A comparative analysis of user behaviour in DeFi applications

Dorottya Zelenyanszki[1][0009−0004−1300−0357], Zhé Hóu[1][0000−0001−7164−0580], Kamanashis Biswas[2][0000−0003−3719−8607], and Vallipuram Muthukkumarasamy[1][0000−0002−6787−6379]

[1] Griffith University, Australia
[2] Australian Catholic University, Australia

Abstract. The Decentralised Finance (DeFi) community is continuously being affected by various types of security incidents that are causing high levels of financial loss. There have been multiple tools proposed for security attack detection, however, these cannot be used for general audit in DeFi applications as they focus on specific attacks or areas. This work aims to present how behavioural clustering and analysis can be leveraged to present insights into common user and smart contract behaviour in DeFi applications. Preliminary results are presented for two commonly known applications, Sushiswap and Uniswap V3 that lay down the foundation for general DeFi security audit.

Keywords: blockchain · DeFi · user behaviour · transaction analysis.

1 Introduction

Decentralised Finance (DeFi) is a financial ecosystem that enables users to perform financial activities in a trustless and borderless manner due to the underlying blockchain [10]. It can offer various types of services including lending and token exchange [1]. As a result of its benefits, DeFi applications became very popular which can be seen by the 77% increase in Total Value Locked (TVL), reaching $103 billion, as reported by DappRadar's 2023 industry report[3]. The potential financial profit also prompted interest from malicious actors. These actors leverage various types of open issues such as vulnerabilities related to smart contracts and design to perform several kinds of attacks. For example, sandwich attacks or Pump-and-Dump (P&D) scams [7]. This caused a significant financial loss in the DeFi community. According to Chainalysis[4], in 2022 $3.7 billion was stolen. In 2023, there was a decrease to $1.7 billion, but the number of individual hacking incidents grew and the malicious actors became more sophisticated and diverse. As a result, the research community has an ongoing interest in providing tools for transaction audit and security protection.

[3] https://www.chainalysis.com/blog/crypto-hacking-stolen-funds-2024/
[4] https://www.chainalysis.com/blog/crypto-hacking-stolen-funds-2024/

For example, leveraging symbolic execution, fuzzing and formal verification to detect smart contract vulnerabilities [4].

Interest in blockchain transaction analysis also increased as it can be utilised for various types of purposes including cryptocurrency market analysis [15], NFT community detection [3] and blockchain forensics [17] where the latter focuses on identifying malicious transactions and the corresponding adversarial actors. It can also be utilised to describe the users' behaviour in decentralised applications (dApps) such as classifying Ethereum users based on past behaviour [2]. The extraction of behavioural information can be useful in the analysis of security-related incidents as well as deducing the participants' habits and social interactions can be leveraged to determine their future actions including potentially malicious activities.

Previous research generally only focused on specific attacks such as sandwich attacks [6] or token leaking vulnerabilities [11] when it comes to security analyses and did not consider establishing general behavioural clusters for both normal and abnormal behaviour. General behavioural analysis can potentially present general adversarial behaviour that is present across various types of applications as the habits and interactions of the malicious actors can be extracted and analysed. For example, actors who usually act together can be revealed if their extracted behaviour is highly similar. It can be also utilised to analyse the long-term behaviour of addresses which can present if they have had a tendency to perform malicious activities in the past or if they are often prone to fall for certain types of scams. Appropriate actions can be taken by the developers of the applications to prevent malicious actors from having access to the application and users who are common victims can be notified of suspicious activities. It can be leveraged to detect DeFi attacks that happened in the past as well. These attacks can be linked to the involved addresses that can help the developers and authorities in their investigations. Behavioural analysis overall can introduce long-term patterns that can be utilised for detection, similarly how malicious transaction patterns and behavioural patterns can be leveraged for blockchain forensics purposes [13]. This work aims to present a comparison of the extractable general behaviour of two well-known DeFi applications, Sushiswap and Uniswap V3 with a reported trading volume of \$6,411,043 and \$1,111,949,320 according to CoinGecko[5]. The results for both were analysed based on selected features.

For this purpose, transaction data from a certain period was collected for both applications and the emitted transaction events were analysed to introduce general DeFi actions that can describe the users' past activities. These were utilised in an unsupervised clustering process which resulted in general behavioural clusters of the user and smart contract addresses involved in the DeFi applications. The clusters were analysed and preliminary results for behaviour that is present in the applications were described.

