UNIVERSITÀ DEGLI STUDI DI MILANO

CORSO DI DOTTORATO IN INFORMATICA
XXXVI° CICLO

DIPARTIMENTO DI INFORMATICA

Mining and learning Web3 platforms: a temporal network perspective

INF/01

Dottorando:
Cheick Tidiane BA

Relatore:
Prof. Sabrina GAITO

Co-relatore:
Dr. Matteo ZIGNANI

Coordinatore del dottorato:
Prof. Roberto SASSI

A. A. 2022/2023

Contents

Part I Background

Part II Network evolution dynamics

Part III The interplay of user behavior and currencies

Part IV Modeling and prediction of user migration

Part V Machine learning on multilayer graphs

Chapter 1

Introduction

Over the last decade, the world of Internet services underwent a set of profound changes, as the actual structure of Web 2.0 has been questioned by novel paradigms trying to reduce the over-centralization around a few big platforms and tech companies. We observed a shifting attention from monolithic centralized services, to open, decentralized, and distributed alternatives [1]. The necessity of taking power and control away from the major centralized web platforms has become more evident, leading to the development of alternative platforms embracing decentralized and open principles [2]. One of the ideas gaining momentum is ***Web3***, i.e. the design of platforms and software systems built on blockchain technologies to promote a decentralized Web [3]. Indeed, blockchain technology offers many design options for decentralized systems, such as decentralized storage, consensus-based validation of stored data, and the very important option to implement economic systems. The application of Web3 principles in social media resulted in blockchain online social networks [4]; in the economy, the application led to Decentralized Finance (DeFi) [3]; whereas the application in governance resulted in Decentralized Autonomous Organizations (DAOs); to cite a few well-known examples. As the number of platforms and services following Web3 principles is still growing, with applications reaching a wider public, with decentralization influencing the future of the Web, it becomes of fundamental importance to better understand Web3; and, although the ideas of Web3 are the heart of a heated debate between enthusiasts and skeptics, the influence of Web3 is undeniable. Nevertheless, Web3 has not been studied much, especially on the new applications that go beyond the decentralized finance systems. From a data standpoint, platforms following this paradigm offer a great opportunity to researchers in different fields thanks to the huge volume of high-resolution data stored in the supporting blockchains, representing a steep change point w.r.t. Web 2.0. Indeed, a broad set of data about these techno-social systems can be easily accessible: by the nature of blockchains, data are publicly available, validated, and affordable by interfacing with the API blockchain. Moreover, data from Web3 platforms offer two advantages: *i)* each piece of information is timestamped since each blockchain block has a validation timestamp; and *ii)* each block reported multi-faceted interactions — social, economic, financial — among people and between people and platform. So, these data sources have all the features to face tasks and issues related to modern techno-social networks and to support detailed and in-depth analysis of users' traits.

However, blockchain-based systems are often highly interconnected and exhibit intricate interdependencies that span different interaction layers within the same blockchain and multiple blockchain networks, at the same time. These unique characteristics present substantial challenges for researchers involved in the analysis and understanding of these innovative platforms, that necessitate novel solutions. Addressing the new open problems requires a comprehensive approach, ranging from data collection methodologies to modeling, analysis, and prediction tasks. There are many *open questions* where the key points are the *interplay between the temporal and the heterogeneous dimensions* characterizing blockchain-based systems. These aspects, still marginal in the current literature, are the main subject of this thesis.

Given the interconnected nature of the Web3 paradigm, in this work, we mainly rely on models, algorithms, and methods from the field of ***network science*** [5]. Network science is a multidisciplinary field examining complex structures and dynamic interactions within various systems. The main focus of network science is to investigate the relationships between interconnected entities, typically represented as nodes (or vertices), connected by links (or edges) that represent interactions or dependencies. These entities span from biological molecules and neurons to individuals in social settings, showcasing the versatility of this representation across diverse domains. Modeling through networks — graphs — has been fundamental for improving our understanding of the fundamental principles governing interconnected systems. The use of mathematical models and computational tools has provided important results, from early studies covering communication networks, urban mobility networks, and scientific collaborations to works focusing on brain networks, biological or genomics networks, financial transaction networks, social media networks or contact networks [6, 7]. Throughout this work, we showcase how network science is a suitable analytics framework to model the complexity of Web3 by tackling different aspects and open questions on Web3 through a network science approach. More precisely, we focus on the following broad aspects which result in the main parts of this thesis:

- **Modeling Web3 as a network**: we provide an extensive background of Web3 platforms, along with details on the datasets retrieved for our works, and we introduce the main concepts required for network-based methodologies to model Web3 data.
- **Network evolution dynamics**: we investigate the dynamical aspects of Web3 systems from a microscopical perspective by exploring the temporal patterns of users, singularly, and high-order temporal patterns leading to the formation of triads.
- **The interplay of currency and user behavior**: we analyze the interplay between user activity and the economic dimensions, i.e. cryptocurrency and reward systems; one of the main key points differentiating Web3 systems from the traditional Web 2.0 solutions.
- **Modeling and prediction of user migration**: the combination of heterogeneous and high-resolution temporal data with a dynamic socio-economic environment provided by Web3 platforms allows us to investigate cross-chain behaviors such as user migration, with a focus on its predictability through different machine learning techniques.
- **Machine learning on multilayer graphs**: we develop a framework to improve graph neural network usage on multilayer graphs where the multilayered nature of Web platforms has provided a real scenario testbench to evaluate the performances of the framework.

1.1 Modeling Web3 as a network

In Part I, we introduce the basic concepts behind blockchain technology and its main applications. First, we describe the building blocks of the Web3 paradigm, including blockchain technology, consensus mechanisms, and token systems. What emerges, is that the Web3 field is constantly changing, with an increasing amount of proposals and innovations, that allow the application of Web3 across new fields. This also leads to an additional challenge in the comprehension of Web3, as each application shows its own characteristics. Within this context, we require a modeling approach characterized by flexibility, allowing for tuning and customization of the data representations ranging from simpler to more complex ones, depending on the dataset and the specific open problem at hand. Network science provides an ensemble of models and algorithms up to the task: in literature, we have different network models that can be leveraged depending on the complexity of the data to represent and the problem. Our main contributions are primarily focused on how to effectively represent the intricate relationships that exist within or among blockchain-based complex systems by network-based models. Throughout the thesis, we provide different methodologies depending on data and tasks to then leverage network science toolkits. We rely on various models, from simple homogenous graphs to heterogeneous and/or multilayer graphs, while we also leverage time information to build temporal networks. Moreover, we focus on **temporal multilayered graph models** (Figure 1.1), an emerging yet powerful tool for handling a more realistic representation of the various and heterogeneous relationships that may characterize an entity in a graph-structured system. Therefore, we illustrate different graph representations to model interaction datasets and some indication on how to leverage them.

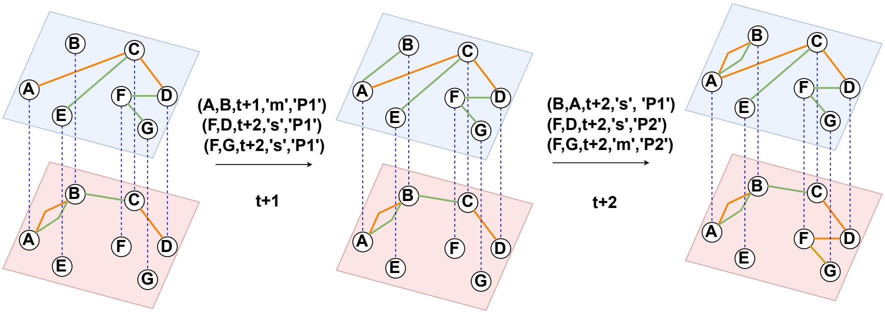


Fig. 1.1: *Example: a temporal heterogeneous multilayer graph to model the most complex scenarios in Web3. Here, we observe some users and their relationships in two platforms, P1 and P2, where they can interact both socially and financially. The layers represent the two platforms, P1 in blue and P2 in red. Social links are shown in green and financial links are in orange. Alongside the arrows, we display the interaction actions, occurring during a time window, which generate the links in the corresponding graphs. In the example, the heterogeneous links allow us to capture the order of social and financial relationships between A and B in Platform 1. Another interesting aspect is the influence across platforms: we can see node how node F has a social link with nodes D and G in P1 and then starts economic relationships in platform P2.*

To sum up, in the first part:

- We provide basic concepts of Web3 paradigm and technology.

- We describe the main applications of the Web3 paradigm such as decentralized finance and blockchain online social media.
- We describe the datasets used across the thesis to test the newly proposed methodologies.
- We provide background knowledge on modeling Web3 with networks.

Part I is partially based on the following publications:

- Cheick Tidiane Ba, Matteo Zignani, Sabrina Gaito. *The role of cryptocurrency in the dynamics of blockchain-based social networks: The case of Steemit. PLOS ONE* 17.6 (June 2022) pp. 1–22.
- Cheick Tidiane Ba, Matteo Zignani, Sabrina Gaito. *Cooperative behavior in blockchain-based complementary currency networks through time: The Sarafu case study. Future Generation Computer Systems* (2023).
- C.T. Ba, A. Galdeman, M. Dileo, C. Quadri, M. Zignani, S. Gaito. *Web3 social platforms: modeling, mining and evolution. Proceedings of the 1st Italian Conference on Big Data and Data Science (ITADATA 2022),* 2022.

1.2 Network evolution dynamics

In Part II we focus on understanding how blockchain-based systems evolve and how their users behave over time, problems of extreme interest. On one hand, understanding Web3 systems' growth will provide valuable insight into the understanding of dynamics in decentralized systems and potentially improve their design process; moreover, understanding the evolution of blockchain-based platforms allows us to understand the impact of the Web3 paradigm and features through comparisons with other systems, from other decentralized systems but not following the same paradigm, to the more traditional centralized ones. *Specifically, in this Part, we investigate how blockchain-based systems evolve from a microscopical perspective both in terms of single-user behavior and high-order structures.*

Studies conducted on systems showed that human dynamics are heterogeneous and characterized by a bursty behavior, i.e. they alternate periods of frequent activity or events and long intervals with a low frequency of activities or events [8]. This behavior has been measured on and crosses different human activities, such as email communications, mobile phone calls, library loans, or even letter correspondence, and more recently online social networks [9]. However, few attempts have focused on the microscopic dynamics of blockchain-based systems. Specifically, we focus on the following research questions: *RQ1)* Is the behavior of users in blockchain-based social networks characterized by bursty nature? *RQ2)* Is there bursty behavior across both social and financial interactions? In Chapter 3, we address these questions by

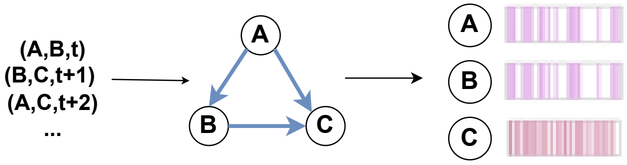


Fig. 1.2: **Analyzing evolution through bursty behavior.** *Starting from the interactions, in format (sender, receiver, timestamp), we construct the temporal network. Relying on the timestamp information we can analyze network evolution as a temporal process. For example, given node A, we can model the activity through a time series derived from the timestamps on outward edges. In the example, nodes A and B exhibit the typical traits of bursty behavior, i.e. periods of high activity separated by intervals with lesser or no activity, while node C does not.*

analyzing the dynamics of the link creation process and the claiming of rewards in the blockchain-based online social network (BOSN) Steemit [10]. We model large-scale blockchain data as a temporal directed network from which we extract the time series characterizing link creation and reward claims for each user in the network. Adopting a user-centric approach (see Figure 1.2), we evaluate the characteristics of the users' time series. By answering the previous research questions we provided the following contributions:

- The outcomes of the analysis highlight that the above processes regarding users' interactions show bursty traits typical of human dynamic *(RQ1)*.
- However, the creation of new relationships and the reward claim dynamics present a few differences concerning the types of models describing their behavior and the time scale of their bursty nature *(RQ2)*.

Another important aspect of the Web3 paradigm is that these new systems are strongly relying on cryptocurrency tokens to sustain themselves and generate profit. Specifically, the growth and success of these platforms are strongly dependent on the growth and evolution of the trade relationships among users. In this context, it is of paramount importance to understand the mechanism behind the evolution and growth dynamics of these economic ties: however, in these systems the trade relationships are strictly intertwined with social dynamics, posing significant challenges in the analysis. In particular, one of the most important mechanisms behind the evolution and the dyanmics of social networks is the triadic closure principle: individuals with a common friend have a higher chance to become friends themselves at some point in the future [11]. Given the strict link between social and economic spheres, the mechanism emerges as a potential candidate among mechanisms in literature [12, 13, 14].

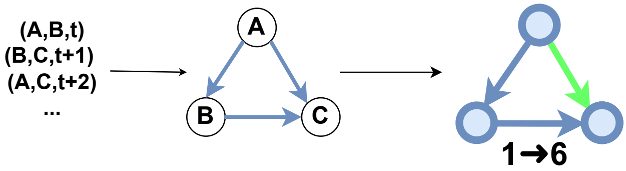


Fig. 1.3: **Analyzing evolution through triadic interactions.** *Starting from the transactions, in format (sender, receiver, timestamp), we construct a temporal network. To understand the evolution of the network structure, we focus on triads, i.e. 3 node subgraphs, and study how they change over time. In this example, nodes A and C have a friend in common at time t+1 and this leads to a triadic closure i.e. the formation of a link between A and C, at time t+2.*

When dealing with the triadic closure process, triads, and their census are the fundamental building blocks for describing the actual state of a network (closed triads) and for identifying where closures may occur (open triads). In Chapter 4, we investigate the impact of triadic closure in Web3 socio-economic networks. The analysis of triadic closure process commonly revolves around 3-node subgraphs known as "triads" (see Figure 1.3). The analysis aims to answer the following research questions: *RQ1)* From a static network perspective, are decentralized socio-economic networks similar in terms of triadic-based structures, or whether each network is characterized by specific triadic-based patterns depending on its nature? *RQ2)* From a temporal standpoint, do specific evolution patterns of the triads characterize different socio-economic networks or do they follow a common growth mechanism? *RQ3)* From a dynamic viewpoint, how does the triadic closure process change over time? Do the different types of triads resulting from a triadic closure process form at the same speed? Is the dynamic of triad formation stable along the evolution of these networks? To do so we extend the existing methodology for triadic closure studies and adapt it to directed networks snd we conduct an analysis both from a static [15] and temporal perspective [16, 17]. The methodology was applied to various decentralized socio-economic networks with distinct levels of social components. These networks include currency transfers from the blockchain-based online social media platform Steemit [18], trade relationships among Non-fungible tokens (NFT) sellers and buyers on the Ethereum blockchain [19], and blockchain-based currency for humanitarian aid Sarafu [20]. We provide the following main contributions:

- From a static standpoint the methodology highlights both similarities and differences across networks where the impact of the social components may vary, both from a static and temporal standpoint (*RQ1*).
- Similarities and differences emerge also from a temporal standpoint, where evolution patterns reflect the type of socio-economic activity of the network (*RQ2*).
- We employ metrics from social network analysis to describe the triadic closure process over time, and we also extended and formalized the triadic closure delay metric [21] for directed networks. Our measurements show how triadic closure is relevant during the evolution of these platforms and, for a few aspects, more impactful than centralized online social networks, where triadic closure is also incentivized by recommendation systems (*RQ3*).

Overall the results show strong evidence that triadic closure is an important evolutionary mechanism in the economic networks present in blockchain-based systems.
Part II is mainly based on the following publications:

- Cheick Tidiane Ba, Matteo Zignani, Sabrina Gaito. *Social and rewarding microscopical dynamics in blockchain-based online social networks. Proceedings of the Conference on Information Technology for Social Good,* 2021.
- Cheick Tidiane Ba, Matteo Zignani, Sabrina Gaito. *Characterizing growth in decentralized socio-economic networks through triadic closure-related network motifs. Online Social Networks and Media* 37-38 (2023) p. 1002662023.

1.3 The interplay of currency and user activity

Web3 gives the ability to set up alternative currency systems in various domains. Depending on the application domain, these currency systems may have different purposes and rules. The presence of this innovative element gives new options to the users, something that can lead to changes in their behavior, i.e. their actions or activities. This can lead to an interesting interplay, where the cryptocurrency system and user behavior are linked and influence each other: this interplay is an interesting open problem, yet not fully understood. *In*

Part III we would like to define new methodologies to analyze the impact of currency systems on user activity and vice-versa, the impact of user behavior on the currency systems.

In the case of blockchain-based online social networks — BOSNs, cryptocurrencies are used for content monetization, i.e. the money reward for the most popular content on a social media platform [22]. In blockchain-based online social platforms, network structure and the dynamics on top of it are strongly coupled with the cryptocurrency markets and reward systems: there is already evidence that the rewarding system influences user content [23, 24]. Indeed, the value of the cryptocurrency may influence users' efforts: the shocks that affect the cryptocurrency can influence user behavior and vice-versa. Therefore, among the many unknown aspects of these techno-social systems, a notable one is the understanding of the impact of the cryptocurrencies linked to the platform on the evolution of its social network and the behavior of its users. Addressing the above unknow aspect, we provide a methodology to answer the following research questions: *RQ1)* what is the interplay between currency and the evolutionary traits of the network; and *RQ2)* does and to what extent the reward system exert any influence on the users'activity?

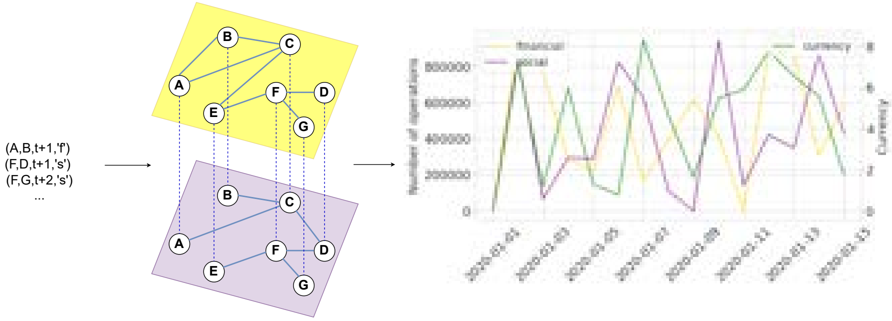


Fig. 1.4: **Interplay of user activities and currency.** *Starting from the transactions, in format (sender, receiver, timestamp, operation type), we can continuously update a multilayer temporal network, where each layer corresponds to an operation type. The activity on each layer can be monitored over time alongside the currency to understand the interplay.*

In Chapter 5, we present a methodology for addressing these research questions by leveraging a network-based approach based on the multilayer network model (see Figure 1.4). Specifically, we investigate the impact of the cryptocurrencies (in particular their value) linked to the platform on the evolution of its social network and on the behavior of its users, in terms of production of content and its promotion through a voting and reward system. To this aim, from Steemit, one of the most widespread BOSNs, we consider a three-year-long high-resolution data on its evolution along with the price of STEEM, its cryptocurrency. On users' activities extracted from these longitudinal data, we applied a time-series correlation analysis. In the case of most central accounts, we proceeded with a correlation analysis between the action allocation strategies and the obtained rewards.

By addressing the above questions, we highlight the following main insights:

- The analysis has highlighted pieces of evidence of the influence of the cryptocurrency price on users' actions, particularly on actions that shape the structure of the social network(*RQ1*).
- We also found that highly rewarded users prefer actions related to the promotion of content rather than the creation of high-quality content, exploiting the reward distribution mechanisms implemented by the platform (*RQ2*).

In general, these findings highlight that the shift of paradigm towards blockchain and cryptocurrency technologies might strengthen the influence of financial and economic factors rather than relational/social aspects on the evolution of these new complex techno-social systems.

Another interesting domain where Web3 leads to new dynamics is the field of humanitarian aid, where blockchain technology has been used in new-generation economic systems [25]. A very interesting example of such systems is complementary currencies, i.e. cooperative currency systems that support national economies to provide humanitarian aid and promote sustainable development [26]. While there are many studies on the principles and case studies of successful complementary currencies [27, 28, 29], many aspects are still unexplored, especially regarding users' behavior. As in the previous case, the introduction of a cryptocurrency-based system means that there could be an interplay between user behavior and the currency value. However, there is a key difference with the case of online social media, i.e. the different purpose of the currency: the currency system's main objective is not the profit for users involved, but to create a sustainable local economy. Complementary currencies are often born out of cooperation among members that face a period of crisis or they usually have the objective of creating bonds of reciprocity and integrating social networks between people, which should lead to increased cooperation. Therefore it is not only the individual's behavior of interest, but even the *cooperative behavior* raising in these systems is a key aspect. However, there is a lack of studies on many aspects of cooperative behavior in complementary currencies, such as how such behavior changes over time, especially in times of a crisis like the COVID-19 pandemic. Moreover how cooperation behavior is affected by time and different geographical locations is still unclear.

