Analyzing Connectivity of Blockchain Markets and DeFi Applications

Jared Gridley
Graph Mining Project Final Report
gridlj@rpi.edu

1. Introduction

With more and more people experimenting with cryptocurrencies, numerous Decentralized Finance applications have been created to imitate the traditional financial system in the crypto space. With many applications being created with similar functionality, naturally some will take a dominant position int he ecosystem and some will fall into a more niche environment. [1] This network of which users are using which services on which platforms can be modelled and analyzed as a bipartite network.

By analyzing these activity patterns, the general health of a DeFi application can be estimated accurately, this is particularly important for people who want to earn money off of DeFi apps. Applications that experience an influx of users will start paying more to liquidity providers as they are validating more transactions. This extends to applications associated with apps that grow in popularity. We can look at Axie Infinity, a decentralized gaming platform build on ethereum, the game somewhat resembles Pokemon or TheSims. From October to December 2021, Decentralised saw a wave of new users come onto the gaming platform, users were trying to by everything from land to materials to just stockpiling the native $AXS token. [2,3] However in the early days of Axie Infinity, most people could not buy into the app directly with USD, they had to go through Ethereum and thus go through an exchange. Particularly, the 1INCH exchange saw a similar rise in users, most likely because Axie Infinity allows you to connect you exchange wallet directly to the app. Nevertheless, the liquidity providers on 1INCH saw a sharp rise in returns as they were almost suddenly sacked with thousands of new users trying to get onto Axie Infinity. [4]

Further, being able to predict where users will visit next based on users with similar transaction patterns has many applications to advertising. When it comes to DeFi applicaitons, they way you sign on is by connecting your wallet. When you connect your wallet, you share your wallet address which is all that is needed to get the entire transaction history on that address, allowing you to target ads to specific user transaction patterns.

With this research I hope to identify key DeFi apps for making the largest returns on investments as well as predict which users will go to which applications based on previous transaction history. The code for this project can be found on my Github[1].

2. Background

2.1. Decentralized Finance in a Nutshell

Since the takeoff of Bitcoin in 2012, numerous other cryptocurrencies have been developed, some for the memes and others for imitating a decentralized micro-economy. These currencies can be split into two major categories. There are coins like Bitcoin and Dogecoin which are almost purely transnational and then there are coins like Ethereum and Solana which allow for Smart contracts. Smart contracts are programs that are stored on the blockchain and runs automatically when certain conditions are met. [5] Smart contracts are the basis for most DeFi applications. Take Yeild Farming for example, Yield farming is typically used on decentralized exchanges to lend, borrow, or stake coins to earn interest and speculate on price swing. [6] The heart of any yield farm is a smart contract that is mediating all of these borrow, lend, and stake transactions among all the parties involved. [7] In general, smart contracts allow user users to take advantage of many of the services provided by banks and other traditional financial institutions but with more transparency and higher returns.

The most common DeFi applications are Open Lending Protocols, Automated Market Makers and Decentralized Insurance.

[1] Github: https://github.com/JGridleyMLDL/GraphMiningProject

2.1.1. Open Lending Protocols

Open Lending protocols allows users to offer crypto loans in a trustless manner by allowing holders to stake the coins they have for lending purposes. This process is controlled by a smart contract that makes the staked tokens available to lend while providing returns in the platforms native token. Nearly all loans are collateralize, however for crypto loans, the collateral is typically higher than the loan itself. [7,8] The most popular lending platforms on Ethereum are Compound and Aave.

2.1.2. Automated Market Makers (AMM)

An automated market maker is a protocol on decentralized exchanges that uses a math-based formula to price assets. Instead of using an order book like a centralized exchange, assets are priced according to a pricing algorithm. When a user wants to exchange tokens the AMM orchestrates the transaction with a Smart Contract associated with a liquidity pool for the exchange pair. This allows users to trade currencies easily with no intermediaries (aside from the algorithm, which is publicly available). The most popular exchanges/AMMs on Ethereum are SushiSwap, PancakeSwap, Bancor and Curve. [9]

2.1.3. Decentralized Insurance

Defi Insurance operates similarly to regular insurance. Users or companies that operate in the DeFi space can pay a certain amount to get your funds covered for a certain event. Not everything can be ensured, typically it only covers exchange hacks, attacks on DeFi protocols, smart contract failures or stablecoin price crashes. [10] The most popular Decentralized Insurance providers are Cover and Nexus. [11]

2.2. *Spectral Transformations*

Spectral transforms are a method of matrix manipulation which in the case considered for this project, pertains to applying kernel functions to the eigendecomposition of an adjacency matrix. There are also applications to signal and image processing where it is used to extract features from sound recording and images, but for this project I only consider the graph applications.

