proceeds.

Overall, these findings indicate that a single-wallet perspective provides too little transactional data for effective clustering. This insight underlines the need to analyse richer structures, such as transaction graphs, to identify distinctive patterns in mixer use.

Graph-Based Clustering Graph-based clustering produced more distinct and interpretable patterns than address-based clustering, particularly for pre-mixing transaction graphs. Incorporating structural features of the transaction network added valuable information, enabling the identification of clear user profiles pre-mixing and broader transaction strategies post-mixing.

Pre-mixing graphs. Four profiles emerged: "Consolidator," "Straightforward Depositor," "Aggregator Funnels," and "Higher-Risk Users." The "Consolidator" collects funds from a few services, both low- and high-risk, before routing them to the mixer. However, fewer than half of these graphs contain a service, suggesting that some consolidation activity may occur entirely outside known service infrastructure. The "Straightforward Depositor" sends funds directly from a KYC-regulated exchange. This is one of the most distinct clusters, with low within-cluster variety. The presence of KYC-regulated exchanges could allow for easy identification of these mixer users. The "Aggregator Funnels" have large, complex graph structures in which funds pass through intermediary hub-like wallets, often from many licit and illicit sources. Their high service presence suggests a possible layering strategy to obscure origins, though the lower service percentage indicates this pattern is not universal throughout the cluster. The "Higher-Risk User" deposits more frequently from peer-to-peer exchanges with weaker KYC, though its structure and attributes closely resemble the "Straightforward Depositor." This may indicate overlap in user patterns despite differing source types, which makes this cluster less reliable.

Post-mixing graphs. Classification was more challenging, leading to the choice of three profiles: "Splitter," "Big-Time Distributor," and "Straightforward User." The "Splitter" disperses funds from a central hub to many addresses shortly after mixing, potentially as an obfuscation tactic. The "Big-Time Distributor" also uses hub-like structures but with fewer output addresses, each receiving larger amounts, resulting in more concentrated withdrawals. Both of these patterns suggest deliberate structuring of post-mixing flows,

though the exact intent remains speculative. The "Straightforward User" sends funds from the mixer to only one or a few addresses, sometimes directly to an exchange, reflecting either low-risk or casual use. Across all three profiles, the consistent presence of both illicit services and regulated exchanges challenges the assumption that anonymity-seeking users avoid KYC endpoints, and may reflect either confidence in the mixer's obfuscation or the use of mixed funds for legitimate purposes.

Reflection on uncertainty. The certainty (i.e. uniformity) of cluster assignment varies notably across profiles. Well-defined clusters allow for more confident interpretation and prediction, while others require caution due to overlapping characteristics or incomplete service attribution. This highlights that although graph-based clustering offers richer insights than address-based methods, not all profiles can be applied equally. Recognising where patterns are robust and where they are ambiguous helps avoid over-interpretation and prioritise clusters in downstream analysis. Table 25 summarises the uncertainty levels for each profile. These levels are not quantified, but they are estimated using visual inspection of the profile variable values and ranges.

Table 25: Profiles by graph type with indicative clustering uncertainty

Graph Type	**Profile Name**	**Cluster Uncertainty**
Pre-mixing	Consolidator	High
Pre-mixing	Straightforward Depositor	Low
Pre-mixing	Aggregator Funnels	Medium
Pre-mixing	Higher-Risk User	High
Post-mixing	Splitter	Low
Post-mixing	Big-Time Distributor	Medium
Post-mixing	Straightforward User	Medium

Prediction of Post-Mixing Clusters The classification results show that the ensemble model performs substantially better than random guessing, even though the overall accuracy is moderate. In this task, there are five possible target clusters. If someone had no information and simply guessed a cluster at random, the probability of being correct would be one out of five, or 20%, assuming the clusters are roughly balanced in size. This 20% therefore serves as a baseline for comparison. The model achieves 48% accuracy, meaning it predicts the correct cluster more than twice as often as random guessing. This improvement demonstrates that the model is identifying real and consistent patterns in the data that can be used to make informed predictions.

Feature importance. However, the feature importance analysis reveals a crucial nuance: the pre-mixing cluster labels contribute very little to predictive performance. This suggests that the profiles derived from clustering do not drive the model's success. Instead, the graph embeddings (i.e. quantified graph structure) carry the bulk of the predictive power. In practical terms, the model is learning from the topology of each graph, rather than from its membership to a profile category. This finding firstly reinforces the value of constructing rich graph embeddings, as they retain relevant discriminative information. Second, it highlights that while pre-mixing profiles are useful for understanding general user types, they are less effective as direct predictive features in linking to post-mixing behaviour as much information is lost between embeddings and clusters.

Reflection on probabilistic results. The modest performance of the model should be interpreted in its operational context, where the goal is not to make high-stakes binary decisions but to generate investigative leads. In such scenarios, false positives and false negatives are less critical than in applications like medical diagnostics or cyberattack prevention, and even with imperfect recall and precision the model can significantly narrow the search space. However, the evaluation metrics still reveal risks: low precision means many predicted addresses in a cluster will be incorrect, potentially diverting resources, while low recall means many true instances are missed, risking the loss of important leads. These trade-offs vary by cluster, and the optimal balance depends on operational priorities, whether it is more important to minimise false leads or to capture as many true cases as possible. Recognising these limitations is essential for responsible interpretation and for integrating the model into a broader, multi-source investigative workflow.

5.2 Implications

In this section we cover both the academic and practical implications of this research.

5.2.1 Scientific Contributions

This thesis adds to the literature primarily in four ways: an understanding mixer users, the usage of extensive ground-truth data, an extension of the conceptual model of (Wu et al., 2021), and the adoption of a new methodology.

Understanding Mixer Users This study offers more insights into how mixers interact with mixers. Firstly, we underscore that users adopt minimal privacy tactics. We thereby

expand on Miedema et al. (2023) by adding that the origins and destinations of funds are also often not obfuscated, and reaffirm the conclusion by Crawford and Guan (2020) that users place a lot of trust in the mixing service.

Additionally, not only do users not use privacy tactics, they appear to exhibit consistent trends and patterns. This is confirmed by our ability to partly predict post-mixing patterns based on pre-mixing patterns, which means these users show non-random, perhaps even habitual patterns in their interactions with the mixer. This could suggest that users also exhibit this behaviour while using other mixers, not just our dataset. However, these differences in patterns are likely more nuanced than captured here, as our transaction graphs represent fund flows toward the mixer rather than direct user actions. We did not verify whether all wallets in a graph belonged to the same user, since this was unnecessary for our prediction task. We can therefore conclude that although the observed transaction patterns are clearly distinct, we cannot say with certainty that this is entirely attributable to a single user.

Data Source We assert that a persistent challenge in the academic study of Bitcoin mixers is the scarcity of ground-truth data, which hampers the ability to rigorously validate research findings. This thesis is based on a unique data source derived from law enforcement access to actual mixer records. This represents a significant improvement in terms of data reliability. Unlike previous work that had to infer mixer detection and de-mixing attempts more indirectly, this dataset contains confirmed links between users, deposits, and withdrawals. The presence of such verified information provides a solid foundation for analysing address behaviour and transaction graphs, and it strengthens the internal validity of our results.

Conceptual Contribution Only a limited number of studies focus directly on the individuals who use Bitcoin mixers, even though understanding user behaviour is essential for interpreting how these systems are used in practice. Some research has examined user perceptions on trust and mixers (Crawford & Guan, 2020), while studies such as Miedema et al. (2023), have looked at the security behaviour of mixer users. However, there is a clear gap in the literature when it comes to analysing the actual transactional patterns that mixer users exhibit on the blockchain.

This study addresses that gap by starting from the assumption that mixer users are

not a uniform group. We propose that different types of users exist, each with distinct transactional patterns. As posited in Section 3.1, we adjusted the conceptual model of Wu et al. (2021) by expanding their input and output phases to include pre- and post-mixing transaction patterns. We hypothesised that there was a relationship between the patterns exhibited pre-mixing (Phase 1) and those exhibited post-mixing (Phase 3). We conclude that this relationship does exist, and that the way the pre-mixing phase looks says something about the way the post-mixing phase looks. We showed that we can even predict what Phase 3 looks like based on Phase 1, without considering Phase 2 (the mixing process).

Therefore, we assert that expanding the conceptual model as proposed by Wu et al. (2021) would more accurately capture the mixing system and its different actors. This would reveal the mechanism where not just the mixing process plays a part in obfuscation; the user itself also plays a big role. Adopting this new perspective should broaden the perspective we have on mixers and how they work and open up new areas to research how to circumvent their obfuscation.

Probabilistic Method Most existing studies that attempt to link mixer inputs to outputs focus on establishing direct, deterministic correlations. This approach underlies many earlier efforts in tracing through mixers on different blockchains (de Balthasar & Hernandez-Castro, 2017; Du et al., 2024; Hong et al., 2018).

In this study, we propose a broader, probabilistic approach to de-mixing. Rather than attempting to pinpoint exact withdrawal addresses, we aim to decrease the pool of candidate post-mixing addresses on a probabilistic basis. By shifting from a deterministic to a probabilistic framework, our methodology acknowledges the limitations of perfect traceability through a mixer. It has the advantage of not needing internal mixer knowledge, at the cost of some uncertainty. However, this uncertainty is nuanced when compared to deterministic methods. Unlike deterministic models, which might identify one address out of 100 with 70% certainty but still risk being wrong, a probabilistic model might narrow 100 candidates to 20 with 50% confidence, increasing the chance of pursuing the right lead. The choice between the two approaches depends on the investigator's priorities. While we don't claim that the probabilistic approach is better than the deterministic approach, we do conclude that it offers a very different, complementary perspective on de-mixing, depending on what the de-mixer needs and what information they have available to them.

5.2.2 Practical Implications

The practical implications of this research stem primarily from its confirmation that pre-mixing transaction patterns can provide meaningful insight into post-mixing patterns, and from the subsequent development of a prediction model. While the user profiles generated in this study offer a view into how users interact with mixers and the flows of funds around them, their direct utility for law enforcement is limited. This is partly due to the uncertainty in the clusters themselves, as each still encompasses a wide range of user behaviours. Applying these profiles without caution risks oversimplifying the diverse transaction patterns present in the data.

In contrast, the prediction model could have significant operational value. Given access to a user's pre-mixing transaction graph, an investigator could embed it, feed the embedding into the model, and receive a predicted cluster label. Post-mixing graphs could then be embedded and clustered, allowing investigators to prioritise those post-mixing graphs with the same label as the prediction. This targeted approach increases the likelihood of identifying a promising lead without exhaustively checking every post-mixing graph. With an accuracy of 48%, this method could meaningfully improve investigative efficiency. Moreover, because the approach does not rely on internal knowledge of a mixer's operations, it may be generalisable to other mixers, although this remains to be tested.