Contributions of this paper are listed as follows:

- Formation of general DeFi actions that can describe past user behaviour in DeFi application.

[5] https://www.coingecko.com/

- General behavioural clusters that group addresses that correspond to similar behaviour that can be utilised to deduce adversarial actors' habits, connections and interactions.
- Comparison of two DeFi applications that was performed based on the behavioural cluster results.

The rest of the paper is structured as follows: Section 2 presents related research works that present the current DeFi scope and also describe current research tools for vulnerability and attack detection. Following that, in section 3 the comparison of the two DeFi applications is presented regarding extractable user behaviour including the data collection, action formation, clustering and analysis steps. Finally, the paper is concluded in section 4 where future directions are also mentioned.

2 Related works

This section presents the current state of the DeFi landscape concerning security-related incidents and it also presents various types of research works that propose tools to detect vulnerabilities and attacks. These works, similarly to this research, aim to propose solutions that can be utilised for security audit and protection.

Liu et al. [8] presented the need to research DeFi users' perceptions as current research is limited to specific attacks or smart contract vulnerabilities. The authors conducted a semi-structured interview with DeFi users and an online survey afterwards. They aimed to research users' security risk awareness, the adequacy of employed security practices, how victims respond to DeFi attacks and how they mitigate these risks. They identified profitability and decentralisation as the main reasons users would use DeFi services. They also added that some of these users often blindly believe in decentralisation. Furthermore, they deduced that users do not apply adequate security controls and are overconfident in the use of two-factor authentication (2FA). Surprisingly, they also discovered that prior accident experience does not change users' perceptions as financial motivation overthrows their concerns, and even experience and education seem to be insufficient in increasing security levels. A large number of the users also do not support introduced regulations because of the potential additional fees which further presents the importance of profit as a major motivator in using DeFi services. They declared that the users' overall behaviour resembles gambling, therefore, the DeFi sector needs a similar type of regulatory system to the one implemented in the traditional gambling industry. They recommended that the community has to collectible work on regulation and also mentioned that DAO can be a potential tool as it can provide membership certificates to verified DeFi services. They also endorsed education and the use of regular reminders as tools that can improve users' security control. For example, a reminder feature that prompts users to review their token approvals.

Wang et al. [14] presented a mixed-method study to quantify the impact of sandwich attacks and show the knowledge gap between user perception and the actual effect of these attacks. Their results showed that miners started actively participating with attackers in relay services and also presented that attackers always achieve maximum profit. In general, they deduced that sandwich attacks achieved an increased influence on DeFi users.

Green et al. [5] proposed a DeFi survival analysis approach to describe user behaviour in lending protocols. Survival analysis uses time-to-event data and has been previously applied to Centralised Finance (CeFi) however, it was stated that its application to DeFi is not straightforward. They focused on seven transaction types: deposits, redeems, borrows, repays, liquidations, interest-rate swaps and reserve collateral usage toggling. These transactions always have one specific coin involved which they categorised into stablecoins and non-stablecoins. They utilised Kaplan-Meier curves for visualisation and when it was also required to show how the variables affect time to the outcome event, Cox regression was used. They used the analysis to deduce the subsequent transaction after the index event transaction. This is almost extensively the deposit transaction as before depositing any currency into an account, there are no other possible actions a user can take. They also utilised the analysis to check how long it takes for borrows to be repaid or liquidated and whether the type of the coin influences liquidations.

Wu et al. [16] presented that existing detection tools cannot detect based on logic as that would require an understanding of DeFi semantics. Their work concentrated on two common price manipulation attacks: the first presents when an attacker forces a DeFi app to perform an unwanted trade inside a DEX pool by finding and exploiting a vulnerability in the dApp and the second refers to attacks when an attacker manipulates the price of the token by exploiting the vulnerable DeFi app's price mechanism that depends on real-time status as the attacker can manipulate said status by trading in the pool. They aimed to detect these attacks by the analysis of invocations between smart contracts and high-level DeFi semantics. However, they identified that there is a gap between raw transactions and high-level semantics. They, therefore, defined semantics for three basic DeFi actions (such as transfer, minting and burning) and five advanced DeFi actions (such as trade, depositing and withdrawal). The basics were identified from the raw transactions and the advanced ones all consist of basic actions. To form these, they collected raw transactions and constructed Cash Flow Trees (CFTs) from them. These include contract invocations, events and basic actions. By applying three operations, connection, insertion and combination, they were able to lift the trees' semantics to advanced actions. They also pre-defined and utilised patterns to detect the attacks. For evaluation, they implemented a prototype system called DeFiRanger which identified four root causes of the price manipulation attacks: access control, design compatibility, slippage check and price dependency.

