ISyE 6740 – Summer 2021
Final Report

Team Members: Tan Jun Xiong – Data engineering, report writing (903554328),
Lim Zi Hong – Modelling (903651169)

Project Title: Characterizing Ethereum wallet behavior via clustering

Problem Statement

Ethereum is a decentralized and open-source blockchain that has smart contract and decentralized application functionality courtesy of the Ethereum Virtual Machine. The ability to execute code stored on the blockchain in a trustless and permissionless manner has led to innovations in various areas/industries, and especially in the area of decentralized finance (DeFi) applications within the past couple of years. This ecosystem in particular has seen an explosion in terms of interest, growth and capitalization. These new protocols, by leveraging on Ethereum, aim to replace the backend processing of financial intermediaries and dramatically reduce the speed and costs of settlements. In the process, the programmatic and modular nature of DeFi smart contracts have also led to novel financial primitives.

Typically, end users interact with these protocols via their own Ethereum wallets. An aspect of wallets that projects are trying to incorporate into DeFi protocols and other projects is the characterization of wallet behavior based on transaction history. This includes things like how active a wallet is, kinds of smart contract functions called, participation in governance, alignment with projects' interests, and socially beneficial behavior (such as voluntarily returning funds that do not belong to a user). This can be a proxy for credit or behavioral scoring, something like a DeFi passport, whereby a history of desirable behavior can lead to benefits from protocols like lower interest rates and lower collateralized loans[1]. New projects may choose to distribute airdrop tokens to these addresses that signal long term investment horizons or active participation. Morally upstanding behavior can be recognized even by non-affiliated projects down the road. This has potential beyond just financial applications. In a Web 3.0 world, a wallet address is one's identity on the Internet (similar to Single Sign-On). Crypto founders have shown interest in potentially hiring individuals who can demonstrate they are a crypto "native" using their wallet as a pseudo-resume[2]. Licenses, physical purchases, collectibles, proprietary ownership, in-person experiences (tickets, memorabilia etc.) can be stored in a wallet as a non-fungible token as proof of a history that organizations can tap on wherever a user goes on the Internet.

For this project, we will examine the users of the Aave[3] protocol which provides a decentralized and permissionless protocol for the lending and borrowing of crypto assets. It is one of the foundational and more established projects with a large number of users and liquidity. We will narrow the focus to the version running on Ethereum, and look at users of the largest USDC (a

---

[1] https://arcx.game/
[2] https://atomic.blue/degenscore
[3] https://aave.com/

stablecoin) asset pool which, at the point of writing, stands at $5.6 billion. Using the transactions of these user wallets, we engineered features that attempt to describe on-chain behavior and cluster wallets with similar behavior together. This can potentially serve as a tool that projects can utilize to identify and reward wallets with desirable behavior. Work has been done by others to cluster and classify wallets, such as whether they belong to exchanges, miners, and investors etc., using certain features[4][5]. In our project, the features engineered were tailored towards clustering wallets based on behavior from interacting with a DeFi protocol, in the hope that it will serve as proxy for general behavior in the DeFi ecosystem. As such, features considered include:

- Frequency/activity of wallet interaction (power user)
- Value of assets locked (power user, trust in protocol)
- Value of assets transacted (power user)
- Participation in governance (shows interest and alignment)
- Staking/holding protocol or farmed tokens (long-term investor, alignment)
- Type of transaction (flash loans, liquidations) (advanced user)
- Interactions with other DeFi protocols (crypto native)

Data Source

The data used in this project is of course sourced from the blockchain itself. Aave and Sushi (another well-known DeFi project) documentation was consulted for smart contract addresses. Etherscan[6] provides a convenient user interface for downloads of processed data relating to token holders and transactions.

In DeFi, liquidity providers are considered premier participants as they provide the assets to make markets, while users provide the revenue that can keep the protocol economically sustainable. For this project, we narrowed the scope of focus to these users interacting with the USDC lending pool[7], as it is a foundational asset of a foundational DeFi protocol and we can view users interacting with this lending pool to be a proxy for general DeFi participation. aTokens[8] are interest-bearing tokens minted when users deposit assets into Aave lending pools, and allows token holders to claim back their assets from the pools together with interest. Debt tokens[9] are interest-accruing tokens representing the debt owed by token holders that are issued when assets are borrowed. We looked at the transactions relating to the interest-bearing and interest-accruing tokens of the USDC lending pool within the last 6 months. The current balance of the holders of the interest-bearing and interest-accruing tokens are also used.

There are two kinds of addresses on Ethereum – user wallet and contract addresses. Transactions executed by smart contract addresses were filtered out manually using Etherscan's labels as the distributed ledger does not label addresses as wallet or contract addresses. Smart contract addresses with significant value tend to be other projects pooling assets from a large number of users and interacting with the Aave protocol. It is possible to trace the origins of the assets in these contract addresses, however it is a non-trivial task and unnecessarily complicates the work.

[4] https://towardsdatascience.com/clustering-ethereum-addresses-18aeca61919d
[5] https://thomasdelatte.com/2021/04/clustering-ethereum-addresses/
[6] https://etherscan.io/
[7] https://docs.aave.com/developers/the-core-protocol/lendingpool
[8] https://aave.com/atokens
[9] https://docs.aave.com/developers/the-core-protocol/debt-tokens

Historical transactions from users interacting with the Aave governance smart contract[10], as well as the smart contract for liquidity mining incentives[11] are retrieved. The current balance of holders of stkAAVE[12], a token that represents users staking their AAVE tokens in the protocol, are also used.