There are, however, important caveats. The first is that an investigator must wait at least 60 days before embedding and clustering the candidate set of post-mixing graphs, due to the graph-construction constraints used in this study. Also, to apply the profiles developed here, the exact same embedding model that was trained on the original dataset must be used along with the same graph constraints. This is essential because GAEs learn a specific mathematical representation of the data during training. A different model, even if trained on similar data, would produce embeddings that are positioned and scaled differently, making them incompatible with the original clusters. The 60-day time window is therefore not just a data requirement but also a methodological constraint, raising the question of how quickly investigative leads must be pursued.

The second caveat is that an investigator still needs to identify where the mixer's outputs occur in order to track outgoing funds. This requires detailed knowledge of which addresses belong to the mixer and which don't. While mixer detection and classification techniques have advanced considerably (as discussed in Section 2.2), perfect accuracy is

unlikely and any errors here introduce further uncertainty into the method.

5.3 Limitations and Future Research

5.3.1 Limitations

Several limitations should be considered when interpreting the results of this research. We distinguish limitations in four areas.

Data Limitations The majority of the data originates from orders placed via Bestmixer's clear-web interface. These users may be less privacy-conscious than those using the more anonymous Tor browser, which could influence transactional patterns. As a result, the identified user profiles and clustering outcomes may not generalise to more anonymity-focused users, placing the findings in a specific behavioural context.

Methodological Constraints Methodological constraints stem from the limitations in constructing the graph and the use of the GAE.

Transaction graphs were limited to a maximum depth of five hops from the deposit address. While this choice follows previous research (Rosenquist et al., 2024), it prevents full reconstruction of user transaction chains and may omit relevant obfuscation strategies that unfold at greater depths. Additionally, to manage computational load and reduce noise, the analysis excluded addresses with more than 189 incoming transactions and limited the transaction window to 60 days after deposit. Although these thresholds were justified, they potentially excluded some meaningful patterns and may have truncated graphs in non-obvious ways.

Secondly, the GAE is not strictly reproducible. Due to stochastic elements in neural network training, repeated runs may produce slightly different embeddings even with the same parameters. Additionally, the constraints mentioned above have to be used when applying the GAE to other datasets. This offers a trade-off; on the one hand, increasing the constraints (especially the time-window) offers more information to the model, which could improve prediction performance. On the other hand, a long time-window means that an investigator has to wait a long time before being able to use the model (as discussed previously). It is unclear what the ideal parameters are that balance practical utility with predictive power.

Explainability The GAE used for creating graph embeddings captures complex structural and feature-based information, but the resulting embeddings are abstract and difficult to interpret. This makes it challenging to link specific user patterns directly to cluster assignments. Additionally, it hampers our ability to fully understand our prediction model. We saw that the embeddings had a high feature importance, so they contribute a lot to the prediction, but other than that the numbers don't tell us much. When used in real-life contexts, this can be dangerous since human oversight (and understanding) is reduced.

Cluster Overlap As mentioned previously, most clusters that were created show overlapping ranges of attributes. This reduces the practical utility of the cluster labels. Additionally, overlap in cluster ranges means a prediction model has more difficulties correctly classifying a point. The results of this research should therefore function as a stepping stone to more distinct clusters that have clearer boundaries.

5.3.2 Further Research

Given the identified limitations and scope of this research, we identified a number of areas for further research.

Model Refinement We believe a lot of gains can be made by further refining the embedding prediction model. Future research could explore alternative graph embedding strategies, such as using a higher number of embeddings or experimenting with different hyperparameters to better capture subtle differences. Additionally, evaluating other embedding methods may yield more interpretable or discriminative representations than the GAE used in this study. This might then also increase the prediction accuracy of the probabilistic de-mixing method.

Broaden Dataset Second, future research should seek to validate these findings using a broader set of mixer services to assess their generalisability across different user groups. Although one might hypothesise that users do not significantly vary their behaviour across mixers, testing this on additional services would strengthen the robustness of the current results. Moreover, given the rapid evolution of this field, a dataset from 2019 may no longer fully reflect present-day practices. Repeating this analysis on more recent data could therefore provide valuable insights and enhance the relevance of the findings.

User Motivations An interesting and challenging direction for future research concerns the underlying motivations that lead users to fall into different behavioural groups. This line of inquiry is based on the conclusion that mixer users make conscious choices about how they interact with the service. However, the reasons behind these choices remain unclear. Understanding these motivations is difficult, primarily due to the lack of direct data on users' intentions or contextual factors. As a result, uncovering the drivers behind distinct transactional patterns will likely require alternative data sources or complementary methods such as interviews, surveys, or forum analyses.

Returning Users Finally, this study did not account for returning users, though they are present in the dataset. Users of Bestmixer would receive a "user_id" they could enter after a first order to make sure they didn't receive their own coins back. Future research could investigate whether individual users exhibit consistent transaction patterns over different uses, providing insight into the evolution (or stability) of patterns over time. This puts the focus even more on individual, unique user behaviour. It could also provide more insight into whether the transaction graphs reflect actual user behaviour instead of just the flow of funds.

6 Conclusion

This thesis set out to determine whether pre- and post-mixing Bitcoin transaction networks display patterns that can be used to narrow the pool of plausible post-mixing addresses for a given pre-mixing address. Using a law-enforcement–seized, ground-truth dataset from Bestmixer.io, we approached the problem in three steps. First, we tested whether aggregated address-level attributes suffice to identify patterns around mixing. Second, we constructed transaction graphs around deposits and withdrawals, learned graph embeddings with a graph autoencoder, and clustered those embeddings to uncover structural typologies before and after mixing. Third, we examined whether pre-mixing information could predict post-mixing patterns with a supervised model.

SQ1

How effectively can pre- and post-mixing wallets be clustered using aggregated address-level attributes?

Address-level clustering offers only limited separation. On the pre-mixing side, wallets form a small number of coarse profiles with many borderline cases; on the post-mixing side, wallets appears relatively homogeneous and cluster assignments are influenced by service categories rather than distinctive transaction routines. In practical terms, address-only features do not provide the discriminatory power needed to meaningfully narrow candidates after mixing, which motivates a shift toward graph-level analysis.

SQ2

How effectively can pre- and post-mixing addresses be clustered using features from their transaction graphs?

Graph embeddings followed by k-means produce distinct, interpretable clusters that capture how funds are routed and redistributed. Before mixing, we observe *Consolidator*, *Straightforward Depositor*, *Aggregator Funnels*, and *Higher-Risk User*. After mixing, patterns include *Splitter*, *Big-Time Distributor*, and *Straightforward User*. These structural signals, derived from graph connectivity and flow, provide substantially clearer differentiation than address-level attributes. We show that this method is suited to analyse transaction patterns, though cluster overlap is still present.

SQ3

How reliably do clusters formed from pre-mixing patterns predict corresponding post-mixing clusters?

Using pre-mixing graph embeddings together with the deposit amount, a tree-based ensemble predicts post-mixing clusters with an accuracy of 0.48 across five classes, which is well above a naive random baseline. Feature-importance analysis shows that deposit size and graph embeddings carry most of the predictive signal, while the pre-mixing cluster label itself adds little incremental value. These results support a probabilistic strategy that focuses follow-up on post-mixing candidates consistent with the predicted patterns.

Main Research Question

To what extent do pre- and post-mixing Bitcoin transaction networks display patterns that can be leveraged to narrow the pool of plausible post-mixing addresses linked to a given pre-mixing address?

Pre- and post-mixing networks do exhibit systematic patterns that are actionable once graph structure is taken into account. Graph embeddings reveal interpretable typologies on both sides of the mixer, and, combined with the deposit amount, support a supervised model that achieves 0.48 accuracy across five classes. This is sufficient to reduce the investigative search space in a probabilistic manner, prioritising a smaller set of plausible post-mixing patterns for further inquiry.

Taken together, these findings imply a practical workflow in which investigators embed a user's pre-mixing graph, obtain a predicted post-mixing label, then embed and cluster candidate post-mixing graphs to prioritise those matching the prediction, thereby improving efficiency without relying on internal mixer knowledge and with potential (yet untested) generalisability to other mixers. Key caveats for application are the need to wait at least 60 days before embedding and clustering candidate post-mixing graphs due to the graph-construction setup, the requirement to use the same trained embedding model and graph constraints for compatibility, and the necessity of accurately locating mixer outputs on-chain. More broadly, interpretation and transfer are bounded by the predominance of clear-web orders in the dataset, graph-construction constraints, less interpretable GAE embeddings, and overlapping cluster boundaries. These limitations indicate where validation on other mixers, periods, and parameter settings is most needed.

References

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. *International conference on database theory*, 420–434.

Ahmed Al-Kerboly, D. M., & Al-Kerboly, D. Z. F. (2024). A comparative study of clustering algorithms for profiling researchers in universities through google scholar. *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*, 1–5. https://doi.org/10.1109/ICMI60790.2024.10585953

Arbabi, A., Shojaeinasab, A., Bahrak, B., & Najjaran, H. (2023, October). Mixing Solutions in Bitcoin and Ethereum Ecosystems: A Review and Tutorial [arXiv:2310.04899]. https://doi.org/10.48550/arXiv.2310.04899

Ashari, I. F., Dwi Nugroho, E., Baraku, R., Novri Yanda, I., & Liwardana, R. (2023). Analysis of elbow, silhouette, davies-bouldin, calinski-harabasz, and rand-index evaluation on k-means algorithm for classifying flood-affected areas in jakarta. *Journal of Applied Informatics and Computing*, *7*(1), 95–103. https://doi.org/10.30871/jaic.v7i1.4947

Blau, B. M., Griffith, T. G., & Whitby, R. J. (2021). Inflation and bitcoin: A descriptive time-series analysis. *Economics Letters*, *203*, 109848. https://doi.org/10.1016/j.econlet.2021.109848

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., & Dougherty, E. R. (2007). Model-based evaluation of clustering validation measures. *Pattern Recognition*, *40*(3), 807–824. https://doi.org/10.1016/j.patcog.2006.06.026

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, *3*(1), 1–27. https://doi.org/10.1080/03610927408827101

Campbell-Verduyn, M. (2018). Bitcoin, crypto-coins, and global anti-money laundering governance. *Crime, Law and Social Change*, *69*(2), 283–305. https://doi.org/10.1007/s10611-017-9756-5

Chainalysis. (2024, February). *The 2024 Crypto Crime Report* (tech. rep.). https://www.chainalysis.com/wp-content/uploads/2024/06/the-2024-crypto-crime-report-release.pdf

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. https://doi.org/10.1145/2939672.2939785

Chordia, S. N., & Shinde, S. (2024). Using unsupervised learning to detect fraud in cryptocurrency. *2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*, 1–6. https://doi.org/10.1109/ICBDS61829.2024.10837066

Cramér, H. (1946). *Mathematical methods of statistics*. Princeton University Press.

