Connectivity of Cryptocurrency Exchange Markets: A Topological Analysis

[redacted]

Fr[redacted]Vogt

[redacted]

***Abstract*—In our research we examine the connectivity of cryptocurrency exchange markets (CEM) at the autonomous system (AS) level of internet topology. Such online services rely on availability and robustness. For this we alter the existing CRISP-DM methodology and develop a web-scraper to gather and process source data of CEM and their AS-numbers. Using the CAIDA AS Links and the web-scraped dataset, algorithms within NetworkX are used to create a network graph and derive graph measures used to analyze the connectivity of CEM. Our results help to identify well-connected CEM, that could potentially suffer from network outages. We conclude that most CEM are well-connected. Additionally, they are oftentimes hosted on AS provided by platforms such as Cloudflare and Amazon. Our approach can be used by providers of CEM to assess their connectivity as well as for customers to select CEM based on availability benchmarks.**

Keywords—cryptocurrency exchange markets, availability, connectivity, network analysis, internet topology, autonomous systems

I. Introduction

The market capitalization of actively traded cryptocurrencies has risen from US-$1.5 billion to around $560 billion since 2013 [1]. Due to this growth the demand for digital services facilitating the trade of cryptocurrencies has risen significantly. One of these services are cryptocurrency exchange markets (CEM), which play a fundamental role in the crypto-ecosystem. These platforms enable the trade of cryptocurrencies in exchange for other digital currencies and additionally for real currencies [2]. Currently over 200 CEM are being actively used by traders and other peers [3].

For e-businesses and platform businesses availability and reliability of their services play a fundamental role given their strategic and operational management [4]. The significance of these fundamentals is further emphasized when costs of downtimes are analyzed. An hour of downtime on average leads to damages of around $0.35 million [5]. Not only do network outages have a direct impact on costs, but they can additionally lead to a reduction of customer satisfaction, which in turn may impact the long term profitability. Furthermore downtimes of larger platforms, such as *Coinbase* or *Kraken*, have resulted in short-term dips in the value of cryptocurrencies [6]. These downtimes can be caused due to several factors, such as but not limited to natural disasters, human error, and malicious attacks.

Despite connectivity not being a guaranteed factor, it ultimately plays a fundamental role in the success of e-businesses, such as CEM. While CEM providers cannot guarantee constant uptimes of their services, they can try to minimize network outages, leading to a higher degree of connectivity. Thus reducing customer dissatisfaction and potentially using their increased degree of connectivity as a competitive advantage in their respective playing field. The key to solving and preventing a large percentage of such issues is given by studying and analyzing the structure of the internet, its vulnerabilities, bottlenecks and probable points of failure [7].

II. Background and Related Work

In the following chapter important theoretical concepts concerning the research topic are explained and illustrated. We begin with an explanation of graph theory and its use in different contexts. This chapter provides additional theoretical information on the applicability of graph theory in the context of internet topology analysis. Furthermore, related work regarding similar research topics is described.

A. Graph Theory

Graph theory is a field of study within mathematics which tries to model and illustrate relations and properties between different graph elements. Its applicability in science varies and ranges from computer science, operations research to social network analysis and even biology. A graph itself consists of nodes and edges [8]. In other terms a graph G is a defined set of nodes N (sometimes also called vertex) and a set of edges E (sometimes also called arcs or links). E, as a finite subset of the graph, may be expressed as the cartesian product of the number of nodes with itself NxN.

Throughout this paper the terms nodes and edges are used as a standard. The meaning of those may vary from their applicability, e.g. a social network is a specific set of linkages among a defined set of actors. The characteristics of these linkages may be used to interpret behavior of involved actors [9]. In this example the nodes are presented by the actors and the edges by the linkages between those actors. Another practical example of graph theory and its components can be found in neural networks. The features in the input layer, the neurons in the hidden layers and the results in the output layer represent nodes. The connections between those neurons are the edges of the computational graph.

Different kind of graphs can be found in theory, where edges may be directed, e.g. feedforward neural networks (features are start nodes and the output is destination node) or undirected, which implies that the direction of an edge can be used interchangeably or no direction is present [8]. Edges can be weighted (weighing edges with a real metric) or unweighted (weighing edges with a binary) [8]. In a directed graph the positioning of the node may have a special meaning. A weighted graph may express a special meaning of the edges (i.e. strong ties vs weak ties in a social network). Moreover, graphs and their characteristics may be presented in form of incidence- and adjacency matrices [8]. Those indicate whether

nodes have edges and/or are linked to one another. To not overcomplicate the modeling process this paper focuses on an undirected and unweighted graph.

The purpose of graphs and their equivalent networks is similar in all scenarios. By modeling and measuring different graph-based measures in order to quantify and compare graphs and their properties, we can derive useful and relevant information on the structure und behavior of the complex graph and its dynamic components, such as connectivity.

To better illustrate the composition of a graph, refer to Figure 1. The graph has both undirected and unweighted edges. It consists of seven nodes (A, B, C, D, E, F, G) and nine edges ({AB}, {AD}, {AE}, {BC}, {BD}, {DE}, {DG}, {EF}, {FG}).

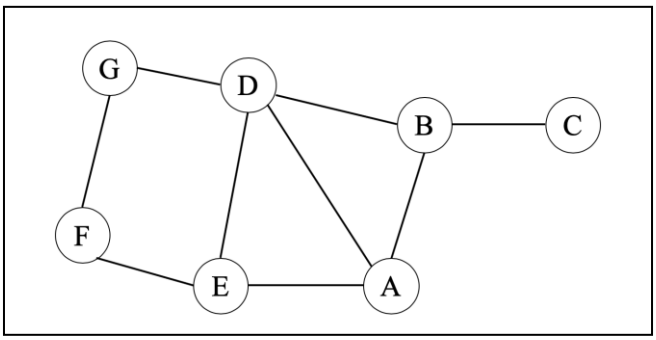

FIGURE 1: EXEMPLARY NETWORK GRAPH

B. *Internet Topology*

Over the last decades the Internet, which started as a small set of interconnected entities for research purposes [10], grew to an immense network of devices which are interconnected to one and other throughout the internet. This was mainly due to the fact that the internet switched from a heavily regulated environment and limited tool in its use until the early nineties, to a less regulated and more dynamic environment [11]. The purpose of using the internet took on multiple forms which led to the fact that it underwent exponential growth and still is. This growth can be measured quantifying the rise of the number of hosts (computer systems with a registered IP address on the Y axis) on the internet [12] [13].

The modern internet grew out of a cluster of different computers used by different entities in the 1970s (Military, R & D etc) [14]. Because of their actual form, the Department of Defense Advanced Research Projects Agency decided to establish a network for connection purposes of those heavily geographically distributed computers [14]. By doing so it was made possible to share data and software across this network. In the 1980s different computers, mostly owned by universities, were added to this network. By the mid-1990s the Internet had become self-sufficient [14]. Due to a growth in numbers of businesses and households purchased computers, there was a rise of subscriptions to internet service providers (ISP) [14]. The network, which was small in the beginning, grew as more institutions established a web presence or set up e-commerce operations [14] which led to the fact that the exchanged traffic increased [15].

CEM are part of this constantly changing network. To understand how the different components of this strongly dynamic environment behave and are interconnected research is needed. Over the past decade the networking research community has shown a growing interest in discovering and analyzing the network topology.

The reason why graph theory is important for the scope of this paper is because the internet and its backbone structure may be viewed as a graph itself. The examination of this network graph is referred to as network topology analysis which is described as the representation of the interconnection between directly connected peers within a network [16]. In other terms the analysis of the graphical representation of the internet components and their connectivity. Donnet et al. [16] state that these components of the internet can be pictured and analyzed at three different layers; the link layer, the overlay layer and the network layer. “The link layer is the description of how data link layer devices, switches and bridges are interconnected and how the different hosts are connected to them” [16]. The overlay topology can be seen as a topology of a peer to peer system [16] whereas the network layer can be referred to as the internet topology.

In the underlying research topic of this paper we focus on the network layer which can be analyzed on multiple abstraction levels: IP level, router level, Point of Presence (PoP) level and AS level. The IP level takes into consideration IP interfaces of routers and end-systems and can usually be measured with tools like traceroute [16].At the router level each router represents a node and each edge is a one hop connection. Data on the router level topology of the Internet can’t be obtained directly, so that inference from traceroute measurements is needed [17]. The PoP level is obtained by aggregating all relevant information from the router level or by direct aggregation of every interface information which is identified as geographically co-located [16]. Finally, the AS level provides information about the connectivity of the components of the internet. The AS level is the most abstract form and therefore the most computable for internet topology research.

Summarizing the above, AS incorporate all points of presence. A point of presence group consists of geographically co-located routers. Every router owns an IP address which serves as an interface for communication between different components. This can be used to create an AS graph, in which each node represents an AS and each edge a connection between two AS. Figure 2 illustrates the above-mentioned structure.

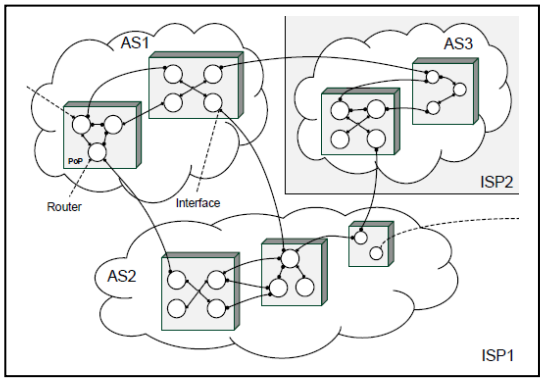

FIGURE 2: LAYERS OF INTERNET TOPOLOGY [26]

An AS can either be owned by an ISP or by any other big organization, e.g. corporate network or an educational institution. The owners may impose specific routing policies to govern the autonomous systems. An AS can be identified by its AS-number which serves as a unique identifier.

As the internet is owned by a large amount of ISPs, which operate in distinct parts of the internet, they engage in both formal and informal relationships to route the traffic in the internet collectively and ubiquitously [18]. Therefore, the links between the various nodes of an autonomous system level graph signify different roles and relationships in between the entities represented in the graph. The relationships can be categorized under provider-to-customer (P2C), customer-to-provider (C2P), sibling-to-sibling (S2S), peer-to-peer (P2P) [19]. Figure 3 illustrates the interaction between the different ISPs [16].

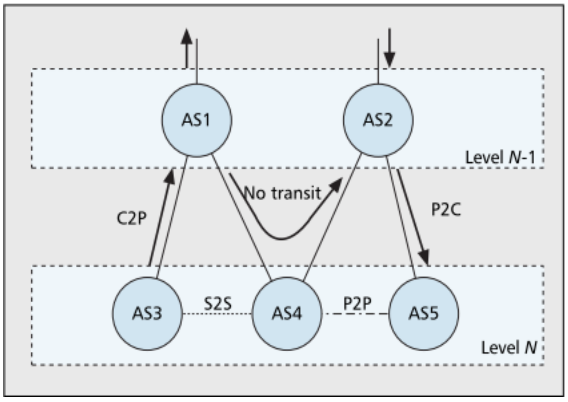

FIGURE 3: AS-RELATIONSHIPS [26]

To reach certain parts of the internet that an AS does not own nor can reach through its customers, it must buy transit services. In other terms it has to pay some provider in exchange for the rest of the internet [18], in this case we can speak of C2P and/ or P2C relationship. If exchange of traffic between different entities is negotiated without the exchange of money, to avoid sending traffic through a provider and by doing so avoid costs, we can speak of a P2P relationship [16]. The last link, S2S, connects two AS which administratively belong to the same ISP and exchange all kinds of traffic not only between their customers [16]. S2S relationship can result out of mergers and acquisitions or under certain network management scenarios [18].

C. Related Work

While internet topology analysis is a relatively new research field it has gained tremendous interest in the past years. An overview of related work and current literature is described in the following.

Kelkel [20] performs an analysis on the internet connectivity robustness, which is of paramount importance considering the fact that the internet is vulnerable to a variety of threats including cybercrime as well as terrorism. Those do not only put the internet and its underlying structure into danger but can also lead to real life failures and consequences. Examples are ATM failures, disturbances and malfunctioning of systems connected to the web. In order to conduct the research, a vulnerability assessment is performed by simulating a worm-based attack on border routers, which interconnect ASs and thus constitute an eventual bottleneck of the internet. In a next step, the impact of the worm-based router attack is assessed and quantified. The data which has been used for the research has been provided by the Center of Applied Data Analysis (CAIDA) [21]. The simulation results show that attacking the border routers is an efficient strategy in terms of deterioration of internet connectivity in total, as they tend to be of high importance for the internet structure. The study suggest that most harm can be caused by targeting Cisco and Juniper border routers [20].