Spectral Tranforms are defined by the following:

1. Given a normalized adjancecy matrix representation of a graph, we can perform an eigen-decomposition on the square matrix:

   $$A = U \Lambda U^T$$

   where U is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues.

2. Now, if we wanted to perform a transformation on the Matrix A, such as applying a power of 3, we get

   $$A^3 = A \times A \times A = U \Lambda U^T U \Lambda U^T U \Lambda U^T = U \Lambda^3 U^T$$

   This then opens up Matrix A to more transformations with exponents, such as the exponential.

When applying spectral transforms to a graph, not all kernel function will work. If we are applying a kernel function to a matrix A, or the eigenvalue matrix, we assume that the eigenvectors will remain the same since our original graph and the transformed graph are still mostly similar. This can be reasonably achieved when we use the original graph as the training data and the transformed graph as the test data. [12] This was shown by Jérôme Kunegis who analyzed how kernel functions change the two matrices and gave some kernel functions that work for graph applications:

Kernel Functions that work for Graph Spectral Transformations:

- Triangle Closing: A^2
- Path Counting: $p(A) = \sum_{k=0}^{\infty} \alpha_k A^k$
- Exponential kernel: $K_{EXP}(A) = exp(\alpha A)$
- Neumann kernel: $K_{NEU}(A) = exp(I - \alpha A)^{-1}$

[12]

After applying the kernel functions, Kunegis showed that the problem can be reduced to the sum of a least squared regression problem:

$$A' = F_{ker}(A) = UF_{ker}(\Lambda)U^T$$

$$min|F_{ker}(A) = A'|_F = |UF_{ker}(\Lambda)U^T - A'| = |F_{ker}(\Lambda) - U^T A'U|$$

$$= min\sum_k (F_{ker}(\Lambda_{kk}) - U_{k,}^T A'U_{,k})$$

Where A' is the transformed matrix and $_F$ is the Frobenius Norm. [12, 13]

In applying spectral transforms to this projects, I use it for link prediction in bipartite graphs. Since we are trying to predict new edges across disjoint sets, we can use it to match users to applications. Traditional neighbor-based approaches will only predict new edges within the same set.

3. Methodology

3.1. Data

The data for this project was obtained through TheGraph, an application that indexes the Ethereum Blockchain and serves requests for information on certain addresses, events and other identifiers. TheGraph is structured by subgraphs. Anyone can make a subgraph by creating a schema that instructs the indexing protocol which events to pull off of the blockchain. Large Apps like Compound, Aave and Uniswap all maintain offical subgraphs. TheGraph is designed to be a decentralized marketplace for Ethereum data but currently offers a centralized service that will index and serve queries for free[2].

TheGraph uses GraphQL to write and return queries, since each protocol has its own subgraph, they each have their own subgraph schemas, meaning that generally queries do not transfer across application subgraphs. To avoid writing custom queries for each subgraph, there are some companies that serve this data with a paid lisence, but for the scope of this project, I use TheGraph to get all the data.

The DeFi applications examined in this project are Uniswap V2, Aave, Compound, 1Inch, Balancer, Axie Infinity, and Decentraland. I looked at the internal structure of Uniswap V2 and Compound. All 7 were used in mapping movement among this subset of ethereum apps.