Crawford, J., & Guan, Y. (2020). Knowing your Bitcoin Customer: Money Laundering in the Bitcoin Economy. *2020 13th International Conference on Systematic Approaches to Digital Forensic Engineering (SADFE)*, 38–45. https://doi.org/10.1109/SADFE51007.2020.00013

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(2), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

de Balthasar, T., & Hernandez-Castro, J. (2017). An Analysis of Bitcoin Laundry Services. In H. Lipmaa, A. Mitrokotsa, & R. Matulevičius (Eds.), *Secure IT Systems* (pp. 297–312). Springer International Publishing. https://doi.org/10.1007/978-3-319-70290-2_18

Directive (EU) 2018/843 of the European Parliament and of the Council of 30 May 2018 amending Directive (EU) 2015/849 on the prevention of the use of the financial system for the purposes of money laundering or terrorist financing, and amending Directives 2009/138/EC and 2013/36/EU (2018, June 19). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32018L0843

Du, H., Che, Z., Shen, M., Zhu, L., & Hu, J. (2024). Breaking the anonymity of Ethereum mixing services using graph feature learning. *IEEE Transactions on Information Forensics and Security*, *19*, 616–631. https://doi.org/10.1109/TIFS.2023.3326984

Emane, C. R. D., Song, S., Lee, H., & Yoo, J. (2024). Anomaly detection based on gcns and dbscan in a large-scale graph. *Electronics*, *13*(13), Article 2625. https://doi.org/10.3390/electronics13132625

Fung, G. (2001). A comprehensive overview of basic clustering algorithms.

Gaihre, A., Pandey, S., & Liu, H. (2019). Deanonymizing cryptocurrency with graph learning: The promises and challenges. *2019 IEEE Conference on Communications and Network Security (CNS)*, 1–3. https://doi.org/10.1109/CNS.2019.8802640

Goldsmith, D., Grauer, K., & Shmalo, Y. (2020). Analyzing hack subnetworks in the bitcoin transaction graph [Number: 1 Publisher: SpringerOpen]. *Applied Network Science*, *5*(1), 1–20. https://doi.org/10.1007/s41109-020-00261-7

Holt, T. J., Lee, J. R., & Griffith, E. (2023). An Assessment of Cryptomixing Services in Online Illicit Markets [Publisher: SAGE Publications Inc]. *Journal of Contemporary Criminal Justice*, *39*(2), 222–238. https://doi.org/10.1177/10439862231158004

Hong, Y., Kwon, H., Lee, J., & Hur, J. (2018). A Practical De-mixing Algorithm for Bitcoin Mixing Services. *Proceedings of the 2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts*, 15–20. https://doi.org/10.1145/3205230.3205234

Huang, D. Y., Aliapoulios, M. M., Li, V. G., Invernizzi, L., Bursztein, E., McRoberts, K., Levin, J., Levchenko, K., Snoeren, A. C., & McCoy, D. (2018). Tracking Ransomware End-to-end. *2018 IEEE Symposium on Security and Privacy (SP)*, 618–631. https://doi.org/10.1109/SP.2018.00047

Huang, Z., Huang, Y., Qian, P., Chen, J., & He, Q. (2023). Demystifying Bitcoin address behavior via graph neural networks. *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 1747–1760. https://doi.org/10.1109/ICDE55515.2023.00137

Hunt, E. L., & Reffert, S. (2021). Improving the open cluster census—i. comparison of clustering algorithms applied to gaia dr2 data. *Astronomy & Astrophysics*, *646*, A104. https://doi.org/10.1051/0004-6361/202039341

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in r*. Springer.

Kehinde, J., Ajayi, O. O., Adetayo, A., Obafemi, J. R., Akinrolabu, O. D., & Ebitigha, A. E. (2024). Machine learning model for detecting money laundering in bitcoin blockchain transactions. *Machine Learning*, *1*(1).

Kethineni, S., & Cao, Y. (2020). The Rise in Popularity of Cryptocurrency and Associated Criminal Activity [Publisher: SAGE Publications Inc]. *International Criminal Justice Review*, *30*(3), 325–344. https://doi.org/10.1177/1057567719827051

Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*. https://arxiv.org/abs/1611.07308

Koronaios, A., & Koloniari, G. (2025). Graph-based bitcoin fraud detection using variational graph autoencoders and supervised learning [8th International Conference on

Emerging Data and Industry (EDI40), April 22-24, 2025, Patras, Greece]. *Procedia Computer Science*, *257*, 817–825. https://doi.org/10.1016/j.procs.2025.03.105

Liu, M., & Dong, B. (2025). Analysis of Cryptocurrencies Mixing Services and Its Regulatory Mechanism [ISSN: 1865-0937]. *Blockchain, Metaverse and Trustworthy Systems*, 95–110. https://doi.org/10.1007/978-981-96-1414-1_7

Lo, W. W., Kulatilleke, G. K., Sarhan, M., et al. (2023). Inspection-L: Self-supervised gnn node embeddings for money laundering detection in bitcoin. *Applied Intelligence*, *53*, 19406–19417. https://doi.org/10.1007/s10489-023-04504-9

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, *1*, 281–297.

Mattke, J., Maier, C., Reis, L., & Weitzel, T. (2020). Bitcoin investment: A mixed methods study of investment motivations. *European Journal of Information Systems*, *30*(3), 261–285. https://doi.org/10.1080/0960085X.2020.1787109

McInnes, L., Healy, J., & Astels, S. (2017). Hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, *2*(11). https://doi.org/10.21105/joss.00205

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 13th ACM Internet Measurement Conference (IMC)*, 127–139. https://doi.org/10.1145/2504730.2504747

Miedema, F., Lubbertsen, K., Schrama, V., & Wegberg, R. v. (2023). Mixed Signals: Analyzing {Ground-Truth} Data on the Users and Economics of a Bitcoin Mixing Service, 751–768. Retrieved November 23, 2024, from https://www.usenix.org/conference/usenixsecurity23/presentation/miedema

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

Nan, L., & Tao, D. (2018). Bitcoin mixing detection using deep autoencoder, 280–287. https://doi.org/10.1109/DSC.2018.00047

Nazzari, M. (2023). From payday to payoff: Exploring the money laundering strategies of cybercriminals. *Trends in Organized Crime*. https://doi.org/10.1007/s12117-023-09505-1

Pakki, J., Shoshitaishvili, Y., Wang, R., Bao, T., & Doupé, A. (2021). Everything You Ever Wanted to Know About Bitcoin Mixers (But Were Afraid to Ask). In N. Borisov & C. Diaz (Eds.), *Financial Cryptography and Data Security* (pp. 117–146). Springer. https://doi.org/10.1007/978-3-662-64322-8_6

Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., & Zhang, C. (2018). Adversarially regularized graph autoencoder [arXiv:1802.04407]. *CoRR*, *abs/1802.04407*. http://arxiv.org/abs/1802.04407

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series 5*, *50*(302), 157–175. https://doi.org/10.1080/14786440009463897

Pecuchova, J., & Drlik, M. (2022). Identification of students with similar behavioural patterns using clustering techniques. In E. Smyrnova-Trybulska (Ed.), *E-learning in the transformation of education in digital society* (pp. 257–267, Vol. 14). University of Silesia in Katowice. https://doi.org/10.34916/el.2022.14.19

Powers, D. M. W. (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation. *Journal of Machine Learning Technologies*, *2*(1), 37–63.

Rosenquist, H., Hasselquist, D., Arlitt, M., & Carlsson, N. (2024). On the Dark Side of the Coin: Characterizing Bitcoin Use for Illicit Activities [ISSN: 1611-3349]. *Passive and Active Measurement*, 37–66. https://doi.org/10.1007/978-3-031-56252-5_3

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Rysin, V., & Rysin, M. (2020). The money laundering risk and regulatory challenges for cryptocurrency markets. In *Restructing Management. Models - Changes - Development.* (pp. 187–202). Retrieved December 5, 2024, from https://www.researchgate.net/publication/347564474_RESTRUCTURING_MANAGEMENT_MODELS_-_CHANGES_-_DEVELOPMENT

Satrya, R. N., Pratiwi, O. N., Fa'rifah, R. Y., & Abawajy, J. (2022). Cryptocurrency sentiment analysis on the twitter platform using support vector machine (svm) algorithm. *2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, 01–05. https://doi.org/10.1109/ICADEIS56544.2022.10037413

Sergio, I., & Wedemeier, J. (2025). Global surge: Exploring cryptocurrency adoption with evidence from spatial models. *Financial Innovation*, *11*(96). https://doi.org/10.1186/s40854-025-00765-0

Shah, R. S., Bhatia, A., Gandhi, A., & Mathur, S. (2021). Bitcoin data analytics: Scalable techniques for transaction clustering and embedding generation. *2021 International Conference on COMmunication Systems NETworkS (COMSNETS)*, 1–6. https://doi.org/10.1109/COMSNETS51098.2021.9352922

Shojaeinasab, A., Motamed, A. P., & Bahrak, B. (2023). Mixing detection on Bitcoin transactions using statistical patterns [_eprint: https://ietresearch.onlinelibrary.wiley.com/doi *IET Blockchain*, *3*(3), 136–148. https://doi.org/10.1049/blc2.12036

Silva Ramalho, D., & Igreja Matos, N. (2021). What we do in the (digital) shadows: Anti-money laundering regulation and a bitcoin-mixing criminal problem. *ERA Forum*, *22*(3), 487–506. https://doi.org/10.1007/s12027-021-00676-4

Sun, X., Yang, T., & Hu, B. (2022). LSTM-TC: Bitcoin coin mixing detection method with a high recall. *Applied Intelligence*, *52*(1), 780–793. https://doi.org/10.1007/s10489-021-02453-9

Tironsakkul, T., Maarek, M., Eross, A., & Just, M. (2020). Tracking Mixed Bitcoins [ISSN: 1611-3349]. *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, 447–457. https://doi.org/10.1007/978-3-030-66172-4_29

United States v. Larry Dean Harmon. https://www.justice.gov/opa/press-release/file/1230246/download

United States v. Sterlingov. https://law.justia.com/cases/federal/district-courts/district-of-columbia/dcdce/1:2021cr00399/231931/38/

van Wegberg, R., Oerlemans, J.-J., & Deventer, O. v. (2018). Bitcoin money laundering: Mixed results? An explorative study on money laundering of cybercrime proceeds using bitcoin [Publisher: Emerald Publishing Limited]. *Journal of Financial Crime*, *25*(2), 419–435. https://doi.org/10.1108/JFC-11-2016-0067

Vlahavas, G., Karasavvas, K., & Vakali, A. (2024). Unsupervised clustering of bitcoin transactions [Number: 1 Publisher: SpringerOpen]. *Financial Innovation*, *10*(1), 1–31. https://doi.org/10.1186/s40854-023-00525-y