Internet robustness is further analyzed by Baumann [22]. As financial losses can be derived from a lack of reliability of services, the robustness of the internet is analyzed in an AS-level graph. Similar to other research in the field of internet topology a network graph is created and analyzed using topological measures. This is followed by a robustness analysis by examining and targeting specific nodes within the graph, for example by random deletion. Following the robustness analysis, a closer look is taken at the vulnerability of individual AS for which the unified-risk score, based on previously used graph measures, is developed. The work concludes with a classification of AS into industry classes as well as geographic locations, in which their vulnerability is further analyzed. The results show that AS can be classified into industries, which are especially dominated by telecommunication- and IT-based firms. Additionally, the classification into geographic locations shows that AS in some locations, such as in Africa, are more vulnerable than others, e.g. Europe [22].

Schulze [23] conducts an industry classification of AS. By integrating different data sources (mostly social networks) Schulze performs a study which tries to categorize AS into different industry classes by using the allocated organizations, aiming to discover which organizations manage their own AS. Therefore, using multiple search patterns, a list containing different industry classes is created. In a next step, a first analysis based on a list of different ASN is implemented. Results imply that approximately nine percent out of 64.640 AS could be classified into 161 different industry classes which show a high dominance of IT related companies [23].

To analyze the structure of the internet from a more holistic perspective Tilch [24] attempts to combine data from different large-scale-TTL-based measurement campaigns into one. Therefore six traceroute data sources are analyzed and uniformly processed. He derives the data from iPlane, CAIDA, Carna Botnet, DIMES and RIPE. iPlane is a research project by the University of Washington in which key figures of path performance are attempted to be predicted as well as the creation of a structural model of the internet is pursued. After traceroute data is collected by the traceroute measurement infrastructure, the raw output is used to create an annotated map of the internet. Carna Botnet is a project conducted by an anonymous researcher. Using botnets, which are a collection of network-connected devices acting under the command of one entity, a 9 TB dataset, the Internet Census 2012, is created. Within the Internet Census 2012 is a sub-dataset containing traceroute data used for the thesis. DIMES is a research project started by the Tel Aviv University with the goal to create a full snapshot of the Internet graph. To do this a software (netDIMES) is installed on volunteer's computers to perform traceroute measurements and ping queries. RIPE is the Regional Internet Registry for Europe, the Middle East, and Central Asia. Two topology measurement projects were hosted by RIPE to create one dataset with traceroute data and another dataset with IPv6 data. Using the data derived from the different sources to create a uniform dataset Tilch models network topology graphs at the five different topology layers. Following the creation of the graphs, topological metrics are derived and analyzed [24].

Ghazaryan [25] conducted a topological analysis of the internet by laying the focus on single institutions. Concretely, the purpose of the study was to derive useful information about the online connectivity of financial institutions, which offer different banking and payment services online, by modeling those as a graph on the AS-level. For the building of the graph as well as the execution of the underlying steps of the analysis a dataset including a list of the AS and their links was needed. That information has been made available by CAIDA. To perform the analysis a variety of graph-based measures have first been described and then calculated. Results of the analysis show that most of the financial institutions which were taken into consideration are well connected in terms of network connectivity. Furthermore the conducted research revealed that the AS widely differed from each other in their connectivity as well as reliability and that some computed metrics tend to correlate with each other. [25].

Fabian et al. [26] focus on the network reachability of cloud services by analyzing and quantifying the topological connectivity of large cloud service providers (CSP) using graph-based measures. Therefore, the connection between AS are constructed and integrated so that they can be described on the least granular level. Findings suggest that the AS CSP appear to show better connectivity measures than those of average AS. The results are used to differentiate between CSP which are well connected and those which show an increased probability to suffer from internet outages due to lacking robustness. By using the researched information CSP can enhance customer experience on the demand side as well as performance evaluation on the supply side [26].

Interest in internet topology research using graph theory and underlying measures has gained augmented devotion in the past years [27]. Baumann et al. have explored the Bitcoin transaction graph using graph measures and NetworkX to investigate the anonymity and economic relationships in Bitcoin [28]. Despite this, at the point of this research no previous scientific work regarding the analysis and especially connectivity analysis of CEM could be found. In general, researchers had different ways of approaching internet topology analysis. Faloutsos et al. [29] approach was to conduct internet topology using power laws, describing the distributions of relevant graph-based measures by power-law components which exist and hold in different Internet instances [30] whereas Mahadevan [27] computes different graph measures which illustrate the Internet structure on AS-level based on a variety of data sources [27].

The importance of research on the correct functioning of the Internet in the context of platform and multi-sided markets can be derived from the fact that availability and reliability of the latter's services play a fundamental role given their strategic and operational management in today's society [4]. Especially the rationales derived from the field of the financial sector, like perceived security, usability and hereby created trust [31] [32] can be taken on to the field of CEM, which can be classified as a disruptive innovation of the financial sector, its middlemen and its products.

After having elaborated relevant theories related work regarding internet topology analysis we define a precise procedure model.

III. Method & Tools

In this chapter we introduce procedure models and methodologies, which we used to create a systematic research approach. Specifically, we illustrate commonly used data mining models, which we have tailored to our research purpose. Furthermore we identify and explain relevant tools and demonstrate their practicality for the analysis..

A. General Procedure Models: CRISP-DM and KDD

The Cross Industrial Standard Process for Data Mining (CRISP-DM) and Knowledge Discovery in Databases (KDD) theory are applicable for our research. We explain relations between the two models and demonstrate the important steps to receive a clear understanding of what the problem resolving process of each model involves.

The main procedure model that has been used is the CRISP-DM which is a process specialized to perform data mining and provides a systematic procedure cycle, that could be used to conduct a data mining project [33]. Chapman et al [33] define six phases which are interrelated to each other and where the sequence of the phases is not rigid. Moreover, it is encouraged to move back and forth between the steps and to iterate through the process model until promising and reliable solutions for a given problem are found. Figure 4 illustrates the CRISP-DM procedure and the individual steps.

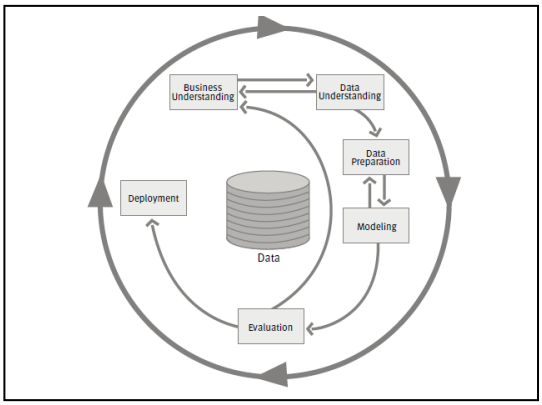

Figure 4: CRISP-DM [32]

The process model follows a step-wise approach, which is structured in six parts: business understanding, data understanding, data preparation, modeling, evaluation and deployment. Iterating multiple times through these steps should be seen as a rule and may be necessary to achieve good results.

The initial step of the methodology, business understanding, is necessary to achieve an understanding of the project objectives and translating them into a data mining problem. Additionally, it requires the design of a preliminary plan to achieve those objectives [33]. It is important to define the exact problem and the benefits of the undertaking. Furthermore, a possible solution must be roughly defined and profound knowledge of the domain and its background must be acquired.

In the second phase, data understanding, the initial process of data collection, followed by tasks which enable the person to fully understand the data they are working with, is central. A first description of the collected data as well as the identification of data quality problems and first insights into the collected data may be used to gain familiarity with the

data basis and gain a deep understanding of the latter [33]. It is important to define what kind of information is available for the analysis and to conclude whether it is relevant or not for previously defined goals. Additionally, it is required to decide if the data quality, quantity and actuality are sufficient enough for the underlying tasks. If that is not the case and the data does not suit the problem well, one should traverse the methodology one step backwards to business understanding to either redefine the problem or to revise the stated objectives.

In the data preparation phase, the final data set is to be built. This data set is going the analysis object derived from the initial raw data. Those preparation tasks are likely to be performed multiple times. Example tasks are the selection of the relevant data, depending on rationale for inclusion or exclusion, cleaning of the data, construction of new features and/or records, integration of data from different sources and data reformatting [33]. It is essential to decide on which data to concentrate on as well as to transform the data in such a way that it is best for the modeling part. Also, techniques to augment the data quality can be applied in this part.

The following phase, modeling, deals with selection and application of different modeling techniques. Typically, there are different data mining methods which are suited for one problem. One can go back and forth between the data preparation and the modeling tasks to experiment with different data mining modeling techniques [33]. It is important to select a modeling architecture which suits the problem best, to choose a good technique and method to create the model and finally to see how good the model performs technically.

Evaluation is the phase which serves the purpose of thoroughly evaluating and reviewing the steps executed to create the model. It should be proven that high quality standards have been achieved from a data analysis perspective and furthermore that the model satisfies the previously defined business objectives [33]. In other terms, it is essential to define how good the model performs subject to the project requirements and to decide on key lessons learned from the project. If the defined business objectives are not achieved, the project is due to be discontinued. If they are partially achieved the objectives may be revised. Upon successful achievement of the defined objectives, one may proceed to the final step of the CRISP-DM methodology.

In this final step, deployment, the organization and presentation of the knowledge gained with the model play the central role. These must be presented and especially deployed in a way, that is satisfactory and extend to the customer. It may be applied live within an organization or can also be as simple as a report summing up the most important insights [33].

Data mining itself is often set in the broader context of Knowledge Discovery in Databases. The CRISP-DM may be seen as a step-by-step and definitive implementation of the KDD [34], which is characterized by the following: selection of relevant data, preprocessing of selected data, transformation of the preprocessed data, data mining within the transformed data and finally the interpretation of patterns derived from the data mining step, leading to a gain of knowledge [35].

Similarities between both procedure models can be identified. It is observable that the CRISP-DM methodology incorporates the steps that must precede and follow the KDD process and gives advice on the practical implementation of those [34].

B. Modified Procedure Model

In this section we use the previously highlighted phases of the CRISP-DM methodology as a basis to create a procedure for our specific research purpose alter it to fit our requirements.

CRISP-DM is primarily meant to be used on data mining problems within a practical context, e.g. in a corporation and is therefore not inherently used for simpler data analysis tasks. Still, most of the steps within the procedure model are well suited to perform data analysis in a highly structured way, so that while some alterations are necessary, a solid base for the underlying network analysis of this paper is already given by the CRISP-DM.

The modeling step of the procedure model has been changed to network analysis. In this phase, we analyze and quantify the connectivity of CEM by using graph measures. No data mining model is implemented. Therefore, the steps known as the selection of different data mining models and techniques as well as the step which deals with the generation of a test design are not required [33]. Only the implementation of the network analysis model as well as the assessment and the summarization of the results are taken into consideration.

A deployment of the underlying research project is not seen necessary for the scope of this project. While CEM could use the results of our analysis to improve their connectivity, this is not a weighty reason for the research to be considered as shaping or reshaping a business environment. We therefore omitted the deployment step from the CRISP-DM to fit our procedure.

Our altered CRISP-DM methodology with dependencies between the procedural steps is shown in Figure 5.

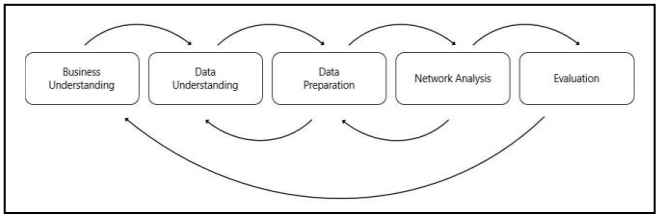

FIGURE 5: ALTERED CRISP-DM METHODOLOGY

C. Tools

In this section of the chapter we present the tools which were used for the realization of this project. Characteristics and the specific use of the tools are elaborated.

The most important tool to perform the given task was the programming language Python in conjunction with its numerous built-in methods as well as openly available libraries which can be used in different scenarios. It was heavily used for this project in terms of data collection, description but also for the network analysis part. “Python is

powerful… and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open" [36]. Python is seen as a programming language which is easy to use and efficient. It was mainly used together with Spyder, a well-known integrated development environment, which is tailored towards data analysis and is part of the computer program Anaconda.

Postman is another tool which was used to test GET and POST requests [37] on different web-addresses mainly in the data collection part of this work. After successful testing, the requests were integrated in the Python scripts, which is detailed in Chapter IV.

Git & GitHub as well as Google Drive were used for project collaboration and task coordination. GitHub is mainly used as a web hosted service which enables version control using Git [38]. Using GitHub we managed to work on the same code and on different features simultaneously without worrying about code-merging, task management etc. Google Drive was solely used to share resources like literature, presentations and other interesting topics regarding the paper.

To organize the group work and creating a roadmap a GANTT chart is used. Setting internal deadlines allowed us to better organize ourselves. The initial project plan is shown in Figure 6.