Su et al. [11] focused on weaknesses that originate from functional bugs such as token leaking vulnerabilities. They proposed a tool called DeFiWarder that traces transactions and provides protection from token-leaking vulnerabilities. Their solution also constructs CFTs and based on those it tracks the handlers of calls and mines the roles of the related addresses which then enables the capture of the real relationship between users and the accurate token flows of the users and the DeFi apps. It also merges the flows of different types of tokens, calculates the corresponding return rates and utilises abnormal cases to reveal token-leaking vulnerabilities. They identified arithmetic weakness, access control, control-flow hijacks, improper token transfer and price manipulation as causes of token leaking-related issues.

The related literature presented the effect and deep impact of DeFi security incidents and that the user adoption lacks accurate precision and security practice. It showed that the users are not educated on appropriate protection techniques and they are also generally against additional regulations. However, the high number of DeFi security incidents presents the need for tools that can provide security audits by the detection of adversarial activities. Related literature presented multiple types of DeFi security tools that were proposed for this purpose. However, these proposals have some limitations that this work's behavioural clusters aim to address. For example, they do not always take all transaction and event types into account as general behaviour is not their focus. They only consider one specific attack type such as price manipulation or sandwich attacks and some of them only show limited results as survival analysis only presents the potential next step after a certain index transaction. They rely on certain standards (for example ERC20), pre-defined rules, thresholds and the use of the flows of financial funds to make deductions regarding malicious intent. Other types of information coming from various types of events can present additional insights into the behaviour of DeFi users. General behavioural clusters can incorporate that to provide a general audit that focuses on a broader scope of DeFi and is not limited to a specific set of security attacks.

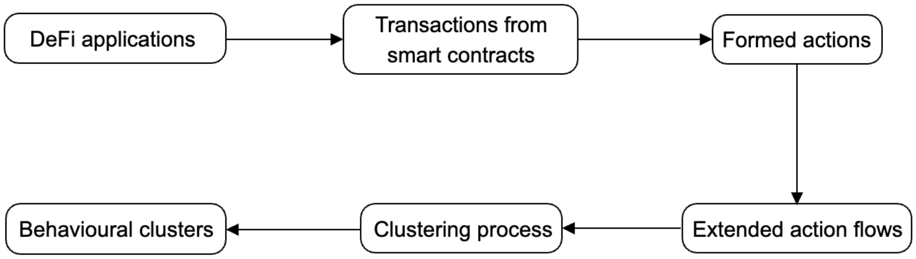


Fig. 1: Process to extract behavioural information.

3 User behaviour in DeFi

This section presents the extraction of user behaviour from the examined two DeFi applications. At first, it is explained how transaction data is collected from the related smart contracts. Following that, the way the actions are formed and utilised to present the activities of the addresses in the form of action flows is also described. Subsequently, the clustering process is presented and then finally, the resulting behavioural clusters are described by calculated features. The overall process to extract behavioural information is presented in Fig. 1.

Table 1: Properties of the action nodes.

Property	**Description**
Total count	total # times action is called
Min call	min # times action is called
Max call	max # times action is called
Mean call	avg # times action is called
Min timestamp	first time action is called
Max timestamp	last time action is called

3.1 Data collection and action formation

For the applications, transactions from smart contracts that are currently deployed[6][7] on Ethereum were collected that correspond to the period covered by block numbers 19993250 and 20207948. The Etherscan[8] and Alchemy[9] APIs were utilised to extract the transactions and decode the emitted events. The collected data then were added to a local Neo4j database so it could be queried to extract the necessary information to form actions and extract action flows. These flows present that in which order what kind of actions were performed by the corresponding address. The establishment of the local database followed the steps detailed in our previous work [18].

By querying the database, each included wallet addresses were extracted and categorised as whether they were smart contract or user-related. For Sushiswap 556 smart contract and 1800 user addresses were found, whereas for Uniswap V3 the ratio is 1211 smart contract and 2459 user addresses. However, not all of them actually initiated transactions within the examined period. By differentiating them and utilising behavioural clustering, both user and smart contract behaviour can be analysed separately. Apart from the wallet addresses, all token-related addresses were extracted as well.

Following this, the events that were emitted in transactions associated with these addresses were also retrieved. Every new unique set of events in transactions was added as a new DeFi action that can be performed by a wallet address.