Methodology

Since the blockchain only provides us with raw transaction data, some data processing and feature engineering is necessary for model building. Features were engineered to represent on-chain activity and wallet behavior. Altogether, 23 features were engineered for over 12000 addresses that had interacted with the Aave USDC lending pool in the past 6 months. They represented information such as:

- Number of deposit+withdrawal transactions in the past 30/90/180 days
- Number of borrowing+repayment transactions in the past 30/90/180 days
- Total number of USDC deposit transactions
- Total number of USDC withdrawal transactions
- Total number of USDC borrow transactions
- Total number of USDC repayment transactions
- aUSDC balance
- Debt balance
- Average size of each deposit/withdrawal transaction
- Average size of each borrow/repayment transaction
- Flash loan transaction
- stkAAVE, xSUSHI balance
- User is staking AAVE, SUSHI, voting in governance
- Number of votes submitted in AAVE governance
- Rewards from liquidity mining
- Number of reward claiming transactions

Since we are interested in characterizing the behavior of wallets by the features engineered, we will not be mapping this high-dimensional dataset to their principal components in order to reduce the dimensionality. Given the number of features, visualization of the distribution of the data is tricky. However, a heatmap of the correlation between the features was plotted (Figure 1). As it turns out, there was no variability in the columns relating to AAVE governance transaction data upon joining different data tables together to get feature information for the filtered addresses. This is likely due to data collection error while parsing the transactions for the relevant smart contract, as it is not probable that active user addresses within the past 6 months did not participate in AAVE governance. Features that are highly correlated with other features (>0.8) was also removed to reduce dimensionality. Features removed include whether a user is holding stkAAVE/xSUSHI, number of borrowing+repayment transactions in the past 180 days, number of deposit+withdrawal transactions in the past 180 days, and total number of repayment transactions. The remaining 16 features were then scaled using a minmax scaler.

[10] https://docs.aave.com/developers/protocol-governance/governance
[11] https://docs.aave.com/developers/guides/liquidity-mining
[12] https://docs.aave.com/developers/protocol-governance/staking-aave