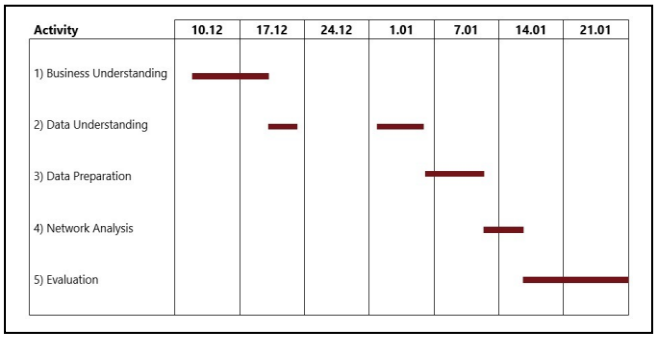

FIGURE 6: RESEARCH PROCESS

IV. DATA & ALGORITHMS

This chapter of the paper illustrates the steps required to perform the network analysis. After elaborating how the data was collected, the data is described, explored and then processed for the analysis. The procedure is explained by diving into the code which has been used for the different steps. Additionally, topological graph-measures are described in detail.

A. Data Collection

Recall, that data collection is the initial step of the data understanding part of the CRISP-DM methodology [33]. All data sources, their previous and actual locations as well as their content are elaborated.

The data of all relevant CEM was collected from different data sources and websites. For this, a web-scraper was developed with the programming language Python. As some steps in this process may take some time, interim results were continuously stored in separate .csv files to facilitate the reloading of the latter into the integrated development environment. The web-scraper iterates through every CEM listed on the CoinMarketCap website [39] and over all the pages. It collects information associated with the CEM, e.g. name and URL. For the implementation of the web-scraper different kind of Python libraries were needed. The most important ones where:

- The "request" library, allowing to send HTTP/1.1 requests, without the need for manual labor. There was no need to manually modify query strings to the URL or to form-encode POST data. Keep-alive and HTTP connection pooling were fully automated [40].
- The "BeautifulSoup" library allowed us to parse the HTML code and extract relevant data from the latter. The Python library was used for pulling data out of HTML and XML files. It works by providing idiomatic ways of navigating, searching and modifying the parse tree [41].
- The Python standard "socket" library is a low-level networking interface used to get the IP addresses of the previously identified URLs [42].
- The "re" library was used to formulate regular expressions in order to modify and extract strings. The library provides regular expression matching operations on patterns and strings [43].
- The "pandas" library provides high-performance, easy to use data structures and data analysis tools. It was mainly used for reading and writing data in between memory data structures, but also different formats as well as column insertions and deletions on the data frame which were needed for the analysis [44].

Packages were installed over the anaconda command prompt called -conda. Later they have been imported into the Python execution environment as seen in following Figure 7:

```
from requests import get
from requests import post
from bs4 import BeautifulSoup
import pandas as pd
import time
import socket
import re
```

FIGURE 7: IMPORT OF RELEVANT PACKAGES

First of all, the crawl function performs a simple GET request on the main webpage containing all the rankings of the CEM. It starts with the first page containing the first 100 CEM names and iterates through until the last page. A simple *getUrl()* function checks the response code of the GET request which is useful for error handling and debugging. If nothing goes wrong and the status code is 200 the content of the html code is returned. The *inspectFile()* function than parses the returned html string and stores the ranking table in form of a Python DataFrame. The *getInfo()* function then goes row by row and first selects the internal CoinMarketCap URL of the given CEM, performs a GET request on that internal CoinMarketCap page containing the website URL information and finally extracts the definitive website URL from that page. When finished the results are exported into a .csv file so that the code does not need to be rerun every time there is a change but the possibility of just reloading the .csv

is existent. The main structure of the code can be seen on the Figure 8 .

```
def crawl():
    global dfs
    dfs = []
    for i in range(1,4):
        print(i)
        url = "https://coinmarketcap.com/rankings/exchanges/"+str(i)
        ans = getUrl(url)
        html = inspectFile(ans.content)
        df = getInfo(html)
        dfs.append(df)
    dfs = pd.concat(dfs, sort = False)
    dfs.to_csv('./out.csv',index = False)
    return dfs
```

FIGURE 8: CRAWL FUNCTION

This process is repeated for every CEM listed on every page of the ranking, so that in the end we have a table with the following columns:

- Name of CEM
- Internal Page of the CEM on the CoinMarketCap website, which may look like: *"https://coinmarketcap.com/exchanges/binance/"*. It is only stored temporarily because it is only needed to extract the website URL of the CEM.
- siteURL: indicating the identified URL of the given CEM

The second column containing the internal page URL is removed from the data frame so that we have a table as illustrated in Figure 9.

	name	siteUrl
2	Binance	https://www.binance.com/
3	Bit-Z	https://www.bit-z.com
4	OKEx	https://www.okex.com
5	Huobi	https://www.hbg.com/
6	ZB.COM	https://www.zb.com/

FIGURE 9: OUTPUT OF WEB-SCRAPING

In a second step the internet protocol (IP) address for the relevant website URL is identified. Therefore, some regular expressions are needed to bring the records of the siteUrl column into a format that is accepted when passed on to the *socket.gethostbyname(url)* function. If there are more than three backslashes in the URL the content in between the second and the last slashes are extracted. E.g. *"https://www.binance.com/"* becomes *"www.binance.com"*. If there are less than three backslashes only the substring after the last two backslashes are selected. E.g. *"https://www.bit-z.com"* becomes "*www.bit-z.com"*. The results are passed on to a function to determine the IP addresses which are then appended to the data frame in form of a third column.

To get the AS-numbers, two different online sources were identified. The first tool which has been used is called UltraTools [45], after putting in an IP address, it returns a set of data containing information about; IP owner, registration date, issuing registrar and the autonomous system number as seen in Figure 10.

```
AS16509
          Country: US
Registration Date: 2000-05-04
        Registrar: arin
            Owner: AMAZON-02 - Amazon.com, Inc., US
```

FIGURE 10: SAMPLE AS INFORMATION

To get the information a simple GET request is performed and the results are stored in a forth column of the table; containing every AS-number for every CEM. The second tool which has been used is called TracerouteOnline [46]. Like UltraTools it only needs an IP address and returns a comma separated list of the AS-number and other details. The AS-number is extracted and appended to the initial table in a fifth column. A sample output can be seen in Figure 11.

```
"16509","52.84.160.0/22","AMAZON-02 - Amazon.com, Inc., US"
```

FIGURE 11: SAMPLE OUTPUT

The only difference between the two sources is that for the TracerouteOnline tool a post request needs to be sent to the website, which then returns the needed information. Therefore, some testing was performed in Postman and then implemented in Python. The post request is illustrated in the following figure.

After having identified all AS-numbers, the information of both sources is crosschecked. In other terms, the AS-numbers which have been extracted from the first source are compared to those which have been extracted from the second source. A sixth column indicating whether the numbers are the same is than added to the final table in form of a last column. The end result is a data set containing all relevant information of all listed CEM.

To perform a network analysis on the connectivity of the scraped AS-numbers an edgelist containing all numbers as well as their links was required. Therefore, the AS-Links dataset from CAIDA was selected and locally stored. CAIDA is one of the most renowned institutions for Internet topology research and based at the University of California San Diego's Supercomputer Center. CAIDA systematically collects topology data, analyzes it and provides the results to the public [21]. Additionally, numerous tools for own topological analyses are made available. As part of the "Macroscopic Topology Project", the AS Links dataset is published regularly. It is constantly updated using the Archipelago measurement infrastructure, which is the direct successor of the "skitter" platform. As part of this project Raspberry Pi-based monitors are placed in geographically dispersed locations, mainly in academic institutions. At the point of our data collection process, the most recent dataset was from the 3rd of January, 2019. The AS Links dataset is also known as IPv4 Routed /24 AS Links Dataset and provides regular snapshots of AS Links derived from the ongoing traceroute-like-IP-level topology measurements, which are part of the Ark infrastructure. These are processed using RouteViews BGP tables which are released to the public to identify AS associated with each corresponding IP address. These are then collapsed into a set of links between AS using the original probed IP paths [21].

In summary, two data sets were built or selected to perform the network analysis: the web-scraped AS-numbers of CEM data set and the CAIDA AS Links dataset containing

a compilation of tracerouted AS-numbers and their links, which are described in the following chapter.

B. Data Description

This section of the chapter deals with the examination of the surface properties of the acquired data and reports on the results. The structure, the format, and the quantity of the data and similar surface features are elaborated to develop a basic understanding of the data which is used for the analysis part [33].

The data which has been scraped from UltraTools and TracerouteOnline are structured. Therefore, it contains rows and columns which can be easily addressed to perform different kind of analyses. In this case, the collected dataset contains 232 rows and 6 columns and can be visualized as seen in Table I.

TABLE I. HEAD OF WEB-SCRAPED DATA

Name (CEM)	**URL**	**IP**	**ASN1**	**ASN2**	**Cross-Check**
Binance	www.binance.com	52.84.161.223	16509	16509	True
Bit-z	www.bit-z.com	104.20.187.100	13335	13335	True
OKex	www.okex.com	104.19.213.87	13335	13335	True
Huobi	www.hbg.com	104.17.199.190	13335	13335	True
ZB	www.zb.com	203.90.247.83	55355	55355	True

The structured dataset has been saved locally in a .csv format. The file size is approximately 15kB and can be imported into Python as a Pandas data frame to perform further analysis.

The CAIDA AS-links dataset is unstructured, meaning the data is not organized or assembled in a predefined manner and does not have a pre-defined data model. It is therefore difficult to save it in a relational database [47]. Managing and analyzing unstructured data may differ from the one of structured data. It has therefore been saved locally in form of a text file, with a file size of 4.6MB. It can be imported into Python for further string manipulations and analysis.

C. Data Exploration

Basic querying, descriptive statistics and other exploration tasks of the collected data are described. Field types, distributions, unique values, frequencies of most occurring values as well as missing values are further elaborated and examined.

The web-scraped data, which was described in the previous section of the paper, counts 232 different records of CEM. The columns name, siteURL contain string values, whereas the identified AS-numbers are integer values. The crosscheck column contains Boolean values. Each name and URL are distinct unique values whereas the IP “107.154.248.133” is found thrice in the dataset. Therefore the unique count of all IP records equals 230.

34 out of 232 of the AS-numbers which were identified with UltraTools are unique. The most frequent AS-number is 13335, with an occurrence of 140. Also, 9 missing values have been identified and marked as 0 in the dataset. Using TracerouteOnline 35 unique AS-numbers were identified. The value occuring the most is the AS13335 with a frequency of 140. No missing values have been identified using TracerouteOnline.

The crosscheck column containing the Boolean variable has two logically distinct values whereas the most occurring value is “True” with a frequency of 222. Therefore, ten AS were not equal when comparing the output of UltraTools and TracerouteOnline. Nine of these values can be explained due to the existence of missing values, which could not be scraped using UltraTools. This implies that 35 unique AS will be analyzed as part of our research. Table II summarizes the results of our data exploration process.

TABLE II. SUMMARY OF DATA EXPLORATION RESULTS

Column	**Type**	**Count**	**Unique**	**Top**	**Freq**	**MV**
Name	String	232	232	-	-	-
URL	String	232	232	-	-	-
IP	String	232	230	s.o	2	-
ASN1	Integer	232	34	13335	140	9
ASN2	Integer	232	35	13335	140	-
Cross-check	Boolean	232	2	True	222	-

The unedited AS Links dataset consisted of 127.201 rows. A majority of these represent links between AS, which do not necessarily have to be one-hop distances. 59.885 indirect AS-links are contained in the dataset. Furthermore metadata such as dataset and measurement specific make up a further 640 rows. Lastly, the dataset also contains monitor keys and multi-origin AS (MOAS). In the following step, the original dataset is cleaned to only include one-hop distance AS links.

D. Data Processing

We will now discuss the preprocessing and preparation of the data that has been done for further analysis. Important transformations are performed on the web-scraped data and on the CAIDA dataset.

The initial state web-scraped data contains 232 different records and six columns. Only the AS-numbers are necessary for the graph analysis. Therefore unnecessary columns were removed from the dataset. This step can be considered to be similar to the feature selection part of data mining models, in which only the relevant data is selected. In this case the AS-numbers are the most relevant ones. Columns containing the web-scraped AS-numbers from different source are merged. Then previously identified missing and all duplicates are deleted from the dataset. The final state of the web-scraped data consists of 1 column and 35 records, which indicate unique AS-numbers.

To both structure the AS Links dataset and remove unnecessary information, a total of 58.534 rows had to be removed. These included links that were not within a one-hop distance, metadata, and other invalid data. Additionally, we created new lines for AS that were previously aggregated within one field due to being connected to a MOAS.

E. Data Quality

The quality of the processed data is described. Alongside, we examine the syntactical and semantical correctness, the completeness and the actuality of this data.

From a syntactical point of view the web-scraped data does not show any errors. The CEM names as well as the other columns values were directly extracted from the different data sources. The columns name, siteURL and IP only contain string values and no other types. The AS-numbers columns contain only integer values. Nine missing values have been identified in the ASN1 column. The failure of identification of AS-numbers by UltraTools, may be due to limitations within the website, that did not allow our script to fully run. These limitations were not set in place by TracerouteOnline as no missing values were found.