3.2. Analysis Methods

To analyze the connectivity and structure of DeFi applications I used a variation of metrics to visualize the structure of an app or set of apps:

- Average Shortest Path Estimation
- Degree Distribution and Power Law estimation
- Hub Detection
- Connected Components/Strongly Connected Components
- Centrality Detection
- Community Detection
- General Statistical Analysis (mapping transaction values and frequency)

These methods were applied to graph representations of Uniswap V2, Compound and User flows between apps. Each graph was structured a little differently, so I go through that below:

[2] TheGraph Hosted Query Service: https://thegraph.com/hosted-service/

- Uniswap: Modeling the liqudity pool structure. Nodes represent tokens and edges are liquidity pools connecting these tokens with edge weights holding the timestamp of creating and the total value locked in at that timestamp.
- Compound

3.3. Link Prediction

For testing the link prediciton method, I used the Spectral Transform method described above with an Odd path counting kernel function. With this kernel, I examine the neighbors 1 hop, 3 hops, 5 hops and 7 hops away to calculate a prediction score for if a node is likely to attach. This prediction score is determined based on a weighted sum of path counting kernels:

$$Pred\_Score = a_1 V + a_3 V^3 + a_5 V^5 + a_7 V^7$$

where V is the adjacency matrix of training data and the a values are predicted eigenvalues.

4. Results and Discussion

In this project, I analyzed the internal structure of two defi apps, Uniswap V2 and Compound. I then investigated the connectivity of the apps in relation to each other, then building a prediction model for where users will perform their next transaction event on.

4.1. Internal: Uniswap V2

Uniswap is decentralized exchange made of liquidity pools that, combined with smart contracts, control the exchange. These pools operate as an exchange mechanism between two tokens, so I initialized my Networkx MultiGraph with edges as Tokens and the edges the liquidity pool creation date and total value locked. From this representation I was able to analyze the connectivity of tokens on Uniswap. I found that generally, the graph was quite well-connected, with an Average Shortest path of 2.12. From mapping the degree distribution and fitting it to a power-law relationship with a coefficient of 1.73, we can deduce that there are some tokens that are connected to nearly every edge and others that are more fringe. This is what we expect to see from the degree

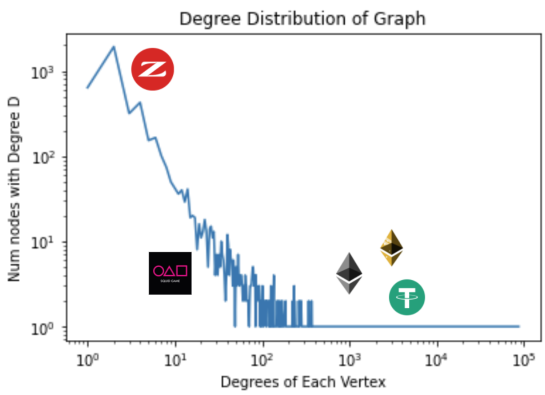

Fig. 1. Degree Distribution of Uniswap Tokens

distribution, there are some tokens, particularly stablecoins like USDC, USDT, and DAI that nearly every node connects to, primarily because these are the coins that centralized financial institutions, like banks are connected to, so if you want to get money in or out, it typically goes through a stablecoin or major cryptocurrency like Bitcoin or Ethereum. This follows with detecting a large number of hubs, out of the 4670 coins, there were 886 hubs detected.
This was further seen when measuring the most central tokens on Uniswap, listed below.
We can see that in both metrics, the most central tokens are often the stablecoins, however when we examine the Degree centrality closer, we start to see the skeletons of dead scams. Particularly with EPRO, which is the ticker for Ethereum Pro, a scam token that was created when the developpers of Ethereum were pitching the idea of an update to Ethereum. The goal of this scam was to imitate Ethereum and get user to put a lot of money into the currency and drive up the value before the majority shareholders can pull all their funds and send it into a crash.

Table 1. Most Central Tokens on Uniswap

Centrality Metric	1	2	3	4	5
Degree Centrality	wETH	USDT	USDC	EPRO	MIST
Closeness Centrality	wETH	USDT	USDC	DAI	UNI

It appears as one of the most central tokens because as part of the scam, they needed to make it look like this new Ethereum token was legit and so they created liquidity pools that mimicked the ethereum pools, hence so many pools. However the value locked into each of these pools is quite low, so it ranks quite low on the closeness centrality.

When looking at the connectivity of Uniswap, there was another anomoly that apeared in the graph, it wasn't fully connected. This contradicted the indicators above, where every coin would go through a central stablecoin, but there were small cliques of 2-14 tokens that were completely disconnected from the rest of crypto. These turned out to be development projects for people testing out the ecosystem.