In Chapter 6, we focus on Sarafu, a complementary currency that went digital and now relies on blockchain technology [30]. Sarafu is a successful case of a complementary currency that was used for humanitarian aid during the COVID-19 pandemic [20]. Moreover, Sarafu is a perfect case study for investigating cooperative behavior, as it implements a special type of account, the *group account*, to support cooperation groups [31]. This feature supports the study of group dynamics and behavior. We target the following issues: *RQ1)* the impact of cooperation groups and how it changes over time as we consider different pandemic situations and restrictions, *RQ2)* how cooperation groups allocate and redistribute resources, considering their *business type*s (such as "food", "farming", etc.), *RQ3)* the impact of geographical location on cooperative behavior, and *RQ4)* the interplay between the geographical location and how users or cooperation groups allocate and redistribute resources. In the Chapter, we propose a methodology to leverage the data on user transactions and user attributes to characterize the behavior of group accounts through their transactions, in terms of their spending behavior as well as to characterize how they are funded (see Figure 1.5). The methodology showcases how to model the transaction data as a temporal network, which is then analyzed over time to study the flow of money across accounts and user behavior changes. We provided the following contributions:

- Sarafu users exhibit a strong reliance on cooperation: group accounts have a crucial role, as they are few (0.38%) and yet handle a significant amount of transactions (36%); moreover, their importance even increases over time, as the amount of money spent by these accounts increases significantly over the observation period (*RQ1*).
- Second, we also found that the allocation of resources by cooperation groups changes the observation period, as we observed variations over the categories of products of interest (*RQ2*).
- Third, we observed that while cooperation is important across different geographic locations, not all areas relied immediately on group accounts (*RQ3*).
- Fourth, we found an interesting interplay between geographic areas and the allocation of resources: geographical areas are characterized by their categories of interest, with urban and periurban areas showing some similarities (*RQ4*).

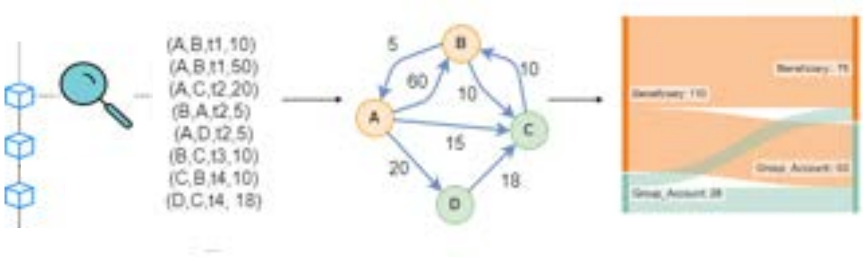


Fig. 1.5: **User activity through currency flows.** *An example outlining the methodology to leverage user attributes to analyze user activity. Starting from the transactions, in format (sender, receiver, amount, timestamp), we filter them on the timestamps to obtain a subset for the period of interest. Then, we construct the transaction network. Relying on the weights and attributes of the transaction network, we can aggregate to construct the Sankey diagrams. In the example of the transaction network, nodes are colored according to the type, while the weights on links correspond to the number of tokens flowing from the source to the destination. Monitoring the networks and the Sankey diagrams over time allows us to detect changes in user behavior.*

Part III is mostly based on the following publications:

- Cheick Tidiane Ba, Matteo Zignani, Sabrina Gaito. *The role of cryptocurrency in the dynamics of blockchain-based social networks: The case of Steemit. PLOS ONE* 17.6 (June 2022) pp. 1–22.
- Cheick Tidiane Ba, Alessia Galdeman, Matteo Zignani, Sabrina Gaito. *Temporal Analysis of Cooperative Behaviour in a Blockchain for Humanitarian Aid during the COVID-19 Pandemic. Proceedings of the 2022 ACM Conference on Information Technology for Social Good,* 2022.
- Cheick Tidiane Ba, Matteo Zignani, Sabrina Gaito. *Cooperative behavior in blockchain-based complementary currency networks through time: The Sarafu case study. Future Generation Computer Systems* (2023).

1.4 Modeling and prediction of user migration

In the world of Web3, we have the data to deepen our understanding of problems that affect traditional platforms. One such example is *user migration*, i.e. the movement of large sets of users from one online social platform to another one [32]. The growing popularity of online social media (OSM) has led to the creation of a wide amount of social media platforms. In this context, the increasing competition among platforms and the emergence of decentralized alternatives such as Blockchain Online Social Network (BOSN), have led to more frequent user migrations: individuals tend to switch platforms in search of improved features, content, or communities. Therefore there has been increasing interest in studies modeling and predicting user migration [33, 32, 34, 35]. In these recent works, user migration has been successfully modeled through networks but with some limitations. *In Part IV, we address the following problems: i) there is a lack of modeling methodologies for user migration, especially in the case of blockchain forks, i.e. bifurcation of the main branch of the original blockchain, that allows users to create new platforms originating from the original one, and ii) a framework for network-based prediction is not defined.*

First, in Chapter 7, we deal with the evolution of BOSNs from the perspective of user migration across platforms as a consequence of a fork event. We propose a general, network-based, user migration model applicable to BOSNs to represent the evolution patterns of fork-based migrations, the multi-interaction structural complexity of large-scale BOSNs, and their growth characteristics. The resulting model is described in Figure 1.6. By this framework, we answer some open questions on user migration such as *RQ1)* what is the impact of fork events on the social and financial networks? *RQ2)* Is user migration predictable through network structure? *RQ3)* Are social or financial structures equally important for prediction?

We applied our framework to the case study of the Steem-Hive fork event [36], and we provided the following contributions:

- By the multilayer temporal network approach, we are able to monitor network metrics and evaluate the effects of user migration on the networked structure of the interactions. In particular, we show that most properties remain similar even after the split, although a few do change (density, diameter). Network metric changes affect both social and financial interactions (*RQ1*).
- We cope with the task of predicting user migration in the case of a fork, i.e. forecasting if users will remain on the original blockchain or they will migrate to the new one. We show how network structure, without any additional information, is enough to obtain remarkable prediction performance (*RQ2*), an important result as in Web3 datasets we usually do not have many features associated with users' account.
- We show that when performing prediction tasks, it is important to consider both social and economic information, regardless of the learning algorithm considered (*RQ3*).

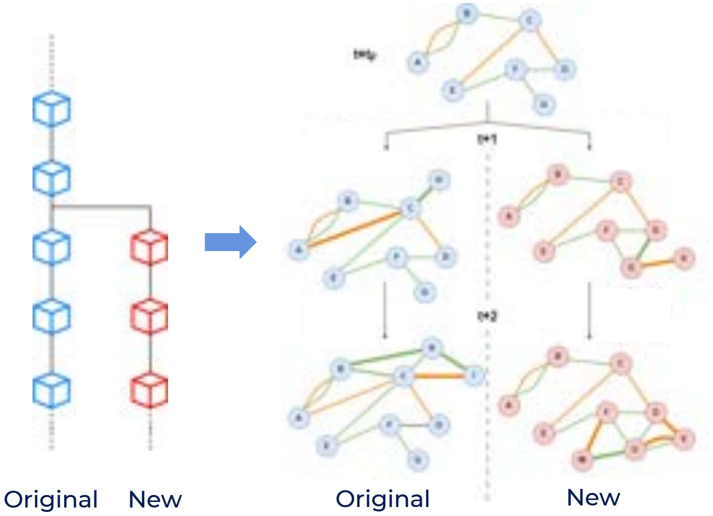


***Fig. 1.6:* The framework for network analysis at fork time.** *Given a stream of transactions in the format (sender, receiver, timestamp, type), we can reconstruct the evolution of the graph over time as sequences of temporal graphs. In the example, the graph on top represents the state of the network at fork time t_F. An interaction will result in a new link (in bold) in the corresponding graph (colored based on their type). Then at fork time, the two sequences evolve independently. The sequence on the left describes the evolution of the original blockchain, while the sequence on the right is related to the new blockchain. By analyzing the sequences, we can monitor network metrics and evaluate the effects of user migration on the graph structure of the interactions.*

Given the promising performance of network structure, we decided to investigate the applicability of prediction methods tailored for graph structure, i.e. *graph neural networks* [37]. Existing methods rely on user activity to derive interaction graphs and then address the user migration prediction problem as a node classification task, where user decisions are encoded as node labels. While the performance looks promising, there are currently two important research gaps: *i)* there is no work using graph neural networks, the state-of-the-art in machine learning on graphs; and *ii)* there is a lack of methods designed to improve prediction performance in the case of class imbalance, i.e. the presence of dominant behavior among the ones to predict [38]. In Chapter 8, we propose a machine learning pipeline utilizing graph neural networks (GNNs)

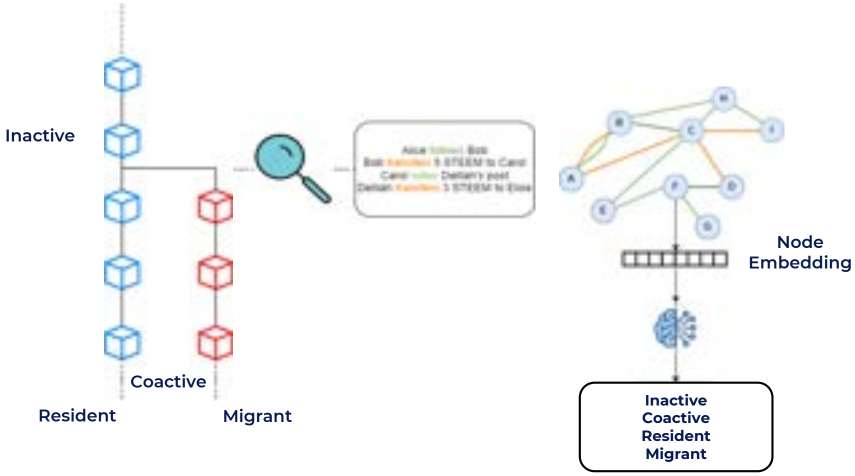


***Fig. 1.7:* Network-based prediction of user migration.** *We represent the prediction setting used for user migration prediction. The graph at fork time is derived from transaction data up to the fork. From the transaction data, we derive user decisions, i.e. a user can stop using the platforms (inactive) or be active only on the original chain (resident), or only on the new one (Migrant) or undecided (Coactive). The machine learning models are trained to predict these cases from a node's network structure.*

to predict user migration in BOSN. Here, our main goal is to verify to what extent graph neural networks are suitable methods for tackling the prediction of users' migration across blockchain-based platforms (*RQ1*). Moreover, we also deal with how to properly handle a severe imbalance of the classes in the graph neural network framework (*RQ2*).

Also in this setting, we model the data as a directed temporal multilayer graph, capturing social and monetary interactions among users. To address the problem of class imbalance in node classification, we introduce a data-level balancing technique following an undersampling approach. In general, we provided the following contributions:

- Graph neural networks are a suitable machine learning approach to perform user migration prediction as shown by an extensive evaluation conducted on data describing user migration across blockchain online social media platforms (*RQ1*).

- We proposed a balancing method following an undersampling approach that produces a more balanced training set and showed how it improves predictive power on severely imbalanced data (*RQ2*).

These results highlight how graph neural networks are effective in predicting user migration, without the need for manual feature engineering and in the absence of user information.
Part IV is based on the following publications:

- Cheick Tidiane Ba, Andrea Michienzi, Barbara Guidi, Matteo Zignani, Laura Ricci, Sabrina Gaito. *Fork-Based User Migration in Blockchain Online Social Media. 14th ACM Web Science Conference 2022,* 2022.
- Cheick Tidiane Ba, Alessia Galdeman, Manuel Dileo, Matteo Zignani, Sabrina Gaito. *User migration prediction in blockchain socioeconomic networks using graph neural networks.* Proceedings of the 2023 ACM Conference on Information Technology for Social Good, 2023, Lisbon, Portugal.

1.5 Machine learning on multilayer graphs

While good results can be obtained considering each layer separately, recent literature showed the superiority of machine learning methods more suited for multilayer graphs. Specifically, we proceed with our work on graph neural networks by focusing on multilayer graph neural networks (MLGNN) [39]. Indeed, to leverage MLGNN on large-scale datasets that are characterized by a wide set of different relationships among a large set of entities, it is fundamental to improve performance and scalability: this can be done through graph simplification approaches [40] to remove noise or redundant information. While various graph simplification approaches based on machine learning are available for single-layer graphs [41, 42, 43], there is a lack of suitable ones for multilayer graphs. This is an important issue when dealing with large-scale datasets, a key issue with Web3 but not limited to the field. *For this reason, in Part V, we focused on the development of a framework to simplify multilayer graphs, i.e. reduce the size of the graph.*

Graph simplification could be useful for the application of graph neural networks in multiple fields, including Web3. The key aspects to investigate can be summarized by the following research questions: *RQ1*) What is the impact of graph simplification performed on multilayer graphs? *RQ2*) How does graph simplification influence the structure of multilayer graphs? In Chapter 9, we propose the MultilAyer gRaph simplificAtion (MARA) framework, a GNN-based approach designed to simplify multilayer graphs based on the

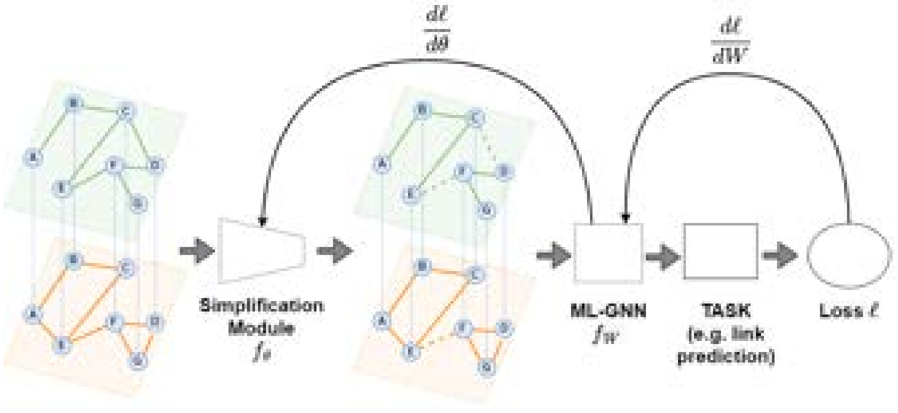


Fig. 1.8: **Proposed multilayer graph simplification framework.** *Overview of the proposed approach. A simplification module f_θ and multilayer graph neural network f_W, are used to generate node embeddings for a downstream task. If the simplification module is trainable e.g. a neural network, it is possible to train the two components jointly: through gradient descent, we update the parameters θ, W backpropagating from the loss function ℓ. In this case the simplification module can learn to detect noisy links specifically for that task.*

downstream task. MARA generates node embeddings for a specific task by training jointly two main components: i) an edge simplification module and ii) a (multilayer) graph neural network (see Figure 1.8). We tested MARA on different real-world multilayer graphs for node classification tasks.

In the context of simplification of multilayer graphs, we provided the following contributions:

- MARA reduces the dimension of the input graph while keeping and even improving the performance of node classification tasks in different domains and across graphs characterized by different structures (*RQ1*) as highlighted by different experimental results.
- Deep learning-based simplification allows MARA to preserve and enhance graph properties important for solving the task (*RQ2*).

Therefore, we are able to tune simplification and machine learning models at the same time: the strategy improves performances not only for user migration prediction but also with other node classification tasks on graphs in other domains.

Part V is based on the following publications:

- Cheick Tidiane Ba, Roberto Interdonato, Dino Ienco, Sabrina Gaito. *TAMARA: a task-aware multilayer graph simplification framework.* Accepted to conference 20th International Workshop on Mining and Learning with Graphs (ECMLPKDD 2023).

1.6 Conclusions

In this thesis, we focused on the study of temporal dynamics in decentralized systems following the Web3 paradigm. One of our primary contributions was the analysis and investigation of the more innovative Web3 applications, that go beyond the classical transaction systems like Bitcoin and Ether. In fact, we focused on complementary currencies — characterized by rich attributes, NFTs that are very different tokens, and blockchain online social media characterized by multiple interaction types and cross-chain behaviour. For these Web3 systems, we provided background and collected new datasets when not publicly available. A key highlight is a novel and unique Steem-Hive dataset that allows the study of decentralized social media, especially from a temporal perspective. Beyond the aspects studied in this thesis, these data can contribute to many more scenarios, from building network models to benchmarking machine learning algorithms for dynamics settings. In addition, we proposed new methodologies encompassing the processing from modeling to mining, using a network-based approach. Indeed, we have made significant methodological contributions to the field of network science. Our contributions include methodologies for representing Web3 data as networks, employing various network models. Specifically, we provided the methodology to model data as temporal networks to investigate different aspects of networks, such as bursty behavior, triadic closure and cooperative behavior. We also showed how to employ a temporal multilayer graph to examine the interplay between currency and user activity. We also introduced the concept of a temporal multilayer heterogeneous network for the analysis of Web3 platforms, an approach that we plan to apply to other complex systems such as knowledge graphs or recommender systems. These modeling methodologies represent a crucial advancement in applying network science to Web3-related challenges. In addition to the application of mechanisms and principles from network science, we also formalized and extended current methodologies, as in the case of triadic closure, where we introduced the directed triadic closure delay measure to gauge triadic closure speed in directed networks. Additionally, we laid the groundwork for the use of the combination of network science and machine learning for prediction tasks in Web3 networks by introducing methodological frameworks for user migration prediction, class-imbalanced learning in graph neural networks, and deep-learning-based graph simplification for multilayer graphs. Overall, we showcased how to perform prediction with state-of-the-art machine learning on graph approaches with convincing results. Finally, the applicability of the work presented in this thesis is not limited to Web3 systems. It provides tools applicable to any domain with similar complexity and challenges in terms of high temporal resolution, heterogeneous action types, or interactions across different systems.

In addition to the previously mentioned publications, during the PhD, we also contributed to the following papers:

1. Manuel Dileo, Cheick Tidiane Ba, Matteo Zignani, Sabrina Gaito. *Link Prediction with Text in Online Social Networks: The Role of Textual Content on High-Resolution Temporal Data. Discovery Science: 25th International Conference, DS 2022, Montpellier, France, October 10–12, 2022, Proceedings,* 2022.
2. Cheick Tidiane Ba, Matteo Zignani, Sabrina Gaito. *The role of groups in a user migration across blockchain-based online social media. 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops),* 2022.
3. Cheick Tidiane Ba, Alessia Galdeman, Manuel Dileo, Matteo Zignani, Sabrina Gaito. *Analyzing user migration in blockchain online social networks through network structure and discussion topics of communities on multilayer networks.* Submitted to journal *Distributed Ledger Technologies: Research and Practice.*
4. Alessia Galdeman, Cheick T. Ba, Matteo Zignani, Christian Quadri, Sabrina Gaito. *City consumption profile: a city perspective on the spending behavior of citizens. Applied Network Science* 6.1 (Aug. 2021).
5. Ben Steer, Naomi Arnold, Cheick Tidiane Ba, Renaud Lambiotte, Haaroon Yousaf, Lucas Jeub, Fabian Murariu, Shivam Kapoor, Pedro Rico, Rachel Chan, Louis Chan, James Alford, Richard G. Clegg, Felix Cuadrado, Matt Barnes, Peijie Zhong, John Pougué-Biyong, Alhamza Alnaimi. *Raphtory: The temporal graph engine for Rust and Python.* Arxiv, Submitted to *Journal of Open Source Software (JOSS)*, 2022.

Work 1 presents a methodology for extracting consumption behaviors from credit card transaction data to create city consumption profiles. Utilizing network-based representations and community detection algorithms, the study identifies unique city profiles influenced by mono-categorical consumption patterns. The city consumption profile serves as a tool for understanding economic behaviors, comparing cities, and informing tailored city services. In works 2 and 3 we further explore our study of user migration in blockchain online social media. In work 2, we performed a network-based analysis centered on the identification of communities on multilayer networks. The paper showed that communities are characterized by different migration behaviors: for example, users in communities formed through economic transactions are more likely to stay. In work 3, we leveraged text mining approaches to characterize communities based on their posted content in the paper: we observed how users are characterized by different discussion topics. We leveraged text mining also in work 4, to perform link prediction tasks focusing on the following relationships in Steemit, isolating the "follow" layer to perform link prediction. In this work, we show how the combination of structural and textual features enhances the prediction performance of traditional models. Deep learning architectures outperform traditional ones and they can also benefit from the addition of textual features. Finally, work 5 introduces Raphtory, a platform for constructing and analyzing temporal networks. It features methods to create networks from diverse data sources, algorithms for exploring structure and evolution, and a GraphQL server for application deployment. The core engine, efficient in Rust, integrates with Python for usability, making Raphtory a versatile tool for temporal network analysis and application development.

Part I

Background

Chapter 2

Web3: background, network-based modeling, datasets

2.1 Web3: the building blocks

In this section, we dissect the essential components of the Web3 paradigm. We start with some concepts on the underlying Blockchain Technology. Next, we describe Consensus Mechanisms, the governing protocols ensuring data validation in Web3 platforms, and how they can change, i.e. the mechanism of forks. Finally, we delve into the pivotal role of Smart Contracts and Tokens driving some of the most interesting features of decentralized applications.