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). *Data mining: Practical machine learning tools and techniques* [Copyright © 2017 Elsevier Inc. All rights reserved.]. Morgan Kaufmann. https://doi.org/10.1016/C2015-0-02071-8

Wu, L., Hu, Y., Zhou, Y., Wang, H., Luo, X., Wang, Z., Zhang, F., & Ren, K. (2021). Towards Understanding and Demystifying Bitcoin Mixing Services. *Proceedings of the Web Conference 2021*, 33–44. https://doi.org/10.1145/3442381.3449880

Ye, C., Li, Q., Gao, R., Fu, Y., Wang, P., Bao, X., Wang, G., Liu, Y., & Tian, Z. (2024). Detecting Mixing Services in Bitcoin Transactions Using Embedding Feature and Machine Learning [ISSN: 1865-0937]. *Network Simulation and Evaluation*, 91–105. https://doi.org/10.1007/978-981-97-4519-7_7

Yeo, I.-K., & Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, *87*(4), 954–959. https://doi.org/10.1093/biomet/87.4.954

A Research Design Appendix

A.1 Chainalysis Category Labels

ATM	Cash–crypto kiosks with tiered KYC and higher fees; fast but risk money laundering if KYC is weak.
Bridge	Protocols linking different blockchains for token and data transfers, either via trusted relays or trustless swaps.
Child abuse material	Hidden forums trading illegal child sexual content, typically on the dark web.
Darknet market	Marketplaces unreachable from the "normal" internet selling illicit goods (drugs, weapons, stolen data) using crypto, often with escrow and ratings.
Decentralised exchange	Smart-contract platforms for peer-to-peer token swaps without custody or intermediaries.
Exchange	Centralised sites for buying, selling, and trading crypto—the source of about 90% of on-chain volume.
Fraud shop	Single-vendor sites selling stolen personal information, cards, accounts; accept deposits only, making outflows traceable.
Gambling	Crypto betting (casino games, sports) with often lax KYC, posing laundering risks; common in permissive jurisdictions.
High risk exchange	Exchanges with no/weak KYC, AML convictions, or unusually high illicit-service exposure.
Hosted wallet	Custodial wallets hold users' keys; convenient but expose users to counterparty and security risks.
ICO	Token crowdfunding events; unregulated IPO analogs, often used for scams.
Illicit actor/org	Persons or groups engaged in illegal activities (darknet, hacking, extremist funding).
Infrastructure as a service	VPNs, VPS, domain registrars, etc.; funds may support bulletproof hosting or legit infrastructure.
Lending	Platforms (centralised or via smart contracts) for borrowing against collateral and earning interest.
Malware	Software (viruses, trojans, ransomware) designed to steal data, disrupt

	systems, or hijack resources.
Merchant services	Crypto payment processors converting to fiat; generally low risk but occasionally abused by scammers.
Mining	Computational process validating blocks and minting new coins.
Mining pools	Collective mining to share rewards; low risk unless they accept non-mining deposits.
Mixing	Services that shuffle funds to break transaction links, used for privacy or laundering.
NFT platform & collection	Marketplaces for unique digital assets; collections are themed token groups by creators.
Online pharmacy	Web vendors of prescription or research chemicals, sometimes without proper licensing.
Other	Miscellaneous entities (donations, bots, seized addresses) with variable risk.
P2P exchange	Peer-to-peer trading sites, often non-custodial and low-KYC, susceptible to laundering.
Protocol privacy	Native privacy features (zero-knowledge proofs, shielded pools) that hide transaction details.
Ransomware	Malware that encrypts data and demands crypto payment for decryption.
Sanctioned entity	Persons or organizations on official embargo lists; transactions with them are prohibited.
Sanctioned jurisdiction	Services based in fully sanctioned regions (e.g. Iran, North Korea, Cuba).
Scam	Fraudulent schemes impersonating legitimate services or promising unrealistic returns.
Seized funds	On-chain addresses holding assets confiscated by law enforcement.
Smart contract	Self-executing code on blockchain that enforces agreements without third parties.
Special Measures	Entities designated under FinCEN section 311 as primary money-laundering concerns, subject to restrictions.
Stolen funds	Crypto stolen in hacks or breaches, typically moved from compromised wallets.

Terrorist financing	Crypto used to finance designated terrorist groups and their operations.
Token smart contract	Contracts (e.g. ERC-20) that define, issue, and manage blockchain tokens.
Unnamed service	Unattributed clusters showing service-like transaction patterns, pending identification.
UTXO	Model where coins exist as discrete outputs that can be spent only once, underpinning Bitcoin-style ledgers.

A.2 Detailed Pre-processing

This appendix provides the pre-processing steps in more detail.

Transaction Log We start with 241,713 unverified data points. We do the following:

1. Remove all data points where `confirmed = False` (-312 data points).
2. Remove rows where `type = payment to yourself` or `amount = 0` (dusting transactions[8]). These transactions are definitely not associated with either incoming or outgoing transactions related to orders, so they are not relevant to the analysis (-131 data points).

Orders We start with 36,083 orders, and do the following:

1. Discard Bitcoin Cash and Litecoin orders because they are out of our scope (-985 data points).
2. Drop records where `order_id` is duplicated. This duplication appears to have been caused by a wiretap fault (-5,677 data points).
3. Exclude rows whose `status = Cancelled` indicating a cancelled order (-157 data points).
4. Cross-verify from transaction log: remove an order if its `deposit_address` is not present in the transaction log with a matching amount *and* Chainalysis can't report a transaction that matches the `deposit_amount` for the `deposit_address` of the

[8]In the case of Bestmixer, these transactions with near-0 values are likely advertisements, with a message attached to the transaction promoting Bestmixer.

order indicating the order never happened (-3,172 data points).

5. Correct entries where `deposit_amount` was recorded as 0 while the order was valid by retrieving the actual value from Chainalysis data (3,227 data points corrected).

A.3 Graph Modelling Cut-off

Figure 8 shows how deep our transaction graphs were able to grow under two different time windows: 30 days and 60 days after the deposit into Bestmixer. Each bar represents how many graphs reached a certain depth, and the colors indicate whether the graph includes a known Chainalysis-labeled service node (such as an exchange or darknet market) or not.

We began with a 30-day window, based on earlier analysis showing that most pre-mixing wallets receive any follow-up transactions within that period. After 30 days, activity tends to level off. Still, around 15% of wallets (2,000) received no follow-up transactions within that window.

Applying the 30-day limit, only about half of the graphs reached the structural depth we aimed for. To test whether the time limit was the main constraint, we extended the window to 60 days. This led to only a slight improvement—around 56% of graphs now met the requirement, indicating that time alone is not the main bottleneck.

Upon closer inspection, we found that many graphs stopped growing early because they reached a known service, which we deliberately filtered out to avoid noisy, high-volume nodes (by implementing the maximum of 189 inbound transactions). However, about 2,000 graphs stopped prematurely *without* hitting any known service node. Two possible reasons for this are: (1) the 60-day window might still be too short (although the earlier plateau makes this less likely), or (2) our limit of including only the 189 most recent incoming transactions might be cutting off useful links too early.

Looking at the figure:

- For depths 1 through 5, we actually see fewer graphs when using the 60-day window. This is likely because more graphs are able to grow deeper, which is positive.

- At depth 6 (the maximum allowed), we see a clear increase in the number of graphs, especially those including a service node. This suggests that the longer window helps more graphs reach full depth.

Even so, many graphs still stop at shallow depths. While early terminations from depth 2 onward can often be explained by hitting a known service, it is more puzzling that some graphs stop already at **depth 1**, and do not include any labeled service. The reason these graphs end so quickly is unclear; it could be due to actual user behavior, or limitations in our data or graph construction.

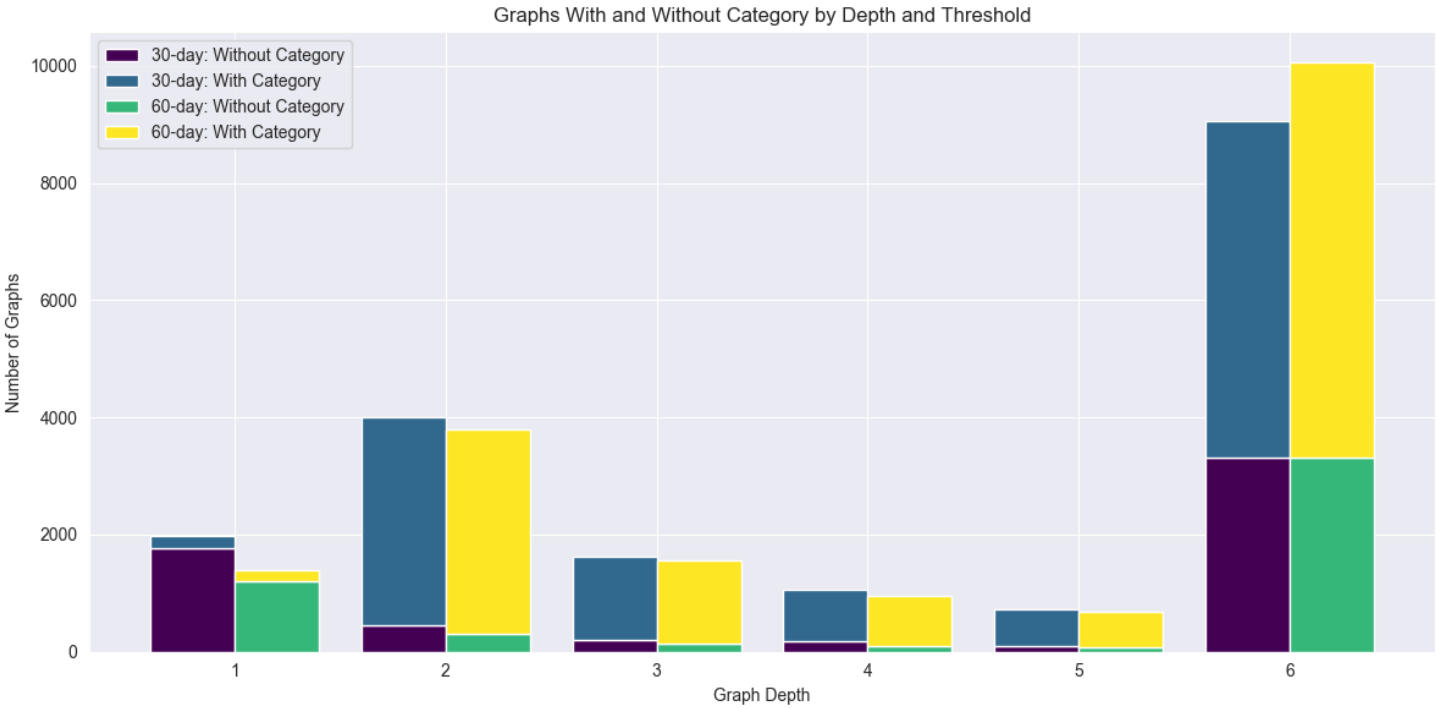

Figure 8: Stacked Bar Chart showing graph types by depth and threshold

B Results Appendix

B.1 Exploratory Data Analysis Figures

Figure 9 shows the distribution of category labels for pre-mixing wallets on a log scale. The data follows a long-tail distribution: a few categories dominate, with most wallets labelled as `missing` or associated with exchanges.