From a semantical point of view the web-scraped data does not contain any errors. These values are therefore subject to change due to the dynamic environment of the internet.

The completeness of the scraped data is limited to the fact that only CEM which were listed and ranked on CoinMarketCap have been included in our dataset. This list may not be complete, yet gives an overview of commonly used CEM based on their daily trading volumes. Furthermore, two different sources have been used to determine the AS-numbers of CEM. For the scope of our analysis, we deem this as satisfactory.

The web-scraped data is considered up to date. Although the data was scraped in January 2019, repeating the web-scraping process at any other moment would likely yield different results than before, due to constantly changing AS-numbers.

The AS Links dataset is, at the current point of time, up to date. The dataset is not complete, as it is impossible to determine every existing edge between two nodes, whether these represent AS, routers, etc., that is found in the internet [24]. In comparison to other datasets, e.g. potaroo, the AS Links dataset provided by CAIDA seems most appropriate, as it contains more links and is updated on a more frequent scale. Both the syntactical and semantical correctness are further limited due to the data collection being out of our hands and the dynamic nature of the internet.

F. Graph Metrics & Algorithms

Recall that complex networks are characterized by large amounts of connected nodes. By using certain metrics, the process of understanding and analyzing the network can be significantly facilitated. As a sheer number of graph metrics exist, which can be used for their individual purposes, a specific selection for the underlying research purpose is necessary. We choose to use topological graph metrics as they provide a method to analyze the topological structure of complex networks. Furthermore, they can be applied either for a global or local view of the network. These topological measures can be used to derive information about the connectivity of a network based on the defined topological criterion, thus allowing to analyze the connectivity of CEM.

Table III gives an overview of existing topological metrics from the field of graph theory, which we have selected for our research purpose. The selection of metrics is based on a review of related work [20] [22] [23] [25] [26].

TABLE III. OVERVIEW OF TOPOLOGICAL GRAPH MEASURES

Distance Measures	**Centrality Measures**	**Neighborhood Measures**
Eccentricity (ECC_i)	Degree Centrality (DC_i)	Local Average Neighbor Degree ($LAND_i$)
Single Source Shortest Path Length ($SSSPL_i$)	Eigenvector Centrality (EC_i)	Clustering Coefficient (CC_i)
	Betweenness Centrality (BC_i)	Local Node Connectivity (LNC_i)
	Closeness Centrality (CLC_i)	

1) Distance Measures

Within graph theory the concept of reachability, which describes the possibility of going from one node to another following the connections given by edges in the network, reflects the inherent design principle of the internet: each device may, in an optimal scenario, reach any other device, that is as far away as possible [48]. If a path between a pair of nodes exists, a connection is present. In most large-scale graphs not all nodes are connected with each other, resulting in a graph with several disconnected nodes. In order to quantify the distances between connected nodes, distance measures are introduced. The number of edges (also: number of hops) when a path is traversed between two nodes *i* and *j*, which must not be disconnected, is known as the path length or distance d_{ij}. The shortest path between two nodes, is therefore the path with the least number of hops. Two local measures for distance with a perspective of a single node were selected accordingly.

The *Single Source Shortest Path Length* (SSSPL) builds on the above concepts and analyzes the average length of shortest paths connecting a source node *i* with all other reachable nodes *j* ($j \neq i$) in the network [49]. Therefore a small SSSPL value of a node *i* is used as an indicator for a better connectivity, as this implies that the source node is close to the center of the network [26].

The second measure is the *eccentricity* (ECC) of a node *I* and represents the greatest distance between *i* and all other nodes *j* in the network. Accordingly, the center of the network can be defined as the nodes with the smallest ECC, which implies that the larger the ECC of a node *i*, the farther away it is topologically from the center of the [26].

$$ECC_i = \max\{d_{ij} : \mathrm{j} \in \mathrm{N}\} \qquad (1)$$

Distance measures are therefore used to analyze the topological closeness of a node to the rest of the network and are a good indicator for the connectivity of a node. The

distance of a node within a complex network is not to be confused with the geographic distance between two points, e.g. server-hosts of CEM [50].

2) Centrality Measures

Centrality measures are used to determine information about the relative importance and connectivity of a node in a network [51]. Commonly used centrality measures are *degree centrality*, *eigenvector centrality*, *betweenness centrality*, and *closeness centrality* [26].

In graph theory one of the fundamental characteristics of a node *i* is the number of other nodes it is directly, therefore within one edge, connected to [52]. This attribute is referred to as *degree centrality* (DC) and denoted as following for a node *i*:

$$DC_i = \sum_{j=1}^{N} a_{ij} \qquad (2)$$

With N being the total number of nodes within the network, and a_{ij} being a binary variable. If a link exists between the source node *i* and neighboring node *j*, a_{ij} is equal to one and else zero [52]. A property of the DC is, that the value uniformly increases with the number of its direct neighbors. As the characteristics of nodes naturally vary, the importance of one node increases with the importance of its neighbors [24].

Therefore *eigenvector centrality* (EC) extends the view of DC and is used to calculate the importance of a source node proportional to the sum of the importance of its neighbors [48]. The measure works best for undirected graphs, due to only one leading eigenvector being calculated in comparison to two in direct graphs [24]. Additionally, only one-hop distances are considered in this measure. EC is calculated as the maximum eigenvalue of the network's adjacency matrix A. The formula is therefore similar to DC, but in addition to the adjacency matrix $A = a_{ij}$ the eigenvector $x = (x_1, \dots, x_N)^T$ of the maximum eigenvalue $\lambda_{\max(A)}\ of\ A$ is derived [26].

$$EC_i = \frac{1}{\lambda_{\max(A)}} \sum_{j=1}^{N} a_{ij} x_j \qquad (3)$$

A central characteristic of the EC is that the importance of a node may be higher, the more well-connected neighboring nodes it has. Thus generally, the higher the EC of a node the better connected it is in the network [53]. It is important to note, that the first connection to a central node may just be a single link, therefore EC and DC do not have the same implications regarding the connectivity of a node [26].

The measure of *betweenness centrality* (BC) is related to the shortest path length. If the shortest paths between all pairs of nodes in the network were to be taken, it is implied that some edges are traversed more often than others [48]. BC quantifies this idea and calculates the number of shortest paths which have passed through a node *i*. Thus BC can also be seen as an inverse measure of network resilience. If a node with high BC were to be removed, the implications for the entire network would be more severe than if a node with low BC would be removed [54]. Therefore BC can be used when looking for potential bottlenecks in the network. With the removal of such bottlenecks the network will in turn become more robust [55]. The measure is commonly normalized by $N(N-1)$, which is the greatest possible BC of a node, turning it into a binary variable. With $n_{j,k}$ being the number of shortest paths between nodes *j* and *k*, and $n_{j,k}(i)$ denoting the number of shortest paths traversing *i*, the BC for a node *i* is defined as [26]:

$$BC_i = \sum_{j,k} \frac{n_{j,k}(i)}{n_{j,k}} \qquad (4)$$

Lastly, the *closeness centrality* (CLC), which measures the average distance from a source node *i* to all other nodes, is introduced [48]. Similar to BC, CLC is related to the shortest paths. The standardized CLC is denoted by the inverse sum of the length of distances from node *i* to all other $N-1$ nodes [48]. A high CLC value implies, that the distance between a source node *i* and all others is shorter, indicating greater centrality and connectivity [26]. The shortest path distance between *i* and *u* is denoted as $d_{i,u}$.

$$CLC_i = \frac{N-1}{\sum_{u=1}^{N} d_{i,u}} \qquad (5)$$

As previously mentioned, centrality measures give an indication of the relative importance of individual nodes. There are two uses of centrality measures with respect to network analysis. Firstly, nodes can be ranked according to their importance and the different values can be compared. Secondly, the individual centrality values can be averaged to compare different networks [24].

3) Neighborhood Measures

Neighborhood measures take a node's neighbors and neighborhood characteristics into account. In this context the *local average neighbor degree* (LAND), *clustering coefficient* (CC), *average clustering coefficient* (ACC), and *local node connectivity* (LNC) will be introduced. We categorize these measures as neighbor measures to ensure a clear structure and classification of graph measures.

LAND takes the amount of neighboring nodes and their individual degrees, thus describing how well the node is connected, regarding the available alternative paths [26]. Additionally, LAND provides information about whether a node, in this context an AS, prefers connecting to high-degree or low-degree neighbors [26]. Therefore a high LAND value is an indicator for a better connectivity. The LAND for a node *i*, where $U(i)$ represents the number of one-hop distance neighbors of *i* and k_u quantifies the degree of those neighbors, is denoted as [49]:

$$LAND_i = \frac{1}{|\mathrm{U}(i)|} \sum_{u \in U(i)} k_u \qquad (6)$$

The *clustering coefficient* is a local metric, which indicates the tendency of neighboring nodes connecting to each other. It is defined as the ratio of the number of edges existing between the direct neighbors of a node *i* (L_i) and the maximum possible value [48]. The maximum value is given by $k_i(k_i-1)$, where *k* represents the degree of *i*. The CC of a node *i* is therefore denoted as [48]:

$$CC_i = \frac{2L_i}{k_i(k_i - 1)} \quad (7)$$

Thus, the CC measures the probability that the neighbors of a node are well connected and takes on values between zero and one [56]. A CC equal to one would indicate, that two neighbors of a node *i* are, in all cases, connected. The CC is an important metric, because it displays that alternative paths can be used to reroute traffic and communication information, which can be helpful regarding potential link failures [55]. It is also possible to calculate the *average clustering coefficient* (ACC) for the entire network, which is the mean CC of all nodes. The ACC is an indicator for the local robustness of the whole network [57].

$$ACC_i = \frac{1}{n}\sum_{i=1}^{N} CC_i \quad (8)$$

The final measure is the *local node connectivity*. To calculate the LNC a source and a target node, that are not direct neighbors, are required. The measure determines the minimum amount of nodes and their respective edges that need to be removed in order to disconnect the source and target node by destroying all paths between [49]. Therefore the higher the LNC, the better connected and more robust the underlying node is.

4) NetworkX: Metrics Example

NetworkX is a Python package used for generation, manipulation and analysis of complex networks [49]. We used NetworkX for our research, as it has successfully been used for previous analyses of network graphs.

To illustrate the presented metrics and demonstrate the algorithms provided by NetworkX, we have used the exemplary graph from Chapter II.A to create a metrics example as seen in Table IV. From the results it is possible to show, that node C is the least well-connected, due to a low DC and high SSSPL score. High CLC scores assumed by node A and D show that they are more central within the structure of the network, which is further underlined by their high EC and CLC scores.

TABLE IV. METRICS EXAMPLE

Node	ECC	SSSPL	DC	EC	BC	CLC	LAND	CC
A	2	1.29	3	0.5	0.1	0.7	3.3	0.7
B	3	1.43	3	0.4	0.3	0.6	2.7	0.3
C	4	2.14	1	0.1	0	0.4	3	0
D	2	1.14	4	0.5	0.3	0.8	2.8	0.3
E	3	1.43	3	0.4	0.2	0.6	3	0.3
F	4	1.86	2	0.2	0	0.5	2.5	0
G	3	1.57	2	0.3	0.1	0.6	3	0

In this chapter, we introduced measures from graph theory, which were specifically selected for the underlying research purpose. Using these measures we can analyze the connectivity of nodes, their neighbors, and of the entire network. In the context of this research nodes are classified as AS, which represent at least one CEM. Lastly, to understand the functionalities and semantics of NetworkX, we introduced NetworkX, which will, as presented alongside the metrics example, be used to:

A) Create a global-scale network graph
B) Calculate the aforementioned graph measures.

V. ANALYSIS RESULTS

In the following section we present the results of our analysis. We begin by giving an overview of descriptive statistics for the relevant graph measures regarding the global-scale network. The main part of this chapter lies on the analysis of individual measures and comparing the values of the CEM subset with the global benchmarks. The AS-level graph is generated using the aforementioned CAIDA dataset of AS Links. Using the web-scraped AS-numbers of CEM we extract their individual topological measures from the global-scale network graph.

We begin by presenting key properties of our global-scale AS-level graph. Using the function *nx.info()*, with *nx* referring to NetworkX, we are returned the amount of nodes and edges within the graph and the average degree (= degree centrality) of our network, as seen in Figure 12.

```
Type: Graph
Number of nodes: 31418
Number of edges: 63348
Average degree:   4.0326
```

FIGURE 12: PROPERTIES OF GLOBAL-SCALE GRAPH

For the global-scale graph, the arithmetic mean, median, and variance are given in Table V. In our analysis these values play a central role towards comparing the graph measures of CEM with the global benchmarks, checking for correctness and sanity, and identifying outliers. An overview of graph-based measures for CEM is presented in Appendix II.