2.1.1 Blockchain technology

A blockchain is one of the possible implementations of a distributed ledger [44]. Its characteristic trait is that single pieces of information, usually called *transactions*, are grouped together into *blocks*, and each block is cryptographically linked to its predecessor as the mails of a chain. More precisely, blocks can be identified through a timestamp or their hash, and the chain is formed as each block has a hash of the previous block (Fig 2.1). The records stored in these blocks are stored publicly and distributed to all the user servers inside the peer-to-peer (p2p) network: therefore, the blockchain is a publicly distributed ledger of records, shared among users. In a blockchain-reliant system, the data is no longer handled by a single company, instead multiple nodes have a copy or replica of the blockchain. The blockchain acts as a public database of all the transactions among the users: we obtain a transparent system, censorship-resistant as multiple nodes have access to the data, making it hard to tamper with information. While Information security protocols, such as encryption and hashing, protect data integrity and safeguard data against unauthorized access [45].

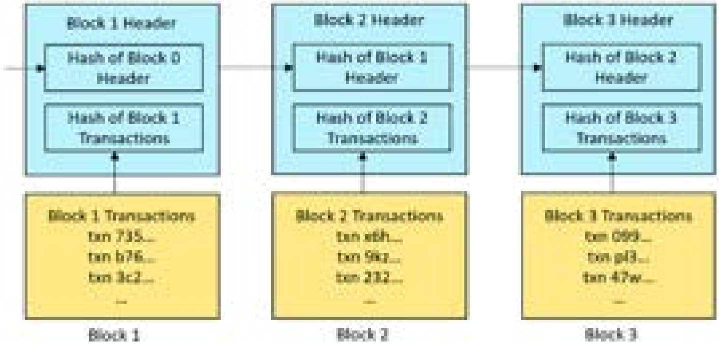


***Fig. 2.1:* Illustration of a blockchain.** *The blocks of the chain, in blue, are cryptographically linked. In each block, we can observe the transactions (in yellow). Inside the transactions, we can find the information about the transactions: for example in Bitcoin, the sender of money, the receiver, the amount of money sent, and so on.*

In this setting, the distributed nodes in the network need to be synchronized with each other and reach an agreement on which transactions are legitimate and how they should be added to the blockchain. Therefore blockchain systems need a consensus mechanism, i.e. a fault-tolerant mechanism to reach an agreement on a single state of the network among the distributed nodes [46]. The term consensus mechanism can be used to refer to the complete stack of ideas, protocols, and incentives that enable a distributed set of nodes to agree on the state of a blockchain [47]. There are many consensus mechanisms proposed in scientific literature and industry publications (more than 130 according to recent surveys [48]). The key aspect of the consensus mechanisms is the consensus protocol, i.e. how the validation of transactions occurs and how the distributed are incentivized to run the network. The main consensus protocols eliminate the need for central authority, as new blocks are verified by the collective of nodes: the transaction is encrypted and set to all nodes; if the transaction is considered valid by the majority of the nodes, a new block with the transaction is generated and sent to all nodes. To ensure correct execution of the consensus mechanism, the most common approach is to reward nodes with *cryptocurrency*, i.e. digital currency tokens. In the following, we'll provide an overview of the most adopted consensus mechanisms.

2.1.2 Consensus mechanisms

There are many consensus mechanisms presented in the literature that vary under different aspects such as the degree of centralization, scalability, and many more factors [48] leading to some systems adopting new mechanisms as time progresses. Moreover, protocols need to address and prevent things such as the 51% attack [49].

The most notorious one is called **Proof of Work (PoW)** [50] and it was introduced in 2008 by Nakamoto's Bitcoin system. In PoW, all the network nodes are called *miners* and maintain a copy of the ledger. Unverified transactions are broadcast to the entire blockchain network. A miner groups the transactions in a potential block while verifying the incoming and outgoing token amounts to avoid double-spending. All miners then compete to solve a complex mathematical problem that ensures the validity of the proposed block and that it follows in sequence the last block in the chain (that the majority of nodes have agreed on). The first miner to complete the task creates and appends the new block for the blockchain and is rewarded with native cryptocurrency tokens: in Bitcoin, miners receive a fraction of Bitcoin, the *Satoshi*. Bitcoin's task consists of discovering a value known as a "nonce": this is a value that is to be inserted in the proposed block, but is unknown. The idea is that the proposed block used as an input of a hash function, should provide a value within a predefined target range and should be prefixed with a number of 0s. Due to the inherent properties of hashing algorithms, finding the right nonce demands a brute-force strategy, i.e. iteratively testing different values, until a valid solution is attained [51]. While effective and diffused, Proof of Work has a key limitation: the production of blocks is very slow, as the validation process is complex and the typical time taken for the formation of a block is around 10 minutes, on average [45]. Some works claim that Proof-of-Work-based blockchains can't scale beyond three transactions per second, which is a fraction of the world's financial traffic [52]. Moreover, as it requires competition among miners, miners need high-cost mining rigs to win the competition, leading to a high waste of energy and computing power.

These limitations are the reason why one of the most popular blockchains after Bitcoin, Ethereum [53] moved on from PoW recently [54]. On September 15th [55], Ethereum moved to the protocol called **Proof of Stake(PoS)** [56]. In PoS, miners do not compete: instead, a miner is selected among the available ones. PoS selects the node based on the number of native cryptocurrency tokens in their possession, i.e. their current “stakes”. The nodes holding more stakes are more likely to be chosen as block producers; the selection however is not purely based on stakes to avoid centralizing the network. Like in PoW, their effort is then rewarded with some transaction fees. To deter malicious behavior, validators are compelled to lock in their stakes, ensuring a commitment to proper block generation. Given the lack of competition, PoS is more energy-efficient overall. However, the importance of the stakes means that token movement tends to be discouraged.

Another popular variant of the PoS mechanism is **Delegated Proof of Stake (DPoS)** [57]. DPoS is a consensus mechanism that uses a democratic voting and election process to select a small group of miners who create and validate blocks, avoiding competition among miners [58]. Users vote for “Witnesses” or “Delegates”, and similarly to PoS, users’ voting power depends on the proportion of owned tokens. The selected witnesses earn transaction fees for their services, while they can be voted out in case of malicious behavior or lack of production. In DPoS The witness produces blocks at high speed: this allows blockchain online social networks to handle the high frequency of actions, at a rate typical of online social networks.

Finally, another commonly used protocol is **Proof of Authority (PoA)** [59]. Proof of Authority (PoA). It’s a mechanism that does not focus on decentralization, in fact it concedes the possibility to produce blocks only to a group of trusted "authorities", i.e. nodes that provide proof of their real-world identity. Like in other mechanisms, authorities are rewarded for their work. This mechanism is which is more suited for blockchain networks formed among enterprises or large institutions, that are interested in blockchain features but with high transaction throughput and energy efficiency.

2.1.3 Forks

Given the huge amount of change in the field, consensus protocols need to be modified and adjusted over time. When miners need to modify their behaviour, they make a ***fork***. Forks in a blockchain occur during significant technical upgrades or alterations to the network protocol, resulting in a shift in the established rules. The nodes in the network are required to update their software to incorporate the modified fork rules. Block creators (miners or validators) and nodes must then produce and validate blocks in accordance with the updated rules. While forks are typically discussed and planned to ensure a synchronized adoption of the changes, occasional disagreements may lead to a temporary network split, with divergent blocks adhering to either the old or new rules. In rare instances, disputes over forks can result in a permanent division of the network [60]. There are two types of fork: **soft fork**, and **hard fork**. A soft fork happens when the change to the protocol governing the blockchain is retro-compatible with the previous version. Indeed, in the case of a soft fork, usually, all the miners keep adding blocks on the same chain. Instead, in the case of a hard fork, miners running different versions of the protocol will see each other blocks as invalid, and therefore they might create two distinct *branches* of the blockchain. Since hard forks are more drastic, miners will usually agree upon a specific time at which they should upgrade the protocol, helping to minimize everyone's loss. Hard forks are events for which a migration phenomenon may happen, depending on the motivation that caused the hard fork. In addition, to introduce small modifications to the consensus protocol, soft forks can be used in very specific scenarios, such as to freeze account funds or revert certain transactions.

2.1.4 Smart contracts and tokens for decentralized applications

As we have seen in the previous sections, **native cryptocurrency tokens**, i.e. tokens generated during the block production process, play a key role for everyone involved in the system. However, relying solely on native tokens has some issues, especially as certain consensus mechanisms do not favour movement. It becomes necessary for these applications to be able to generate different tokens, to sustain their functioning: a popular solution relies on **smart contracts** [53]. Smart contracts allow the development of platforms that do not need to rely on the native currencies and can run in a decentralized setting — the **Decentralized Applications (DApps)**. A smart contract is a piece of code whose execution outcome is agreed upon by all the nodes of the network, and executed directly on-chain in a decentralized fashion. We can write the smart contract code in a smart contract language, high-level languages for implementing smart contracts. The contract then needs to be deployed i.e. written on the blockchain. This code can do different things, including the creation of new assets or tokens. Indeed, smart contracts can be more complex: contracts can be composed which means that developers can reuse existing contracts and combine them to make more complex ones. Common actions are the creation of tokens (*minting*) and the destruction of tokens(*burning*). After deployment, smart contracts can be activated after a transaction making it so that the smart contract conditions are met.

Tokens are fungible when we can substitute any single unit of the token for another without any difference in its value or function [61]. The most common usage of smart contracts is for the generation of **Fungible Token** that flank the native cryptocurrency tokens and that can be spent by the user to unlock various functionalities within decentralized applications (DApps). In Ethereum, there are many fungible tokens, they follow the same ERC-20 standard, which is the most widely adopted standard, indicating a series of properties that the smart contracts need to uphold to ensure compatibility within the Ethereum ecosystem. Among the digital currencies, we also have stablecoins, i.e. cryptocurrencies that are pegged to fiat currencies, providing a reliable and less volatile value. The most known stablecoins are USDC (USD Coin), DAI, and Tether (USDT). Another interesting type of token is **Non-fungible token (NFTs)**. Non-fungible tokens are tokens that each represent a unique tangible or intangible item and therefore are not interchangeable [61]. So, an NFT provides a certificate of ownership of a digital object [19]. An NFT is linked to a given digital asset to attest to its uniqueness and non-transferability: in practice, an NFT can represent a variety of digital items, including photographs, movies, and audio. As a consequence, several fields, such as art, gaming, and sports collectibles, utilize NFTs to regulate and control digital objects. Currently, most NFTs follow the standard ERC-721 [62], which defines the smart contract structure that generates the NFT assets (more details in section 2.2).

2.2 Web3 applications

Initially, Web3 principles were applied in the economy, leading to Decentralized Finance (DeFi). A well-known example is Bitcoin, where blockchain technology was applied to store economic transactions among a network of untrusted nodes. But, Web3 principles have been applied to many more fields: to cite a few well-known examples, the application in governance resulted in Decentralized Autonomous Organizations (DAOs); or in social networks, it resulted in Blockchain online social networks (BOSNs).

2.2.1 Web3 for social media: blockchain online social networks

Blockchain-based online social networks (BOSNs) are an emerging application of blockchain-supported technologies and present some novel and interesting characteristics [10]. These platforms offer i) a set of “social actions” — following, commenting, and voting — which facilitate online interactions among accounts; and ii) whose core functions are rooted in an underlying blockchain that guarantees the persistence and validity of operations.

Because of their decentralized nature, BOSNs alleviate certain frequent issues with regular online social networks, such as the so-called *Single Point of*

Failure. From the point of view of the users, BOSNs are particularly resilient to *censorship*. One of the most enticing features of blockchain technology in this sector is its ability to bring value and usefulness to social platforms by establishing a *Rewarding System* for good contributions. These incentive systems can be designed to encourage positive behavior in many elements of the platform, but their main and major focus is on the awarding of outstanding content and its thoughtful evaluation [63, 64]. Rewards are generally issued as cryptocurrency tokens adding a new dimension compared to traditional OSNs. In fact, in traditional OSNs, user interactions are only "social". Users post and share content on the platform, other users interact using comments or votes to express likes or dislikes. In BOSNs, users can also interact through “economical” or “financial” interactions, as users can share the cryptocurrency tokens by asset transfer actions, i.e. they can move a certain amount of tokens from a source account to a destination account. Nevertheless, blockchain technology also has some disadvantages: it is affected by issues concerning the consensus protocol, such as the 51% attack [49], for instance. Another limitation common to a decentralized social system is the eternal dilemma of content moderation, for which a clear solution has not yet been found. Lastly, since the blockchain is an append-only structure, it is hard to modify it in case some illegal content is put on it.

As blockchain technology became more studied, more and more social media platforms started experimenting with the blockchain for some functionalities or as a core part of their design. One of the most prominent Blockchains, Ethereum, is the host of 3 different Blockchain Online Social Networks: Sapien, Peppeth, and Minds. Sapien[1] is a social media platform focused on article publishing. The system rewards active users with a token produced by the Ethereum Blockchain, called SPN. The chain is used for both storage and cryptocurrency transactions. This social embraces user privacy: users can set different levels of visibility to their content. Minds[2] is a similar content-sharing platform. It runs on the Ethereum blockchain. We have an encrypted messenger and some anonymity options. This is more similar to centralized Patreon, a website where users hide content behind a paywall and money is required to subscribe and view more content. Similarly, in Minds, users pay content creators with tokens to view more content, through tips and paid subscriptions. Another service running on Ethereum is Peepeth[3] a Twitter-like social media platform. The platform relies on the Ethereum blockchain mainly for

[1] https://www.sapien.network/

[2] https://www.minds.com

[3] https://peepeth.com/welcome

storage, with a focus on the execution of Smart Contracts. Another important blockchain home of several dApps is the Steem Blockchain. The main platform hosted is the BOSN Steemit[4] a content-sharing platform with a Reddit-like interface and a large user base. It relies on the Steem blockchain for both storage and financial sustainability. The stored data is completely public, as the emphasis is on the reward system, described as tamper-free and transparent. The Steem Blockchain is also home to other platforms, like Appics [5] another content-sharing platform reliant on Steem. Recently, the Hive Blockchain has gained relevancy. Hive [6] blog social su hive blockchain, it's very similar to steemit, as it was born as a community-driven hard fork of the Steem blockchain, on March 18th, 2020. This means that the data for the 2 blockchains is identical up to that date. The main web interface, is very similar to Steemit, with a Reddit-like interface, where users can share articles, comment and gain rewards. A more in-depth description of the Steemit and Hive platforms will be presented in the following sections.

Steemit

Steemit is regarded as a pioneer for the Web3 ecosystem since it introduced the seminal concepts of the rewarding system in a social network [65, 22] and delegated proof-of-stake (DPoS) consensus algorithm for block validation in social networks apps. Steemit has been one of the first and most successful platforms in the blockchain-based online social network ecosystem, and it has introduced most of the fundamental mechanisms that characterize modern BOSNs.

Launched in 2016, the platform supports the creation and sharing of content, as well as a social network based on "follow" relationships. In Steemit, users create original blog posts, that can be shared or upvoted/downvoted by other users. Users can be *creators* — content producers — or *curator*s — content promoters. The promotion and evaluation of content are made through social actions, such as upvoting (e.g. Facebook's "like", Twitter's "heart" button), downvoting (dislike), and sharing. The role of a user towards content determines how rewards are distributed. In fact, all these actions not only increase the visibility of posts but also have an economic impact. But, unlike other popular online social networks, the economic impact of these actions is explicit and measurable through the amount of gained tokens. In fact, at the

[4] https://steemit.com
[5] https://appics.com
[6] https://hive.blog

end of a 7-day period, the most popular posts are awarded through cryptocurrency tokens, and both creators and curators of the most liked posts get a share of this reward. These mechanisms are inspired by the attention economy and token economy principles [66]. Indeed, active users have a financial incentive for their participation, as they are rewarded for their contributions to the platform. Rewards are distributed in the form of cryptocurrency, which can be traded among users and can be exchanged for traditional currencies like the US Dollars — USD. This way the economic value of posts and users is easily quantifiable and publicly available. This last point constitutes the pivotal link between the socio-economic dynamics internal to the platform and the external financial ones, first of all, the trend of the cryptocurrency market.

In the next sections, we focus on two specific aspects, common to most of the current BOSNs:

1. a token system based on proprietary cryptocurrency used both for fostering high-quality content and users and supporting the validation of all social and economic actions; and
2. a rewarding system for distributing the wealth of the platform

The token system

The rewarding system, the importance — influence — of the users, and the inter/intra financial relations are mainly based on the cryptocurrency system of Steemit, which includes three different tokens, each with a specific purpose [67]:

1. STEEM[7];
2. Steem Based Dollar — SBD; and
3. Steem Power — SP.

A summary representation of the token system is shown in Fig. 2.2, reporting the possible conversion methods as well.

The first token, STEEM, is the liquid cryptocurrency at the base of the token system. This token can be exchanged by users as a form of payment and it is tradable on different exchanges with other cryptocurrencies or more traditional currencies like US dollars. These characteristics cause the STEEM value to fluctuate. This is a key point of this study which has precisely the purpose of investigating how such fluctuations affect the usage of social actions, and, consequently, the structure of the social network.

As described in the Steem Blue Paper [52], there is often confusion behind the relations between tokens and their real-world values. Steem Based Dollar

[7] Capitalized to avoid confusion with the Steem blockchain.

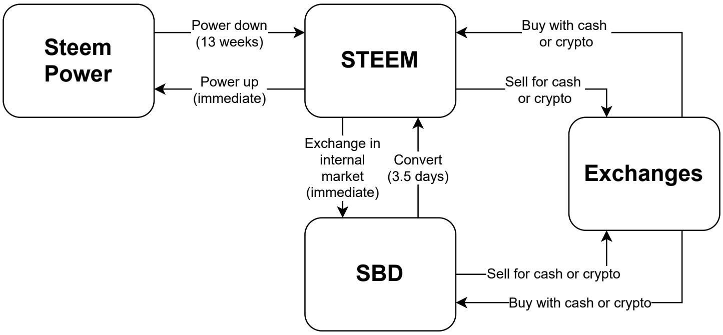


Fig. 2.2: **Tokens and conversion operations.** *Main currencies (rounded rectangles) in Steemit. Possible conversion operations are depicted as arrows. For each conversion or exchange operation, we report the type of the operation and temporal constraints, when available. In fact, many operations are instantaneous, while some others require more days.*

(SBD), a coin that represents the value of STEEM as US Dollars, has been introduced to mitigate this issue. In fact, SBD makes the economic system more accessible to newcomers: for example, rewards are usually shown as SBD. As STEEM, SBD can be bought and traded outside the Steemit platform through exchanges.

The third currency is Steem Power (SP). This currency quantifies the amount of investment in the platform, i.e. the amount of wealth staked in the platform. Steem Power is the equivalent of market shares of Steem. As in common shares, if the value of the company increases, so does the value of users' shares. Users are incentivized to invest in the platform as holding more Steem Power gives them more influence in the network in different ways. In fact, Steem Power plays a key role in the rewarding process, as shown later: posts voted by users owning a large volume of Steem Power gain more visibility and the top posts also get a larger share of the reward pool. Steem Power cannot be acquired or traded, a substantial difference from the other currencies. In short, the only way to get Steem Power is either as a reward or by investing in the platform.

The rewarding mechanism

A further central element in Steemit, and in other BOSNs, is the rewarding system, i.e. the set of rules and mechanisms regulating how Steem Power and other tokens are distributed among the users who actively participate in the platform through the production/interaction with content. The wealth distribution is based on the roles introduced so far: *creators* and *curators*. Creators publish content, either as posts or comments on posts, while content promotion — *curation* — is made through different social actions, such as *upvote*, *downvote*, or *sharing*. Upvotes are key in promoting high-quality content since more upvotes give more visibility on the main pages of the platforms. Also, if the post enters into the most popular chart, the curator will gain a reward, as "users get paid for figuring out who should get paid" [67]. Reward assignment is not an instant operation, in fact, rewards for content are computed after 7 days. The basic rationale behind the reward assignment procedure is that "most popular posts get more from the reward pool". The total payout pool for a single post — content payout pool — depends on the Steem Power of the curators and how much Steem Power was used for the vote. The content payout is taken from the overall reward pool — a reward to be assigned to Steemit users — which is derived from the collection of tokens produced by the Steem blockchain. Then, as summarized in Fig. 2.3, each content payout pool is split into two parts: 50% goes to the creator and 50% to curators. Each user can decide to cash out the prize in two ways: turn the full amount in Steem Power or claim it as a 50/50 split in STEEM and Steem Power.