Figure 10 shows the same distribution for post-mixing wallets. The shape is similar, though there is a noticeably higher share of wallets labelled as `darknet market`.

Figure 11 displays ECDFs for the non-exposure attributes of pre-mixing wallets. The curves indicate a high level of sparsity: most wallets have low values for all variables, with a small number of high-value outliers. This pattern is particularly pronounced for the `missing` category.

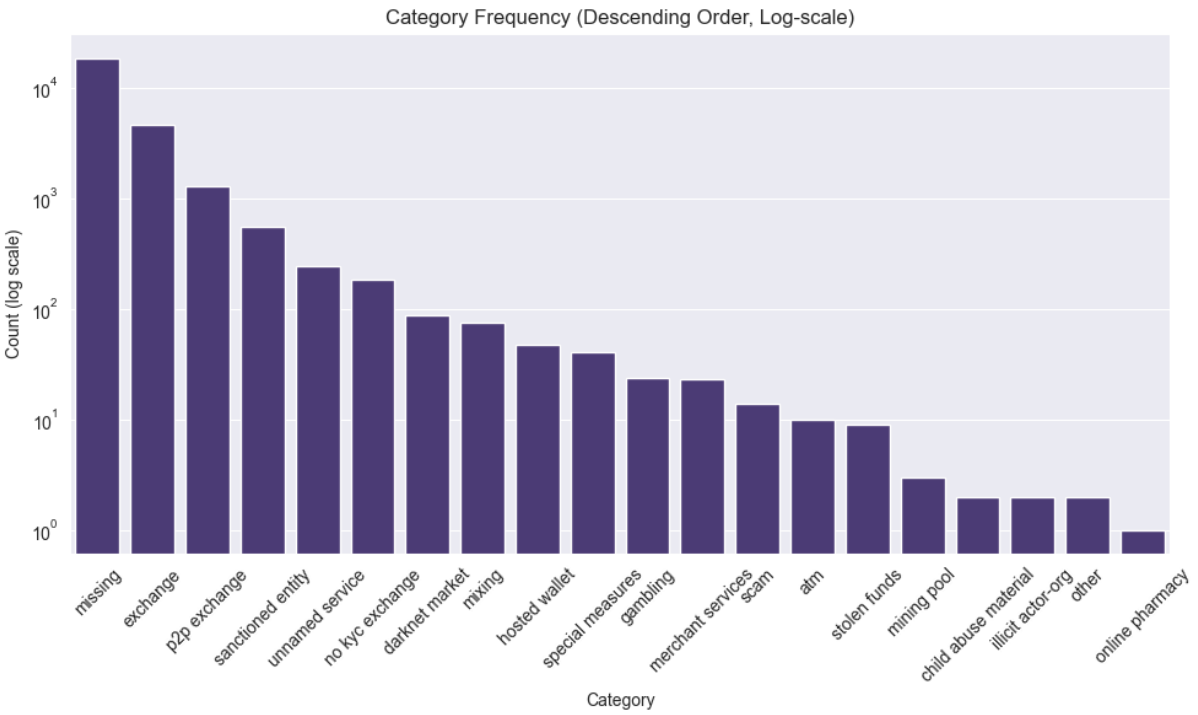

Figure 9: Distribution of pre-mixing categories (log-scale)

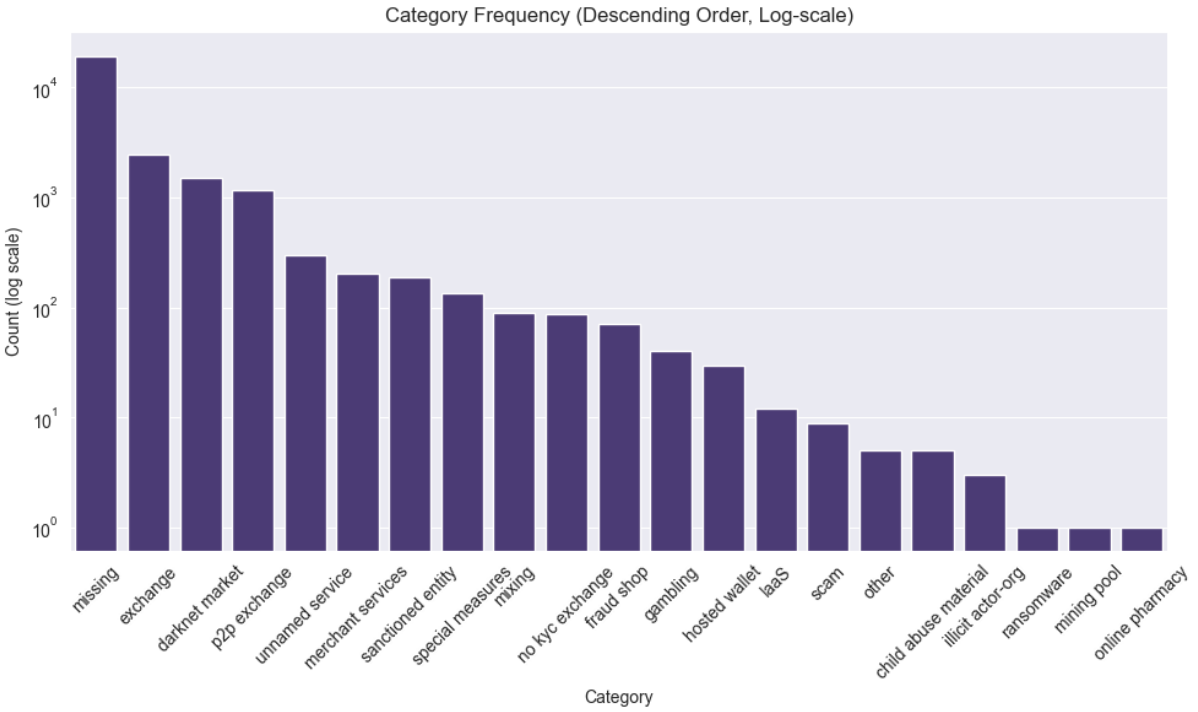

Figure 10: Distribution of post-mixing categories (log-scale)

Figure 12 shows the ECDFs for direct and indirect exposure attributes. The majority of wallets have very low exposure values, except for `exchange` and `Unknown`, which dominate both exposure types.

Figure 13 shows similar ECDFs for post-mixing wallets. These curves are even steeper

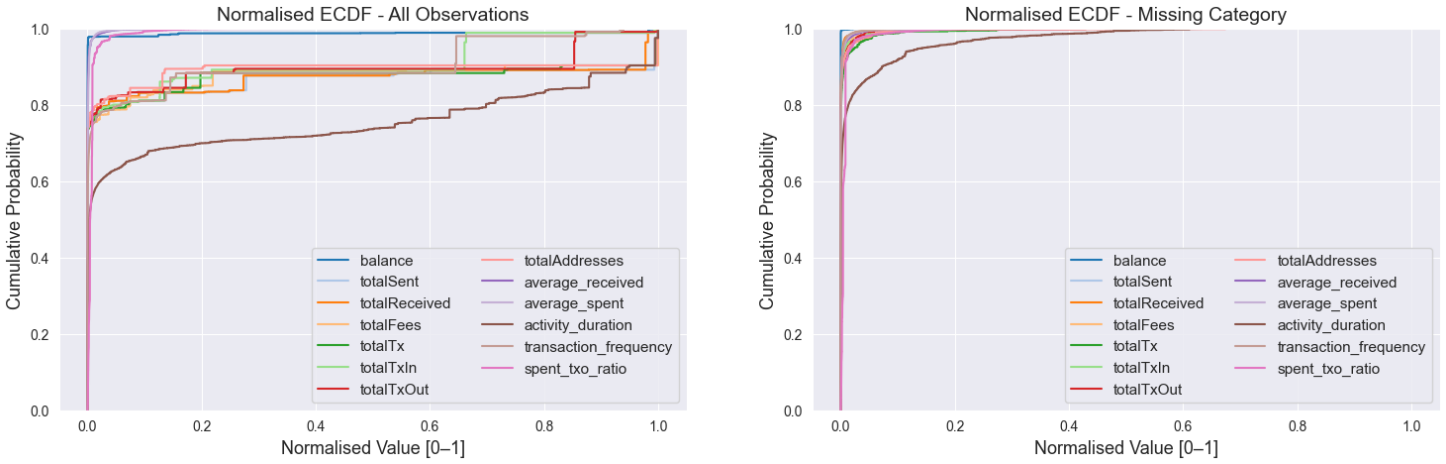

Figure 11: Normalised ECDF of pre-mixing wallet descriptives

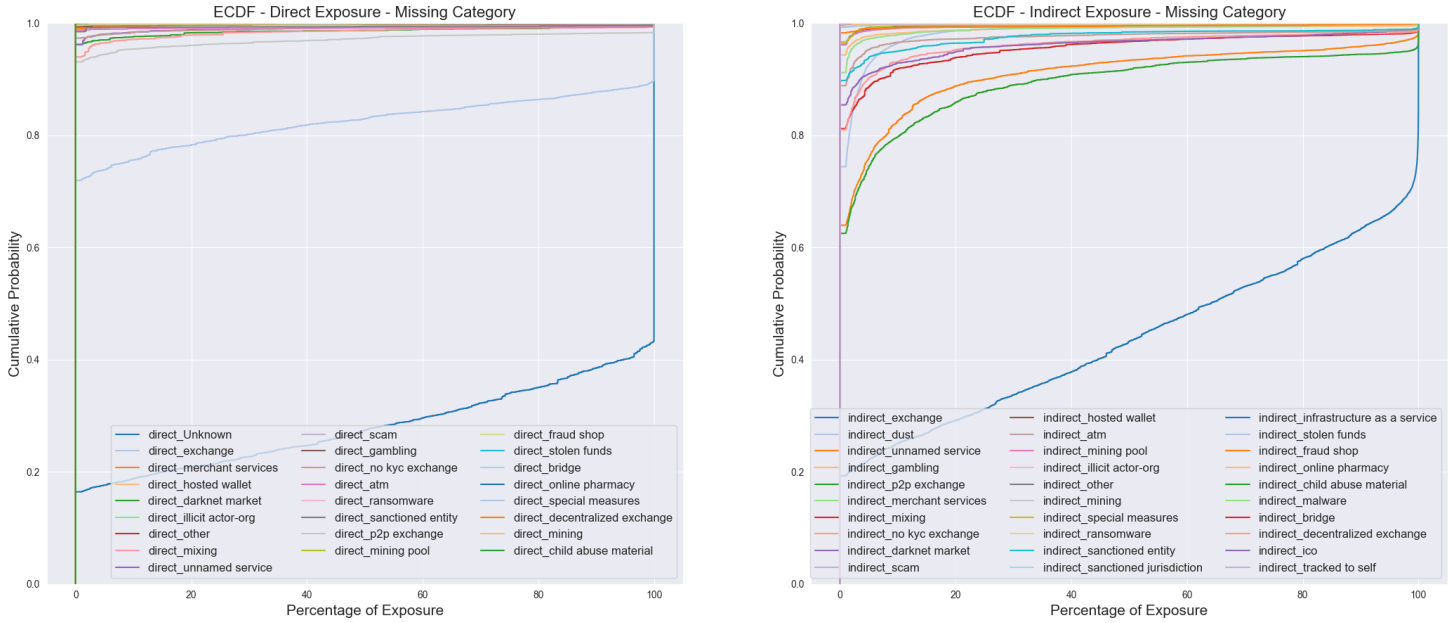

Figure 12: ECDF of pre-mixing wallet exposures

at the lower end, indicating stronger sparsity than in the pre-mixing set.