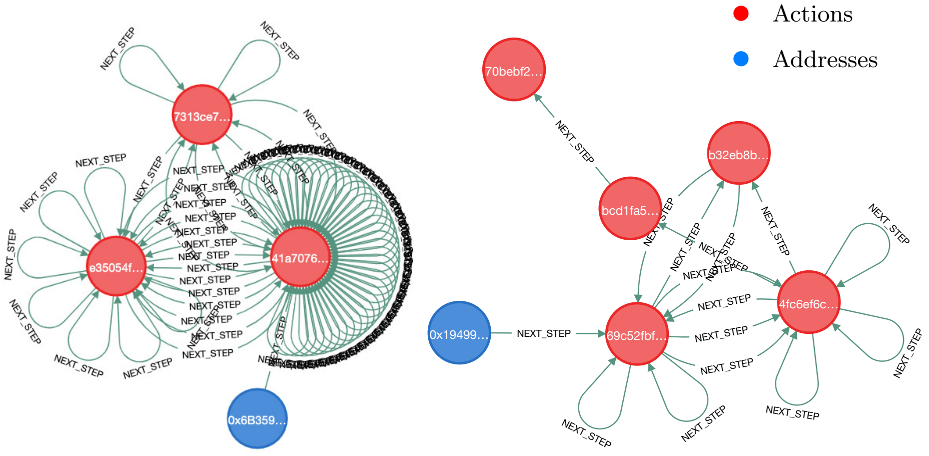
[6] https://docs.sushi.com/docs/Developers/Deployment%20Addresses

[7] https://docs.uniswap.org/contracts/v3/reference/deployments/ethereum-deployments

[8] https://docs.etherscan.io/

[9] https://docs.alchemy.com/reference/api-overview

This is a simplified action formation process as further analysis of DeFi semantics is needed to introduce more defined actions. Overall, 89 actions were added for Sushiswap and 128 of them were formed for Uniswap. Among these actions, 21 of them were present in both applications. Since all transactions per address can be extracted from the local database, they were also all added as an action step made by either a user or a smart contract address. Each step consists of the action ID, wallet address, whether the address is a contract and the corresponding timestamp. The same was also performed for token addresses as well, however, in this case, the step only includes the action ID, token address and timestamp.



(a) Smart contract action flow

(b) User action flow

Fig. 2: Action flows in Sushiswap.

After this, a second local database was established that includes three types of nodes: Address, token and action. For the action nodes numeric features were calculated that can be seen in Table 1. The address nodes have the wallet address and a boolean value that presents whether the address is a contract as properties. The token nodes have the token address and a boolean value that represents whether the token was used in multiple actions as property values. Two relationships were introduced between the nodes: i) NEXT_STEP which presents an action step performed by an address at a specific timestamp and it also covers the order of this step in the entire action flow that corresponds to that address; ii) USED_BY which represents the relation between an action and token that shows that this token was utilised in an action step that corresponds to that particular action. By filtering the NEXT_STEP relations through the address property, the corresponding action flows can be extracted. Through filtering, ac-