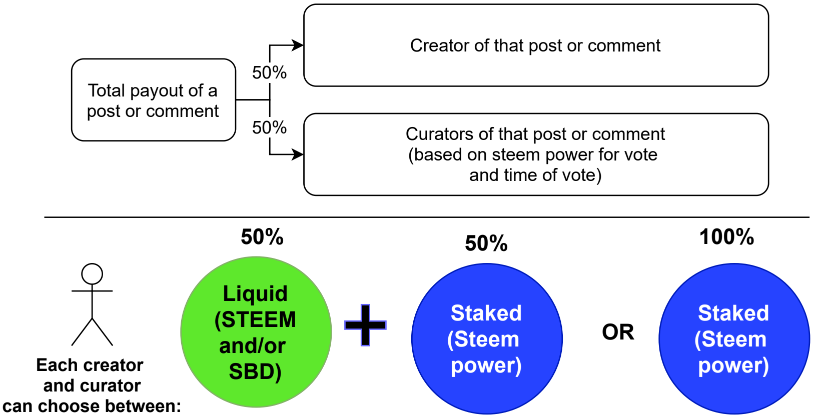

Total payout of a post or comment
50%
50%
Creator of that post or comment
Curators of that post or comment (based on steem power for vote and time of vote)
Each creator and curator can choose between:
50%
Liquid (STEEM and/or SBD)
+
50%
Staked (Steem power)
OR
100%
Staked (Steem power)

Fig. 2.3: **Allocation of the content payout pool.** *Summary of the allocation of the content payout pool between creator and curators. On top, the distribution of the rewards between the creator and curators. On the bottom, rewards can be received in two ways: full amount as Steem Power or as a 50/50 split in STEEM/SBD and Steem Power.*

Note that Steem Power has a key role in the reward assignment process since it is a stake-based voting system, where vote operations are backed by the user's Steem Power. In fact, when a curator casts a vote, s/he also has to decide how much weight to put behind a vote. Finally, as votes are not all equal the curators are not rewarded evenly: the more weight behind the vote, the bigger the reward to the voter is.

The mechanism designed for wealth distribution may have an important impact on how creators "strategically" decide which accounts to follow. In fact, different strategies based on the creation of new "follow" relationships may be adopted to get the attention of wealthy users or to collect votes from a large volume of not very powerful users. All these network-based strategies have the general goal to gain more STEEM, that can be exchanged for traditional currencies. From this perspective, the exchange value of the STEEM cryptocurrency w.r.t. USD is equally important, since a higher price favors behaviors that collect more Steem Power and exchange it into traditional currencies.

Hive

The Hive blockchain and its BOSN platform **Hive blog** are the result of a hard fork, that happened on the 20th of March 2020, originating the new blockchain Hive from Steem, after a 51% attack. Everything began in February 2020 when TRON, a company that owns a gambling-oriented blockchain, led by Justin Sun, acquired Steem [68]. Since the beginning, Steemit's founder allocated a reservoir of tokens that were supposed to be used solely for the development of the Steem ecosystem and to be non-voting in governance issues [69]: however, after the acquisition, there were no guarantees, therefore some of the most active users tried to freeze the tokens acquired by TRON through a soft fork [70]. Nevertheless, TRON was able to temporarily amass a significant amount of voting power on the platform with the aid of some cryptocurrency exchangers, reaching the point where it was able to elect its selected witnesses because it owned more than 51% of them. With its witnesses in place, TRON managed to revert the effects of the soft fork [71]. In response to the hostile takeover, the old witnesses of Steem announced a hard fork [72], which happened on the 20th of March 2020, originating Hive [73]. Because Hive shares the same blocks prior to the hard fork, Hive witnesses froze or confiscated all funds owned by the perpetrators of the hostile takeover to prevent issues on the new platform. Hive, among other innovations, introduced a delayed voting influence mechanism to address potential future 51% attacks, giving the community time to respond in advance. The derived **Hive blog** web platform has similar characteristics. users post and share multimedia content, and users can interact with it. rewarded with cryptocurrency tokens, the *HIVE*. A stablecoin token is also issued, called the HDB Hive-Based Dollar.

2.2.2 Web3 for finance: Complementary currencies.

At the start of the decade, the United Nations changed the global development goals to emphasize the necessity of sustainable growth and social good [74]. The 17 Sustainable Development Goals by the United Nations (UN Agenda 2030 for Sustainable Development [75]) have incentivized the good use of ICT and emerging technologies in many fields and scenarios. At the same time, we have seen the emergence of novel paradigms that are trying to reduce the over-centralization around a few big platforms and tech companies, a trend that has been very noticeable in different fields, like in finance [50] and in online social media [1, 10]. In this scenario, one of the paradigms gaining momentum is Web3, i.e. the design of platforms and software systems built on blockchain technology has emerged as an effective solution for decentralized financial and industrial services [76]. The overlap of the need for more ICT for Good and the emergence of blockchain-based solutions has led to the concept of "Blockchain for Good" [25]. With this term, we refer to the many projects that have been developed over the years, focused on the application of blockchain technology's main features, including cryptocurrencies and smart contracts, to help humanity and the environment [77]. For example, there are blockchain-based solutions utilized to combat corruption and gender inequality[78], to the creation of transparent and sustainable supply chains [79], promoting financial inclusion [25] and social collaboration [80]. Moreover, several publications have examined the possibilities and limitations of blockchain for sustainable development, such as [77] and [81]. Even the United Nations organization has promoted different blockchain-based programs [82] to help refugees, fund non-governmental organizations, and promote the collaboration and coordination of humanitarian aid and social development initiatives. Furthermore, blockchain technology has been utilized to promote social development and local economies [20]. Complementary currencies (CCs) are currencies that originate in various geographic situations to supplement the official national currency [26]. CCs can also be viewed as a fungible “voucher” or credit obligation redeemable for products and services, [20]. There are many instances of CC systems all around the world, with an estimated 3,500 to 4,500 CC initiatives in more than 50 nations since the 1980s [83, 27].In fact, they can be often referred to by many different names such as local currencies, alternative currencies, parallel currencies, community currencies [27], or community inclusion currencies [84] in the literature. While several studies have been conducted on the economic and social principles as well as the analysis of some case studies, there are presently few studies that focus on the impact of CCs during the COVID-19 epidemic. Gonzalez *et al.* [85] investigated the success of a Brazilian CC named Mumbuca E-Dinheiro during the epidemic. Stepnicka *et al.* [86] investigated the Zielony CC in Poland, claiming that the CC was not as successful during the epidemic as it was during times of true financial crises. While Ussher *et al.* [20] investigated Sarafu [30], a Kenyan CC that transformed into an improvised COVID-19 response system: during the crisis, Sarafu has proven to be quite beneficial in assisting the local population.

Sarafu, complementary currency on a blockchain.

Sarafu [30] ("currency" in Kiswahili) is a digital community currency token created by the Grassroots Economics (GE) Foundation[87], a humanitarian aid foundation. Complementary or community currencies (CCs) are currency systems, often born out of cooperation among members that face a period of crisis and introduced in a certain community, with the objective of creating bonds of reciprocity and integrating social networks among people [28]. In both cases, there is a strong interplay between social and economic aspects. In the case of Sarafu, users may perform payments using mobile phones to transfer Sarafu digital tokens to other registered users [31]. As described in Ussher *et al.* [20] the Kenyan Red Cross relied on Sarafu tokens to provide humanitarian aid during the COVID-19 pandemic: users registering were given free Sarafu tokens, backed by donors' money, to maintain the system running.

The use of blockchain technology is a key component of Sarafu. While the Sarafu project has not used blockchain technology from its inception, it has used it to solve several important issues [20]. Among the motivations, we have enhanced transparency, as transaction data allows contributors to fully disclose the impact of their donations. Furthermore, data analysis can lead to more informed decision-making processes regarding, for example, future investments and project functioning, while it also helps the GE Foundation to find ways to improve user welfare and minimize potential misuse. The system first moved to a blockchain maintained privately called *POA*. The name is derived from its consensus protocol, Proof of Authority [88]. The project then switched to a public blockchain named xDai blockchain in 2020 to lower transaction costs[20]. Finally, in May 2022, the project transitioned to a new blockchain built by the GE Foundation to better meet its objectives. Kitabu ("Ledger" or "Book" in Kiswahili) is the name of the new blockchain, which is based on the Proof of Authority consensus protocol.

2.2.3 Web3 for ownership: Non-fungible tokens.

Non-fungible tokens are tokens that each represent a unique tangible or intangible item and therefore are not interchangeable [61]. So, an NFT provides a certificate of ownership of a digital object [19]. An NFT is linked to a given digital asset to attest to its uniqueness and non-transferability: in practice, an NFT can represent a variety of digital items, including photographs, movies, and audio. As a consequence, several fields, such as art, gaming, and sports collectables, utilize NFTs to regulate and control digital objects.

Ethereum NFT ecosystem

Once deployed on the Ethereum blockchain, a smart contract enables the creation (minting) of NFTs. NFTs created (minted) by the same smart contract constitute a collection. Each NFT within a collection is uniquely identified by a token ID. The combination of the smart contract and token ID forms a tuple, serving as a distinctive identifier for each NFT [89]. However, users do not need to interact directly with the blockchain, as several web platforms, known as NFT marketplaces, act as intermediaries between users and blockchains, facilitating the exploration of existing NFTs, their sales, and ownership transfers. Notable among these marketplaces is OpenSea, a decentralized platform widely used for discovering, buying, and selling NFTs. Decentraland is another prominent platform, particularly known for its integration of NFTs within a virtual world. OpenSea provides a user-friendly interface and supports a diverse range of NFTs, including art, collectibles, and virtual real estate. Decentraland, unique in its approach, offers a virtual world where users can buy, sell, and build on virtual land parcels represented as NFTs, combining the concept of virtual reality with blockchain technology. In addition to these platforms, numerous famous NFT collections have emerged. Cryptopunks, an early and iconic collection of algorithmically generated pixel art characters, holds historical significance in the NFT space. CryptoKitties, another notable collection, allows users to buy, sell, and breed virtual cats, each represented as an NFT. These collections have not only contributed to the popularity of NFTs but have also become cultural phenomena within the blockchain and digital art communities.

Smart contract creation and execution

In Ethereum, the most popular NFTs follow the standard ERC-721 [62]. The standard outlines a specific set of functions and events that must be implemented so that a smart contract can generate NFT assets considered valid by the Ethereum Blockchain.

In Ethereum, smart contracts function as computer programs executed on the Ethereum Virtual Machine (EVM), an emulated computer. Each node in the Ethereum network runs a local copy of the EVM to validate contract execution, with the blockchain recording the evolving state of this de-facto decentralized computer as it processes transactions and executes smart contracts. The code is written in programming languages specific for smart contracts, such as Solidity and Vyper. There are two types of accounts for interaction with the Ethereum blockchain: Externally Owned Accounts (EOAs) and contract accounts. EOAs are accounts that can be managed by individuals, that can have access to the funds of the account through their private key. Contract accounts instead are not controlled by individuals, but by the logic of its smart contract code: this type of account does not have a private key. Creating a smart contract corresponds to creating a contract account. The deployment of a contract to Ethereum Blockchain is performed through a transaction, thus requiring the payment of an ETH gas fee similar to a simple ETH transfer. Then, to register a contract on the Ethereum blockchain, a distinctive transaction is executed, targeting the address 0x0000000000000000000000000000000000000000, known as the zero address [61]. The smart contract deployed on Ethereum is composed of a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain [47]. Post-deployment, transactions can be employed to interact with the smart contract. When a transaction's destination is a contract address, it prompts the contract to run in the EVM, utilizing the transaction and its data as input. Transactions may include data specifying the function to run and parameters to pass, allowing them to call functions within contracts. Notably, only EOAs can initiate transactions, so the execution is always started by real users; however smart contracts are then free to respond to transactions by invoking other contracts and creating intricate execution paths.

2.3 Network-based models for Web3

In the context of Web3 systems, network-based modeling has emerged as an effective methodology to analyze the growth and dynamics of trade relationships in these complex economic systems [51, 90]. Indeed, the exchange of tokens in Web3 systems is often modeled through networks where users/wallets are treated as nodes and links represent money transfers between them. While this is an effective approach, the more innovative applications that go beyond the simple cryptocurrency transactions, like blockchain online social media or complementary currencies, require the use and combination of more complex network-based models. More complex models also allow better modeling for open problems considering multiple systems. In this section, we provide background knowledge regarding the formal definition of networks/graphs, while we explain when each graph modeling approach is more suitable and how to combine them when needed. Indeed, across this thesis, we select and combine the available graph models, based on the task. We will address in the respective chapters the motivation behind each selection, providing a detailed description of the models and the data processing steps.

2.3.1 Static graph models

We can use networks to model complex and large connected systems. Modeling data with networks requires choosing the following:

- what is a node, what is an edge
- handling additional information (time, attributes of users or transactions)

Based on these decisions, we can have many types of graphs but the starting point is the static network.

Definition 1 (Static network/ graph). *A graph is $\mathcal{G} = (V; E)$ where V is the set of nodes and E is the set of edges, where an edge is $e = (u, v) \in E$ with $u, v \in V$.*

Definition 2 (Directed (Undirected)). *A directed graph is a graph in which an edge $e_{ij} \in E$ is an ordered pair, i.e. the edge e_{ij} is oriented. Otherwise, the graph is undirected. [91]*

Definition 3 (Weighted (Unweighted)). *A weighted graph is a graph in which a weight function $w_t : E_t \to R$ is assigned to it. Therefore, each edge has a weight associated with it, and it is possible to define a weight matrix W such that $Wij = w(e_{ij})$. Otherwise, the graph is unweighted.*

A graph can have auxiliary Information associated with node/edge/whole graph. A common definition is *attributed* graphs:

Definition 4 (Attributed graph). *A graph with additional information for a node/edge/whole graph. We can find types, labels, attributes, text, images, location, and other features.*
Node-labeled: X is a $|V| \times f$ matrix of node attributes, with f the dimension of attribute vectors.
Edge-labeled: X is a $|E| \times f$ matrix of edge attributes, with f the dimension of attribute vectors

Indeed, certain attributes can describe a node or edge "type". We can distinguish a *homogenous* and *heterogenous* graphs, defined as follows:

Definition 5 (Homogeneous graph). *A Homogeneous graph is a graph in which nodes and edges belong to a single type.*

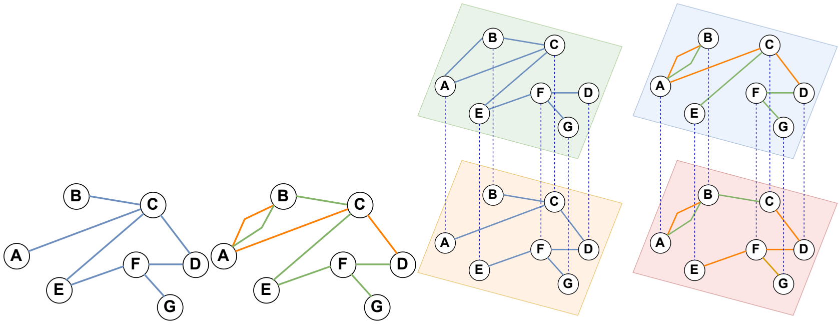
Definition 6 (Heterogeneous graph). *A heterogeneous graph is a graph in which either nodes and/or edges belong to more types, described by a set of T_v node types and T_e edge types.*

Finally, to provide a more realistic representation of the different and heterogeneous relationships we can rely on the multilayer graph model [92] We can define a multilayer graph as follows:

Definition 7 (Multilayer graph). *Given a set $\mathcal{V}$ of entities, and a set of layers $\mathcal{L} = L_1, ...L_l$ with $\mid \mathcal{L} \mid = L >= 2$, a multilayer graph is $\mathcal{G}_{\mathcal{L}} = (\mathcal{V}_{\mathcal{L}}, \mathcal{E}_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$, where $\mathcal{V}_{\mathcal{L}} \subseteq \mathcal{V} \times \mathcal{L}$ is the set of entity-layer pairings or nodes (i.e., to denote which entities are present in which layers), and $\mathcal{E}_{\mathcal{L}} = \mathcal{V}_{\mathcal{L}} \times \mathcal{V}_{\mathcal{L}}$ is the set of directed edges between nodes within and across layers.*

Note that multilayer networks where all cross-layer links (connecting nodes in different layers) are pillar links (links connecting nodes with the same index but in different layers) are usually referred to as **multiplex networks**[93].

In figure fig. 2.4 we present a visualization of all the models. When the object of the study is the transaction networks, the interest is in users/wallets/smart contracts and how they interact. A common approach for the analysis is to rely on a homogeneous graph. In case of important additional information, an attributed or weighted version of the graph can be used to encode it. If the multiplicity of interaction types matters, then a multiplex/heterogenous or multilayer graph is needed. In this case, as well, they can be temporal and/or attributed and/or weighted according to the tasks. Finally, for more complex scenarios like, in the case of migration in BOSNs, where it's easier to perform the analysis if we can model both different types of interactions and the presence of multiple blockchains, we should use the heterogeneous multilayer graph. Note that in many use cases, time is an important aspect, so many works rely on temporal versions of these graphs. In the next section, we'll provide some background on how to leverage temporal information.



(a) *(b)* *(c)* *(d)*

Fig. 2.4: *From (a) to (d) visualization of different types of static undirected graph models. In order: (a) homogeneous, (b) heterogeneous, (c) multilayer, and (d) heterogeneous multilayer.*

2.3.2 Temporal network modeling

There are many complex systems where time is critical. In metabolic process modeling [94] we have brief chemical reactions; in brain networks, the time of transmission of neural impulses is critical [95]. Time information can help us model mobility through networks when planning urban mobility [96]. When fighting disease spreading [97], modeling the time of contacts and their duration is critical. Studying time on social networks can help us understand information diffusion and news spreading dynamics. Finally, in financial studies, the ability to study trading times and behavior can be the difference between gaining a return from investments or losing money [98].

Temporal information allows the modeling of graphs changing over time, where nodes and edges can appear or disappear over time, nodes' properties can change and most importantly the connectivity varies over time. Leveraging temporal information, we can construct **temporal networks or dynamic networks** [99, 9]. A dynamic graph will describe interactions between nodes over the lifetime of a system. E.g. (i, j, t) node i had contact with node j at time t. Edges with temporal information can be called **dynamic links**, **events**, **contacts**. We can define a dynamic graph formally following Barros et al. [91]:

Definition 8 (Temporal network / dynamic graph). *A temporal or dynamic graph is a mathematical structure $\mathcal{G} = (\mathcal{V}; \mathcal{E}; \mathcal{T})$, where $\mathcal{V} = V(t)_{t \in T}$ is a collection of node sets over time span $\mathcal{T}$ and $\mathcal{E} = E(t)_{t \in T}$ a collection of edges sets over time span $\mathcal{T}$. Hence, for each $t \in T$, it is possible to define a graph snapshot $G(t) = (V(t); E(t))$, i.e. a static graph representing a fixed timestamp t of the dynamic graph[91].*

According to recent taxonomies[91, 100] we can subdivide the main approaches into two groups, based on their vision of the time:

- *Continuous-time approach*, where $\mathcal{T}$ is a continuous set, more suitable for events at real-time temporal granularity.
- *Discrete-time approaches*, where $\mathcal{T}$ is a discrete set, meaning that the evolution of a dynamic graph can be represented as a sequence of static graphs, called **snapshots**, each of them with a fixed timestamp.

However, the most common approaches share the discrete vision of time: they differ in the way they derive the snapshots. We can define two types of snapshots: *Interval* and *Evolving*.

Definition 9 (Interval snapshot). *Given a time interval $[t_0, t]$, the snapshot graph $\mathcal{G}_{[t_0,t]}$ represents the directed graph where for each link $e = (u, v, t) \in E$, we have that $t \in [t_0, t]$*

Definition 10 (Evolving snapshot). *Given a time t, the snapshot graph $\mathcal{G}_t$ represents the directed graph where for each link $e = (u, v, t) \in E$, we have that $t \in [t_0, t]$, where t_0 is the smallest t in time span $\mathcal{T}$*

Both models of snapshots are commonly used in literature. The definition from [100] is an example of a dynamic graph considering what we defined as evolving snapshots. While the dynamic graph model in [91] relies on what we defined as interval snapshots.

Therefore, for clarity, in this thesis, we define two types of dynamic graphs, based on the snapshot model underneath.

Definition 11 (Evolving graph). *An Evolving graph is a sequence of evolving snapshots derived from a dynamic graph. Formally, we define a Evolving DTDG as a set* $\mathcal{G}^1; \mathcal{G}^2; ...\mathcal{G}^T$ *where* $\mathcal{G}^t = \mathcal{V}^t; \mathcal{E}^t$ *is the graph at snapshot* t, $\mathcal{V}^t$ *is the set of nodes in* $\mathcal{G}^t$, *and* $\mathcal{E}^t$ *is the set of edges in* $\mathcal{G}^t$.