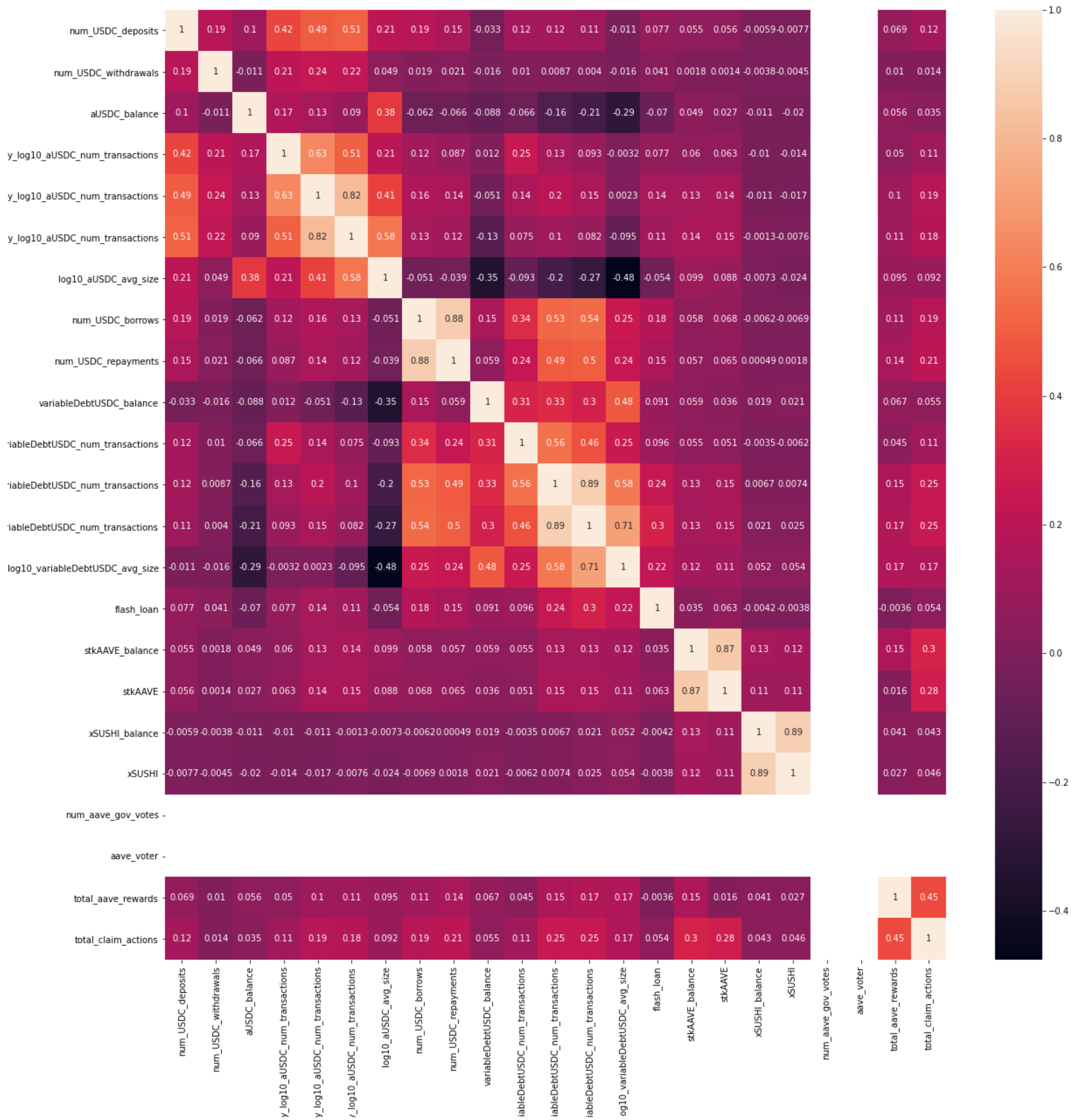

Figure 1- Correlation between features

Various algorithms that are computationally scalable with the number of samples in the dataset and can handle uneven cluster sizes were chosen to cluster the wallets. Some properties of the algorithms tried are listed as follows:

1) Density-Based Spatial Clustering (DBSCAN)
    a. No need for pre-set number of clusters
    b. Suitable for non-flat geometry
    c. Identifies outliers as noise
    d. Does not perform well when clusters are of varying density

2) Agglomerative Clustering
    a. Non-Euclidean distances

3) BIRCH
    a. Memory-efficient

Since this is an unsupervised learning problem, there isn't a definitively objective metric to be used for comparison between the various models. One possible evaluation can be performed using the silhouette coefficient, where a higher score relates to better defined clusters (the closer to 1 the better). It can be calculated by taking the average of the coefficient of each sample in a set, where each coefficient is calculated using

$$s = \frac{b - a}{max(a, b)}$$

where a is the mean distance between a sample and all other points in the same class and b is the mean distance between a sample and all other points in the next nearest cluster

The DBSCAN algorithm requires the hyperparameter epsilon which defines the maximum distance between two points. An approach described in this paper[13] (see footnote) is used to automatically determine the optimal value by calculating the distance to the nearest n points for each point, and then sorting and plotting the results such that epsilon can be determined from the point of maximum curvature. For agglomerative clustering and BIRCH, the number of clusters were tuned to maximize the silhouette score.

Evaluation & Final Results

The silhouette scores for each model is shown in Table 1.

Table 1 - Silhouette scores for clustering models

	DBSCAN	**Agglomerative clustering**	**BIRCH**
N clusters	3	3	3
Silhouette score	0.505	0.408	0.516

[13] https://iopscience.iop.org/article/10.1088/1755-1315/31/1/012012/pdf

Table 2 shows the number of datapoints in each cluster for each model.

Table 2 - Number of datapoints in each cluster for each model

N datapoints	DBSCAN	Agglomerative clustering	BIRCH
Cluster 0	11070	7803	11171
Cluster 1	1337	3390	1387
Cluster 2	15	1437	72

As shown in Table 2, the cluster sizes can be quite uneven, that's why these particular clustering algorithms were chosen in the first place. Uneven cluster sizes are expected since reality tends to follow a power law distribution in which a small number of users have more capital and are more active than most of the population. They are more industrial in their operations than the typical casual user and tend to have specialized skills in trading and smart contract engineering.

In order to better understand the characteristics of each cluster, the mean value of each feature for each cluster is plotted using polar charts that represent the data along the radial and angular axes. This will be done for the DBSCAN and BIRCH models which have higher silhouette scores.