TABLE V. GRAPH MEASURES FOR GLOBAL-SCALE GRAPH

Graph-based measure	Mean	Median	Variance
Eccentricity (ECC)	6.96	7.00	0.44
Single Source Shortest Path Length (SSSPL)	3.82	3.84	0.33
Degree Centrality (DC)	4.03	1.00	1228.70
Eigenvector Centrality (EC)	1.91-e3	3.47e-4	2.81e-5
Betweenness Centrality (BC)	8.94e-5	0.00	4.15e-6
Closeness Centrality (CLC)	0.26	0.26	1.61e-3
Local Average Neighbor Degree (LAND)	446.97	99.50	55.57e4
Clustering Coefficient (CC)	0.23	0.00	0.15

A. *Analysis of Connectivity Results*

For the graph analysis we have computed nine graph-based measures. While the LNC has been calculated only for the AS of CEM, the other measures have been calculated for every node in the network. It is important to note that for six CEM no measures were able to be calculated, due to not being included in the CAIDA dataset. The specifics and possible causes for this occurrence will be discussed in Chapter VI. It is also important to note that despite a total of 232 CEM being web-scraped, only 35 unique AS exist, as mentioned in Chapter III.

1) Distance Measures

We begin by presenting the analysis results for distance measures, namely ECC and SSSPL.

a) Eccentricity

Recall that the ECC is used to indicate how close an AS is to the center of the network. Specifically, small values imply that an AS is closer to the network's center and thus better connected [26].

The ECC for AS of CEM ranges between 6 and 8 with a right-skewed distribution as shown in Figure 13. 25 of the 35 (73.53%) AS have an ECC of 6, with only four AS surpassing this value.

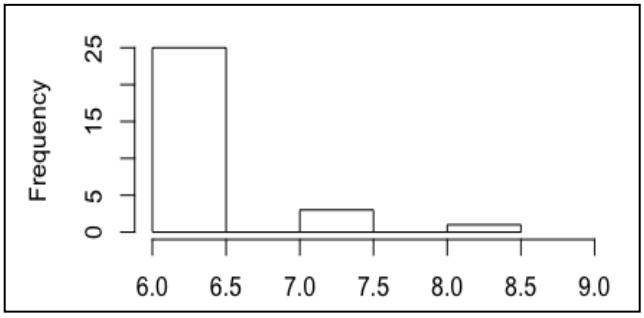

FIGURE 13: ECCENTRICITY OF CEM

This results in a mean of 6.17, which in comparison with the global mean is a significant improvement. The global mean is equal to 6.96. Using summary statistics we derive, that 86.2% of CEM are better connected than the average global AS. An interesting point is that while the distribution of CEM is right-skewed, the global-graph distribution is, graphically seen, not skewed at all and thus normally distributed as shown in Figure 14. This is likely due to the small sample size of AS in a CEM context.

Regarding the ranking, AS 51558 of Simex is the worst performing CEM with an ECC of 8. Despite the average ECC of CEM being significantly better (lower) than the global mean, Simex only ranks in the 20th percentile of the global ECC benchmark. Other low performers are AS 133296 of Instant Bitex, AS 50473 of Graviex, and AS 39287 of BarterDEX with an ECC of 7. Although in a CEM context these AS are not top performers, they place in the global median and can thus be considered averagely well connected. Overall this shows, that regarding ECC most CEM are well connected, with one low ranking outlier.

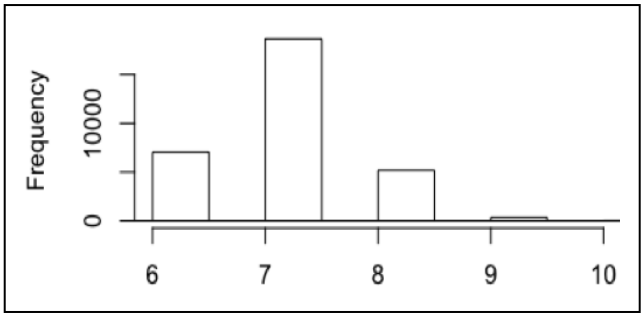

FIGURE 14: GLOBAL ECCENTRICITY

b) Single Source Shortest Path Length

Similar to ECC the SSSPL indicates how close an AS is to the center of the graph. We derive that 96.5% of CEM are better connected than the global mean of 3.82. The mean SSSPL for CEM is 3.078, which can be seen in Figure 15. While some outliers exist, their global placing cannot be considered bad, as they are between the 25th and 50th percentile of the global graph.

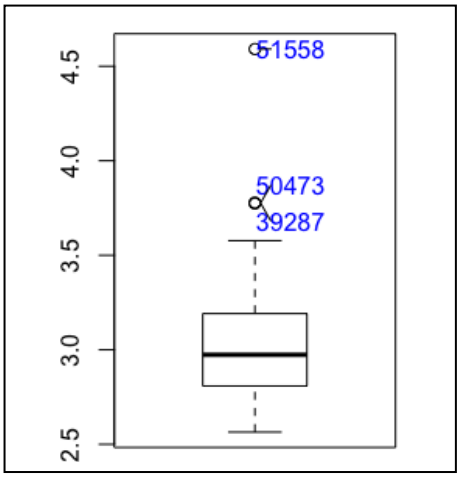

FIGURE 15: SINGLE SOURCE SHORTEST PATH LENGTH OF CEM

The most significant outlier is AS 51558 of Simex with an SSSPL of 4.59. Additionally, Graviex and BarterDex with a respective SSSPL of 3.77 could be classified as outliers in a CEM context. Considering the global mean of 3.82 both of these CEM are better connected than the average global AS. These outliers do not have a strong effect on the mean, as there is only a slight difference when comparing the mean and median values of CEM, which at 2.97 is only 0.1 points smaller.

Thus most CEM are significantly better connected than an average AS. Additionally, bad-ranking CEMs are still well connected in a global view. These results most comply with our analysis of the ECC. Simex remains the worst-performing CEM, while Graviex and BarterDex can still be considered low performers in the context of CEM connectivity. Interestingly, Instant Bitex, which alongside Graviex and BarterDex did not rank well regarding ECC, comparatively performs better with an SSSPL of 3.57.

In the context of distance measures AS20940 of Simex (ECC: 6; SSSPL: 2.55), AS16625 of multiple CEM (ECC: 6; SSSPL: 2.67) and AS4776 of Coinbit (ECC: 6; SSSPL 2.69) were the best connected AS of the underlying CEM.

2) Centrality Measures

Centrality measures determine the relative importance of a node within the network. In the following the analysis results for the aforementioned measures are presented.

a) Degree Centrality

The average DC for CEM is 39, with a median of 19. This dispersion is caused due to outliers as seen in Figure 15.

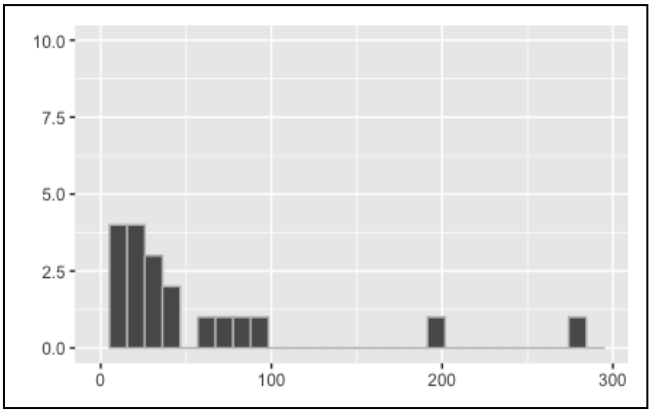

FIGURE 15: DEGREE CENTRALITY OF CEM

Namely, AS 4766 of Coinbit and AS 20940 of Bibox with a DC of 278 and 200 have a large impact on the mean DC of CEM. If we were to exclude Coinbit and Bibox from our analysis, the mean would be much closer to the median with a value of 24.19.

Despite the inflation of the mean being somewhat reduced by excluding outliers, this value is still far from being close to the global mean DC of 4.03. Both for CEM and for the global graph, a pattern can be seen regarding the effect of outliers on the mean. While the global mean DC is relatively small compared to the mean DC of CEM, the difference to the global median DC of 1 is still significant and not be underestimated. This is due to some AS having large degrees, such as 3533 (AS with DC values above 25 have been excluded to reduce graph size), while the majority have small values of either 1 and 2 as shown in Figure 16.

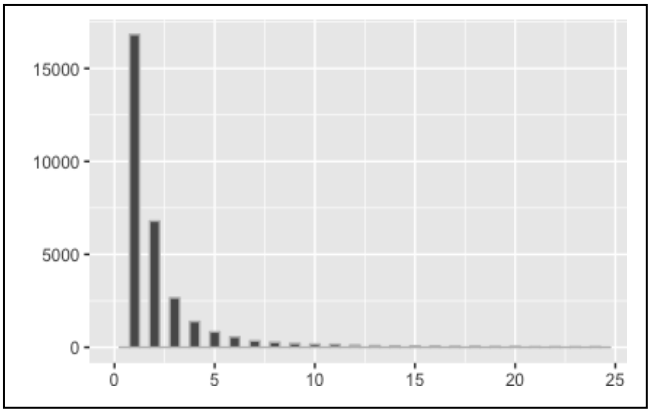

FIGURE 16: GLOBAL DEGREE CENTRALITY

Summarizing our findings regarding DC, CEM are on average better connected than a AS on the global network due to having a higher degree. Coinbit and Bibox are the best connected CEM regarding their DC, while also having low values for ECC and SSSPL, which complies with the characteristics of these measures. Further underlining the correlation of these measures, AS 51558 of Simex has the lowest DC of CEM with a value of 1, while also being the worst-connected CEM in reference to ECC and SSSPL. Unlike previously analyzed measures, outliers have a strong impact on the mean DC value, as underlined by the high variance of 1228.7 and the dispersion between mean and median.

b) Eigenvector Centrality

Nodes with a high EC score are connected to other central nodes, which implies a better connectivity of the source node due to having well connected neighbors.

A majority of CEM, namely 82.76%, have a higher EC than the global benchmark as shown in Figure 17. This is underlined by the mean EC for CEM being 1.98e-2, while the global mean is 1.92e-3. The highest score for a CEM is Bixbox (0.071), followed by Coinbit (0.052) and AS 16625 (0.048), which includes five CEM. Furthermore, AS 15169, which hosts seven CEM, and AS 13335, in which 140 CEM are included, are within the highest ranking with EC scores of 0.042 and 0.039 respectively.

Only three CEM have lower EC scores than the global mean. These are Simex (1.45e-6), BarterDEX (6.12e-5), and Graviex (8.16e-5), all of which were within the worst performing CEM in regards to the distance measures.

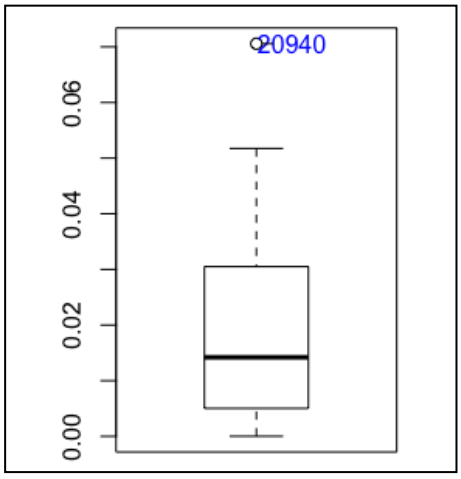

FIGURE 17: EIGENVECTOR CENTRALITY OF CEM

c) Betweenness Centrality

A high BC score for a node indicates that it is more frequently traversed within a shortest path of two other non-adjacent nodes. This implies that the removal of such a node would negatively impact the connectedness of the network from a holistic view and additionally disrupt the flow of network traffic.

The global mean BC score is 8.94e-5, while for CEM this value is 1.36e-3 as shown in Figure 18.

Despite the difference in mean values, only 58.26% of CEM have a higher BC than the global benchmark. Especially in the case of BC analysis the effect of outliers on the mean can be seen. The median BC score for CEM is 2.08e-4, which is significantly closer to the global mean.

A total of five CEM demonstrate prominent BC scores. These are Bixbox (0.0122), Coinbit (0.0118), as well as AS 13335 (0.004), AS 16625 (0.003), and AS15169 (0.002). As shown in our analysis of EC these AS host a large portion of CEM. Unlike for the EC score, AS13335 displays a larger BC

score than AS16625 and AS15169. This is likely due to the fact, that AS13335 has a higher degree (93) than the other two AS (87;76) and is thus closer to the network's center.

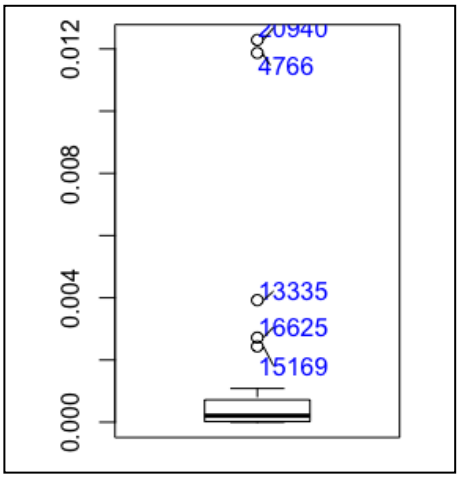

FIGURE 18: BETWEENNESS CENTRALITY OF CEM

d) Closeness Centrality

The final centrality measure is the closeness centrality, which quantifies the importance and role of a node within the network. Therefore nodes with a high CLC score are closer to the network's center, participate in more traffic, and are therefore better connected than nodes with a lower CLC.