Definition 12 (Interval graph). *An Interval graph is a sequence of interval snapshots derived from a dynamic graph, i.e.* $\mathcal{G} = G(t_0), ..., G(t_{NS})$, *where* $G(t_k) = ((V(t_k); E(t_k))$ *is a static graph with timestamp* $t_k (k \in 0, ..., N_S)$, N_S *is the total number of snapshots,* $V(t_k)$ *is the node set at timestamp* t_k *and* $E(t_k)$ *is the edge set including all edges within the time interval* $[t_k; t_{k+1})$.

When all datasets have discrete time, even at the higher resolutions, we usually refer to them as Evolving graphs or Interval graphs. Figure 2.5 provides a toy example visualizing the differences in the approaches.

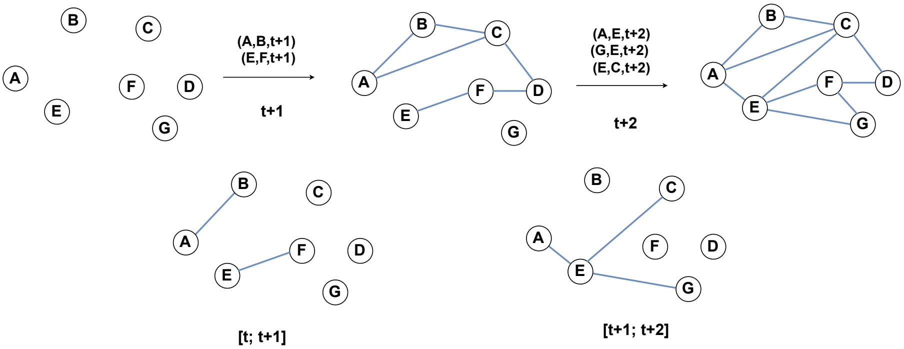


Fig. 2.5: *Example of Evolving vs Interval vision. On the top side, we showcase an evolving graph, on the lower side the interval graph. Alongside the arrows, we display the interaction actions, occurring during a time window, which generate the links in the corresponding graph.*

2.4 Datasets

In this section, we present the datasets collected or retrieved for our study. We focused on various blockchain-based systems with interesting characteristics. We chose Steemit for its prominent position among blockchain social networks.

A pioneer in the field, the platform's success spans both social and economic dimensions, boasting a sizable and active user base. Alongside Steemit, we focus on Hive a platform of great interest due to its unique origin a big fork in social media platforms. Moreover, the subsequent user migration makes for an intriguing case study. The data collected from these platforms constitutes the Steem-Hive dataset . Alongside this dataset, we focus on Sarafu, a humanitarian aid cryptocurrency, that was selected for its distinctive application of Web3 principles in the field of humanitarian aid. and deviation from traditional cryptocurrency systems. The detailed attributes and activity during the crucial COVID-19 pandemic period offer opportunities to explore various open problems. Throughout the work, we refer to the data as Sarafu dataset . Finally, as a different application of Web3 principles, we shifted our attention to NFTS. Our examination of NFTs focuses on Ethereum, home to the most popular collections. As a pioneering platform for this concept, Ethereum boasts a mature and thriving ecosystem of collections and supporting platforms. The dataset retrieved for the analysis is referred to as NFT dataset .

2.4.1 The Steem and Hive transactions data

Conducting our analysis and experiments, required the collection of a dataset. Some subsets are available in the literature, but to address the open problems a complete vision of user interaction was required. As seen in 2.2.1, user actions in blockchain online social networks are stored in the supporting blockchain. In the case of Steemit and Hive, they are stored as transactions recorded in the supporting blockchains Steem and Hive. Specifically, we recovered data describing the actions made by users. The details about blocks and operations for both platforms can be gathered through official public APIs, whose structure and usage are similar for the two platforms. We recall that data between the two blockchains are identical up to the fork event, i.e. to block 41818752, with timestamp 2020-03-20T14:00:00. From there, Hive and Steem have different data, as they have become two different blockchains. So, we collected operations from the very first block on the Steem blockchain, produced on 24 March 2016, up to January 2021. For Hive, we start from the first block after the fork (20/03/2020), and up to January 2021. Users on Steemit can perform many different actions, called *operations*. According to the official documentation [101], there are more than 50 different operations on the blockchain: an overview of these operations and a taxonomy can be consulted in [102], and a complete list can be consulted in the official documentations [101, 103]. The collection of these operations composes a detailed temporal dataset, that describes user activity with a time granularity of 3 seconds[8]. All the usernames have been pseudo-anonymized as soon as they have been collected and stored in their pseudo-anonymized version.

Table 2.1: *List of social and financial operations used in the study. Each operation is characterized by its name, its type and a full description. We do not differentiate between Hive and Steem because the name and the meaning of the operations in the table are common.*

Operation	Group	Description
`comment`	social	A user publishes content or comment on an post.
`vote`	social	An account upvotes or downwotes a content. Users can vote on both posts and comments.
`custom_json`	social	A general-purpose operation designed to add new functionalities without the need for new operations. Social functionalities include: i) **"follow"** to receive updates on what other users are posting; ii) "unfollow" to stop following other users; iii) "mute" to block users from the feed in the case of harassing or unwanted content; and iv) **"resteem/reblog"** to share content with all the followers.
`transfer`	financial	Transfer an asset from one account to another.
`transfer_to_vesting`	financial	Convert an asset to a vesting share and give it to another account.
`delegate_vesting_shares`	financial	Borrow vesting shares to another account, so that it gain the rights to vote contents.
`set_withdraw_vesting_route`	financial	Withdraw vesting shares and transfer the amount to another account.
`transfer_to_savings`	financial	Place assets into time locked savings balances.
`transfer_from_savings`	financial	Transfers assets from the time locked savings balances.

Focusing on interactions. In many of our works, we are interested in *interaction actions* those actions that represent an interaction between two users, either explicit or implicit. A complete description of the operations generating interaction actions is reported in Table 2.1. As shown in the Table, we also

[8] The timestamp of an action is derived from its block, and each block is verified every 3 seconds.

distinguish between two main groups of interactions: *i)* financial and *ii)* social operations. Financial operations are those operations designated for rewards token management, and asset and share transfer; whereas social operations are those that users usually do on traditional social media platforms, like posting, rating, voting, sharing, and following. Overall, from Steem, we extracted 993641075 operations related to social interaction actions and 72370926 operations related to financial actions; from Hive we have a total of 206224132 social actions and 4041060 financial actions.

2.4.2 Sarafu transactions data

The Sarafu dataset includes detailed and anonymized information on token transactions, along with a rich set of user features. The data spans the period from January 2020 to June 2021, totaling $930,161$ economic transactions involving around 55,000 users. Each economic transaction specifies its source and its target as anonymized IDs of the sender and receiver of the cryptocurrency token. Alongside that, we have important additional information for this study: one being the timestamp, i.e. the date and time of when a transaction happened, with a granularity of *ms*. In the following, we fully describe transactions and users' data.

User information. Every user is mainly described by the following attributes:

- **held roles**: the role of the user. *Beneficiary*, which stands for a standard user, is the most prevalent. Another important role is the *group accounts*,i.e. accounts representing cooperation groups. Moreover, there are accounts used by management (*Token Agent*, *Vendor*, *Admin*) described in detail in [31]);
- **business type**: standardized category of economic activity generated from the occupation information provided by users. Examples of possible values include *labor, food, farming, shop, fuel/energy* and so on (see Table 2.2;
- **area type**: the area type determined from user-provided information about the residence place. The provided options are *rural*, *urban*, *periurban* or *other*;
- **area names**: the region or province the user lives in. The possible values span different spots across the whole nation of Kenya. More precisely, we have four urban areas (*Mukuru Nairobi*, *Kisauni Mombasa*, *Misc Nairobi*, *Misc Mombasa*), four rural (*Kinango Kwale*, *Nyanza*, *Turkana*, *Misc Rural Counties*), one classified as periurban (*Kilifi*). Users without a specific location are labeled as *other*.

Table 2.2: *Description of user's business types, derived from the additional information provided with the dataset in [30].*

Business type	**Description**
Labour	Non-farm workers of any kind. Carpenters, bakers, electricians, tailors, chefs, housekeeping, shepherds, beauticians, barbers, artists, engineers, managers, programmers, mechanics, security guards, insurance agents, waiters/waitresses, artisans, employees, bricklayers, masons
Food	Sellers of any kind of local food
Farming	Users registered as farmers or working on farms
Shop	Kiosks, boutiques, phones, cafes, pubs, clubs, clothing, furniture, jewelry, detergent, electric tools, perfumery, flower
Fuel/Energy	Sellers of firewood, kerosene, petrol, biogas, charcoal, paraffin, and diesel
Transport	Drivers, bicycle rental, bike, motorbike, and car services
Water	People in charge of managing the water tanks and other water re-sellers
Education	Teachers in schools, coaches, booksellers, tutors, facilitators, Red Cross volunteers, consulting, babysitters
Health	Traditional and official doctors, nurses, pharmacies, laboratories, first aid operators, and veterinarians
Environment	Waste collection, gardening, seeding, tree planting, cleaning, recycling
Savings	a member of a Chama, or a Chama not yet officially recognized by GE staff
Government	Community authorities (e.g. elders), governmental employees, governmental and military officials, soldiers
Faith	Religious chiefs or religious groups
Other	Unknown
System	Accounts run by GE Staff members

Transaction information. Each economic transaction specifies its **source** and its **target** as anonymized IDs of the sender and receiver of the cryptocurrency token. Alongside that, we have important additional information for this study: one being the **timestamp**, i.e. the date and time of when a transaction happened, with a granularity of milliseconds *ms*. Another useful feature in the dataset is the **weight** of each transaction, corresponding to the amount of tokens moved from source to target. Finally, we find different types of transactions, described by the **transfer subtype** attribute, whose main values are:

- **standard**: the regular token transfer, the most frequent transaction;
- **disbursement**: the creation of tokens and transfer to an account;
- **reclamation**: the removal of Sarafu from an account;
- **agent out**: exchange of tokens with Kenyan Shillings, (only available for group accounts, that can send tokens to a system account to receive money).

2.4.3 Ethereum NFT sales data

The dataset of NFTs sales was collected and analyzed in [19]. The dataset aggregates NFT trades from different marketplaces (APIs): Cryptokitties, Atomic, Opensea, Gods-unchained, and Decentraland. The data collection is composed of 6.1 million trades of 4.7 million NFTs in 160 cryptocurrencies, primarily Ethereum and WAX, covering the period from June 23, 2017, to April 27, 2021. **Transaction information.** The dataset provides a rich set of features for each transaction:

- Unique_id_collection: unique ID for a given NFT
- Crypto: cryptocurrency used to acquire the NFT
- Price_Crypto: amount the NFT was sold for
- Price_USD: price in US Dollars, conversion is done with a daily resolution
- Seller_address: addresses for sellers
- Seller_username: seller username used on the NFT marketplace (when available)
- Buyer_address: addresses for buyers
- Buyer_username: buyers username used on the NFT marketplace (when available)
- Image_url_1, Image_url_2, Image_url_3, Image_url_4: urls to the digital object associated with the NFT.
- Datetime_updated: the time of the transaction with a day resolution
- Datetime_updated_seconds: the time of the transaction with a seconds resolution
- Smart_contract: smart contract of the given NFT
- ID_token: ID of the NFT asset within a given smart contract
- Transaction_hash: hash of the transaction involving an NFT sale
- Collection: the collection in which the NFT belongs to
- Collection_cleaned: the field Collection after some preprocessing e.g. misspellings removal.

- Market: data source (the API).
- Name: title of the NFT listing
- Description: description of the NFT listings
- Permanent_link: a link that allows to verify the NFT authenticity (valid only for the OpenSea Market)
- Category: category to which the NFT belongs. Examples are: Art, Games, and Collectible

Part II

Network evolution dynamics

Chapter 3

Bursty dynamics in decentralized social networks

3.1 Introduction

During the past 15 years, digital systems have increasingly become the main media supporting social interactions and relationships. Platforms such as online social networks, emails, and mobile phones are now producing an invaluable volume of digital footprint data which is unveiling different aspects of human behavior. Among these aspects, the availability of large-scale data tracking the interactions of many individuals has pointed out that human dynamics are heterogeneous and characterized by a bursty behavior.

This behavior has been measured on and crosses different human activities, such as email communications, mobile phone calls, library loans, or even letter correspondence. However, few attempts have been focused on the dynamics of large-scale online social networks; especially on microscopical and high-resolution temporal dynamics of link creation or economic interactions. In studying these properties the main obstacle is a lack of publicly available data capturing the detailed evolution and growth of online social networks. A few studies have analyzed these microscopical evolution data, but datasets have been kept private or not easily shareable due to constraints enforced by the data owner.

Our primary objective is to provide an analysis and characterization of the microscopical dynamics of the link creation and reward collection processes, based on publicly available data provided by blockchain-based online social networks (BOSNs). BOSNs include an ecosystem of social platforms where interactions and social activities are supported by a blockchain, often linked to cryptocurrency. By cryptocurrency, these platforms are able to implement a reward system that aims to promote high-quality content or trusted users. For our study BOSNs represent an interesting socio-technological system since

1. every social interaction, such as "follows", likes, and comments, is recorded in an accessible blockchain with a high-resolution timestamp (seconds); and
2. besides social interactions, BOSN platforms track actions related to economic aspects and how users engage with the rewarding system. The latter point represents a novel aspect in human dynamics studies since the temporal characterization of how people interact with and within techno-economical systems is partially unexplored.

Our goals can be summarised by the following research questions:*RQ1)* Are blockchain-based social networks characterized by bursty behavior? *RQ2)* Is there bursty behaviour across both social and financial dimensions? To answer these questions, among the social networks in the BOSN ecosystem, we focus on Steemit, one of the most successful platforms. Steemit — a blogging platform with more than 1 million users — offers incentives for users to participate, either as *creators* — content producers — or as *curator*s — content promoters. At the end of a 7-day period the most popular posts are awarded through cryptocurrency tokens, and both creators and curators of the most liked posts have a share of this reward. Steemit relies on the Steem blockchain for data validation, data storage, and action tracking. As the first contribution, we model the blockchain data as a temporal network to represent the evolution of relationships in the online social network. We also take into account the reward claims requested by users, so to understand how people interact with the reward system. Our analysis of the temporal characteristics of these two aspects has highlighted that

1. in general, the bursty trait typical of human dynamics also holds in the creation of "follow" relationships as well as in claiming cryptocurrency rewards.
2. Although the two processes follow the same general behavior, they present a few peculiarities mainly due to automated processes that facilitate reward claims.
3. Differences between the dynamics of how people create new online relationships and how they claim their rewards also emerge from a temporal user-centric analysis: more users establish new relationships in a highly bursty manner than in reward claim cases.
4. The bursty behavior in the creation of new relationships spans different time scales, while the bursty dynamics in claiming rewards have a narrower scale.

The study of a property like burstiness is critical for the BOSN scenario, as this is a "cross-field" property observed in many complex systems, thus highlighting similarities and differences with respect to other complex systems; it has also practical consequences on architectural and performance aspects of BOSN platforms since high-throughput I/O is a bottleneck of many blockchain-based solutions. If, on the one hand, the above findings confirm previous observations about bursty dynamics of the link creation process in online social networks [9], on the other, in this Chapter, we show that other behaviors are bursty in this kind of social network (*RQ1*). However, we stress that each process has its own characteristics (*RQ2*). These differences open up studies on the modeling of the processes leading to different kinds of bursty behaviors and on the role played by services and mechanisms in shaping the dynamics of the interactions in online social networks.

3.2 Related work

Blockchain-based online social network

BOSNs, such as Steemit, are complex systems where social and economic aspects are intertwined. As every action is stored on a blockchain, these platforms provide a detailed data source for studying the dynamics guiding the system. Such characteristics have made BOSNs, especially Steemit, the subject of recent studies. For example, Li *et al.* [104] released a dataset paper, stressing the potentiality of this network, meanwhile highlighting difficulties in the extraction and processing of the high volume of produced data. Other works focus on the characteristics of this type of social network [10, 105, 106]. User content has been also useful for text mining tasks [107] and bot detection ([108]). Other works are more focused on the economic aspects: Ciriello *et al.* [23] and Thelwall *et al.* [24] analyze the relationship between rewards and content, while Li *et al.* [22] describe and analyze the networked structures behind the Steemit rewarding system. There is also a growing interest in the social network structure. Chonan [109] and Kim *et al.* [65] focus on the structure of its social network and its characteristics. Also, Guidi *et al.* [110] delve into a study of the follower–following graph, and study other operations in Steemit [102]. Aside from the relationships among users, [111] studies block producers (witnesses) and highlight their social impact on the platform. Nevertheless, there is still a limited number of works focused on network dynamics and temporal aspects. For instance, Jia *et al.* [112] focuses on the diffusion of content, while in previous work we discussed the interplay between cryptocurrency and graph evolution [113].

Microscopical dynamics in networks

There are many complex systems where time is critical; such fields include mobility networks, disease spreading, information diffusion, financial studies, and biological systems [114]. Only recently, research efforts have begun to focus on the even more complex problem of studying networks that include temporal information, called temporal networks or dynamic networks [99]. Currently, network-based approaches designed to deal with temporal information are few — each one targeted to specific research areas and with different objectives. However, studies have highlighted the presence of cross-field properties, found in different contexts: one such property is *burstiness*[8]. Bursty behavior or burstiness describes a system that alternates periods of frequent activity or events and long intervals with a low frequency of activities or events. Bursty behavior has been found in nature, for example in physical phenomena (earthquakes, solar flares), in biological systems (neuronal firing, biological evolution), and in ecology (animal movements, ecosystem evolution). Moreover, man-made systems such as router traffic and financial markets, show bursty patterns. Finally, bursty behavior emerges in human dynamics, with earlier studies detecting burstiness in emails, phone calls, and messages, and more recently in online social networks [9].

3.3 Research questions

While many studies are observing bursty behavior across multiple fields and domains there is a lack of studies focused on this important property in blockchain-based systems. In this work, we address this gap, focusing on the following research questions:
Research question RQ1: Are blockchain-based social networks characterized by bursty behavior?
Research question RQ2: Is there bursty behaviour across both social and financial dimensions?

3.4 Data Preprocessing

BOSNs rely on their supporting blockchain to store, validate, and manage the interactions occurring on them. From this viewpoint, they represent a fundamental data source to study the dynamics of different kinds of interactions.

In this chapter, we consider the Steem-Hive dataset presented in Section section 2.4. We consider a subset covering a three-year period of users' activities: precisely, from December 6, 2016, up to March 20, 2020. The starting date has been chosen according to the Steemit white paper, i.e. when regular production of STEEM cryptocurrency started. The end date corresponds to the Hive Hardfork [115], an event that led to a migration of some users to a similar social media platform based on the Hive blockchain. Such an important event is bound to influence social activities and network structures, thus we limit our analysis to this date.

The goal of this study is to understand which dynamics characterize different actions on blockchain-based online social networks. Here, we focus on two different aspects: *a*) the "follow" relationships, capturing the social dimension; and *b*) the reward claiming, more related to economic and content production aspects. As for reward dynamics, we study the action of reward claims, i.e. a user asks to add the cryptocurrency to her/his balance, obtained from content creation and curation[1]. As rewards can be traded for actual currency and spent on goods, the reward system plays a critical role in users' activity on BOSNs. In Steemit, reward claims are tracked as `claim_reward_balance` operations. As for social dynamics, we study "follow" actions, i.e. a user starts following another user to receive blog posts directly in her/his feed. Follow operations provide insight into the evolutionary dynamics of the social network. In the dataset, "follow" actions, alongside other operations, are tracked as `custom_json` operations.

Overall, we considered 31,609,874 reward claims and 134,721,867 "follow" operations, each associated with a timestamp.

Steemit social network

While reward claims are modeled by a time series associated with each Steemit user, "follow" actions are represented by a temporal directed network $G = (V, E)$ where the set V includes Steemit accounts and $(u, v, t) \in E$ denotes a directed "follow" link from u to v created at time t, so that each link has a timestamp. The final snapshot of the network is made up of $1,347,905$ nodes and $134,721,867$ links. An in-depth analysis of the Steemit social network is out of the scope of this Chapter, however, we report the distribution of the degrees to highlight the nature of Steemit as a social media. In Fig. 3.1 we plot the distributions of the degree (a), the in-degree (b), and the out-degree (c) for the final snapshot of the network. All the degree distributions exhibit

[1] In Steemit up/down-votes, comments and sharing are curation actions.

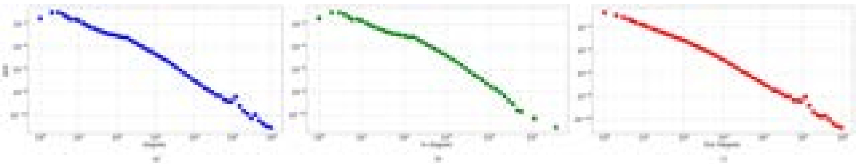


Fig. 3.1: Probability density function (PDF) of the degree distributions of the final snapshot of the Steemit temporal network built from "follow" actions. In a) the degree distribution, in b) the in-degree distribution, and in c) the out-degree distribution. PDFs are built from a logarithmic binning of the sample. On the y-axis: the PDF of the degree. On the x-axis: the node degree.

the typical shape of scale-free networks, that characterizes most traditional social networks. So, the Steemit "follow" network is characterized by the presence of hub accounts — super-users — and a large fraction of accounts with medium/low connectivity.