<u>*DBSCAN*</u>

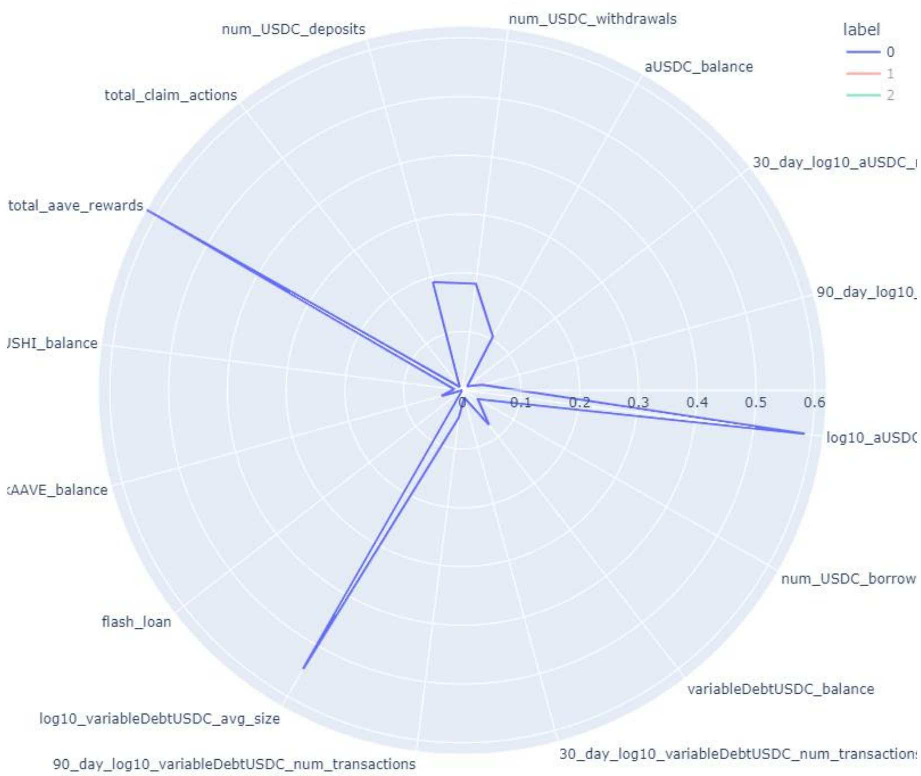

Figure 2 - DBSCAN cluster 0 polar chart

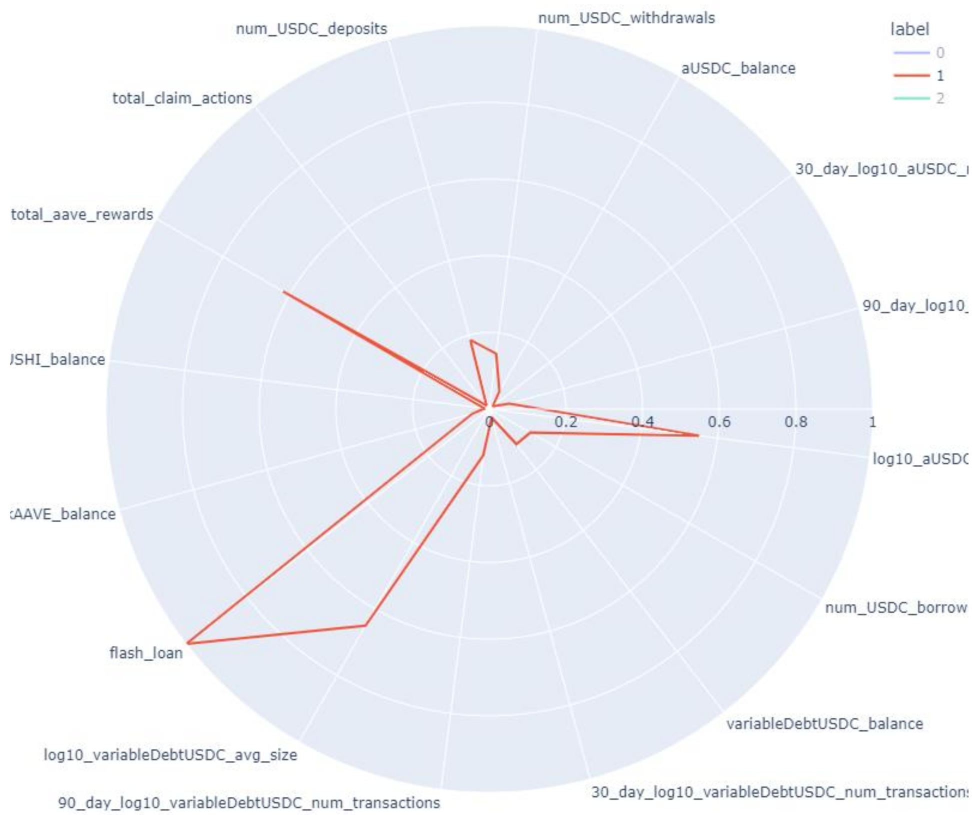

Figure 3 - DBSCAN cluster 1 polar chart

Figure 4 - DBSCAN cluster 2 polar chart

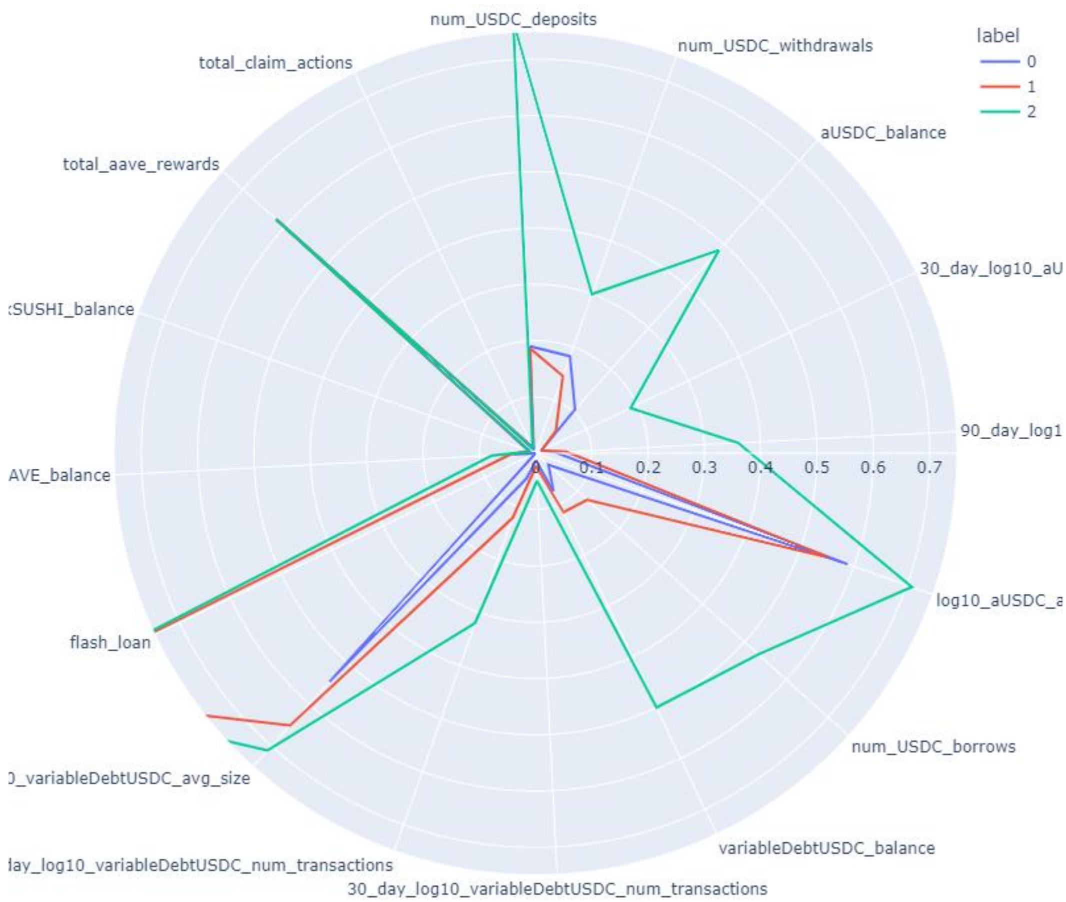

Figure 5 - DBSCAN combined polar chart

Figures 2, 3 and 4 shows the isolated polar charts of each cluster, while Figure 5 overlays them together. The figures show that DBSCAN clusters 1 and 2 (the smaller clusters) contain flash loan users, while the more numerous cluster 0 does not. This is indicative of more advanced users since it is a specialized function of the AAVE protocol that goes beyond just simple USDC transfers to/from the lending pool. Some smart contract engineering is possibly involved from these user wallets as flash loans are typically used for arbitrage within very short timeframes.

Figure 5 shows that cluster 2 wallet addresses on average make much greater number of deposit, withdrawal and borrowing transactions, and currently has the greatest number of USDC deposited in the lending pool. They also hold the greatest debt and greatest amount of AAVE staked. They make more lending/borrowing/depositing/withdrawing transactions on average, and the average size of each transaction is also greater.

Clusters 0 and 1 have similar characteristics. A few differences include cluster 1 having 1) flash loan users, 2) greater number of borrowing/repayment transactions, 3) greater average debt balance, 4) greater average size of each borrowing/repayment transaction. Cluster 0 which represents the majority of wallets (11070/12422) differs in having a greater average deposit balance and number of deposit withdrawal transactions. This might suggest that cluster 0 users tend to be casual users that are simply looking to deposit USDC into the lending pool in order to earn interest. On the other hand, cluster 1 contains users that are looking to take on more debt and borrow USDC for leverage in order to seek greater returns elsewhere.

Based on the above analysis, the clusters can be characterized as follows:

- Cluster 0: Casual users looking to earn interest on deposits
- Cluster 1: Users borrowing USDC for leverage
- Cluster 2: Advanced power users, high frequency of transactions, greatest amount of capital, likely to be professionals