Standardized CLC scores for AS of CEM are shown in Figure 19. Bibox (0.388), AS16625 (0.372) and Coinbit (0.370) are closest to the topological graph center. On a global scale, Bibox ranks within the top 0.05% of AS. Simex is the only CEM with a CLC score (0.217) below the global mean of 0.262. With a mean CLC score of 0.328, 96.55% of CEM are better connected than the average AS.

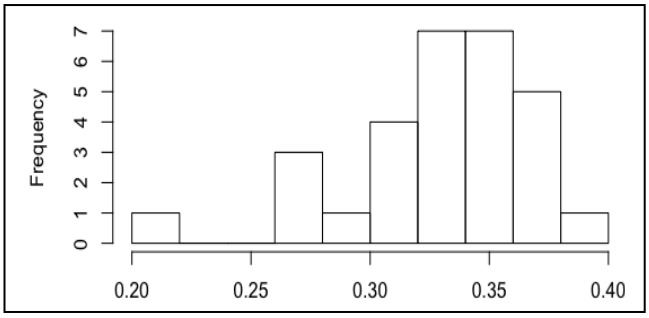

FIGURE 19: CLOSENESS CENTRALITY OF CEM

Summarizing our analysis results of centrality measures, we were able to identify certain patterns of well-connected as well as poorly-connected CEM. Namely Bibox, which for all measures has been the best-connected CEM and has ranked within the top percentile of global connectivity for CLC. Furthermore, Coinbit as well as AS16625, AS15169 and AS13335 demonstrate a high degree of connectivity. Simex, BarterDEX, and Graviex are on the lower spectrum of connectivity derived from centrality measures, when compared to the other CEM. Compared with the global benchmark, their connectivity is somewhat close to the mean values.

3) Neighborhood Measures

Neighborhood measures take a node's neighbors into account. We will present our analysis results for LAND, CC, and LNC.

a) Local Average Neighbor Degree

The LAND measure quantifies the connectivity of a node by averaging the degrees of all its one-hop neighbors. This implies that a source node with a high LAND score has more alternative paths to choose from when one or more paths fails, e.g. due to network outages.

The mean LAND for the entire network is 447. This could imply that, on average, a node has 447 alternative paths to choose from. Although such an interpretation may seem intuitive, it must be stated with caution. As the median is only 99.5, most nodes do not have the previously stated amount of alternative paths. This is caused by outliers with LAND scores well over 3000, oftentimes by institutions such as colleges or corporate networks, while most AS have rather low scores as seen in Figure 20.

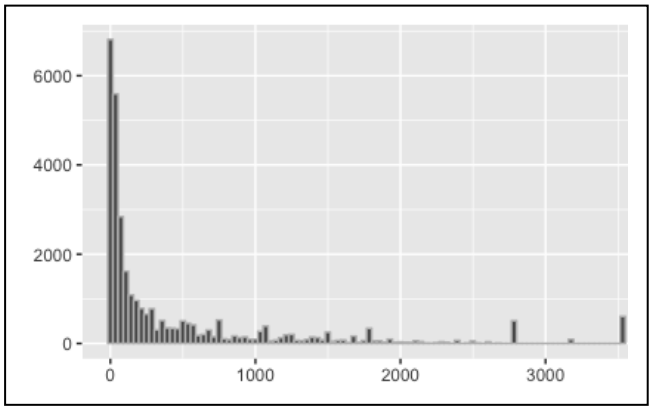

FIGURE 20: GLOBAL LOCAL AVERAGE NEIGHBOR DEGREE

In comparison, the mean LAND score for CEM is 675.6 with a median of 413.4. A dispersion between the mean and median is evident, but not as significant as with the global-scale graph. 44.83% of CEM have a higher LAND score than the average AS, resulting in a more robust network in case of e.g. network outages. CEM with high LAND scores are Exmo (2799), Bits Blockchain and Cryptomate (2527) as depicted in Figure 21. All of these CEM have rather low DC scores between 1 and 5. This is due to the implication, that AS with low degrees tend to connect to well-connected, i.e. in terms of degree, AS. An exception to this rule is Simex, which is as well as a LAND score of 1, is additionally not well-connected regarding centrality and distance measures.

An inherent problem with the implication that nodes with a low connectivity connect to nodes with higher degrees is that they are dependent on their neighbor's robustness. Should a network outage occur and the neighboring node be affected, the probability of the source node not being reachable is, on average, higher.

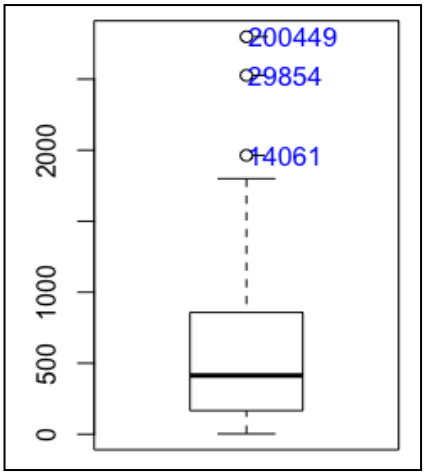

FIGURE 21: LOCAL AVERAGE NEIGHBOR DEGREE OF CEM

b) Clustering Coefficient

The clustering coefficient measures the probability that the neighbors of a node are connected as well, i.e. how densely the neighborhood of a source node is connected. The advantage of inter-connected neighbors is that traffic can be rerouted in case of network outages. Similar to some of the previously mentioned measures, e.g. CLC, CC scores range between 0 and 1.

The mean CC, also known as the average clustering coefficient, is 0.233 for the entire network. Thus, a probability of 23.3% that two neighbors of a node are interconnected. For CEM a 34.7% chance exists, that neighboring nodes are connected with each other. On average, CEM are therefore better connected and more robust to network outages. This is further underlined as 13.8% of CEM have a CC score of 1, resulting in a fully inter-connected neighborhood as shown in Figure 22. These CEM are Bits Blockchain, Bisq, CRXzone, and BarterDEX, which furthermore have a low degree centrality, ranging between 2 and 5. Despite a low DC score, a high CC score can be an indicator for a well-connected node. While such nodes are not connected to many neighbors, the interconnection and high DC scores of their neighbors, which range from 178 (AS47605) to 1858 (AS3257) underline the possibility of a high connectivity and robustness despite a low DC score.

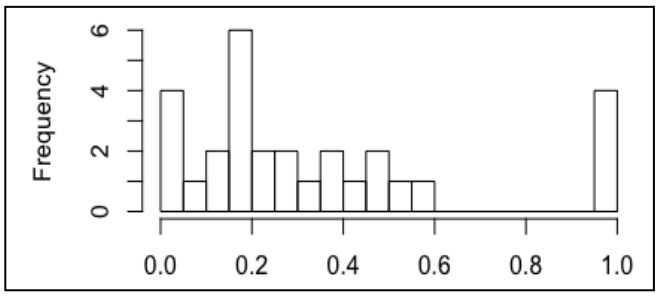

FIGURE 22: CLUSTERING COEFFICIENT OF CEM

While most CEM are quite robust due to an above average CC score in terms of the global benchmark, three CEM have a CC of 0, indicating that no connections exist between their neighbors. Simex and Exmo additionally only have a DC score of 1, while LakeBTC has a DC of 3. Interestingly, some CEM with below average CC scores, such as Coinbit (0.032), rank well regarding their DC (278). This could be due to the neighboring AS being customer AS with few links between them.

c) Local Node Connectivity

The LNC measure can be used to determine the required amount of removed nodes to disconnect a source node from a target node. The source nodes are AS of CEM.

For our research we have chosen three target nodes at different locations, to allow for a comparison of connectivity based on geo-location. These are AS701 of Verizon (USA), AS3320 of Deutsche Telekom AG (GER), and AS45102 of Alibaba Group (CHN). An overview is presented in Appendix III. All three nodes are well-connected regarding their degree of 1387, 631, and 190 following the order above. We have chosen these AS as they represent some of their respective countries' major companies. Furthermore, a large share of their business is web-based, which requires a high degree of connectivity.

According to our analysis 72.42% of AS have identical LNC scores for all three locations. 24.13% of LNC scores are equal between Verizon and Telekom, but not Alibaba. The remaining 3,45% are not equal in all three locations. A correlation between Verizon and Telekom could therefore be present.

From our analysis we derive, that Verizon is the hardest target node to be disconnect from with an average LNC of 29.24. AS20940 of Bibox is the highest ranking with an LNC of 169. AS4766 of Coinbit ranks second highest with an LNC score of 99. Additionally, AS 16625, which includes 5 CEM, displays a high LNC of 85. Two AS, specifically AS51558 of Simex and AS200449 of Exmo score a LNC of 1 and are thus within one removed node of being disconnected from Verizon's AS.

The average LNC to AS3320 of Deutsche Telekom AG is 28.2 and therefore displays the second difficult target node to be disconnected from. AS20940 of Bibox, which with an LNC of 139 is the best connected CEM to a German AS. Bibox also performed well in regards to the previously explored graph measures. It must be underlined, that while Telekom ranks behind Verizon in terms of LNC, the disparity is minimal, despite Verizon's degree being close to 600 higher than Telekom. Additionally, the lowest LNC score of 1 is again displayed by Simex and Exmo.

With a mean LNC of 16.06 the China-based AS of Alibaba Group is the easiest target node to disconnect from. 25.71% of CEM share the highest LNC of 32, which include AS16509, AS20940 of Bibox, and AS4766 of Coinbit. CEM such as Exmo and Simex are rather easy to disconnect from their respective target nodes, as their LNC of one is equal for all locations. These results are consistent with the previously analyzed graph measures, in which Exmo and Simex showed the lowest LNC score of 1 regarding their connectivity.

In general, AS that are hard to disconnect from a location-specific target node, are also more difficult to disconnect from the other target nodes. This implies that a high degree of connectivity is, on average, not specific to one location, but rather independent. This assumption can also be made for AS that are not-well connected.

In this chapter we analyzed the connectivity of CEM. We began by presenting an overview of global benchmarks for the graph measures, which we used to compare with the individual scores of CEM. The chapter was concluded with a

location-based connectivity analysis, where the removal of nodes within the path of a source and target node was explored

VI. Discussion

A. Interpretation of Results

The mean values for CEM and the AS graph for the respective type of graph measure are shown in Table VI. Both the analysis results and the comparison of mean scores demonstrate that generally CEM are better connected than an average AS.

TABLE VI. Results of graph measures

Type	Measure	Mean CEM	Mean AS Graph
Distance	ECC	6.17	6.96
	SSSPL	3.08	3.82
Centrality	DC	24.19	4.03
	EC	1.98e-2	1.91-e3
	BC	1.36e-3	8.94e-5
	CLC	0.33	0.26
Neighborhood	LAND	675.57	446.97
	CC	0.35	0.23

This is likely due to the fact, that CEM are a relatively new type of platform and therefore use popular service providers as their internet hosts. Examples of these are Cloudflare, Amazon, Google, Incapsula, Akamai, and Alibaba US. 138 CEM use Cloudflare, 22 use Amazon, seven use Google, six use Incapsula and ten CEM are evenly split amongst Akamai and Alibaba US. Moreover, this was the cause for a mere 35 unique ASN being analyzed, despite having web-scraped 232 CEM. For a detailed overview regarding AS and respective CEM see Appendix I.

While most CEM did perform above average regarding their connectivity, some were well below these scores, while others were well above. CEM such as Bibox (AS20940) and Coinbit (AS4766) displayed a substantially higher degree of connectivity for all measures, compared to the global average. For instance in regards to CLC, Bibox (0.388) ranks within the top 0.05% of all AS, while CoinBit (0.370) ranks within the top 0.25%. Other well connected CEM were part of AS13335 of Cloudflare, which includes 138 CEM. These, alongside the highest ECC score of 6, were within the top five AS in a CEM context for the centrality measures DC (93), EC (0.039) as well as BC (0.0039). Additionally, 5 CEM which are part of AS16625 of Akamai, are close to the graph's center and therefore well connected due to their centrality and distance scores (DC of 87, ECC of 0.039, EC of 0.048, CLC of 0.372, and BC of 0.0027). These values are further underlined by their low SSSPL scores (Cloudflare: 2.75; Akamai: 2.67), placing them close to the network's center.