3.5 Methods

Here, we take a node-centric approach and deal with two sequences of actions made by each node: the creation of new "follow" relationships and the claim of rewards gained by content creation and curation actions.

User level dynamics

We model the link creation and the reward claim processes as point processes on time so that each user u is characterized by two-time series with irregular timings, i.e. two sequences of events $ev_f(u)$ and $ev_r(u)$ such that $ev_{(.)}(u) = \{t_0, t_1, ..., t_{n-1}\}$ is an ordered list of n timings, where t_i is the timing of the $i-th$ event generated by user u. The construction of the time series describing the behavior of users when it comes to reward claims — ev_r — is straightforward. By inspecting the `claim_reward_balance` operations of user u, we extract their timings and sort them. Instead, the analysis of "follow" operations requires a processing step on the directed temporal network G. For each user $u \in G$, we extract all the out-going temporal links $e_{uv} = (u, v, t)$ and we get the timings ts associated with the links. Thus we obtain a set of timing events for each user u. This way we model how users follow other accounts.

Inter-event time distribution

In a time series, an inter-event time is the elapsed time between two consecutive events. Formally, given a time series of events ev, we can define the inter-event times sequence as $iet = \tau_1, ..., \tau_{n-1}$, where each inter-event time $\tau_i = t_i - t_{i-1}$ is the time interval between two consecutive events, with $i = 1, ..., n-1$. The inter-event distribution $P(\tau)$ accounts for the type of dynamics of the process, since different dynamics are described by different, specific inter-event times distributions. For instance, a Poisson process where events occur independently and at a constant average rate λ (homogeneous Poisson process) is characterized by an inter-event time distribution that follows the exponential function [8]. However, the analysis of many empirical datasets [8, 116], has revealed that many systems cannot be characterized by a Poisson process, rather their inter-event times are characterized by a heavy-tail distribution $P(\tau)$ with a wide temporal span. Specifically, systems with a bursty behavior have been successfully described by means of inter-event times modeled by a power law distribution, most often by a Pareto distribution with exponential cutoff $P(\tau) = \tau^{-\alpha} e^{-\tau/\langle\tau\rangle}$. Thus, the analysis of $P(\tau)$ for both follow and reward claims time series can provide insights into the dynamics of the processes, specifically whether it is homogeneous (Poisson-like) or not.

Burstiness

The heterogeneity of inter-event times can be quantified by a single quantity named *burstiness*, introduced by Goh and Barabasi [117], and extended by Kim *et al.* [118]. Given a time series of n events, we compute the standard deviation σ of the inter-event times, and the mean inter-event time $\langle\tau\rangle$. The burstiness B_n is a function of the coefficient of variation $r = \sigma/\langle\tau\rangle$; more precisely, the burstiness B_n of a time series with irregular timings is defined as:

$$B_n = (r\sqrt{n+1} - \sqrt{n-1})/(r(\sqrt{n+1} - 2) + \sqrt{n-1})$$

We get positive burstiness values when the time series has more heterogeneous inter-event times than a Poisson process, with $B_n \to 1$ for extremely bursty cases (with $\sigma \to \infty$). On the contrary, the measured burstiness $B_n \to -1$ for regular time series (where $\sigma = 0$), while $B_n \to 0$ for random Poissonian time series, where $\sigma = \langle\tau\rangle$. Since our goal is a characterization of per-user dynamics, we compute the burstiness B_n of $ev_f(u)$ and $ev_r(u)$ for each user u, and then compare the burstiness distributions for both “follow” and reward claim actions.

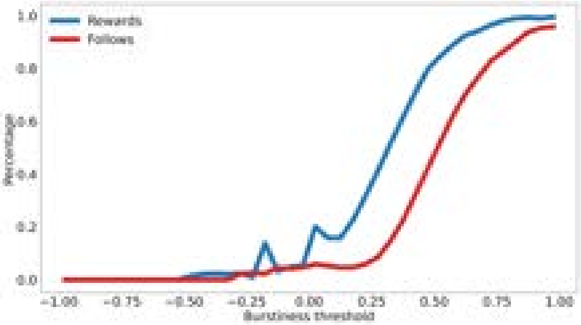


Fig. 3.2: *Percentage of users following a power law in the distribution of inter-event times for "follow" (red line) and reward claims (blue line), given a fixed level of burstiness. We used 0.05-length bins and computed the fraction of users following a power law model.*

Burstiness similarity

The burstiness is a general indicator of heterogeneity in the time series of the users, however, it does not suggest which model has generated the heterogeneity, since it depends on point estimates of the inter-event time distribution. For this reason, we also adopted a model-based approach to investigate how users are similar in terms of model describing their event timings. Specifically, we wonder whether the inter-event time series of the users follow power law distributions (model) with close parameters — accounting for similar time scales — or very different parameters. This latter case would suggest a different behavior in users' dynamics, despite the shared bursty behavior. According to the methodology used by Gaito *et al.* [9] for the analysis of bursty behavior in social networks, the approach provides two steps. First, we have a fitting phase: we fit each user's inter-event time series using the model $P(\tau) \approx \tau^{-\alpha}$, so we estimate the best α parameter. Then, according to [116], we run a goodness-of-fit test to quantify the plausibility that the inter-event times are drawn from the fitted power law model. As the second step, for the time series which have passed the test, we analyze the distribution of α values, to evaluate the users' similarity in terms of the parameter of the fitted model. If the distribution of α values were concentrated, with a very limited dispersion, we could say that users have a homogeneous behavior, that can be described with a small set of α values. On the contrary, we might find a distribution with a higher dispersion: in that setting, the behavior might not be described with a small set of alpha values, hence we say that the bursty behavior is more heterogeneous among the users.

Since the burstiness and the above model-based approach are related, for a given level of burstiness we measure the percentage of users whose inter-event time distribution follows a power law. This way, we quantify the predominance of the power law model given a certain level of heterogeneity of the temporal behaviors. In Fig. 3.2, we display the above percentage for "follow" actions and reward claims. The trend for both the curves is quite expected. For values of burstiness less or equal to 0, only a very low percentage of users follow a power law distribution, while we observe a pronounced increase in the ratio for burstiness values greater than 0.25. In particular, we highlight that most of the users with heterogeneous event timings have a temporal behavior that can be modeled by a power law. This observation is even more evident for "follow" actions.

Temporal correlations

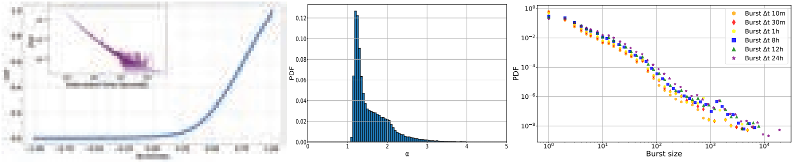
Burstiness measures the heterogeneity of inter-event times, however, it does not highlight potential correlations between consecutive events. In particular, it is assumed that two consecutive events e_{t_i}, $e_{t_{i+1}}$ are related if they follow each other within a short time interval Δt s.t. $t_{i+1} - t_i <= \Delta t$. Similarly, we can consider bursty periods or *bursty trains*, i.e. sets of events where the maximum inter-event time between consecutive events is less than a threshold Δt. The assumption is that consecutive events in the same bursty period are causally correlated. Note that different values of Δt can generate different bursty trains. Therefore, we consider temporal correlation at different time scales and different values of Δt can capture different temporal aspects and correlations among events. Karsai *et al.* [8] have studied bursty periods by analyzing the size of bursty trains and have shown that temporal correlation is captured by the bursty train size distributions $P_{\Delta t}$. More precisely, $P_{\Delta t}$ generally has an exponential distribution when generated from sequences of independent events. Thus, any deviation from an exponential distribution of $P_{\Delta t}$ indicates a correlation between inter-event times. Different works [8] have found power law distributed train sizes.

We construct $P_{\Delta t}$ with a user-centric approach in this Chapter. Specifically, given a user's sequence of inter-event times and a Δt, we detect the set of the bursty periods where consecutive inter-event times are less than Δt. Then for each bursty period bt, we consider its burst size $|bt|$, i.e. the number of events in that period, and compute the median burst size, which characterizes each user. Finally, from the set of median values from all the users, we can plot the distribution $P_{\Delta t}$.

Burstiness, burstiness similarity, and temporal correlations have been computed for both "follow" and reward actions, so that we highlight potential similarities or differences among the two types of operations from both aggregated and user-centric perspectives.

3.6 Results

This section presents results and insights obtained by applying the methodology introduced in Section 3.5 onto the Steemit data. In our analysis, we discard users who performed too few actions. For "follow" actions we exclude nodes with final degree[2] lower than the 70% percentile (40) of the out-degree distribution. Similarly, we filter out users who claimed rewards only a few times, lower than the 70% percentile length of the event sequences — 17 reward claims.

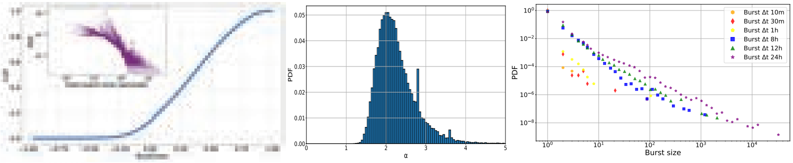


a) b) c)

Fig. 3.3: *Dynamics of the "follow" relationships: a) burstiness CDF, in the inset figure the inter-event time distribution, reported as a probability density function (PDF) on a log scale on both axes; b) PDF of the scale parameter α (bins of length 0.05) for time series that have passed the goodness-of-fit test; and c) PDF of the median bursty train sizes.*

As for the inter-event time distributions, in the inset figure of Figures 3.3a and 3.4a, we report the probability distribution function (PDF) for "follow" and reward claim time series. As for "follow" actions, the aggregated dynamics are in line with that observed in other online social networks [9], suggesting a general heterogeneous temporal activity in the creation of new "follow" relationships. On the contrary, the inter-event time distribution for reward claims shows a different trait (see inset Fig. 3.4a). In fact, we observe several spikes that may be due to automated reward-claiming services. In fact, Steemit users

[2] The final degree of a node corresponds to the node degree computed on the last snapshot of the "follow" temporal directed network.



a) b) c)

Fig. 3.4: *Dynamics of the reward claims: a) burstiness CDF, in the inset figure the inter-event time distribution (PDF), on a log scale on both axes; and b) distribution (PDF) of the scale parameter α (bins of length 0.05) for time series which have passed the goodness-of-fit test; and c) PDF of the median bursty train sizes.*

do not receive rewards until they claim them manually. However, to speed up the claiming procedure, several automated services are available[3]: these services perform periodical checks and claim automatically available rewards. As the operations happen at fixed rates, it could explain the spikes in the distribution.

Burstiness

The inter-event time distributions have highlighted an overall heterogeneity of the temporal behaviors in Steemit, however, they do not capture the specific patterns of the users, since they aggregate the inter-event times of all users. Otherwise, burstiness enables a user-centric analysis. Here, we report our analysis on burstiness showing the cumulative distribution function (CDF) for both a) "follow" and b) reward claims time series in the main plots of Figures 3.3a and 3.4a. As for "follow" actions we observe an overall high level of burstiness. In fact, only 5% of users have negative burstiness values, while the remaining users have positive values. Specifically, we find very positive values, with only 20% of users with $B_n < 0.5$ and around 50% with a burstiness level greater than 0.75. This confirms the previous observation from a user-centric perspective: *the "follow" actions are indeed bursty.*

Similar but less distinct behavior can be observed for reward claims in Fig. 3.4a. The distribution shows that a high percentage of users have high levels of burstiness. In fact, the values are mostly positive, suggesting bursty

[3] Automated services and social bot behaviors are also evident in the "follow" time series. In fact, we removed bursts of "follow" events occurring on the same block (same timestamp) whose length is greater than 10. This kind of behavior is very likely generated by bots.

behavior for this action as well: we find around 70% of users have burstiness measures over 0.5, and 90% of users have burstiness measures over 0.75. They are in a lower range compared to "follow" actions. To sum up, *reward claims are characterized by less bursty dynamics* and about 25% of users generate more homogeneous time series.

In general, the analysis of burstiness points up *most of the users are characterized by bursty behaviors when they "follow" other profiles or claim rewards.* The phenomenon is more evident in the former case than in the latter, suggesting that *different actions determine different users' behaviors.*

Burstiness similarity

Here we focus on classifying users according to a power law model. As detailed in Section 3.5 for each user's inter-event time series we fit data to a heavy tail distribution such as power law, log-normal, and exponential. For each time series, we can obtain an estimated value of α from the data; we also obtain a significance value p, s.t. very low values of p are associated with reliable estimations, while bigger values suggest that the fitting should be ignored. Thus, we consider alpha values with significance $p < 0.1$ and discard α values outside the range $[0, 4]$ [119]. The distribution of α for both "follow" and reward claim actions are displayed in Figures 3.3b and 3.4b. The distribution for follows, in Figure 3.3b, shows a long tail. The overall distribution is asymmetrical, with values scattered around a peak at 1, with a tail on the right side. These two characteristics are typical of systems with heterogeneous bursty behaviors in the network, as behavior can not be described with a small set of alpha values. As for reward actions, we find a distribution over a wider range. While the distribution is more symmetrical, the wider range suggests different types of bursty behaviors. Finally, a direct comparison of the two distributions of α, suggests that in terms of "follow" actions and reward claims users behave differently. The "follow" dynamics suggest a trait quite common to all users, while users claim their reward in different ways. We might hypothesize that different processes drive the dynamics of the two actions.

Temporal correlations

We deal with the study of temporal correlations through the analysis of the bursty train size distribution $P_{\Delta t}(E)$. Since different values of Δt capture different temporal correlations, we varied Δt over the set $\{10', 30', 1h, 8h, 24h\}$, for both "follow" and reward claim actions. Then, we compare the different distributions, so that we analyze potential differences between diverse time scales. The distributions $P_{\Delta t}(E)$ are presented in Figures 3.3c and 3.4c. As we can see in Fig. 3.3c, for “follow” actions, we find heavy-tail distributions for each Δt. This implies a likely temporal correlation between consecutive elements. However, we do not find significant differences in the distributions for different Δt values. Similarly, for reward claim actions, shown in Figure 3.4c, we obtain heavy tail distributions for each Δt. However, we observe a stronger impact of the choice of Δt: different values of Δt do change the scale of the distributions. In fact, lower values of Δt lower the size of the bursty train. This leads to a shift in distribution values for different time scales. For the lowest values of Δt, $P_{\Delta t}(E)$ does not have a heavy tail trait, suggesting a lack of temporal correlation when we use short intervals.

To sum up, for both actions there are temporal correlations across different temporal levels. However, we find a very different response to the variation of Δt, suggesting different dynamics behind “follow” and reward claim operations.

3.7 Conclusion

This Chapter studied the microscopical dynamics for both social and economic aspects of blockchain-based online social networks. The novelty introduced by BOSNs is twofold:

1. the software architecture of the platforms, the supporting blockchain, the social actions, and the economical aspects are strictly intertwined as well as their dynamics; and
2. the supporting blockchain represents an invaluable data source to capture and analyze the above aspects, providing high-resolution temporal information that eases the analysis of microscopical dynamics.

Our findings, based on the Steemit case study, on the dynamics behind social and rewarding operations, show that dynamics are characterized by bursty behavior(*RQ1*), but with a few differences(*RQ2*). First, burstiness spans different time scales in the creation of new “follow” relationships, while reward claim dynamics have a shorter time scale. Moreover, the variability of the model describing how users claim rewards is larger than in the “follow” case. From an architectural viewpoint, bursty dynamics on different properties may represent a bottleneck in the performance of the writing operation on the blockchain due to overhead introduced by consensus algorithms, while differences in the bursty dynamics w.r.t. other mainstream social platforms may indicate not properly human behaviors are acting on BOSNs, an aspect which should be further investigated.

Chapter 4

Evolution dynamics through triadic closure-related network motifs

4.1 Introduction

The exchange of tokens — both fungible and non-fungible — has a key role in blockchain-based systems. In fact, the widespread circulation of tokens leads to the formation of trade relationships among users, which can be seen as a complex network structure: in these networks, nodes are users/wallets and links represent the beginning of an exchange relationship. The key aspect is that in many Web3 systems relying on these tokens, exchanges are often strongly intertwined with the more social side of the platforms, making blockchain-based platforms very complex and interesting socio-economic systems. However, there are only a few studies on them from a socio-economic network perspective, and their structure and growth dynamics have been partially studied only. In particular, there is no study focused on triadic closure, one of the main mechanisms driving social network evolution [11]. Such an evolution mechanism is present in online and offline social networks, and indeed it could be a driving factor in Web3 systems as well, where the social structure is strictly tied to the economic structure.

In this chapter, we analyze triadic closure in decentralized socio-economic networks supported by blockchain technology. The analysis requires us to analyze the network structure by observing *triads*, i.e. 3-node subgraphs. We aim to answer these research questions: *RQ1)* From a static network perspective, are decentralized socio-economic networks similar in terms of triadic-based structures, or whether each network characterized by specific triadic-based patterns depending on its nature? *RQ2)* From a temporal standpoint, do specific evolution patterns of the triads characterize different socio-economic networks or do they follow a common growth mechanism? *RQ3)* From a dynamic viewpoint, how does the triadic closure process change over time? Do the different types of triads resulting from a triadic closure process form at the same speed? Is the dynamic of triad formation stable along the evolution of these networks? To answer these questions, we extend the current methodology for triadic closure studies and adapt it to the analysis of decentralized networks. Moreover, we conduct an in-depth analysis of network structure centered on both *triads* and *triadic motifs*, i.e. statistically significant triads, both from a static and temporal standpoint. We conduct our analysis on different decentralized socio-economic networks characterized by different levels of social components: the leading blockchain online social media Steemit [18], NFT trades on Ethereum [19], and a blockchain-based currency for humanitarian aid — Sarafu [20].

Our insights on triadic closure and triads from a static viewpoint have highlighted evident differences among decentralized socio-economic networks mainly due to their main scopes and functionalities. Differences are so remarkable that the distribution of the closed triads may represent a footprint of the network since each socio-economic network has its specific distribution. In defining the footprint an important role is played by the “feed-forward” loop and by fully or almost fully reciprocated triangles. In fact, socio-economic networks where the social and economic traits are more intertwined are characterized by more reciprocal relationships and triads, while feed-forward loops are dominant where the interplay is weaker. The centrality of “feed-forward” loops and reciprocity has been further confirmed by the analysis of the patterns forming closed triads. In fact, all the closing temporal triads forming a feed-forward loop are the most frequent in all the networks. Despite the importance of patterns related to the “feed-forward” loop, the distribution of the closing temporal triads is a further footprint of a network: NFT network is mainly built around patterns leading to “feed-forward” loops, while distributions of the closing temporal triads in Steemit and Sarafu are more uniformly spread over all the possible patterns, with temporal triads leading to the creation of fully reciprocal triangles frequent and significant. To sum up, both in a static (*RQ1*) and dynamic setting (*RQ2*), each network has its own specific profile which depends on the nature of the socio-economic actions it supports. Finally, we found that triadic closure has impacted the evolution and the growth of these platforms even more than in traditional and centralized online social platforms. The closure process is not stable, rather each network is characterized by its own dynamics, sometimes influenced by external conditions. However, there is a characteristic common to all these networks: the closure process is very fast, faster than in the centralized counterparts. So, even though in decentralized socio-economic networks social and economic relationships and interests mix up, the triadic closure, one of the main mechanisms behind the formation of social ties, emerges as an important factor contributing to the growth in trade relationships; even much faster than in centralized online social networks (*RQ3*).

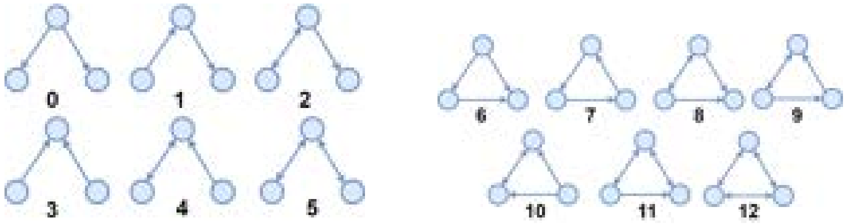
The Chapter is organized as follows. Section 4.2 provides an overview of the main related works in the field and some background. In Section 4.3 we introduce the main research questions, we have on the evolution of these networks. The approach for modeling and analyzing the socio-economic networks is presented in Section 4.4. In Section 4.5 we describe the selected datasets and their preprocessing and details on the experimental setting. Section 4.6 reports the main findings regarding the impact of triadic closure. Finally, Section 4.7 concludes the Chapter, pointing out possible future works.