BIRCH

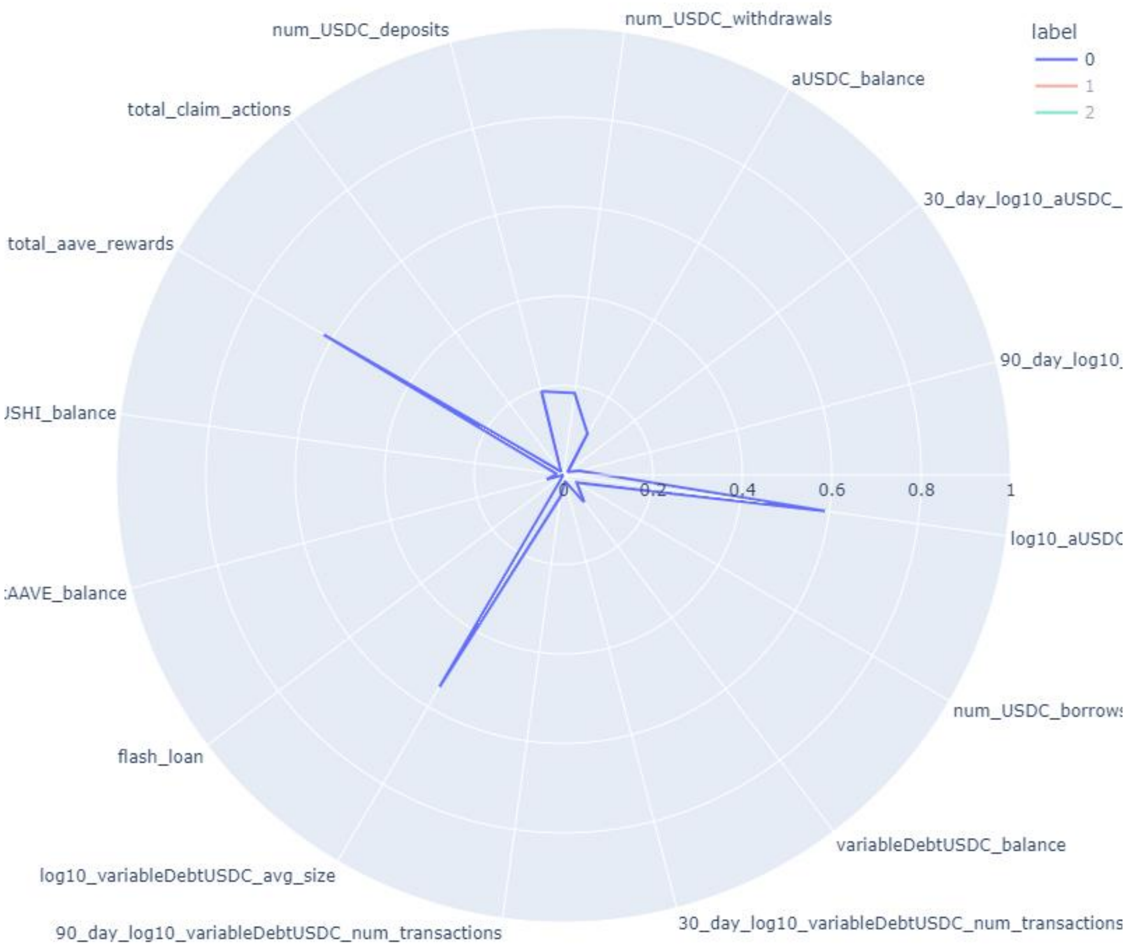

Figure 6 - BIRCH cluster 0 polar chart

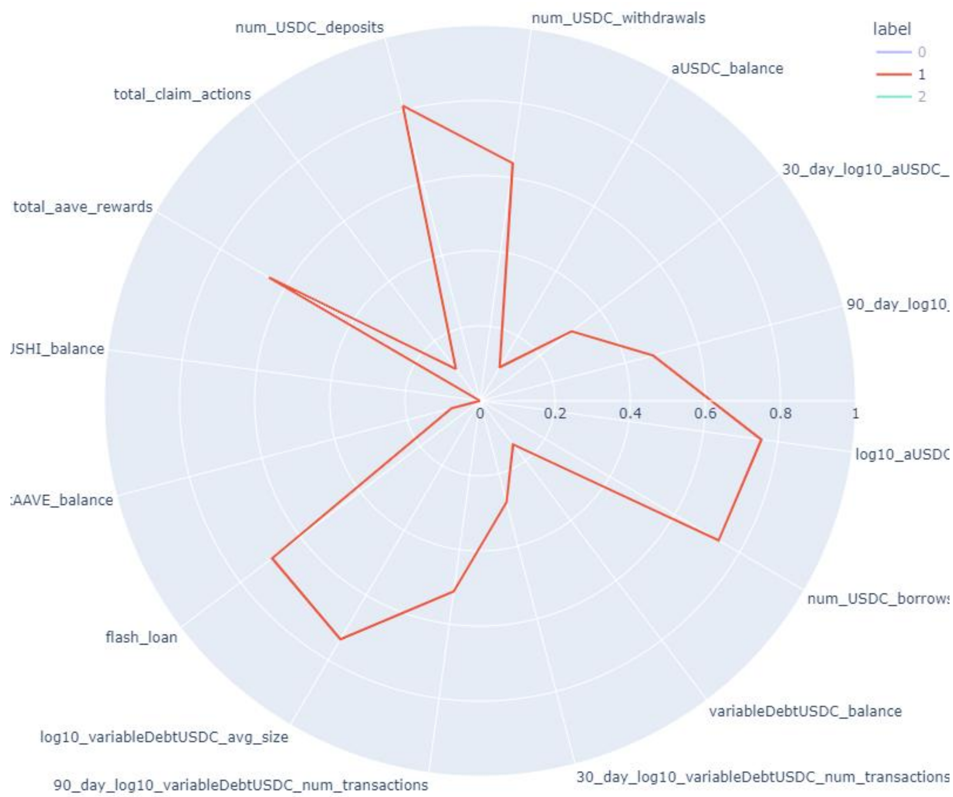

Figure 7 - BIRCH cluster 1 polar chart

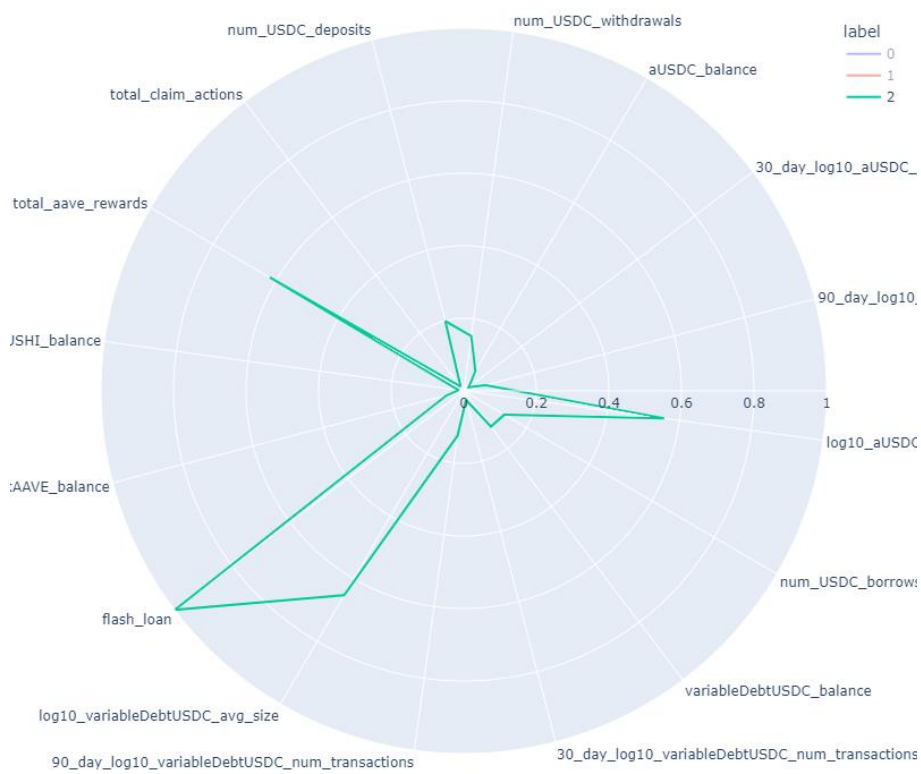

Figure 8 - BIRCH cluster 2 polar chart

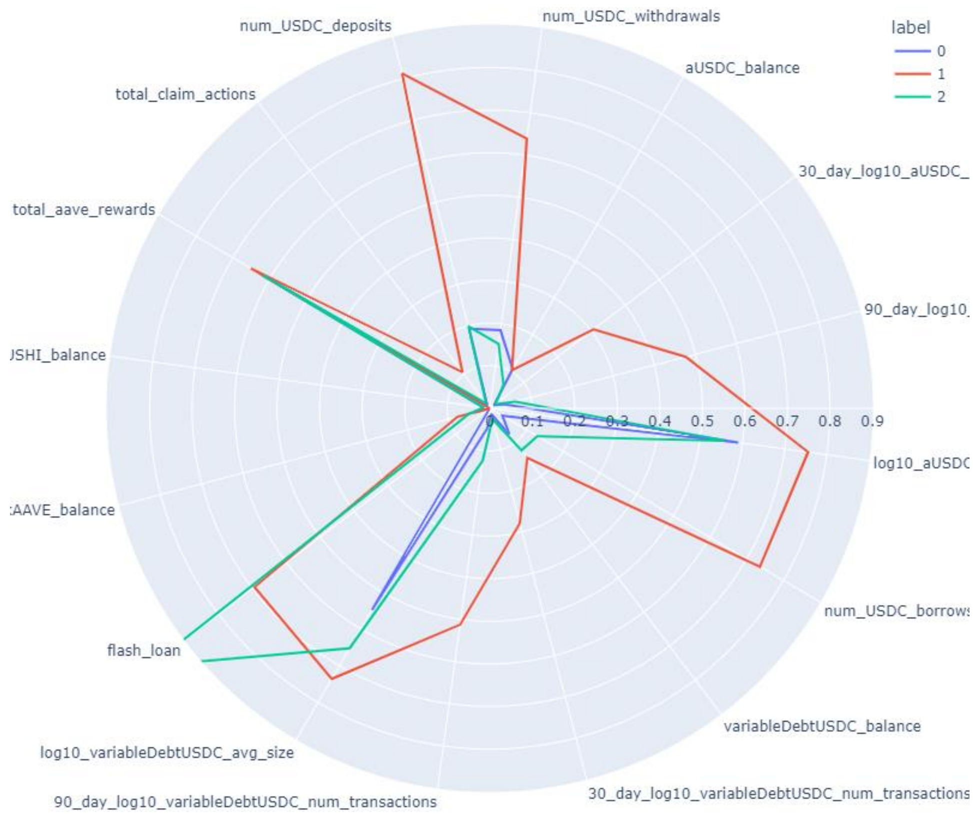

Figure 9 - BIRCH combined polar chart

Figures 6, 7 and 8 shows the isolated polar charts of each cluster, while Figure 9 overlays them together. The figures show that BIRCH clusters 1 and 2 contain flash loan users, while the more numerous cluster 0 does not. This is similar to the clusters obtained from DBSCAN and is indicative of more advanced users.

Figure 9 shows that the most distinguishing characteristic of cluster 2 is the number of wallets associated with flash loans. However, the main difference between the DBSCAN and BIRCH clusters here is that the middle cluster (cluster 1) of BIRCH is the cluster with the greatest average number of deposit, withdrawal, and borrowing transactions. Cluster 1 makes more lending/borrowing/depositing/withdrawing transactions on average, and the average size of each transaction is also greater than clusters 0 and 2.

Based on the polar charts, cluster 0 (biggest cluster, 11171 wallets) is actually more similar to cluster 2 (smallest cluster, 72 wallets). The only major difference is that cluster 2 is characterized by flash loan users, as well as a higher amount for each transaction related to borrowing/repayment. This might suggest that the clustering is suboptimal even though the silhouette score is similar to that of DBSCAN, with the usage of flash loans being the major difference, and that two clusters might be a better option since there are so few datapoints in cluster 2. This will result in two clusters that look like clusters 0 and 1 produced by BIRCH, where cluster 0 represents casual retail users with smaller capital, while cluster 1 represents users with more capital making more transactions on average.