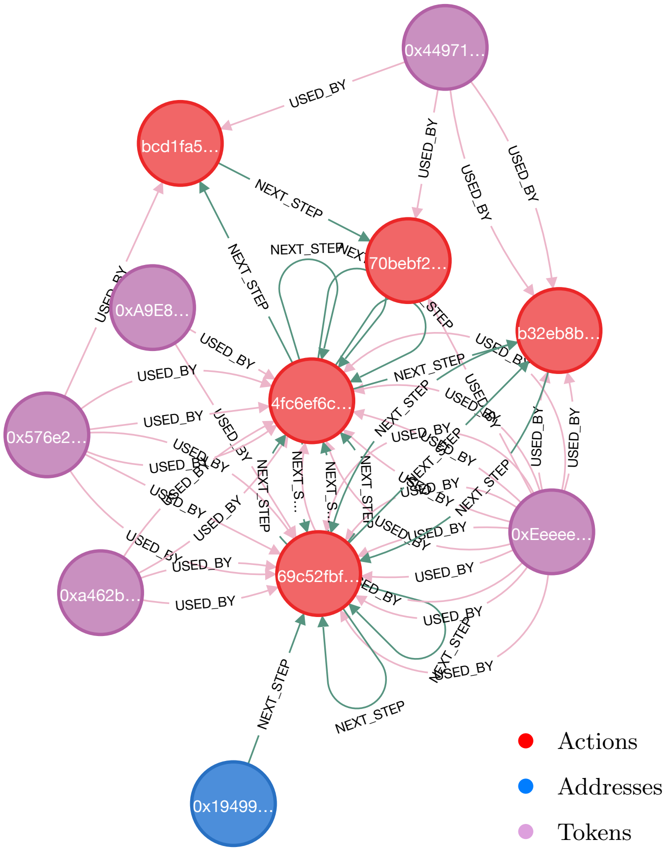
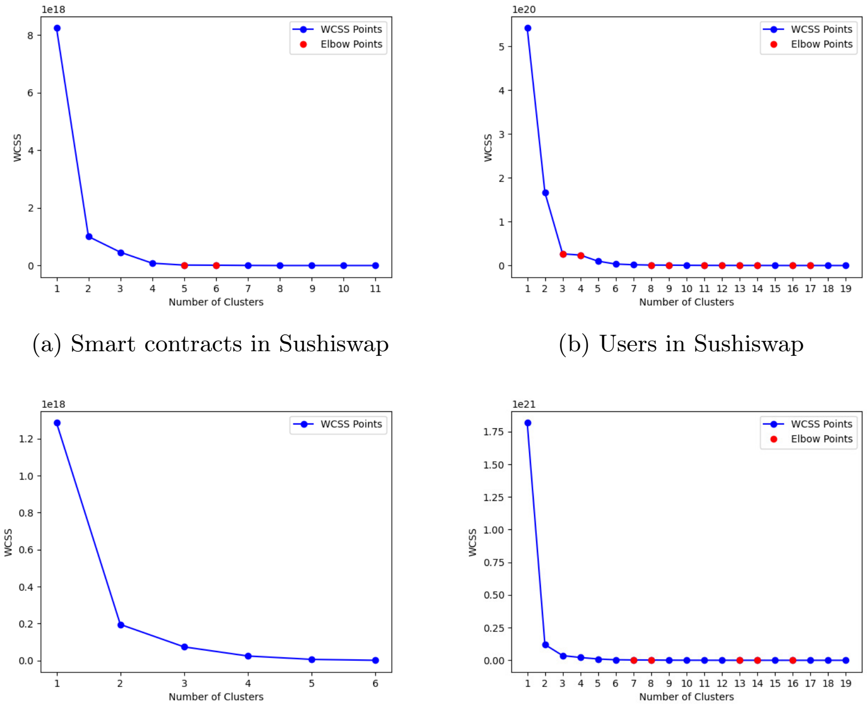


Fig. 3: An extended action flow.

tion flows for both user and contract addresses can be visualised. Fig. 2 presents an example for both user and contract address action flows in Sushiswap.

3.2 Behavioural clustering

As the token usage is also presented by the USED_BY relations, the action flows were extended by all USED_BY relations that had the particular address as a relation property. This way an action flow not only shows the order in which the actions performed but also present the kind of tokens the address interacted with. An example can be seen for this in Fig. 3. These action flows were utilised as input for a GNN model that provided graph-level embedding for them. Each embedding was added to two separate lists of embedding, one for the user and another for the smart-contract-related addresses. By separating them, user and smart contract behaviour can be examined separately. As there were no predefined labels that could be leveraged for the clustering process, the elbow method [12] was utilised to determine the required number of clusters for all four lists of embedding. The results for that are presented in Fig 4. Multiple elbow points were calculated that are visualised by red colour in the corresponding figures. There were no elbow points found for the contract addresses in Uniswap as presented in Fig 4c. This is the result of the fact that among many addresses extracted from event data, only a few of them initiated transactions within the examined period. As a result of this, the Uniswap contracts were excluded from the clustering. For the rest of them, the first identified elbow point was chosen as the value for the required number of clusters. However, for the Suhiswap contracts the addresses were only clustered into four clusters which probably is the result of the lower number of used contract addresses as well.