4.2 Related work

Network evolution trough triadic closure

Many models, mechanisms, and measures describing network growth from a link formation perspective have been proposed. Among them, triadic closure has emerged as one of the most important mechanisms [13]. The main assumption of triadic closure is that individuals with a common friend have a higher chance to become friends themselves at some point in the future [11]. Although the triadic closure has been recognized as one of the fundamental mechanisms driving the formation of dense groups and communities [12] in social networks, their properties, and laws are still scarcely studied at a large scale, due to the limited availability of temporal-annotated datasets capturing the growth of large social networks.

From a static standpoint, triadic closure influences graph structure on the level of *triad*s, i.e. 3-node directed subgraph. Specifically, in a directed network, we have 13 possible triads (if isomorphous subgraphs are counted only once) that can be divided into the 2 categories of closed and open triads: there are 6 possible open triads (see Fig. 4.1a) and 7 closed triads (see Fig. 4.1b). Indeed, the structure of a network can be characterized by the distribution of these triads: for example, Milo *et al.* [15] rely on triads and other subgraphs to characterize networks in different domains, showing that similar networks have similar characteristic subgraphs. For example, focusing on triads in the



(a) *Open triads* ***(b)*** *Closed triads*

Fig. 4.1: *The 13 possible triads in a directed network (if isomorphous subgraphs are counted only once). They can be divided into 2 categories: open triads (a) and closed triads (b).*

field of online social networks, Huang *et al.* [14] confirmed some similarities among centralized online social networks such as Twitter and Weibo.

While frequency is an important indicator of the importance of a triad, it could be frequent simply because of the size of the network. Therefore many studies focus on the analysis of *motif*s, i.e. classes of isomorphic induced subgraphs whose frequency is higher in the data than in a reference null model [15]. There are many ways to test whether a subgraph is a motif [120], however the most common in literature are the significance tests based on the *z-score* and the *p-value* [15]. The idea is that the count of each subgraph in the original network should be compared with the same counting in a randomized version of the original network (the reference model or null model): a subgraph could be *i)* over-represented, i.e. its frequency is higher in the original dataset than in the reference model, *ii)* under-represented , i.e. its frequency is significantly lower in the original network than in the null model, or *iii)* similarly represented, which corresponds to a non-significant subgraph. In the literature, the most common approach is to consider a subgraph g as significant when $|z(g)| > 2.0$, i.e. the absolute value of its *z-score* is greater than 2 [120]. So, the combination of triad frequencies and motifs could be used to characterize decentralized socio-economic networks highlighting common traits and differences in their network structure and in their evolution.

In fact, while static structure already provides some insights into the effects of the triadic closure process, leveraging temporal information is essential to obtain a more complete analysis and characterization of the network evolution. Zignani *et al.* [21] proposed some temporal metrics to quantify triadic closure in undirected networks. The first one is the *triangles/link ratio*, i.e. the fraction of triangles produced over the links. Monitoring the ratio at regular intervals, like daily observations, provides an overview of how much the links tend to form closed triads. A further important measure is the *triadic closure delay*, a measure quantifying the "eagerness" of users in building social structures. The value of delay provides insight into the speed at which users act in building and extending their social neighborhoods by closing triangles. Both measures are able to capture and quantify the presence and dynamics of the triadic closure mechanism in a network, and they can also be used to compare different networks.

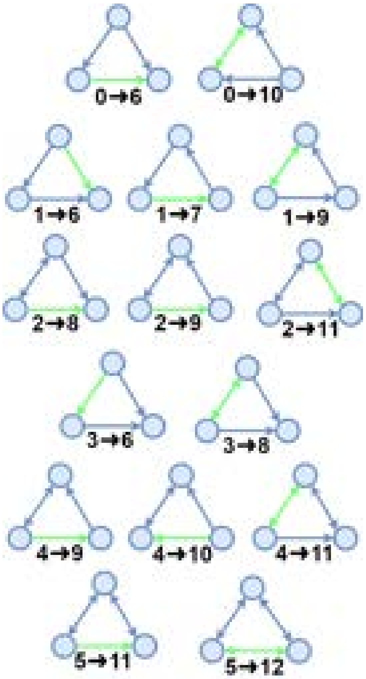


Fig. 4.2: *Closing temporal triads capturing the triadic closure. Blue links are established before time t, green links are established at timestamps $t' > t$ to form the closed triad.*

Furthermore, temporal information favors the study of the evolution of network structure from a temporal standpoint. In this case, networks are modeled as temporal networks, a representation that combines both topology and time. From the triadic closure viewpoint, we can therefore focus on *temporal triads*, i.e. 3-node temporal subgraphs. A subset of *temporal triads* that represent triadic closure are displayed in Fig. 4.2. The identification of such temporal subgraphs is less straightforward than in the static case since the introduction of the temporal dimension has led to different definitions of temporal subgraphs and motifs. One of the most important works on the subject is by Kovanen *et al.* [16]. In their work, they consider a subset of temporal subgraphs in which *i)* the time difference of consecutive events is less than an input interval Δc, and *ii)* the events in the subgraphs are all consecutive. A further definition is in Paranjape *et al.* [17], where they use as a starting point the previous definition[16] but they remove the constraint on consecutive events, as it allows to study more subgraphs that tend to occur in short bursts. They also use a time window Δw to bound the time difference between the last and the first events in a subgraph. There are a few other models in the literature [121], but the key aspect is that the distribution of temporal subgraphs can be used for comparison and characterization of networks [122], similarly to the static scenario. Similarly to the static setting, we can also detect *temporal motif*s [16], i.e. temporal subgraphs that result as statistically significant compared to a null model. However, among the works studying temporal subgraphs and temporal motifs, the term *motif* may be found even for not statically significant subgraphs. Indeed, not all the works actually perform a statistical significance test, both for computational reasons (exact temporal subgraph counting is expensive, and performing it multiple times may not be computationally feasible) or because of the difficulty of selecting a meaningful null model. In fact, as noted in different works [16, 121], the selection of a null model for temporal networks is not trivial. In general, there are many possible reference models for temporal networks, and each model randomizes certain parts of the network, with the goal of preserving some features of the original one. Among the many classes of models presented in the survey by Gauvin *et al.* [123], the most frequently used model in many fields are the “topology-constrained link shuffling” methods, also known as edge randomization or link shuffling. Indeed, it preserves most of the characteristics of the original temporal network: it preserves the original graph structure while eliminating all causal correlations between events taking place on adjacent links.

Network evolution in Web3

Steemit has gathered the interest of researchers for its characteristics and has been dissected in many aspects. However, only some works have studied network structure and evolution. For example, a few studies have focused on the features of different types of social networks resulting from diverse interactions or specific subsets of accounts. Guidi *et al.* have studied "follow" network and other operations in Steemit [102] and have delved into a study of the follower–following graph and the token transfer graph [110]. Other works focused on economic aspects and network structure: for instance, Li *et al.* [22] have analyzed the rewarding system in Steemit from a network perspective, while Ba *et al.* discussed the interplay between cryptocurrency price and the link creation process [18], the impact of user migration on the social networks [36], the role of groups network structure in migration [124], and the bursty dynamics of the link creation process [125]. Also Tang *et al.* [126] model voting and currency transfer data to study user collusion behavior. Moreover, Galdeman *et al.* [127] studied the network growth using transfer operations, subgraphs of up to 4 nodes, in a span of 3 months. They highlighted that in Steemit, network structure is characterized by rules that increase network transitivity and reciprocity. In this Chapter, we rely on the transaction dataset used in [18]. We consider Steemit's *transfer operations*, the most common type of financial action, that allows the exchange of the two main tokens, STEEM and SBD; covering four years of user activity, for a total of 55033746 transactions. For a transfer operation, we consider the users involved, and the action timestamp.

Following their gain of popularity, there has been an increasing amount of studies on NFTs [128]. For instance, Nadini *et al.* [19] have conducted a comprehensive quantitative overview of the NFTs market, including a network-based analysis. Franceschet *et al.* [129] focused on the creators-collectors network, while Galdeman *et al.* [127] highlighted the presence of frequent trading chain patterns.

There are currently few studies on Sarafu from a network standpoint. The GE organization realized a dataset [30] which includes detailed and anonymized information on token transactions. Ussher *et al.* [20] presented an accurate description of complementary currencies, the Sarafu project history, and an analysis of the dataset. Mattsson *et al.* [130] proposed an analysis modeling the entire dataset through a static network structure: their analysis highlights that money circulation is highly modular, geographically localized, and occurring among users with diverse jobs. While Ba *et al.* [131] model the dataset as a sequence of temporal networks to study currency flows and cooperation patterns.

4.3 Research questions

There are few studies that deal with decentralized socio-economic systems from a network and evolutionary dynamics perspective. Currently, there are no works exploring the mechanism of triadic closure and the presence of triadic network motifs in decentralized networks. Specifically, here our hypothesis is that the intertwined nature of social and economic relationships in blockchain-based social networks should lead to an evolution of the economic relationship networks with traits similar to social networks. On the other side, we also investigate the specificity of each economic network asking whether different socio-economic networks are characterized by different network characteristics or patterns, from a microscopically and triadic closure-related perspective. In particular, in this Chapter, we will answer the following research questions:
Research question RQ1: When dealing with the triadic closure process, triads, and their census are the fundamental building blocks for describing the actual state of a network (closed triads) and for identifying where closures may occur (open triads). From this perspective, and in a static setting, we ask whether decentralized socio-economic networks are similar in terms of triadic-based structures, or whether each network is characterized by specific triadic-based patterns depending on its nature.
Research question RQ2: From a temporal standpoint, are different socio-economic networks characterized by specific evolution patterns of the triads or do they follow a common growth mechanism?
Research question RQ3: From a dynamic viewpoint, do the different types of triads resulting from a triadic closure process form at the same speed? Is the dynamic of triad formation stable along the evolution of these networks?

4.4 Methods

Modeling

In general, transactions can be modeled as a set of tuples $I = \{(u, v, a, t)\}$ where u and v are users that "moved" tokens: user u transferred to user v an amount a of tokens at time t. Our focus is on the relationships between users determined by token transfers, by modeling them as a network: transactions over a time interval $[t_0, t_1]$ can be modeled as a temporal network [99]. More precisely, the transaction data over time can be represented as a temporal network $\mathcal{G}_{[t_0,t_1]} = (V, E)$, where:

- V is the set of users[1],
- E is a set of timestamped directed links $(u, v, t) \in E$ where $u, v \in V, t \in [t_0, t_1]$; in other words, links represent a transfer/trade relationship: two users are linked if they performed at least a transfer/trade in the time interval $[t_0, t_1]$, and $t \in [t_0, t_1]$ is the timestamp of the first transaction between u to v.

It is worth noting that the direction of links captures the flow of money from a source to a destination — in the case of transfer — or from a buyer to a seller in the case of NFT trading. As for NFT trade, it is a complementary modeling approach w.r.t. the seminal work on NFT trade networks by Nadini *et al.*, where links are directed from the seller to the buyer. in this Chapter, we do not consider the amounts a of each transfer/trade but the model could be extended to include them as edge attributes. The evaluation of network statistics can give us an insight into the similarity of the datasets.

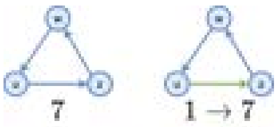


Fig. 4.3: *On the left, a static close directed triad among the vertices u,w and z. Number* 7 *corresponds to the ID assigned to each kind of triad. On the right, is the corresponding closing temporal triad. From the open triad (blue link) by the insertion of the green link* (u, z) *we move to the close triad* 7*.* $1 \to 7$ *indicates that me move from the open triad with ID* 1 *to the closed triad with ID* 7

Frequent triads and triadic motifs

For RQ1 we need to analyze the structure of decentralized socio-economic networks. As detailed in Section 4.2, we can compare the structure of different socio-economic networks from a static standpoint, by studying the frequency of *triad*s, i.e. 3- node directed subgraphs. Therefore, we consider $\mathcal{G}_{[t_0,t_1]}$ as a static network, in this case, discarding the temporal information from the structure. For each triad, we obtain g_i the frequency $N(g_i)$. Then, we can compare the distributions of triads to assess the similarity between the two networks. We separate open and closed triads for an easier comparison so that

[1] In economy, they are referred to as economic agents.

each network is assigned to two distributions: the distribution of open triads and the distribution of closed triads.

Then, we study whether frequent triads are also statistically significant and if there are differences across the selected networks. We consider a *triadic motif* to be a triad that is also statistically significant. As introduced in Section 4.2, to assess the significance of triads, we have to define a proper null model. Here we adopted the null model defined in [132] and since we do not have a closed formula for the null model, we rely on bootstrap by performing N times the randomization of the original network, obtaining for each triad g_i, N outcomes $N_{rand}(g_i)$, corresponding to the counting of g_i in the N realizations of the random model. These counts are confronted with the count of each triad in the original network. We evaluate the statistical significance of the countings both through the *z-score* and the *p-value* [15]. For the former, denoted $\bar{N}_{rand}(g_i)$ as the average count with standard deviation σ_{rand}, we can compute the *z-score* of a triad g_i w.r.t. the null model as:

$$z(g_i) = \frac{N(g_i) - \bar{N}_{rand}(g_i)}{\sqrt{\sigma_{rand}^2}} \tag{4.1}$$

. Finally, a triad can be regarded as statistically significant in a network if its associated *p-value* is less than 0.01 and the absolute value of its *z-score* is greater than 2, $|z(g_i)| > 2$.

Temporal subgraphs and temporal motifs

Answering RQ2 asks for studying how network structure evolves, more precisely how an open triad becomes a closed one, i.e. what is the sequence of link insertion operation transforming an open triad into a closed one? Here, we focus on a special case of *temporal triads* — temporal subgraph of 3 nodes — denoted as *closing temporal triads* $g_{i\rightarrow j}$, i.e. temporal triads that represent the transition from an open triad g_i to a closed one g_j, as shown in Fig. 4.3 on the right. We count the closing temporal triads in the different socio-economic networks, obtaining for each of the possible closing temporal triads the value $N(g_{i\rightarrow j})$. We can compare the distribution of closing temporal triads, for each network. This way, we are able to assess the similarity of the networks in terms of how the triadic closure process has closed open triads.

We also assess how significant each temporal triad is by identifying *closing temporal triadic motifs*, i.e. temporal triads that are statistically significant w.r.t. a null model for temporal networks [16]. We obtain the frequency of each temporal triad $(g_{i\rightarrow j})$, denoted as $N_{rand}(g_{i\rightarrow j})$, one for each of the N

randomized versions of the network. Their average $\bar{N}_{rand}(g_{i\to j})$ and standard deviation σ_{rand} are used for computing the z-score and p-value tests. Similarly to the static case, the z-score of a closing temporal triad $g_{i\to j}$ is:

$$z(g_{i\to j}) = \frac{N_(g_{i\to j}) - \bar{N}_{rand}(g_{i\to j})}{\sqrt{\sigma^2_{rand}}} \tag{4.2}$$

Finally, we evaluate which temporal triads can be considered closing temporal triadic motifs, i.e. as statistically significant in the selected network. Similarly to the static case, we need to evaluate if the associated *p-value* is less than 0.01 and if $|z(g_{i\to j})| > 2$.

Measuring triadic closure

For RQ3, we analyze the triadic closure as a temporal process by leveraging the temporal information of the edges. Specifically, to understand how impactful triadic closure is, we leverage a few temporal metrics for triadic closure [21]. First, we study the impact of closure focusing on the number of triads that become closed (n_closed_triads), compared to the formation of new links (n_links) and monitoring their *ratio* over time as:

$$ratio = \frac{n\_closed\_triads}{n\_links} \tag{4.3}$$

In short, the above measure indicates the overall contribution of new links in the formation of new triangles, and it is strictly related to the densification of the network as time goes on. However, it only returns a general trend in the evolution of the network, since it is counting-based.

A more specific measure based on the temporal information of the links forming a triangle is the triadic closure delay [21], a property characterizing each temporal triangle in a network. Viewing the triadic closure as a dynamic process, it measures the speed of the formation of closed triads. Through triadic closure delay, we can capture the nature of the triadic closure process acting in online social networks: for instance, if only fast closed triads are forming, or if latent triangles are woken up by external mechanisms, such as seasonal events or recommendations systems [21]. The measure has been defined only for undirected graphs. In the undirected setting, we deal with triangles, i.e. an undirected closed triad of vertices u, w, z, where each edge u, w has a timestamp $\tau(u, w)$. So, a triad g will move from an open triad with two links — for example, (u, w) and (w, z) — to a closed triad (triangle) when the last pair ((u, z) in the example) connects. Consequently, undirected close triads are characterized by opening and closing times. The delay accounts for the time the triad g needs to close, namely:

$$delay(g) = \tau(u, z) - max(\tau(u, w), \tau(w, z)) \tag{4.4}$$

where $\tau(u, z)$ is the closing time and $max(\tau(u, w), \tau(w, z))$ is the opening time.

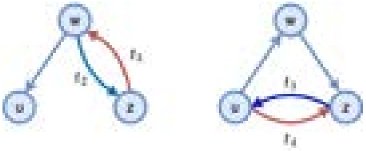


Fig. 4.4: *Example of open and close triads. On the left, an open triad where the blue link forms before the red one, which reciprocates the relationship between u and w: both links may be considered for defining the opening time. On the right, is a closed triad where both links (blue and red) can be considered for the definition of the closing time of the directed triad.*

The above definition does not hold for the directed case, since the time of opening and closing is not as straightforward as in the undirected case: they can be interpreted in different ways because of the presence of bidirectional links. The presence of bidirectionality means that the creation of two links does not imply the presence of an open triad, as we could observe a bidirectional link and an unconnected node. Similarly, the addition of a link, may not lead to a closure: as displayed in Fig. 4.4, in the case of opening time, when a link to an open triad is added, we may not have a closure, because the new link may reciprocate an existing link, hence we have more opening times. Whereas for the case of closed triads (see the example in Fig. 4.4 on the right) that form by bidirectional links, we may be interested in either the earliest (t_3) or the latest (t_4) closing time. This is an important limitation for the analysis of decentralized networks: the importance of tokens in these systems means that we need to distinguish the sender or seller of the token/s from the receivers or buyers. Therefore it's of paramount importance to extend the current approach for directed graphs. In general, to measure the triadic closure delay in directed networks, we have to adapt the formulation to include the direction of links. Here, to measure the delay we consider the earliest opening time and the earliest closing time. Formally, given a closed triad g, with vertices u, w, z, and where each edge u, w has a timestamp $\tau(u, w)$ denoting its creation time and $\tau(u, w) = \infty$ for non existing links, we denote the *earliest closing time* $\tau_c(g)$

as:

$$\tau_c(g) = min(\tau(z,u), \tau(u,z)) \tag{4.5}$$

In this case, we assume that u, z is the last pair to form a link. Given the assumption that $\tau(u,w) = \infty$, by definition, in an open triad, the $min()$ always returns a real number. In the same setting, the *earliest opening time* $\tau_o(g)$ is defined as:

$$\tau_o(g) = max(min(\tau(u,w), \tau(w,u)), min(\tau(w,z), \tau(z,w))) \tag{4.6}$$

where we assume an existing at least a link between u, w and at least one between w, z, formally $min(\tau(u,w), \tau(w,u)) \neq \infty, min(\tau(u,w), \tau(w,u)) \neq \infty$. Then, the directed triadic closure delay can be extracted as:

$$directed\_delay(g) = \tau_c(g) - \tau_o(g) \tag{4.7}$$

Once the triadic closure delay is defined for each closed triad, we can study its distribution and compare it to other online social networks to assess similarities and differences in the dynamic aspects of the closure process. With our proposed approach we can now analyze every directed graph as it's the case with most of the decentralized networks.

4.5 Data Preprocessing

For our study. We focused on three blockchain-based systems presented in section 2.4: the Steemit blockchain online social network, Sarafu, and Ethereum NFTs. From the Steem-Hive dataset , we consider Steemit's *transfer operations*, the most common type of financial action, that allows the exchange of the two main tokens, STEEM and SBD; covering four years of user activity, for a total of 55033746 transactions. For a transfer operation, we consider the users involved, and the action timestamp. From the NFT dataset we consider every transaction, the ID of sellers and buyers, as well as the time of sale/transfer. Similarly, for the Sarafu dataset , we consider each economical transaction available, extracting its source and its target (sender and receiver of the cryptocurrency token). Alongside the timestamp, i.e. the date and time of when a transaction happened.

Preprocessing and Experimental setting

Before delving into the identification of the triads of interest, we proceeded with a data preparation step. For Steemit we limit the analysis to the first

2 years (2016 and 2017), both due to computational constraints as well as to obtain a number of transactions similar to the other datasets. We limit to 8327832 operations. For the NFT trades dataset, we consider all the 6071027 transactions in the original dataset. Finally, for Sarafu we utilize the same preprocessing steps as in [131], overall getting 412050 operations.