After looking at the liquidity pool structure of Uniswap, I turned my attention to the general statistics of transactions on Uniswap. I found that, since the start of Uniswap, the value passing through each day remained relatively the same (exclusing the data from when it was in beta). However the number of transaction had grown significantly, this indicated that transactions on Uniswap have shifted from fewer large transactions to many more smaller ones.

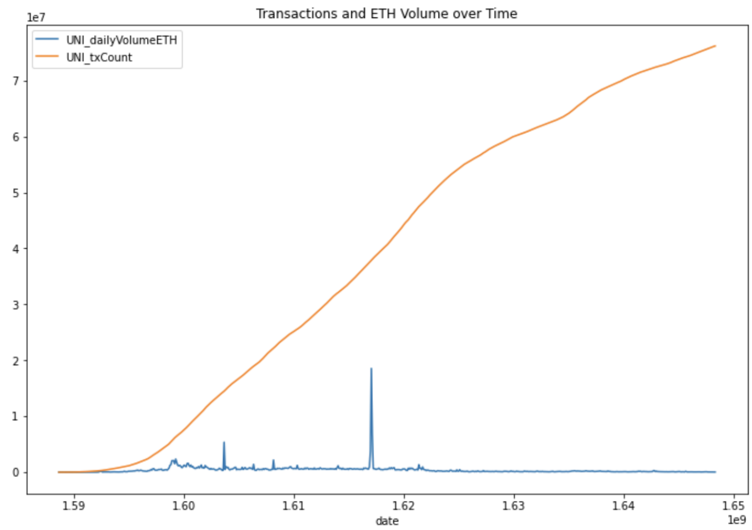

Fig. 2. Uniswap transactions over time

4.2. *Internal: Compound*

Compound is a much different app than Uniswap. While Uniswap is a exchange, Compound is a lending protocol so while there are still smart contracts that control most of the activity, it is still structured much more around users.

When analyzing the connectivity of Compound, I found that the graph was split into different components, so rather than having all the traffic coming from or going to a set of nodes, there are more individuals who manage and control their own smaller lending pools. From the graph of 12,949 addresses with 18,345 transactions, the largest Strongly connected component was just 2,332 nodes while the largest Weakly Connected Component was 12,552. This indicates that there are many users who interact with lending pools across the platform but the largest cluster hasn't emerged as dominating the platform. We can see this when we start plotting the nodes and transasctions.

We can see in the figure that most of the central communities are around contract that Compound has deployed that allow users to trade in a native version of popular stablecoins and cryptocurrencies. Particularly with cETH,

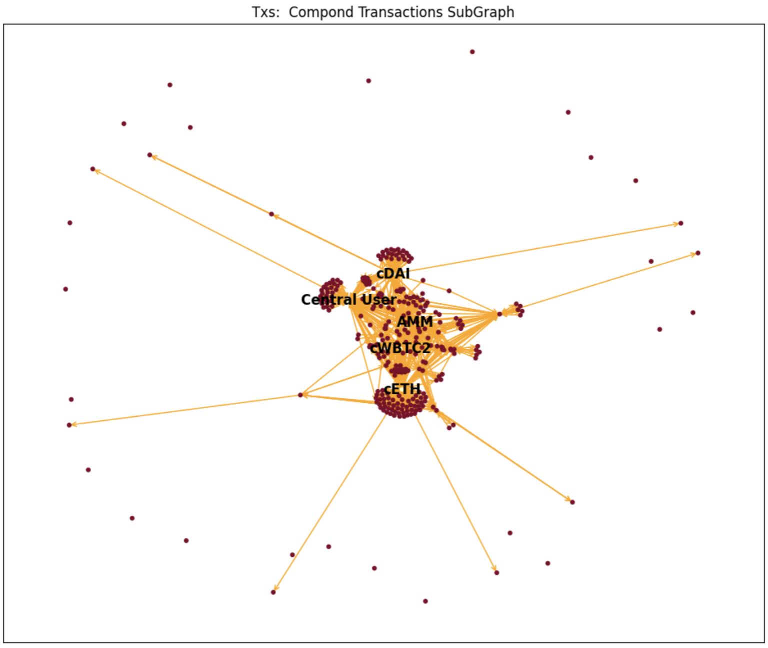

Fig. 3. Compound transaction subgraph (random)

which is Compounds native Ethereum token, it is back almost entirely by ethereum so it will mimic any price changes in ethereum. Similarly with cBTC and cDAI which are the native Bitcoin and Dai Stablecoin tokens on the platform. However, within this subgraph, we see that one of most centrally located nodes is actually a user address, this often occured when generating random subgraphs, but it is an indicator that even though Compound is more decentralized, it still has some very active users that control relatively large portions of the money flowing through the app.