(a) Smart contracts in Sushiswap

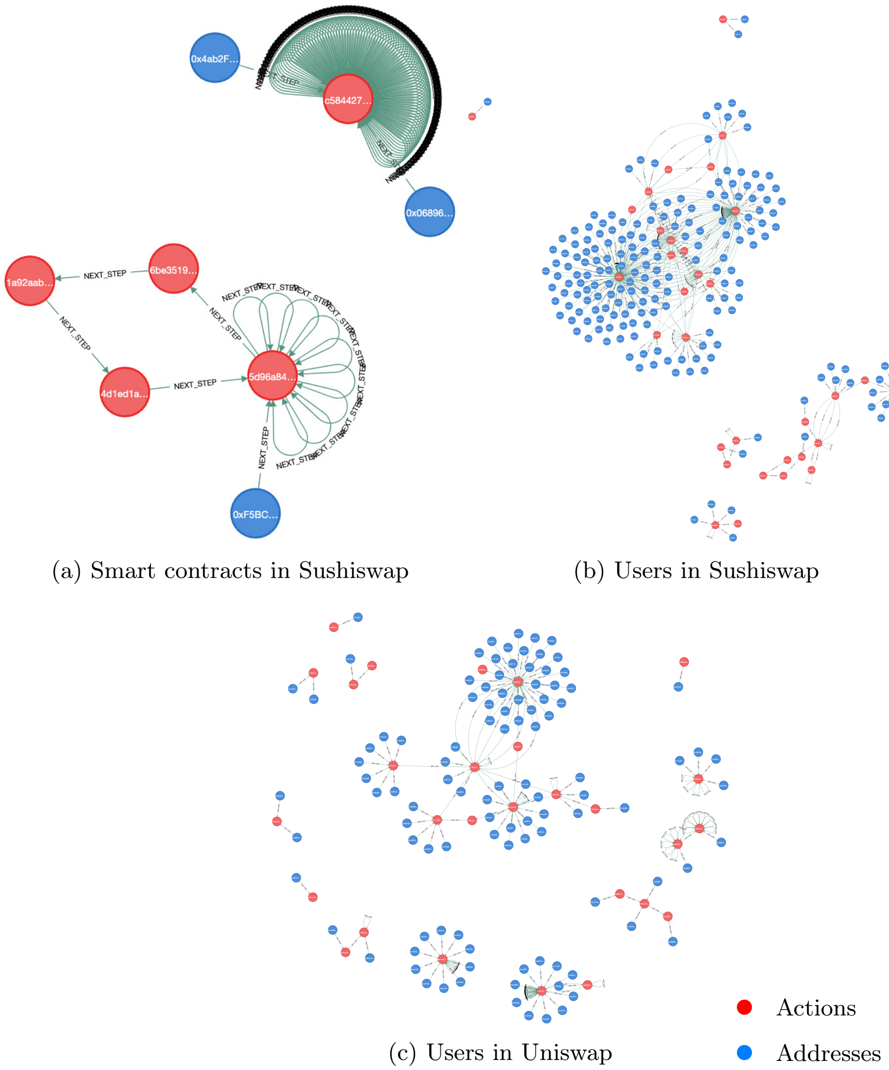
(b) Users in Sushiswap

(c) Smart contracts in Uniswap

(d) Users in Uniswap

Fig. 4: Results of the elbow method.

The k-means clustering [9] clustering algorithm was utilised to cluster addresses that performed similar types of activities into separate behavioural clusters that can be analysed later. We have to note that given the short period that was examined, many addresses only had one performed action which is not sufficient enough to determine their past behaviour, thereby, the results presented



(a) Smart contracts in Sushiswap

(b) Users in Sushiswap

(c) Users in Uniswap

Fig. 5: Cluster C_0 for all three embedding lists.

Table 2: Clusters for Sushiswap contracts.

#	**Number of addresses**	**Number of actions**	**General action flow length**	**Number of unique tokens**	**Time (hr)**
0	5	8	9	179	351.85
1	2	7	7	212	452.81
2	1	8	9	66	500.64
3	4	31	32	311	246.79

in this paper are only preliminary and further data collection is required to show behaviour on a wider scale. Fig. 5 presents examples (Cluster C_0 for all of them) for the created clusters where all included addresses and their NEXT_STEP relations are being visualised.

For the resulting behavioural clusters, multiple features were calculated as presented in Table 2, 3 and 4. The columns are explained: The number of addresses presents the addresses that belong to the particular cluster. The highest count action shows which action was performed the most by the addresses within that cluster. The number of actions highlights how many unique actions are performed in general by the included addresses, whereas the general action flow length presents how many actions were performed by them overall on average. The latter is based on a constructed general action flow which presents that at each possible action step which action will be performed most likely. The number of unique tokens shows how many unique tokens were included in the activities of each cluster and the highest count token presents which one was used the most. Time presents that on average how much time the addresses spent in the DeFi application within the examined period. It is presented in seconds. The highest count action and token columns were excluded from the Sushiswap tables as they presented the same values (action: 4fc6ef6c-8e83-41c3-bb7e-d81fbf184df4, token: 0xEeeeeEeeeEeEeeEeEeEeeEEEeeeeEeeeeeeeEEeE) for all clusters.