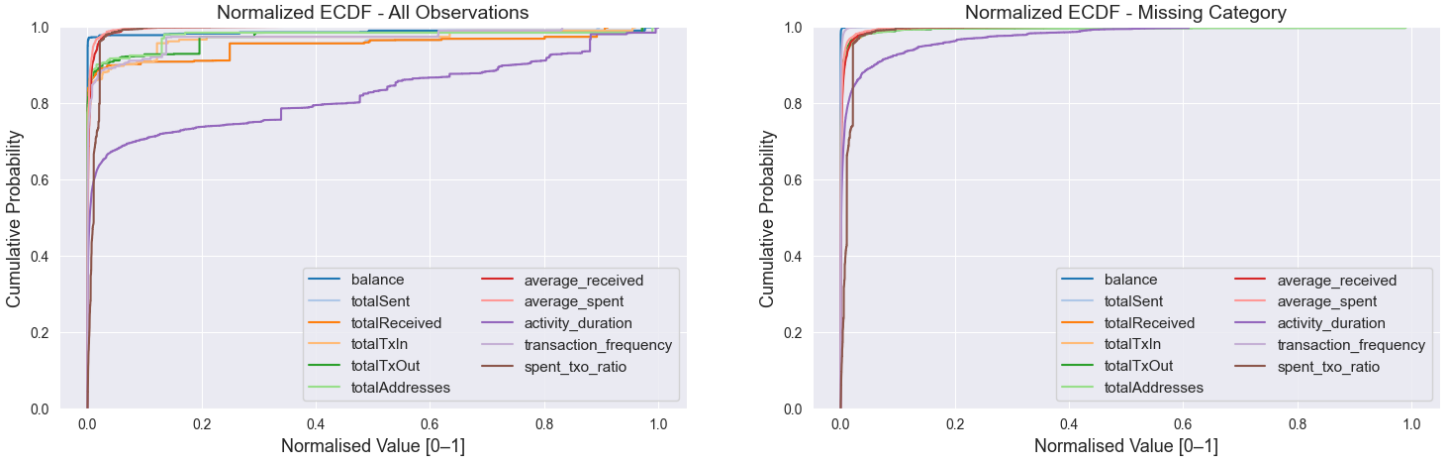

Figure 13: Normalised ECDF of post-mixing wallet descriptives

Figure 14 shows that post-mixing wallets exhibit a similar exposure pattern: most values are low, with concentration in `exchange` and `Unknown`, and slightly higher representation of `darknet market` in the indirect exposures.

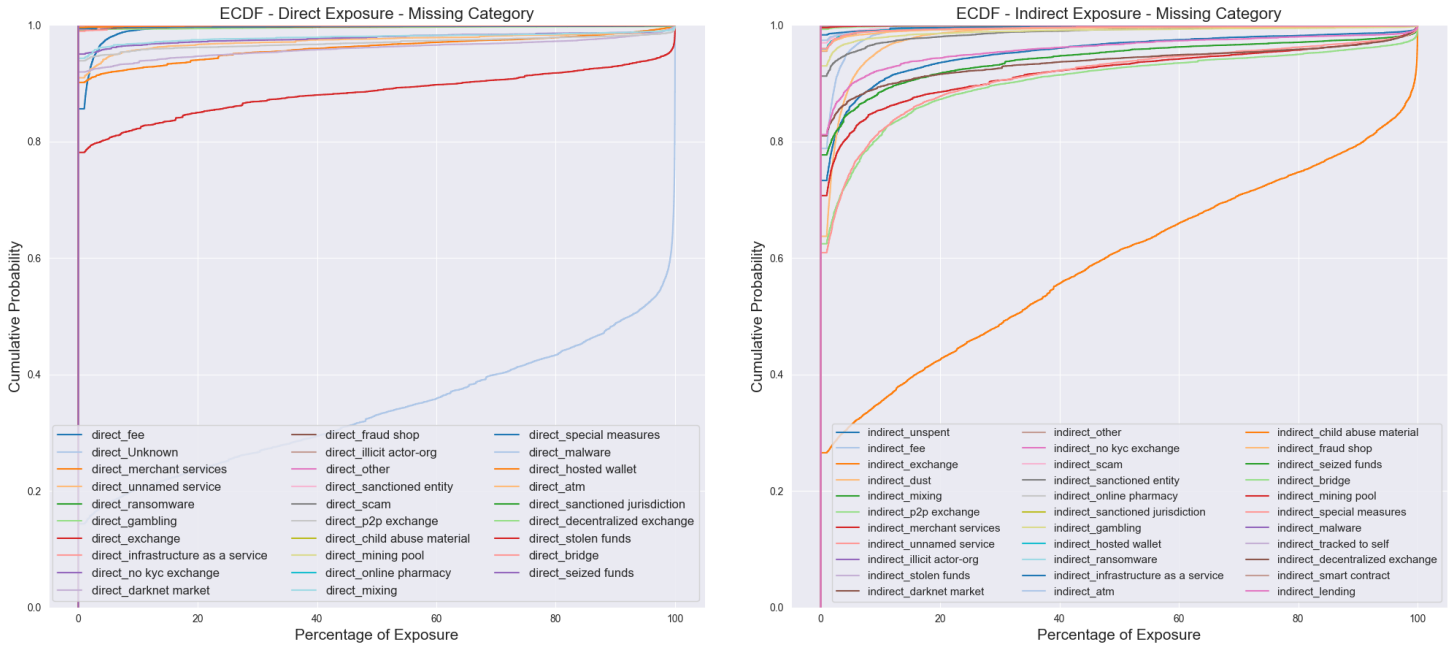

Figure 14: ECDF of post-mixing wallet exposures

B.2 SQ1: Exposure Data

B.2.1 Pre-Mixing Wallets

Analysing the exposure data confirms that high dimensionality lowers the effectiveness of the clustering algorithm. The top ranking parameters from our grid search were a `min_cluster_size` of 252 and `min_samples` of 502. This parameter combination yields the following evaluation scores:

- Silhouette Score: 0.905
- Davies-Bouldin Score: 0.310
- Calinski-Harabasz Score: 112,304.391

Although these scores are strong (a silhouette of 1 indicates perfect separation), Figure 15 reveals many outliers (Cluster -1).

The high prevalence of outliers makes it difficult to draw conclusions on the types of pre-mixing wallets, and could suggest that using exposure data is not suitable for clustering bitcoin wallets. Especially the fact that there are so few wallets from the `missing` category that were clustered hampers our ability to conclude anything on user

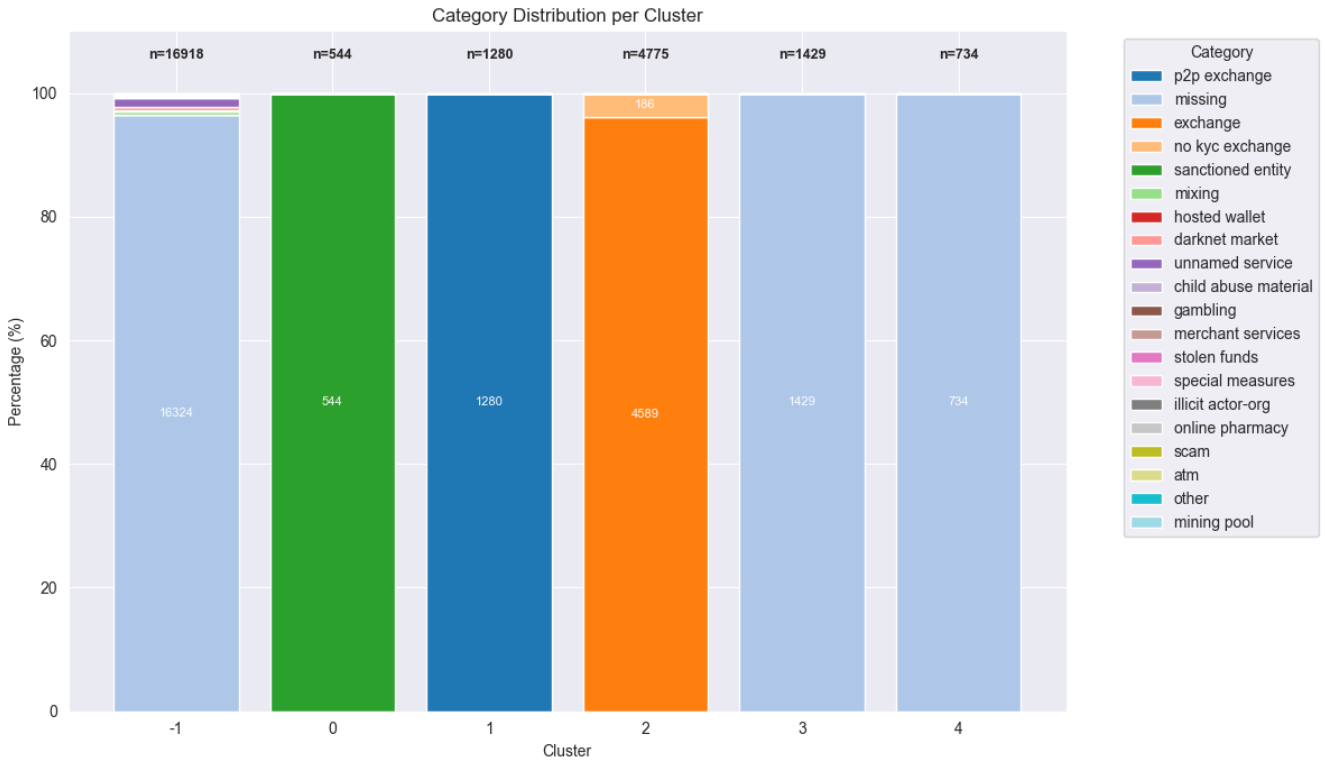

Figure 15: Stacked bar chart of pre-mixing exposure clusters

patterns, as these wallets have a high chance of belonging to an actual person instead of a service (because it would have been labelled as such by Chainalysis).