On the other hand, CEM such as Simex, BarterDEX, InstantBitex, Graviex and Exmo were noticeably lower ranked and therefore not well-connected in terms of the analyzed measures. This was demonstrated by their high ECC scores, such as Simex with an ECC of 8, and low DC ranging between one and three. To put these scores into context, the average AS has a DC of 4 and an ECC of 7. Additionally, we could see that CEM with poor connectivity, tend to link with well-connected AS, as shown by the CC metric. For example BarterDex had the highest possible CC score of 1. Poorly-connected AS (and CEM) therefore attempt to bypass their connectivity and low robustness. It is also worth mentioning that none of the poorly performing CEM were hosted on well-known servers (Cloudflare, etc.).

Recall that for six out 35 identified AS graph measures were not able to be computed, as they were not included in the AS Links dataset. Upon further analysis we deducted that out of those AS, five are owned by Chinese companies. These are Pingtan (AS136782), Alibaba (AS134963), SonderCloud Hong Kong (AS133199), Hong Kong Yunify Technologies (AS134366), and lastly Hong Kong FireLine Network (AS136950). The specific CEM are Coinsuper, LBank, Dobi Exchange, Neraex, CoinTiger and ZB, all of which are also based in China. China is known for its heavy internet regulations and its dramatically different network structure. It is tempting to affirm that China strongly filters its network and therefore traceroute measurements from institutions such as CAIDA are not able to reach these AS [58]. Lastly, the sixth AS, while not being owned by a Chinese company, is owned by Amazon (AS14618) and uses an "Advanced Encryption Standard". This standard may be the cause as to why no information regarding the links of this AS were able to be deducted by CAIDA's traceroute. AS14618 is used by OpenLedger, a Danish CEM.

B. Limitations and Future Work

A limitation of our research is that the data gathering was only done statically. Both the web-scraping process of CEM and the AS Links dataset refer to one specific day. Due to the dynamic nature of the internet, AS and their individual links as well as the ASN of CEM may change. Moreover, the CEM ranked on CoinMarketCap are likely to change, which could result in different analysis results. In future work the data gathering process could be extended to a longer period of time to not only create a one-time snapshot of the connectivity of CEM. It is likely that doing so would require the web-scraping script to be updated, as the script is bound to the website layout of CoinMarketCap for the current point of time of this research.

The data gathering process could be further optimized by not only web-scraping CEM from one source, but rather multiple sources. For our research we used CoinMarketCap as a source, where 200 CEM are ranked by their market capitalization. By doing so, we were able to include the more commonly used CEM, but not niche exchange markets. Alongside sources that rank CEM by different criteria, in future work such data could be gathered by using search engines for which specific string combinations are defined. The integration of different sources could be done via the embedding of public APIs into a Python framework. This would allow us to standardize the web-scraping process with multiple sources as well as reduce the amount of required updating to ensure full functionality of the script over a prolonged period of time.

A further limitation is that our edgelist was limited to one source. As mentioned in Ch. III, researchers have compiled different data sources in order to create a more accurate

dataset of links on different topology levels [24]. Using a compiled dataset could lead to a more accurate analysis and perhaps solve the issue regarding AS that were not included in the CAIDA AS Links dataset. Ultimately though a perfect dataset does not exist due to the topology of the internet not being fully explored.

Lastly, due to computational constraints our analysis was done with a simple graph, in which edges were undirected and unweighted, thus additionally reducing the complexity of our research. In future research undirected and unweighted edges could be included to increase the complexity of the analysis.

C. Contributions

With our research we have expanded the current point of internet topology research from a different perspective. Using the foundations of prior research we analyzed the connectivity of cryptocurrency exchange markets, a relatively new playing field of e-businesses and platforms.

Regarding a more practical side, in addition to popularity rankings and critical reviews the results of our research could be used by users of CEM to make informed decisions regarding their choice of exchange platform.

Additionally, our results may also be interesting for the owners of CEM, as these can identify their weak spots and potential bottlenecks to increase their current connectivity and robustness. By decreasing network outages the customer satisfaction could potentially rise, resulting in more business throughput. Moreover increasing the customer satisfaction could lead to longer business relationships, which in a dynamic and fast-moving playing field such as the cryptocurrency market may result in a competitive advantage over other CEM.

VII. Conclusion

Our research investigates the topological connectivity of CEM, which have been selected based on their market capitalization. By creating a web-scraping script we were able to automate the selection process. To analyze the connectivity we used the Python module NetworkX, which allowed us to create a network graph on AS level and derive previously selected graph measures.

Through this analysis we discovered that CEM are, on average, well connected and that the results of graph measures do not tend to vary regarding the better performing and worse performing AS. Additionally, we showed that a majority of CEM share their respective AS, due to being hosted on large platforms, for example Cloudflare, Amazon, and Akamai.

CEM can use our methods to investigate and potentially increase their network connectivity by selecting a better hosting service as well as increase customer satisfaction.

References

[1] Statista, "Market capitalization of cryptocurrencies from 2013 to 2018 (in billion U.S. dollars)," 2019. [Online]. Available: https://www.statista.com/statistics/730876/cryptocurrency-maket-value/. [Accessed: 24-11-2018]

[2] X. Li and C. A. Wang, "The technology and economic determinants of cryptocurrency exchange rates: The case of Bitcoin," *Decision Support Systems*, vol. 95, pp. 49-60, 2017.

[3] CoinMarketCap, "24 Hour Volume Rankings (All Exchanges)," 2019. [Online]. Available: https://coinmarketcap.com/exchanges/volume/24-hour/all/. [Accessed: 20-12-2018]

[4] W. H. Delone and E. R. Mclean, "Measuring e-commerce success: Applying the DeLone & McLean information systems success model," *International Journal of electronic commerce*, vol. 9, no. 1, pp. 31-47, 2004.

[5] Statista, "Average cost per hour of enterprise server downtime worldwide in 2017 and 2018," 2019. [Online]. Available: https://www.statista.com/statistics/753938/worldwide-enterprise-server-hourly-downtime-cost/. [Accessed: 29-11-2018]

[6] Bloomberg, "One of the Biggest Crypto Exchanges Goes Dark and Users Are Getting Nervous," 2018. [Online]. Available: https://www.bloomberg.com/news/articles/2018-01-12/crypto-exchange-kraken-goes-dark-and-user-anxiety-surges. [Accessed: 25-11-2018]

[7] H. Burch and B. Cheswick, "Mapping the internet," *Computer*, vol. 32, no. 4, pp. 97-98, 1999.

[8] R. Merris, Graph Theory: Series in Discrete Mathematics And Optimization, 2001.

[9] J. C. Mitchell, "The concept and use of social networks," *Social networks in urban situations*, 1969.

[10] J. C. Guedon, "A brief history of internet," *Studies in Health Technology and Informatics*, vol. 36, pp. 121-132, 1997.

[11] S. Greenstein and F. Nagle, "Digital dark matter and the economic contribution of Apache," *Research Policy*, vol. 43, no. 4, pp. 623-631, 2014.

[12] ISC, "Internet Domain Survey, Januar, 2019," 2019. [Online]. Available: https://ftp.isc.org/www/survey/reports/current/. [Accessed: 18-3-2019]

[13] R. H. Zakon, "Hobbes' Internet Timeline 25," 2018. [Online]. Available: https://www.zakon.org/robert/internet/timeline/.[Accessed: 16-03-2019]

[14] S. Greenstein and C. Snively, "Net Neutrality : A Managerial Perspective," pp. 1-15, 2016.

[15] Cisco, "VNI Global Fixed and Mobile Internet Traffic Forecasts," 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html.[Accessed: 14-03-2019]

[16] B. Donnet and T. Friedman, "Internet topology discovery: a survey," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 4, pp. 56-69, 2007.

[17] CAIDA, "Router Level Topology," 2019. [Online]. Available: https://www.caida.org/research/topology/router-level-topology.xml. [Accessed: 20-02-2019]

[18] CAIDA, "AS Relationships," 2019. [Online]. Available: http://www.caida.org/data/as-relationships/ [Accessed: 24-02-2019]

[19] Lixin Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 733-745, 2002.

[20] S. Kelkel, "Internet Robustness Analysis – Simulation of a Worm-Based Router Attack," 2016. unpublished

[21] CAIDA, "About - Center for Applied Internet Data Analysis," 2019. [Online]. Available: http://www.caida.org/home/.[Accessed: 07-01-2019]

[22] A. Baumann, "Internet Resilience and Connectivity Risks for Online Businesses," 2013. unpublished

[23] J. Schulze, "Industry Classification of Autonomous Systems," 2015. unpublished

[24] G. J. Tilch, "Topology of the Internet Graph," 2015. unpublished

[25] Z. Ghazaryan, "Topological Analysis of Internet Connectivity for the Financial Sector," 2017. unpublished

[26] B. Fabian, A. Baumann and J. Lackner, Topological Analysis of Cloud Service Connectivity, vol. Volume 88, 2015, pp. 151-165.

[27] P. Mahadevan, D. Krioukov, M. Fomenkov, X. Dimitropoulos and A. Vahdat, "The Internet AS-level topology: three data sources and one definitive metric," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 17-26, 2006.

[28] A. Baumann, B. Fabian and M. Lischke, Exploring the Bitcoin Network, vol. 1, 2014, p. .

[29] M. Faloutsos, P. Faloutsos and C. Faloutsos, "On power-law relationships of the internet topology," in *ACM SIGCOMM computer communication review*, 1999.

[30] B. Fabian, Z. Ghazaryan and T. Ermakova, "Internet Connectivity of Financial Services – a Graph-Based Analysis," *SSRN Electronic Journal*, no. July, 2018.

[31] L. V. Casaló, C. Flavián and M. Guinalíu, "The role of security, privacy, usability and reputation in the development of online banking," *Online Information Review*, vol. 31, no. 5, pp. 583-603, 2007.

[32] C. Kim, W. Tao, N. Shin and K.-S. Kim, "An empirical study of customers' perceptions of security and trust in e-payment systems," *Electronic commerce research and applications*, vol. 9, no. 1, pp. 84-95, 2010.

[33] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer and R. Wirth, "CRISP-DM 1.0 Step-by-step data mining guide," 2000.

[34] A. I. R. L. Azevedo and M. F. Santos, "KDD, SEMMA and CRISP-DM: a parallel overview," *IADS-DM*, 2008.

[35] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996.

[36] Python, "About - Python," 2019. [Online]. Available: https://www.python.org/about/. [Accessed: 03-03-2019]

[37] Postman, "About - Postman," 2019. [Online]. Available: https://www.getpostman.com/. [Accessed: 04-02-2019]

[38] GitHub, "About - GitHub," 2019. [Online]. Available: https://github.com/). [Accessed: 15-02-2019]

[39] CoinMarketCap, "Crypto Exchange Market Rankings," 2019. [Online]. Available: https://coinmarketcap.com/rankings/exchanges. [Accessed: 04-01-2019]

[40] Python-Requests, "Python Request Library," 2019. [Online]. Available: http://docs.python-requests.org/en/master/. [Accessed: 06-01-2019]

[41] BeautifoulSoup, "About - BeautifulSoup," 2019. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. [Accessed: 04-01-2019]

[42] Python, "Socket lib," 2019. [Online]. Available: https://docs.python.org/3/library/socket.html. [Accessed: 05-01-2019]

[43] Python, "RE," 2019. [Online]. Available: https://docs.python.org/3/library/re.html. [Accessed: 06-01-2019]

[44] Pandas, "Pandas," 2019. [Online]. Available: https://pandas.pydata.org/. [Accessed: 03-01-2019]

[45] UltraTools, "UltraTools - ASN LookUp," 2019. [Online]. Available: https://www.ultratools.com/tools/asnInfo. [Accessed: 04-01-2019]

[46] TraceRouteOnline, "IP-ASN-LookUp," 2019. [Online]. Available: https://traceroute-online.com/ip-asn-lookup/. [Accessed: 05-01-2019]

[47] Ashish Vishwakarma, "Data Structures," 2019. [Online]. Available: https://www.geeksforgeeks.org/difference-between-structured-semi-structured-and-unstructured-data/. [Accessed: 05-03-2019]

[48] R. Pastor-Satorras and A. Vespignani, Evolution and structure of the Internet: A statistical physics approach, Cambridge University Press, 2007.

[49] A. Hagberg Schult, D. & Swart, P., *NetworkX Reference Release 1.11*, 2016.

[50] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vázquez and A. Vespignani, "Exploring networks with traceroute-like probes: Theory and simulations," *Theoretical Computer Science*, vol. 355, no. 1, pp. 6-24, 2006.

[51] C. G. Ghedini and C. H. C. Ribeiro, "Rethinking failure and attack tolerance assessment in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 23-24, pp. 4684-4691, 2011.

[52] J. M. Hernández and P. Van Mieghem, "Classification of graph metrics," *Delft University of Technology: Mekelweg, The Netherlands*, pp. 1-20, 2011.

[53] P. Bonacich and P. Lloyd, "Eigenvector-like measures of centrality for asymmetric relations," *Social networks*, vol. 23, no. 3, pp. 191-201, 2001.

[54] B. Edwards, S. Hofmeyr, G. Stelle and S. Forrest, "Internet topology over time," *arXiv preprint arXiv:1202.3993*, 2012.