Table 3: Clusters for Sushiswap users.

#	Number of addresses	Number of actions	General action flow length	Number of unique tokens	Time (hr)
0	225	8	9	179	351.85
1	384	7	7	212	452.81
2	72	8	9	66	500.64

Table 4: Clusters for Uniswap users.

#	Number of addresses	Highest count action	Number of actions	General action flow length	Number of unique tokens	Highest count token	Time (hr)
0	407	b15b12be...	5	5	90	0x365Ac...	25.49
1	274	12b31f16...	4	4	102	0x365Ac...	1.17
2	44	12b31f16...	3	3	22	0x382ea...	3.48
3	19	12b31f16...	2	2	7	0x9e204...	0.07
4	55	4366058d...	6	6	52	0xD533a...	515.29
5	36	4693b2d2...	3	4	13	0x4e3FB...	509.90

3.3 Discussion

Based on the calculated features, preliminary behavioural information can be deduced for the addresses involved in these applications. In this section, we describe these results for both applications and also present some comparisons between them.

Sushiswap For this application, the created clusters for the contract and user addresses are basically the same. This can be a consequence of the lower number of used addresses that can be leveraged in the clustering process but can also show that there is a connection between the behaviour of user and contract addresses. It is generally also shown that even though the addresses' behaviour is separated as that is clearly shown in the differing general action flows, they still have similarities as the most popular actions and tokens are the same across all clusters.

Uniswap In this case, the clusters present very separate behaviours as the popular actions and tokens differ. However, the general action flows have many similarities as almost all of them start with the same action and some of them even have the first two action steps as the same. Information regarding users' activity levels can be also extracted. For example, users of Cluster C_4 can be generally viewed as active as they spent the longest time in the application within the examined period and performed the most actions on average, whereas Cluster C_3 can be described as inactive as the results present the opposite of Cluster C_4.

Comparison of the two DeFi applications As the smart contract addresses could only be clustered for Sushiswap, the comparison was only conducted on the user-related addresses. It can be generally said that the users in Sushiswap were more active as they spent more time performing a higher variety of actions that utilised more unique tokens. However, in Uniswap the users' behaviour varies more. It was also deduced that although there are commonly formed actions that are present in both applications, only one of them was present in the most popular actions performed by the users in the clusters. This means that the activities available are similar types of DeFi services but they also differ which suggests that DeFi applications have their unique characteristics.

As the collected data only covers a short time period, these preliminary behavioural results cannot be considered as long-term DeFi behaviour in the selected applications. Further data collection will present how often addresses perform similar types of activities that were presented in this section. The preliminary results are not definite, some of the addresses that are currently included may present false or insufficient results. As this is also a result of the small dataset, future data collection and the consequent behavioural clustering will be able to address it.

4 Conclusion

DeFi applications offer several types of services that are similar to traditional banking which resulted in a high volume of funds associated with them. This also prompted the interest of scammers and hackers who exploit vulnerabilities in DeFi applications to gain financial profit. As the participating users' security practice adaption is lacking, various types of tools have been introduced that can be utilised for a security audit. However, these focus on specific attacks and they do not consider all information that can be extracted from transactions.

This work presents preliminary results of general behavioural clustering of addresses involved in DeFi applications. Similar behavioural analysis can be utilised to provide a general audit for DeFi applications. In this work transactions from two applications were extracted that cover a short period. The results presented separate address behaviour, however, the small dataset consisting of these transactions cannot be utilised to present long-term address behaviour. Therefore, in the future, a larger dataset will have to be leveraged to perform behavioural clustering that presents more comprehensive descriptions for the involved addresses. This can address current insufficient results like the lack of analysis of smart contract behaviour for Uniswap V3. At this stage, only two DeFi applications were utilised for behavioural clustering, however, other applications also have to be considered to present behaviour that is consistently present across DeFi. Behavioural analysis on an extended dataset can introduce emerging behavioural patterns that can be utilised in machine learning techniques to classify and predict DeFi-related security attacks and eventually be utilised in real-time analysis. Apart from that, DeFi semantics have to be analysed to form realistic actions as the current actions were added through a simplified process.