2-D visualization of DBSCAN clusters using t-SNE

The full dataset for the DBSCAN clusters is represented in 2 dimensions below in Figure 10 using t-SNE. The algorithm performs non-linear dimensionality reduction into a 2-dimensional embedding visualized here.

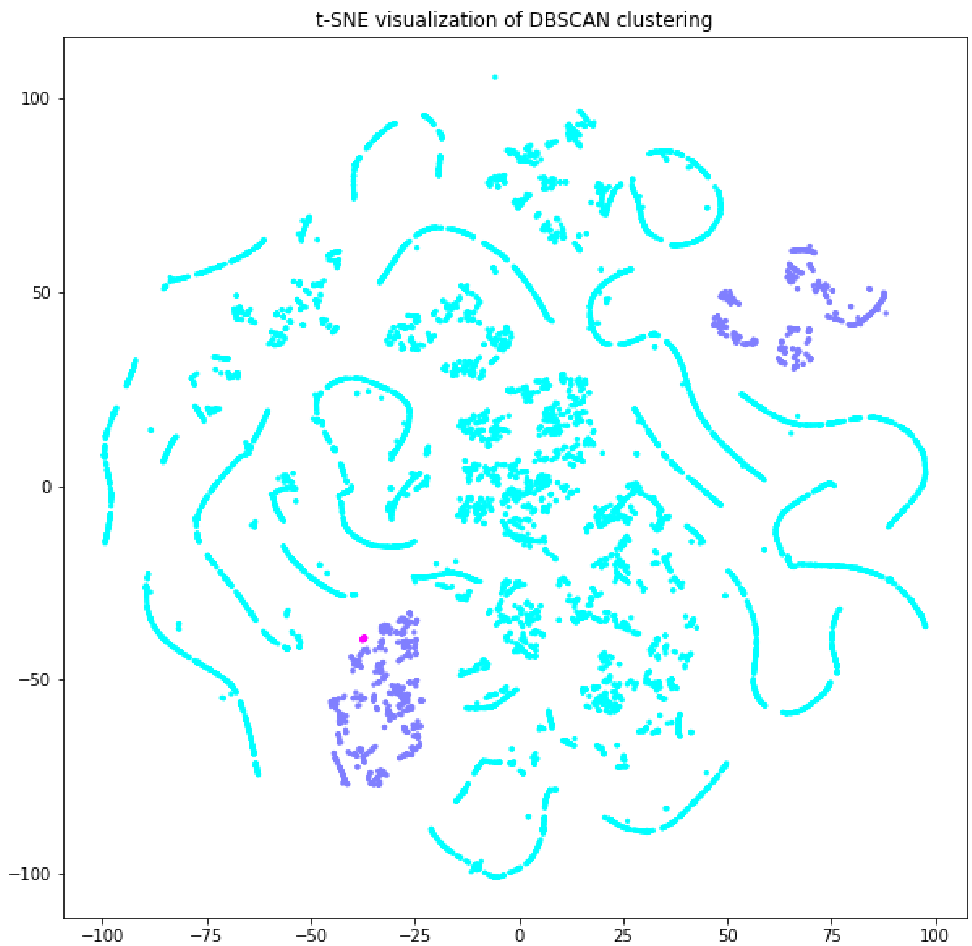

Figure 10 - t-SNE visualization of DBSCAN clustering

The light blue datapoints represent cluster 0, darker blue datapoints represent cluster 1, and the red datapoints represent cluster 2. Figure 10 shows that the clusters are well-differentiated in this non-linear feature space as there isn't any overlapping of datapoints from different clusters, even though the cluster sizes are highly uneven (which is natural).

Conclusion

Even though DBSCAN and BIRCH produced clusters with similar silhouette scores, the polar charts show that DBSCAN seems to have better clustering. 3 clusters were obtained using DBSCAN with these distinguishing characteristics:

- Cluster 0 (11070 wallets): Casual users looking to earn interest on deposits, mainly deposit/withdrawal transactions
- Cluster 1(1337 wallets): Users taking on debt for leverage purposes, greater average borrowing/repayment frequency and debt balances than cluster 0
- Cluster 2 (15 wallets): Advanced power users, high frequency of deposit/withdrawal/borrowing/repayment transactions, greatest amount of deposits/debt, likely to be professionals

These clusters also seem to reflect reality. Uneven cluster sizes are expected since there should be somewhat a power law distribution in which a small number of users have more capital and are more active than the rest when using DeFi to maximize profits. Thus as the actions taken become riskier and more sophisticated with more capital and expertise required, we can expect the number of wallets performing such actions to decrease.

Based on the results, it seems that it is possible to utilize on-chain transactions to characterize the behavior of wallets. However, features have to be engineered in order to suit the objective for the characterization. In this case, we focused on transactions specific to a DeFi protocol to cluster wallets according to their usage. The sample of transaction data used is only a small fraction of the millions of historical transactions in DeFi. Currently, there are more than a million transactions a day on Ethereum, of which more than half are probably linked to the ever-growing DeFi ecosystem. To get better behavioral characterization results that can generalize across projects, it would be necessary to tap into the transaction data of more wallets and DeFi protocols.