B.2.2 Post-Mixing wallets

Exposure data Analysing the exposure data for post-mixing wallets also confirms that exposure data might not be the best data to use when clustering bitcoin wallets. The top ranking parameters were a `min_cluster_size` of 2 and `min_samples` of 802. This parameter combination yields the following evaluation scores:

- Silhouette Score: 0.942
- Davies-Bouldin Score: 0.161
- Calinski-Harabasz Score: 46,637.791

Just like the pre-mixing wallets, these scores are very good (a silhouette of 1 indicates perfect separation). However, Figure 16 also reveals many outliers (Cluster -1).

The high prevalence of outliers gives the same conclusion as for pre-mixing wallets; that exposure data might not be suitable to use for answering our research question.

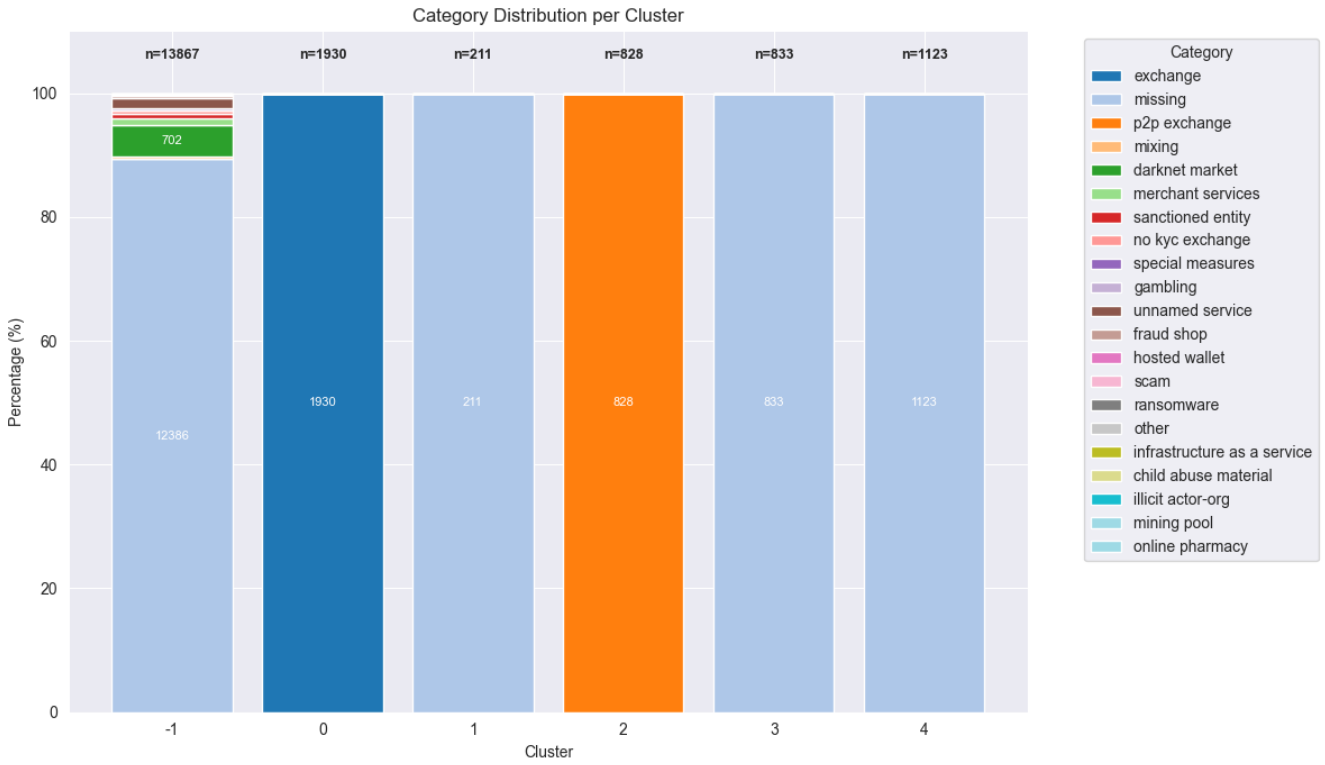

Figure 16: Stacked bar chart of post-mixing exposure clusters

B.3 Sub-question 1 Grid Search

The results of the grid search for sub-question 1 are shown in this section. The xaxis denotes the `min_cluster_size` parameter and the yaxis the `min_samples` parameter. The number inside the square denotes the ranking of the parameter combination. Darker numbers indicate a higher ranking.

B.3.1 Pre-Mixing Wallets

Both exposure and non-exposure wallets have good local optimums, as we can see in Figure 17. This is shown by the darker areas in the figure. While the non-exposure data shows some interesting behaviour from a `min_samples` of 852 and `min_cluster_size` of 352 onwards, there is also a local optimum at a `min_samples` of 602 and `min_cluster_size` of 552 and 602. The data with a `missing` label seems very difficult to cluster. This is likely due to the sparsity of the data.

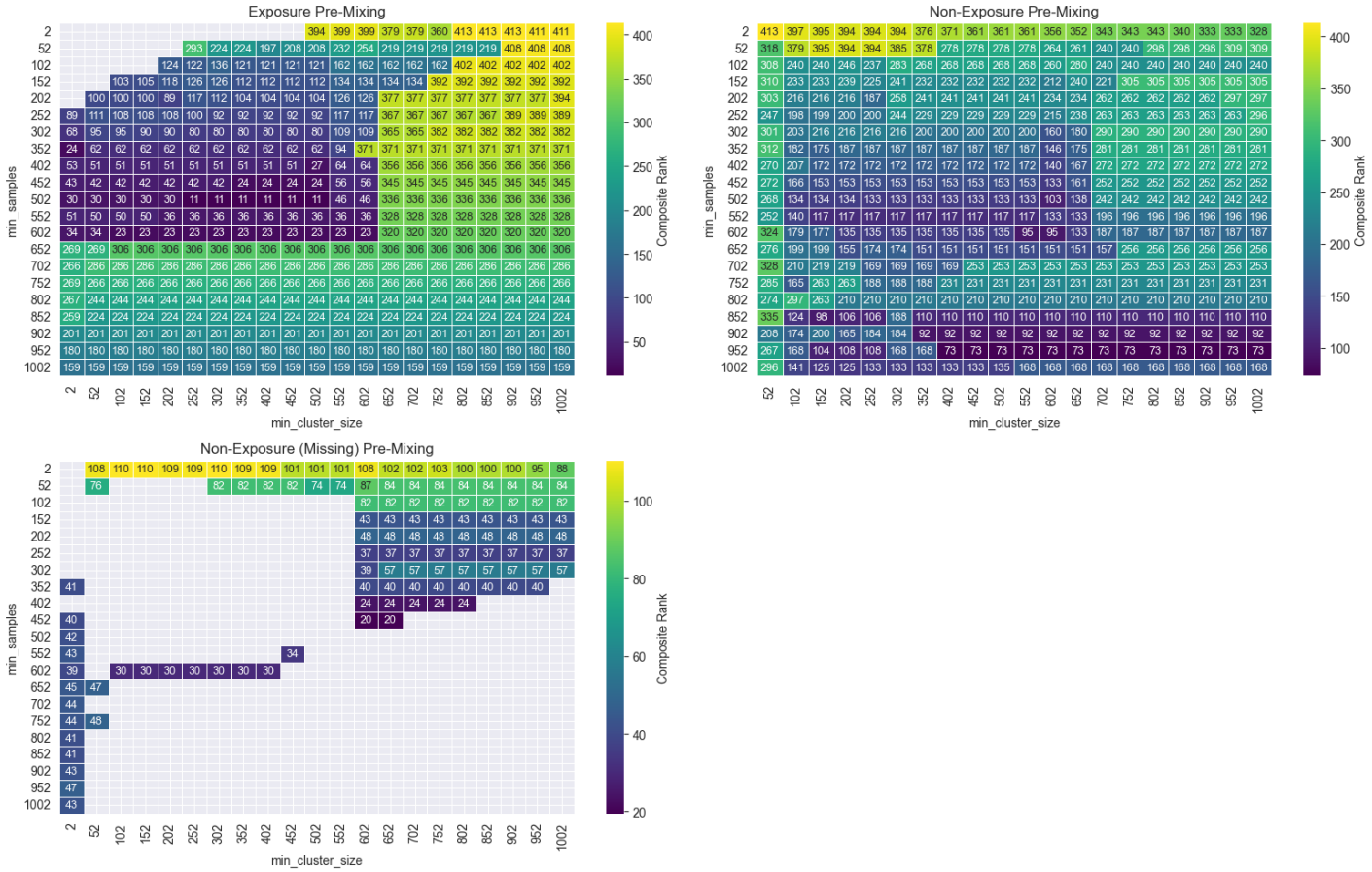

Figure 17: Grid search results of HDBSCAN algorithm on pre-mixing wallets

B.3.2 Post-Mixing Wallets

Figure 18 shows that the post-mixing wallets have a less obvious local optimum, and it has more difficulty with clustering the non-exposure data. Similar to the pre-mixing wallets, the wallets with a `missing` label are difficult to cluster.