This tendency towards centralization is something that most DeFi apps struggle with. Their goal is to remain as decentralized as possible, however as they scale they tend to develop some internal centralization whether it be through a central smart contract or central user pool.

4.3. DeFi Connectivity

After examining the interal structure of Uniswap and Compound, I turned my attentiton to the relation between the apps themselves. I chose to look at 7 applications with a range of uses within the ecosystem. The applications are:

- Uniswap: One of the earliest decentralized exchanges
- Compound: A popular lending protocol
- Aave: A lending protocol/exchange
- Balancer: An automated portfolio manager and trading platform
- Decentraland: A gaming platform similar to Minecraft
- Axie Infinifity: A gaming platform similar to Pokemon
- 1Inch: A newer decentralized exchange

The nodes of this graph were users addresses on the platform and the edges were transactions. Plotting a random subgraph of this network, I found that there were some distint connections between the applications.

As I mentioned in the introduction, changes in market conditions in an app like Axie Infinity was rippled through the 1Inch Exchange. This is seen in the graph above. We can see that there is a distinct connection between Axie Infinity users and those on the 1Inch exchange. After looking into it more, it is almost natural that nearly all the Axie traffic goes through 1Inch. When a new user wants to play Axie Infinifty, they need to make

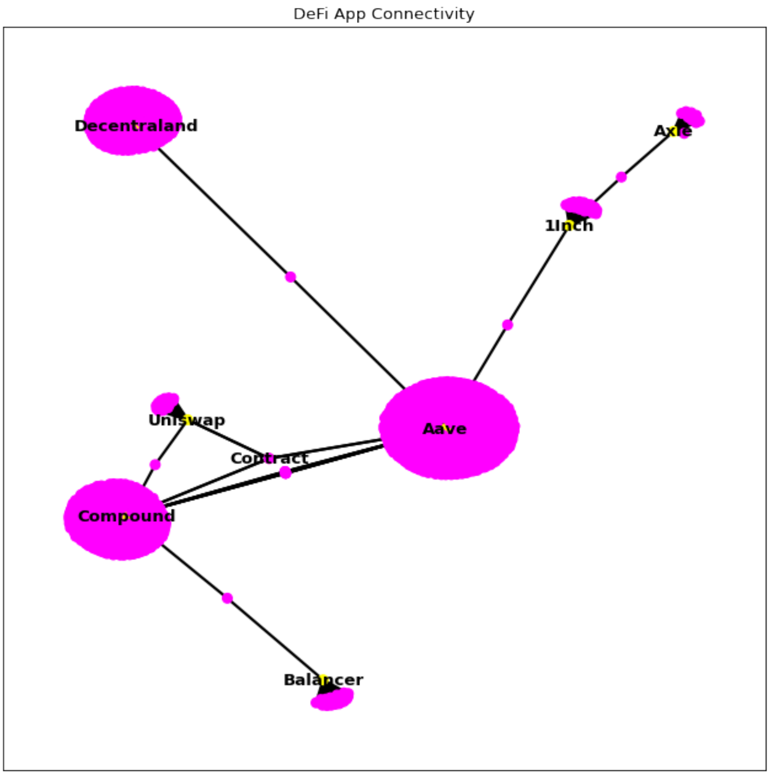

Fig. 4. Plot of DeFi addresses and their transactions across platforms

a Ronin Wallet and transfer $AXS into it. When you go to tranfser funds into the Ronin Wallet, it directs you to the 1inch exchange to do this. While another exchange like uniswap would work, many of the users just go with the suggestion and use 1Inch. The contibutes to the re-centralization of crypto as it is pushing users of one app to only certain other apps. If 1Inch were to go down, many of the Axie users would likly be flustered and upset that they have to find a new app, but that number would be inflated beyond what it could be users picked their own exchanges.