References

1. Amler, H., Eckey, L., Faust, S., Kaiser, M., Sandner, P., Schlosser, B.: Defi-ning defi: Challenges & pathway. In: 2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS). pp. 181–184 (2021). https://doi.org/10.1109/BRAINS52497.2021.9569795
2. Bonifazi, G., Corradini, E., Ursino, D., Virgili, L.: Defining user spectra to classify ethereum users based on their behavior. Journal of Big Data **9** (04 2022). https://doi.org/10.1186/s40537-022-00586-3
3. Casale-Brunet, S., Ribeca, P., Doyle, P., Mattavelli, M.: Networks of ethereum non-fungible tokens: A graph-based analysis of the erc-721 ecosystem. In: 2021 IEEE International Conference on Blockchain (Blockchain). pp. 188–195. IEEE Computer Society, Los Alamitos, CA, USA (dec 2021). https://doi.org/10.1109/Blockchain53845.2021.00033, https://doi.ieeecomputersociety.org/10.1109/Blockchain53845.2021.00033
4. Chu, H., Zhang, P., Dong, H., Xiao, Y., Ji, S., Li, W.: A survey on smart contract vulnerabilities: Data sources, detection and repair. Information and Software Technology **159**, 107221 (2023)
5. Green, A., Cammilleri, C., Erickson, J.S., Seneviratne, O., Bennett, K.P.: Defi survival analysis: Insights into risks and user behaviors. In: Pardalos, P., Kotsireas, I., Guo, Y., Knottenbelt, W. (eds.) Mathematical Research for Blockchain Economy. pp. 127–141. Springer International Publishing, Cham (2023)
6. Heimbach, L., Wattenhofer, R.: Eliminating sandwich attacks with the help of game theory. In: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. p. 153–167. ASIA CCS '22, Association for Computing

Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3488932.3517390, https://doi.org/10.1145/3488932.3517390

7. Li, W., Bu, J., Li, X., Peng, H., Niu, Y., Zhang, Y.: A survey of defi security: Challenges and opportunities. Journal of King Saud University-Computer and Information Sciences **34**(10), 10378–10404 (2022)
8. Liu, M., Huh, J.H., Han, H., Lee, J., Ahn, J., Li, F., Kim, H., Kim, T.: I experienced more than 10 defi scams: On defi users' perception of security breaches and countermeasures (2024), https://arxiv.org/abs/2406.15709
9. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. Oakland, CA, USA (1967)
10. Meyer, E., Welpe, I.M., Sandner, P.G.: Decentralized finance—a systematic literature review and research directions. ECIS (2022)
11. Su, J., Lin, X., Fang, Z., Zhu, Z., Chen, J., Zheng, Z., Lv, W., Wang, J.: Defiwarder: Protecting defi apps from token leaking vulnerabilities. In: 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE). pp. 1664–1675 (2023). https://doi.org/10.1109/ASE56229.2023.00110
12. Thorndike, R.L.: Who belongs in the family? Psychometrika **18**(4), 267–276 (1953)
13. Ul Hassan, M., Rehmani, M.H., Chen, J.: Anomaly detection in blockchain networks: A comprehensive survey. IEEE Communications Surveys & Tutorials **25**(1), 289–318 (2023). https://doi.org/10.1109/COMST.2022.3205643
14. Wang, Y., Zuest, P., Yao, Y., Lu, Z., Wattenhofer, R.: Impact and user perception of sandwich attacks in the defi ecosystem. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. CHI '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3491102.3517585, https://doi.org/10.1145/3491102.3517585
15. Wu, J., Liu, J., Zhao, Y., Zheng, Z.: Analysis of cryptocurrency transactions from a network perspective: An overview. Journal of Network and Computer Applications **190**, 103139 (2021)
16. Wu, S., Yu, Z., Wang, D., Zhou, Y., Wu, L., Wang, H., Yuan, X.: Defiranger: Detecting defi price manipulation attacks. IEEE Transactions on Dependable and Secure Computing **21**(4), 4147–4161 (2024). https://doi.org/10.1109/TDSC.2023.3346888
17. Wu, Y., Tao, F., Liu, L., Gu, J., Panneerselvam, J., Zhu, R., Shahzad, M.N.: A bitcoin transaction network analytic method for future blockchain forensic investigation. IEEE Transactions on Network Science and Engineering **8**(2), 1230–1241 (2021). https://doi.org/10.1109/TNSE.2020.2970113
18. Zelenyanszki, D., Hóu, Z., Biswas, K., Muthukkumarasamy, V.: Linking nft transaction events to identify privacy risks. In: International Symposium on Distributed Ledger Technology. pp. 82–97. Springer (2023)