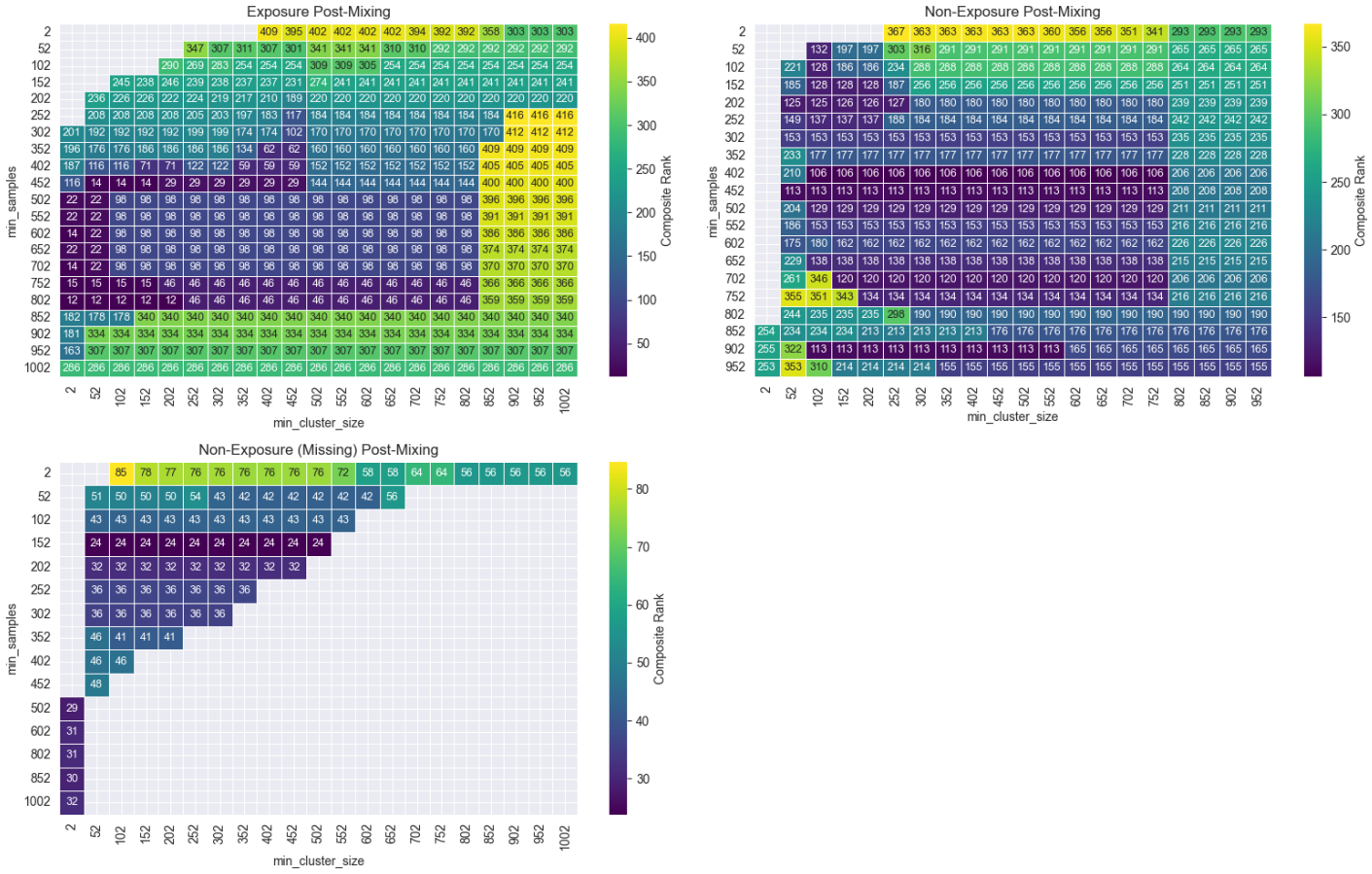

Figure 18: Grid search results of HDBSCAN algorithm on post-mixing wallets

B.4 Sub-Question 2 Grid Search

Both grid searches show the same results. Learning rate does not have an effect on the total loss of the model, and higher hidden and embedding dimensions result in a lower total loss. Higher hidden and embedding dimensions could result in an even lower loss.

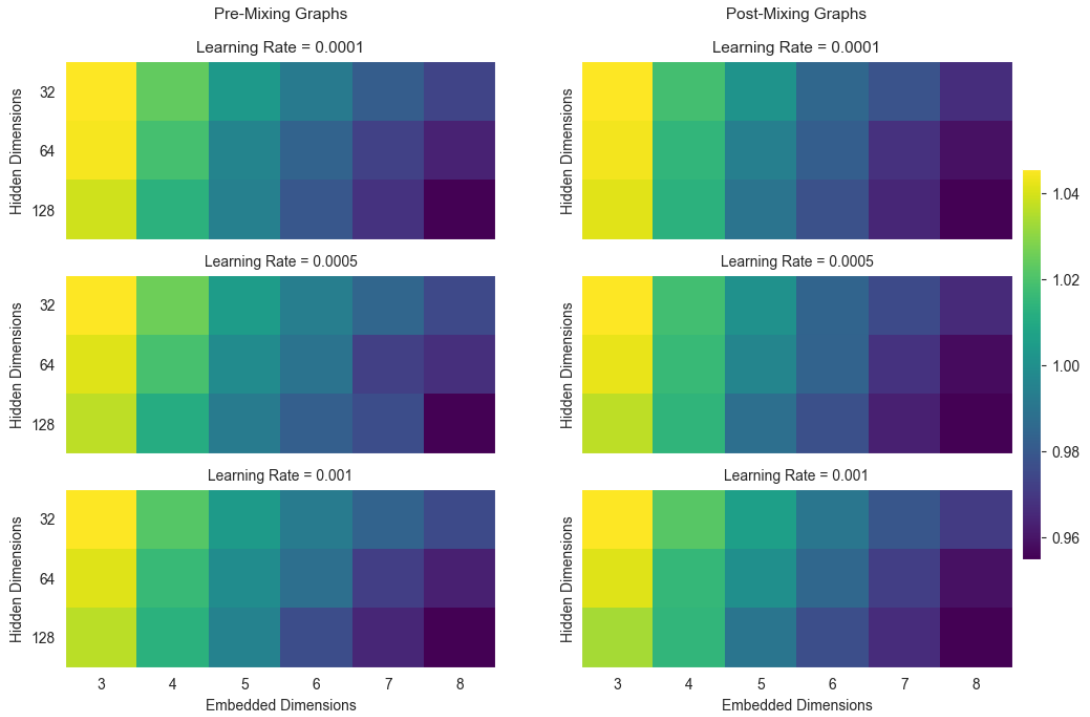

Figure 19: Grid search results of GAE on pre-mixing graphs

B.5 K-means Test

The figures below show the two elbow graphs for pre- and post-mixing wallets. Both graphs show 4 as the optimal number of clusters, though that is less clear for the post-mixing figure.

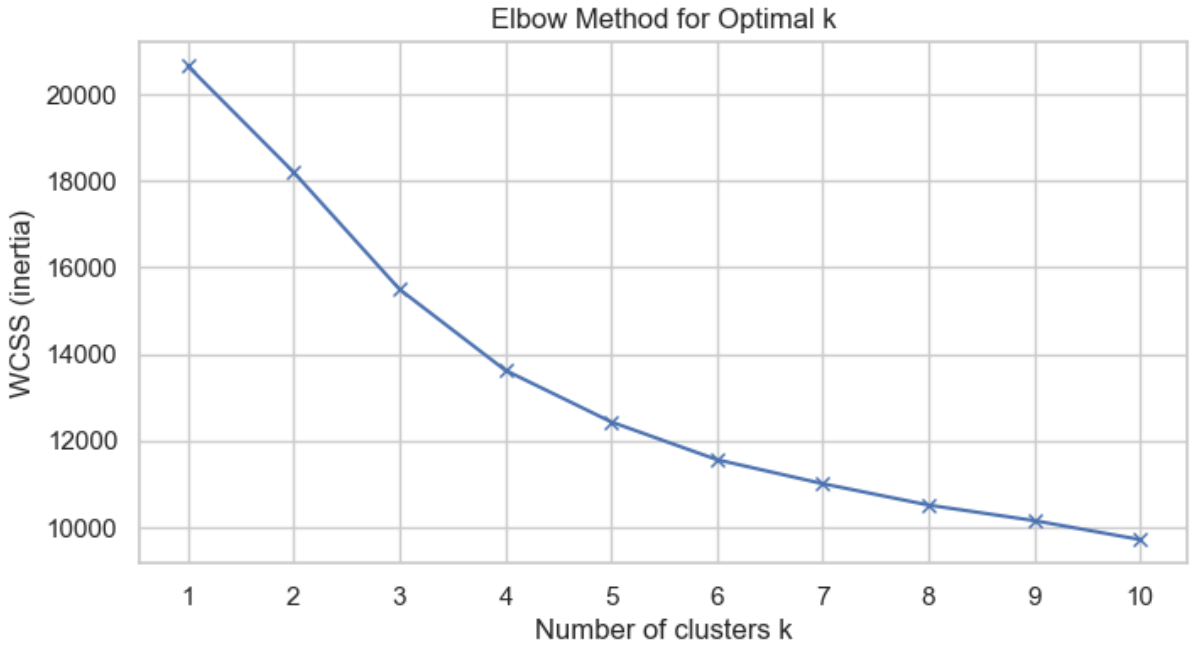

Figure 20: Elbow method for pre-mixing graphs

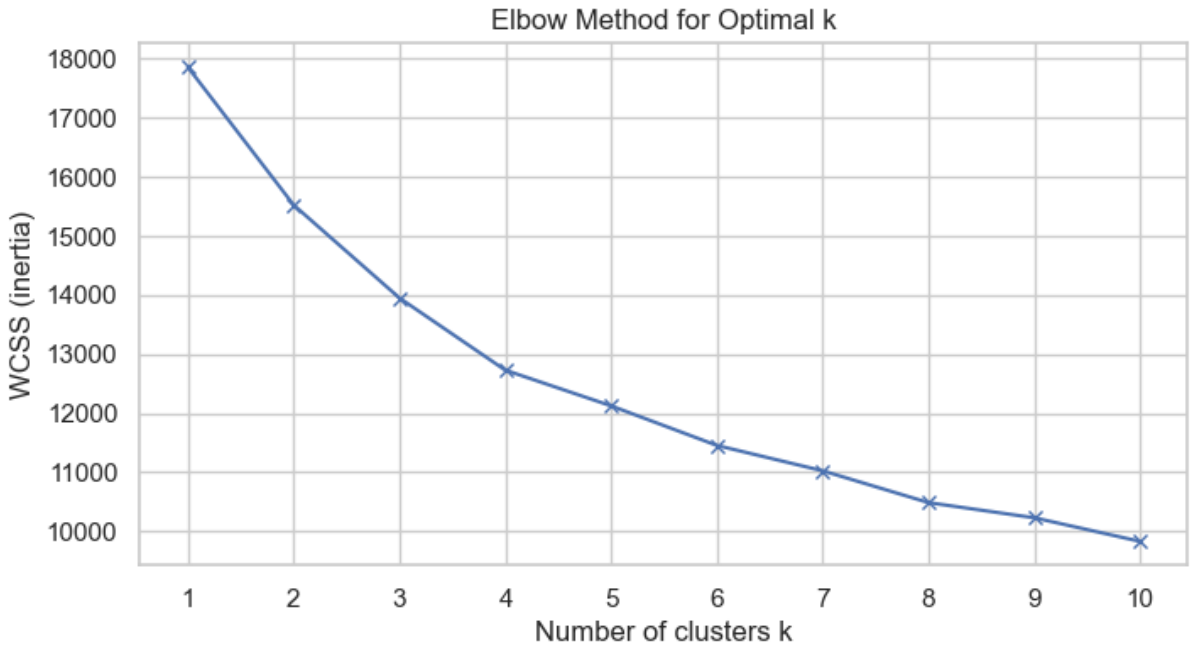

Figure 21: Elbow method for post-mixing graphs