An interesting feature of this connection graph is the nodes between major applications. These are most often contracts that are designed to get the best trade. Specifically with the Smart contract node between Compound, Aave, and Uniswap. When looking at this further, it actually belongs to the 1Inch protocol who uses it to ensure they are providing the cheapest and most efficient exchange routes. While it may seem like its trying to pressure the competiton, its actually more innocent than that, if it finds an exchange route that goes through Uniswap, the contract will act as a proxy and process tha transaction on Uniswap for 1Inch user. 1Inch has other contracts across other exchanges like PancakeSwap and SushiSwap but it reveals that 1Inch is actually much more connected that it leads on.

Having contracts that interact with other platforms is quite common among newer DeFi apps, as there are already numerous apps for each function, it can just look for the best price among those available.

4.4. User Transaction Prediction

To predict user transactions, I used the spectral transformation method described above to ensure predictions would cross the sets and be between users and apps. To get the training and testing data/graphs, I split my graph of users temporally with the earliest 75% of transactions making up the training graph and the last 25% making up the test set.

When applying the odd-path counting kernel to the graph, we can plot the eigenvalues with the transformed eigenvalues to see how well the kernel will represent the transformation.

From the regression fit of the eigenvalues, we can see that it fits quite well. After applying the transformation described above and getting the predictions for the new edges formed, I plotted the Precision-recall Curve and the

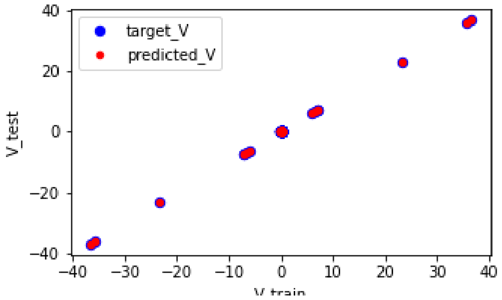

Fig. 5. Plotting odd-path counting regresison fit (Λ_{kk}, $U_{k,}^T A' U_{,k}$)

ROC curve, with the method achieving an AUC score of 0.87. This indicates that the model does pretty well as distinguishing between the connections that will form and the connections that will not.

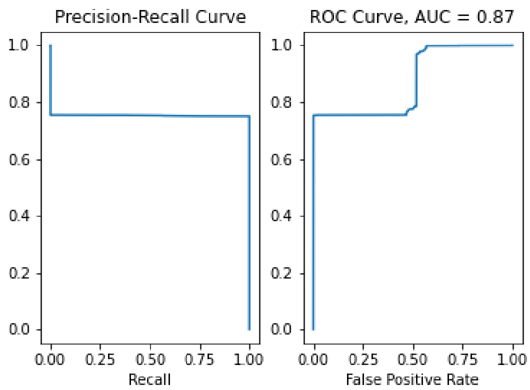

Fig. 6. Precision-recall Curve and ROC curves

A caveat of using the method of spectral transformations is that is makes predictions between every class. On large graphs this can be computationally expensive, to speed this up, I used CuPy, which is a GPU version of NumPy. By making predictions between every class, there will be a disporportionate number of edges that do not occur than those that do. So when looking at the confusion matrix we can see that there are many more edges that are not formed than ones that are, and in this case, the prediction method often misclassified False Negatives.

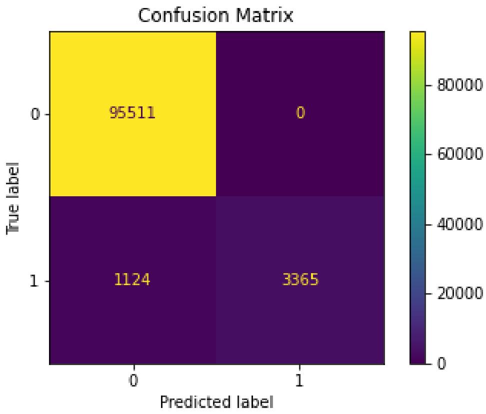

Fig. 7. Confusion matrix

5. Conclusions

In this paper I analyzed the internal structure of Uniswap V2 and Compound. Specifically I used graph mining to estimate the degree of centralization on each platform. I also gave a method for predicting new interactions in this bipartite graph representation of Users and the applications they use and when. To do this I used a spectral transformation to make predictions from the adjacency matrix with an odd path numbers, this way new interactions would cross the two disjoint sets. I saw with this solution that many of the predictions were with users staying with the app they are currently using, which generally follows with real-world patterns.