[55] W. Ellens and R. E. Kooij, "Graph measures and network robustness," *arXiv preprint arXiv:1311.5064*, 2013.

[56] M. Newman, Networks, Oxford university press, 2018.

[57] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos and A. Vahdat, "Lessons from three views of the Internet topology," *arXiv preprint cs/0508033*, 2005.

[58] H. Roberts, D. Larochelle, R. Faris and J. Palfrey, "Mapping Local Internet Control," *Computer Communications Workshop*, pp. 1-14, 2011.

APPENDIX

I. Overview of ASN, ASN Owners, cumulative frequency and respective CEM

ASN	ASN Info	Freq.	CEM
13335	CLOUDFLARENET - Cloudflare, Inc., US	138	Bit-Z, OKEx, Huobi, DigiFinex, HitBTC, BCEX, BitMart, Fatbtc, Cryptonex, OEX, Exrates, UPbit, LocalTrade, LATOKEN, Bitfinex, TOKOK, CoinEgg, Coinbase Pro, Sistemkoin, Kraken, Hotbit, P2PB2B, BiteBTC, B2BX, Bilaxy, CoinsBank, Bittrex, Coinhub, Trade By Trade, Bitlish, Coinbe, BTCBOX, YoBit, Poloniex, UEX, Coinone, Livecoin, itBit, CryptalDash, Mercatox, BitBay, BtcTrade.im, KuCoin, Coinroom, Liquid, Indodax, Vebitcoin, BTC-Alpha, Paribu, Tidebit, Ovis, OKCoin, BtcTurk, CEX.IO, C2CX, Tidex, BX Thailand, Altcoin Trader, Ethfinex, Negocie Coins, CoinEx, Cryptopia, Tokenomy, Trade.io, Bitso, BTC Markets, OOOBTC, QuadrigaCX, Crex24, Bitibu, Bitkub, CryptoBridge, STEX, OasisDEX, Koineks, Coinfloor, Bytex, IDEX, Stronghold, Bitbns, LiteBit.eu, Independent R..., BigONE, WazirX, Trade Satoshi, Liqui, BitMarket, GBX Digital, CoinPlace, Waves Decentr..., BitcoinToYou, TradeOgre, HBUS, CoinMate, Koinex, DDEX, AidosMarket, Kuna, CoinCorner, Kyber Network, Buda, Bittylicious, Gatehub, ezBtc, Coinut, Braziliex, Bithesap, Koinim, CryptoMarket, C-CEX, CoinFalcon, Bleutrade, Allbit, EtherDelta, SouthXchange, Bit2C, Bitex.la, Tripe Dice Ex, Paymium, Coinnest, Nocks, Gatecoin, RuDEX, Coingi, Nanex, FreiExchange, Tux Exchange, Novaexchange, GuldenTrader, Coinrate, ISX, Bithumb, Cobinhood, Zebpay, Bgogo, Coinall, BitMax, CoinMex
16509	AMAZON-02 - Amazon.com, Inc., US	22	Binance, IDAX, RightBTC, Bitrue, Bitbank, Cryptology, Coindeal, CPDAX, GOPAX, DSX, Stellarport, ACX, BitcoinTrade, Switcheo Network, Coinrail, Token Store, Coincheck, BitMEX, FCoin, AirSwap, ABCC, GDAC
NaN	-	9	BitForex, Coineal, Gate.io, Zaif, OTCBTC, RippleFox, Iquant, DEx.top, Fisco
15169	GOOGLE - Google LLC, US	7	Kryptono, Gemini, Coinsquare, COSS, Bancor Network, YunEx, Radar Relay
19551	INCAPSULA - Incapsula Inc, US	6	Bitstamp, bitFlyer, Coinexchange, Mercado Bitcoin, Bitstamp, EXX
16625	AKAMAI-AS - Akamai Technologies, Inc., US	5	55 Global Mar…, Korbit, Rfinex, BITBOX, CoinZest
45102	CNNIC-ALIBABA-US-NET-AP Alibaba (US) Technology Co., Ltd., CN	5	Coinbene, TOPBTC, MBAex, Ripple China, Bcoin.sg
136782	PINGTAN-AS-AP Kirin Networks, CN	2	ZB.COM, CoinTiger
20773	HOSTEUROPE-AS, DE	2	Luno, Bitshares
61157	PLUSSERVER-ASN1, DE	2	Stellar, Lykke Exchange
14061	DIGITALOCEAN-ASN - DigitalOcean, LLC, US	2	Bisq, Cryptomate
134963	ASEPL-AS-AP Alibaba.com Singapore E-Commerce Private Limited, SG	2	LBank, Coinsuper
64050	BCPL-SG BGPNET Global ASN, SG	2	CHAOEX, Allcoin
16276	NCORE-AS Hochstadenstr. 5, DE	2	Bitinka, Bitsane
24940	HETZNER-AS, DE	2	InfinityCoin, Heat Wallet
54113	FASTLY - Fastly, US	1	Satang Pro
20473	AS-CHOOPA - Choopa, LLC, US	1	C-Patex

133296	WEBWERKS-AS-IN Web Werks India Pvt. Ltd., IN	1	Instant Bitex
133199	SONDERCLOUDLIMITED-AS-AP SonderCloud Limited, HK	1	DOBI Exchange
51558	SMTLB-AS, LB	1	Simex
20940	AKAMAI-ASN1, US	1	Bibox
55355	ISP-AS-AP ISP, HK	1	ZBG
134366	YTL-HK Yunify Technologies (HK) Limited, HK	1	Neraex
8075	MICROSOFT-CORP-MSN-AS-BLOCK - Microsoft Corporation, US	1	DragonEX
136950	HIITL-AS-AP Hong Kong FireLine Network LTD, HK	1	IDCM
50473	ECO-AS, RU	1	Graviex
30496	AS-TIERP-30496 - TierPoint, LLC, US	1	Escodex
53667	PONYNET - FranTech Solutions, US	1	LakeBTC
31727	NODE4-AS, GB	1	CRXzone
54994	QUANTILNETWORKS - QUANTIL NETWORKS INC, US	1	BTCC
39287	FLATTR-AS, SE	1	BarterDEX
60781	LEASEWEB-NL-AMS-01 Netherlands, NL	1	The Rock Trading
14618	AMAZON-AES - Amazon.com, Inc., US	1	OpenLedger DEX
200449	QRATOR-, CZ	1	Exmo
4766	KIXS-AS-KR Korea Telecom, KR	1	Coinbit
9286	KINXIDC-AS-KR KINX, KR	1	Cashierest
49349	DOTSI, PT	1	BTC Trade UA
49981	WORLDSTREAM, NL	1	Bitonic
29854	WESTHOST - WestHost, Inc., US	1	Bits Blockchain

II. Overview of graph-based measures for CEM

Measure	Mean	Median	Variance
ECC	6,17241379	6,00000000	0,21921182
SSSPL	3,07782989	2,97374908	0,18289287
DC	39,00000000	19,00000000	3847,14285714
EC	0,01979462	0,01420023	0,00031937
BC	0,00135524	0,00020815	0,00000969
CLC	0,32808119	0,33432779	0,00146855
LAND	675,56887968	413,44000000	528094,68814653
CC	0,34687196	0,23736969	0,09540443

III. Results of LNC analysis

ASN	AS3320	AS701	AS45102
16509	33	37	32
20940	139	33	32
4766	99	33	32
16625	85	33	32
13335	83	33	32
15169	71	33	32
8075	55	33	32
16276	43	33	32
45102	32	32	32
20473	25	25	25
19551	25	25	25
9286	22	22	22
54994	20	20	20
54113	19	19	19
24940	14	14	14
30496	8	8	8
60781	6	6	6
133296	6	6	6
36024	5	5	5
14061	5	5	5
55355	4	4	4
49349	4	4	4
50473	3	3	3
31727	3	3	3
29854	3	3	3
53667	2	2	2
39287	2	2	2
51558	1	1	1
200449	1	1	1

IV. Webscraper Code

```
"""
Aim of script is to get all relevant crypto exchange platform URL's and
identify their ASN.
Export to CSV with the objective to use for network analytics.
"""

from requests import get
from requests import post
from bs4 import BeautifulSoup
import pandas as pd
import time
import socket
import re



def main():
    global df

    df = crawl()
    df['ip'] = df.apply(lambda row: getIP(row),axis=1)
    df.to_csv('./ips.csv',index = False)
    df['asn'] = df.apply(lambda row: getASN(row.ip), axis = 1)
    df.to_csv('./asn.csv',index = False)
    df['asn2'] = df.apply(lambda row: getASN2(row.ip), axis = 1)
    df.to_csv('./asn2.csv',index = False)
    df['crosscheck'] = df.apply(lambda row: crosscheck(row), axis = 1)
    df = df.fillna(0)
    df.asn = df.asn.astype('int64')
    df.to_csv('./crosschecked.csv',index = False)

def crosscheck(row):
    if row.asn == row.asn2:
        return True
    else:
        return False

def getASN2(targetip):
    url = 'https://traceroute-online.com/ip-asn-lookup/'
    headers = {
    #"Connection": "keep-alive",
    #"Content-Length": 129,
    "Origin": "https://traceroute-online.com",
    "X-Requested-With": "XMLHttpRequest",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36",
    "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8",
    #"Accept": "*/*",
    "Referer": "https://traceroute-online.com/ip-asn-lookup/",
    #"Accept-Encoding": "gzip,deflate,sdch",
    #"Accept-Language": "fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4",
    #"Accept-Charset": "ISO-8859-1,utf-8;q=0.7,*;q=0.3",
    "Cookie": "_cfduid=d8e23cac28fd39df56e2ad0cf241a5b211546522071;
csrftoken=oLba5l6YRivh0elUFUDAqUXqH4MZyqOy; _ga=GA1.2.1855903888.1546522075;
_gid=GA1.2.1412935247.1547319788; _gat=1",
    }

    payload = {"csrfmiddlewaretoken":"oLba5l6YRivh0elUFUDAqUXqH4MZyqOy",
    "targetip":targetip}
    r = post(url, data = payload, headers=headers)
    time.sleep(4)
```

```
    html = BeautifulSoup(r.content, "html.parser")
    try:
        asn = int(html.find('div', {'class':'box-
body'}).text.split(',')[1].replace('"','"))
    except Exception as e:
        print(e)
        asn = None
    return asn

def getASN(row):
    try:
        url = 'https://www.ultratools.com/tools/asnInfoResult?domainName='+row
        resp = getUrl(url)
        time.sleep(4)
        html = BeautifulSoup(resp.content, 'html.parser')
        asn = int(html.find('div', {'class':'tool-results-
heading'}).text.replace('AS',''))
        return asn
    except Exception as e:

        print(e)
        return None


def getIP(row):
    if row.siteUrl.count('/') >2:
        try:
            found = re.search('//(.+?)/', row.siteUrl).group(1)
        except AttributeError:
            found = 'Not Found.' # apply error handling
    else:
        try:
            found = re.findall('//(.*)', row.siteUrl)[0]
        except AttributeError:
            found = 'Not Found.' # apply error handling

    try:
        print(found)
        resp = socket.gethostbyname(found)
    except Exception as e:
        print(e)
        resp = None

    if resp is not None:
        return resp

def getUrl(url):
    try:
        print("Perform HTTP GET request on: ", url)
        ans = get(url, stream = True)
        if ans.headers['Content-Type'].lower() is not None and ans.status_code ==
200:# and ans.headers['Content-Type'].lower().find('html') > -1:
             return ans
        else:
            print("smth went wrong")
    except Exception as e:
        print(e)


def crawl():
    global dfs
    dfs = []
    for i in range(1,4):
```

```
        print(i)
        url = "https://coinmarketcap.com/rankings/exchanges/"+str(i)
        ans = getUrl(url)
        html = inspectFile(ans.content)
        df = getInfo(html)
        dfs.append(df)

    dfs = pd.concat(dfs, sort = False)
    dfs.to_csv('./out.csv',index = False)
    return dfs

def inspectFile(content):
    try:
        html = BeautifulSoup(content, 'html.parser')
        table_body = html.find('tbody')
        rows = table_body.find_all('td', {'class':'no-wrap currency-name'})
    except Exception as e:
        print(e)
    return rows

def getInfo(rows):

    names = []
    urls = []
    siteUrls = []
    try:
        for i in range(0,len(rows)):
            data = rows[i].text.strip()
            name = data
            names.append(name)
            tmp = rows[i].find('a').get('href')
            url = "https://coinmarketcap.com" + tmp
            urls.append(url)
            time.sleep(4)
            site = getUrl(url)
            insides = site.content
            getHTML = BeautifulSoup(insides, 'html.parser')
            links = getHTML.find('div', {"class":'col-xs-12'})

            links = links.find_all('a')
            print(links[0].text)
            siteUrls.append(links[0].text)
            time.sleep(4)
    except Exception as e:
        print(e)
    myData = {"name" : names, "siteUrl" : siteUrls}
    myData = pd.DataFrame.from_dict(myData)
    return myData

main()
```