Future work in this area particularily surround the scope of the DeFi applications examined. I investigated 7 DeFi Applications but in reality, there are many more DeFi applications. The main bottleneck I found while doing this project was with collecting the data. I was using TheGraph which is broken up by DeFi application, there are other sources of this data but with a licensing cost, so exploring more DeFi apps will likely result in more distinct clusters of applications, with each cluster having its preferred lending and exchange platform.

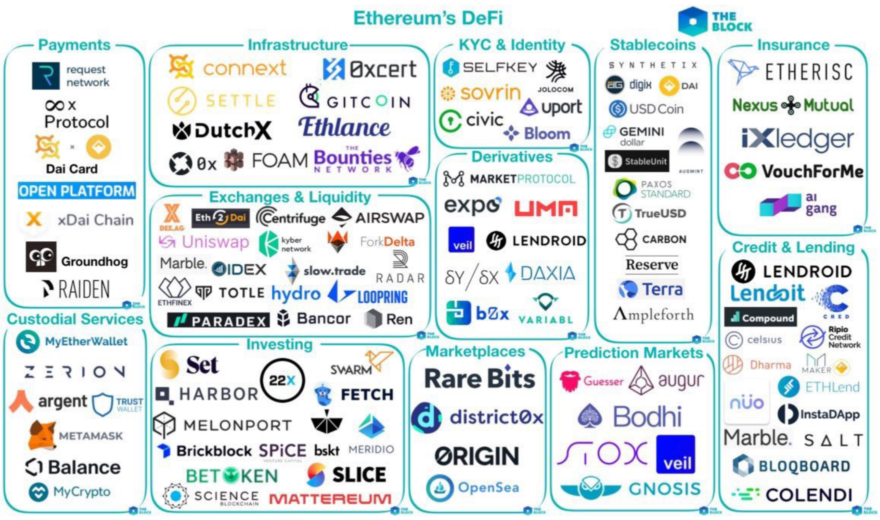

Fig. 8. Ethereum Defi Ecosystem (TheBlock, 2019)

Following the expanion into more DeFi apps, there are also other blockchains with their own ecosystem of DeFi applications, like Solana. Examining these could be insightful into further getting the largest APY as well as measuring general activity across platforms.

To help yeild better insights from the link prediction task and also better understand the centralized nature of the DeFi ecosystem, we could force new connections to be made by removing a DeFi application and then using that predict where the original users will flock to next. Further classification of DeFi applications will likely help with this, classifying them as lending, AMM, portfolio management, etc.

References

1. S. Protocol, “Defi market on the rise: 47% growth in one year,” *Yahoo!*, Feb 2022.
2. M. Quiroz-Gutierrez, “The cheapest real estate on popular metaverse platforms is already selling for more than $13,000,” Feb 2022.
3. “Blockchain gaming and the rise of axie infinity,” *Niko*, Oct 2021.
4. M. Pechman, “1inch price rallies 30% as its monthly dex volume nears $6 billion,” *Cointelegraph*, Jan 2021.
5. I. Blockchain, “What are smart contracts on blockchain?,” *IBM*.
6. “What is yield farming? what you need to know,” *Blockworks*, Feb 2022.
7. “What is defi (decentralized finance)?,” *Ledger*, Nov 2021.
8. “What are lending protocols? the rise of defi lending,” *Market News amp; Forecasts, Charts, Broker Reviews*.
9.
10. M. Bijkerk, “Defi insurance: Simply explained,” *BlockData Research*, Dec 2021.
11. GB0514, “Decentralized insurance: These 4 companies are raising the bar in 2021,” *HackerNoon*, Jun 2021.
12. J. Kunegis, *On the Spectral Evolution of Large Networks*. doctoralthesis, Universität Koblenz-Landau, Campus Koblenz, Universitätsbibliothek, 2011.
13. Y. Yang, R. N. Lichtenwalter, and N. V. Chawla, “Evaluating link prediction methods,” *Knowledge and Information Systems*, vol. 45, pp. 751–782, oct 2014.