For the computation of the frequencies of triads, we implemented a parallelized version of the triad census algorithm presented by Batagelj *et al.* [133]. It is a sub-quadratic algorithm for large and sparse networks able to not enumerate every possible 3-node sub-graph in the network, and whose complexity is $O(m)$, where m is the number of links. As for the evaluation of the significance of the triad frequencies through a null model, we proceeded with a network structure randomization of the static network done using the greedy algorithm of Havel and Hakimi, which was extended to directed graphs by Erdos *et al.* [132]. Instead, when we deal with temporal triads, we implemented a strategy based on the topology-constrained link shuffling method, a randomization method for temporal networks presented by Gauvin *et al.* [123].

4.6 Results

In the following sections, we report and discuss the main outcomes resulting from applying the methodology discussed above to the selected datasets. Transaction networks are modeled as temporal networks, where a trade/transfer relationship is established when the first exchange happens: we have a link between users if they exchange a token or non-fungible token, with the source being who is sending or selling the token/s and the target of the link will be the receiver. The main network characteristics are displayed in Table 4.1.

Table 4.1: *Overview on socio-economic network properties. For each network we report the number of nodes($|V|$), the number of links ($|E|$), density ($x10^5$) (de), diameter (di), average local clustering coefficient (cc), and reciprocity (r),*

	$\|V\|$	$\|E\|$	de	di	cc	r
Sarafu	40343	143239	8.80	22	0.16	0.52
NFT	532944	2991601	1.05	53	0.05	0.02
Steemit	200913	1356011	3.36	14	0.17	0.25

First, we observe that in Steemit we have more repeated transactions between the same users. Indeed Steemit network has a size less than NFT one even though there are more transactions in the former. Further, Steemit and Sarafu differ from NFT trades in terms of density: they are much denser and likely their structure may be characterized by more cohesive structures than in the NFT networks. Sarafu and Steemit also differ from NFT trade networks for other properties: they are characterized by a higher level of reciprocity than NFT trades. These last two features are coherent with the nature of the platforms: Steemit and Sarafu are more social by nature since they revolve around social media or cooperation groups, so more connected structures and reciprocal exchanges are to be expected; while in NFT trade networks there is a distinction between buyers and sellers, and it is unlikely that an account has both roles since there is only a single type of asset to trade. A further consequence of the different nature of NFT trade networks is reflected in the diameter of the networks. They all have larger diameters compared to established OSNs, but the more social Steemit has the lowest value, followed by Sarafu, while the NFT is by far the largest. A similar trait is also observable when considering connected components: both weakly and strongly largest connected components in the NFT trade network span only a subset of the network, while in Sarafu and Steemit the network has a huge largest connected component ($> 95\%$). Finally, the separation between social-like networks, such as Steemit and Sarafu, and NFT trade networks has been also captured by the average clustering coefficient, computed on an undirected version of the graph. Indeed, we observe higher values for Steemit and Sarafu, while in the NFT trade network, it is less likely to observe clustered neighborhoods.

In short, from a network-level standpoint, socio-economic networks such as Sarafu and Steemit express characteristics more resembling online social networks than the NFT trade network; the latter being less clustered, less connected, and probably characterized by more chain-like structures. As for the triadic closure process, the results on the average clustering coefficient offer of first hint at the diversity of how the closure process acts, and its impact on the structure of the network.

Triadic structure to characterize socio-economic networks

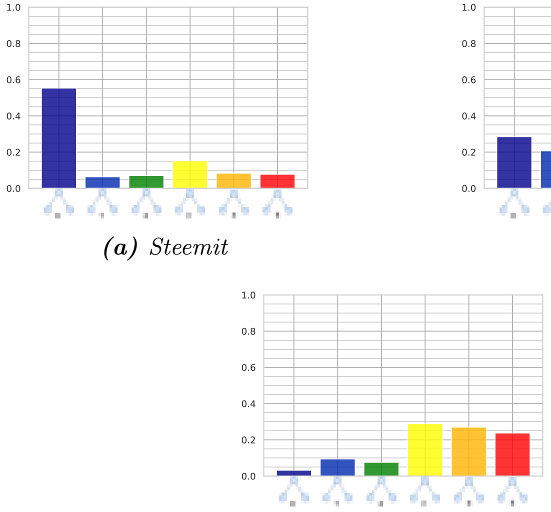
Addressing RQ1, i.e. to what extent decentralized socio-economic networks are similar in terms of static triads — asks for an enumeration of all the possible triads making the structure of the decentralized socio-economic networks; and an evaluation of their statistical significance. Then, we can compare the structure of different socio-economic networks from a static standpoint, by focusing on the most frequent and significative *triads*, i.e. 3-node directed subgraphs, common to all networks, or specific for one network only. Our analysis of open and closed triads and significative triads has highlighted the following main findings:

- Open and close triad distributions are very different among the socio-economic networks. The main scopes and functionalities of the platforms, these networks have been derived from, largely determine the formation of characterizing patterns. For instance, in the case of open triads, the high or low frequency of "collector" or "spreader" patterns (triads 0 and 3) depends on the nature of the socio-economic network, e.g. buying from creators is very common in the NFT trade network. Moreover, open and closed triads are also influenced by the level of reciprocity, i.e. a trait merely linked to more social behaviors of the accounts.
- The distribution of the closed triads represents a footprint of the network since each socio-economic network has its specific distribution. In particular, the main discriminative characteristics are the frequencies of "feed-forward" loops and fully or almost fully reciprocated triangles. Socio-economic networks where the interplay between social and economic traits is stricter are characterized by more reciprocal relationships and triads, while where the interplay is weaker, such as NFT networks, feed-forward loops are dominant.
- All patterns are significative, thus not explainable by a random behavior of the accounts. In particular, the tendency of reciprocating impacts the formation of fully reciprocated open triads, especially in socio-economic networks where the interplay between social and economic actions is stricter. The significance of closed triads is a further discriminative element of the type of network, indeed there is a pronounced difference for under- and over-represented close triads between Steemit and the remaining networks.

From now on, we separately consider open and closed triads *i*) to highlight similarities and differences both in terms of these two types of triads; and *ii*) because of the skewness of the triad distribution (see Table 4.2) towards open triads, which would make the visual exploration of closed triads harder.

Open triads. First, we report the distribution of the frequencies $N(g_i)$ of open triads (triads with index from 0 to 5 in Fig. 4.1a) in Fig. 4.5. As discussed above, the distribution is limited to the possible open triads only. At first glance, we can observe that each network has its own profile, i.e. open triad distributions are different from one another. So, we can comment triad by triad, in order to highlight specific differences but even similarities.

Triad 0 is the most frequent triad in Steemit. This open triad can be seen as an "out-flow" triad, where tokens are only transferred to two other users, or as a "buying" triad, representing a user buying to two different users, in the



(a) Steemit

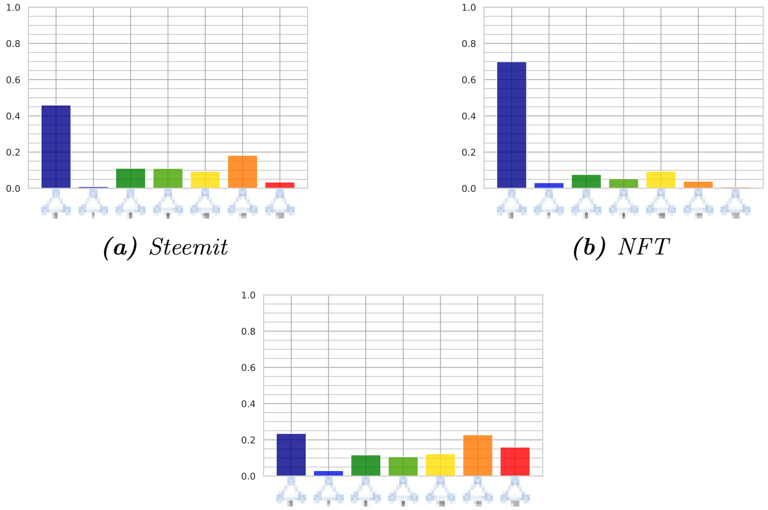
(b) NFT

(c) Sarafu

Fig. 4.5: *The distribution of open triads in the three datasets. On the y-axis: percentage of each open triad. Each open triad, along with its index, is reported below its color bar. The distributions have been computed on the set of all the open triads.*

case of NFT trades. This triad is very frequent in Steemit and the second most frequent in NFT, while it is very rare in Sarafu. In blockchain social media, this pattern is typical of accounts that are sources of resources: in the case of Steemit those might be content creators with money to spend or "whales", i.e. the richest accounts, who used their money as an influence mean; while in the case of NFTs, it should be a triad where the resource spreaders are likely NFT collectors. In contrast, in Sarafu, this kind of triad is much less frequent but it is expected due to the cooperative nature of the platform. Indeed, most of the accounts in Safaru are targets of micro-credit transactions or donations, while there are only a few donors or cooperative groups lending crypto-tokens. This observation also impacts the level of reciprocity of the Sarafu network.

A further evident difference involves triad 1. This triad represents a chain of currency transfers or sales. The frequency of this triad differentiates between Sarafu/Steemit, and the NFT trade network. In fact, in the latter, it is more relevant — 3rd most frequent, than in the former networks. Even for



(a) *Steemit*

(b) *NFT*

(c) *Sarafu*

Fig. 4.6: *The distribution of closed triads in the three datasets. On the y-axis: percentage of each closed triad. Each closed triad, along with its index, is reported below its color bars. The distributions have been computed on the set of all the closed triads.*

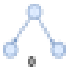
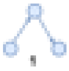
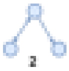
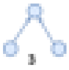
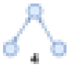
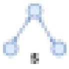
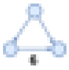
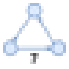
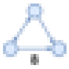
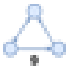
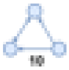
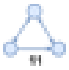
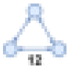
this triad, the difference in frequencies is due to the nature of the NFT network. Triad 1 mirrors a typical chain of sales, especially in the case of "wash trading" a.k.a. the practice of selling among coordinated users to inflate the price of an NFT. Such a trait is less frequent in networks more affine to social networks, where other patterns characterized by a higher degree of reciprocity are to be expected. Even the distributions of Triads 4 and 5 are very specific to each socio-economic network. In fact, these types of triads are very frequent in Sarafu, less in Steemit, and rare in the NFT network. Both triads are characterized by the presence of reciprocal links, which can justify the low frequency in NFTs, where users tend to be either sellers or buyers. In particular, triad 5 captures an interesting situation where there is an open triad composed of two users strongly connected by reciprocal links; and yet, the two unconnected nodes end up not forming any link among them. According to the triadic closure principle, this situation should resolve in a closed triad; or an eventual breaking of the triangle whereas the two unconnected nodes are actually not on friendly terms. In Sarafu, this triad may be also representative of good practice in cooperation and microcredit-based systems: the lender is the central node and the two unconnected nodes have been able to repay the loan to the lender.

The above triads and their frequencies represent distinctive elements among the various networks. However, we also observe open triads which are among the most frequent in all networks. In fact, triad 3 is a very important triad across all networks. In this triad, we have well-defined roles: a node is a target or "collector" of token transfers while the remaining two nodes are not connected to each other but send tokens to the collector. This triad is very frequent in all networks: indeed, for Steemit, it could be a content creator receiving money, a service provider receiving a payment, or even a content promoter or a whale being reached by other users in need of visibility for their posts. In Sarafu this pattern is probably caused by the presence of "group accounts", special accounts handled by more users, that are saving up money. Finally, in the case of NFTs, the target node may represent an NFT creator or an owner of interesting NFTs. A further common trait among all networks revolves around triad 2: it is not very frequent in all of them, with a small increase in Steemit and Sarafu. This triad can be seen as a chain with some reciprocity, and since the difference across the networks is not large, the difference could be simply a byproduct of the higher levels of reciprocity of Steemit and Sarafu compared to the NFTs.

Closed triads. In Fig. 4.6 we report the distribution of closed triads, i.e. triads with index from 6 to 13 in Fig. 4.1b. Similarly to open triads, we can observe that each network is characterized by a different profile, a plausible consequence of the diverse nature of the networks. However, in the case of closed triads, it is more difficult to semantically characterize the overall pattern as it strongly depends on how they are formed — an aspect we shall focus on in the following sections. Nevertheless, we can still highlight similarities and differences triad by triad, as discussed above.

Starting from closed triad 6, we can observe it is the most frequent in all scenarios, even if there are significant differences in its frequency: in the NFT trade network, it is very frequent — about 70%, quite important in Steemit (45%), and less frequent and comparable to other closed triads in Sarafu (about 25%). It is worth noting that Triad 6 corresponds to the well-known "feed-forward loop" pattern, characterizing diverse types of networks, such as biological and regulatory networks [134] or land trade networks [135]. Closed triad 7, the loop, is rare in all networks: it is the least present in Sarafu and Steemit and among the least frequent in the NFT network. In the case of financial networks, the 3-node cycle is strictly related to suspicious

Table 4.2: *Significance of the 13 possible directed triads, in all three socio-economic networks. For each directed triad, we compute the z-score (z) and report the scores with p-value* $<$ 0.01*, while the rest are not significant (NS). In each cell, the first line reports the count of the pattern, the second one its frequency, and the third line reports the z-score, between parenthesis.*

Pattern	Steemit	NFT	Sarafu
	2978073772 57.29% (-85.74)	327981715 28.18% (-64.55)	197348 3.07% (-64.79)
	343201721 6.60% (-135.27)	238232822 20.47% (-78.15)	581008 9.05% (-66.97)
	379895328 7.31% (-100.07)	18235194 1.57% (32.60)	466798 7.27% (-38.26)
	627116917 12.06% (-181.06)	540106984 46.40% (-100.39)	1780132 27.72% (-63.57)
	448251409 8.62% (-11.77)	25306422 2.17% (71.12)	1662086 25.88% (9.05)
	417169005 8.03% (236.97)	1094385 0.09% (52.51)	1460816 22.75% (45.40)
	2017335 0.04% (-129.44)	9094191 0.78% (75.88)	64221 1.00% (-20.73)
	44441 0.00% (-46.06)	405589 0.03% (-35.84)	8218 0.13% (-36.19)
	495953 0.01% (-34.22)	996757 0.09% (73.56)	31942 0.50% (23.31)
	503554 0.01% (-49.22)	688699 0.06% (108.03)	29076 0.45% (-21.17)
	428919 0.01% (-86.03)	1232177 0.11% (159.98)	33677 0.52% (12.39)
	834229 0.02% (-64.32)	507460 0.04% (112.64)	62277 0.97% (31.55)
	157261 0.00% (-14.57)	82705 0.01% (130.42)	43590 0.68% (70.29)

money laundering activities [136]. Further, there is a strong similarity across the networks with regard to triads 8, 9, and 10. These triads tend to be in the middle of the pack in terms of frequency, with very similar rankings across the three networks. While the ranking and the frequency associated with the above triads are traits common to the three networks, the frequencies of triads 11 and 12 are specific to each network. For instance, triad 11 is very frequent in Steemit and Sarafu — the second most frequent — while marginal in NFT. Even in this case, the high frequency in Steemit and Sarafy is a consequence of the high degree of reciprocity. This is also confirmed in the case of triad 12: very rare in the NFT network and quite frequent in Sarafu, otherwise.

Triadic motifs. Finally, we deepen the study of triads by focusing on their significance and identifying *triadic motifs*, i.e. statistically significant triads. Here, we discuss each socio-economic network separately, and then highlight similarities and differences. In Table 4.2, we observe the z-scores (see Equation 4.1) for both open and closed triad motifs. We first observe that all the triads can be considered statistically significant with regard to the selected null model since most of the z-scores are greater than 10 (absolute values). However, there are differences in the z-scores throughout the different networks. For open triads (0 to 5) we can observe that shuffled graphs (random models) end up containing more open triads. Indeed, open triad motifs 0, 1, 2, and 3 are under-represented, except triad 2 in the NFT network. Differences are more evident for open triad motifs 4 and 5. For instance, in Steemit, even open triad motif 4 is under-represented, while in NFT and Sarafu networks we actually have more open triad motif 4 compared to random networks. Finally, there are more open triad motifs 5 in all three networks, where in Steemit the z-score is particularly higher. In short, the tendency of reciprocating relationships in Steemit and Sarafu is far from being the outcome of random behaviors: in socio-economic networks, such as Steemit and Sarafu, where the interplay between social and economic actions is stricter, the reciprocity impacts the formation of fully reciprocated open triads. The tendency of reciprocating links even impacts the significance of reciprocated open triads (2 and 4) in the NFT scenario.

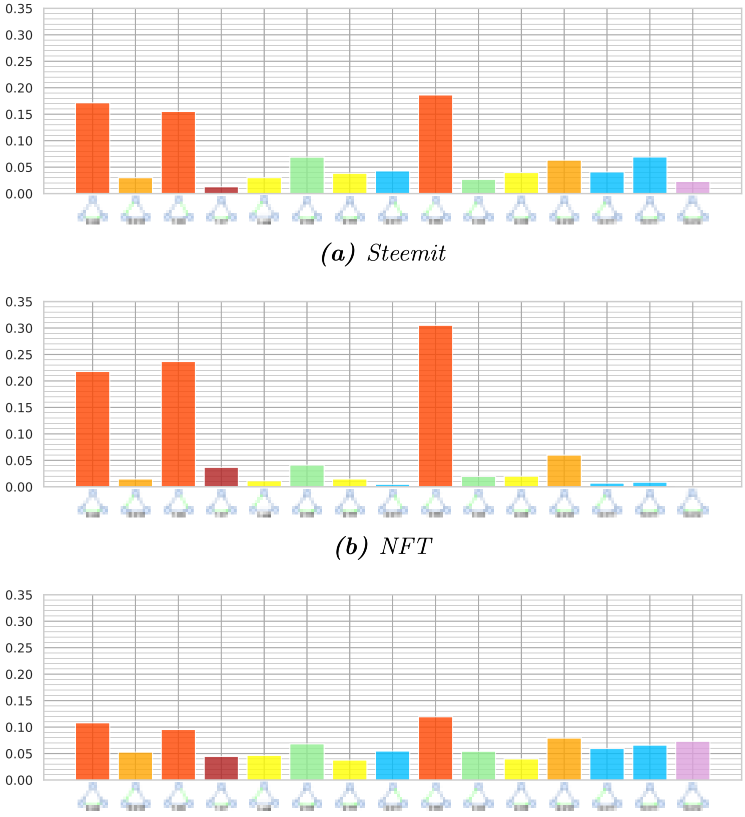
A structural difference in terms of the significance of closed open triads separates Steemit from Sarafu and the NFT network. In fact, for Steemit, all closed triad motifs are actually underrepresented w.r.t. the randomized networks. Given the nature of the network, it is quite an unexpected outcome since one would have expected over-represented triadic closure structures. A possible explanation of this outcome is two-fold: i) the period covered by the dataset captures the early stages of the network where accounts mostly joined other accounts without any attempts to consolidate their neighborhoods through closing triads; and *ii*) open triad motifs 5 are over-represented according to its z-score, and when randomized, those triads tend to turn into closed triad motifs 11 and 12, increasing the average frequency of triangles in the random model. On the contrary, the NFT trade and Sarafu networks are characterized by over-represented closed triad motifs. Specifically, all of the closed triads in the NFT network are more frequent than in the null model, while Sarafu has only some actually more present (8, 10, 11, and 12), those characterized by the presence of bidirectional links. In short, various kinds of triangles in NFT and Sarafu are not the outcome of random actions of the accounts, rather users are more likely to form a close triad. In particular, in Sarafu the tendency towards reciprocating links and the formation of triangles act together.

In a nutshell, to answer the first research question RQ1, each decentralized socio-economic network is different from the others. In a static setting, each network has its own specific profile based on the distribution of open and closed triads.

Closing temporal triads and triadic motifs

Although the analysis of triads on the static network representation highlighted how triad distributions differentiate a network from another one, our comprehension of the mechanisms leading to the formation of these specific patterns is only partial since we lose the sequentiality of the formation process provided by the temporal dimension. For this reason, herein we cope with temporal triads in order to answer RQ2, i.e. how triadic structures evolve and change over time, and whether there are growth patterns common to all the socio-economic networks. The main findings, detailed and discussed in the following, highlight:

- the central role of triadic closure processes leading to the formation of "feed-forward" loops, fundamental directed closed triads characterizing many directed networks in different domains. In fact, all the closing temporal triads ending into a feed-forward loop are the most frequent in all the networks; even if in Sarafu and Steemit some of these patterns are not statistically significant;
- the distribution of the closing temporal triads is a footprint of these socio-economic networks: distributions are different from one another, especially excluding the three most frequent closing temporal triads. For instance, the NFT network is mainly built around patterns leading to "feed-forward" loops while other patterns are irrelevant. On the contrary, the distributions of the closing temporal triads in Steemit and Sarafu are more uniformly spread over all the possible patterns. In particular, the temporal triads



(a) *Steemit*

(b) *NFT*

(c) *Sarafu*

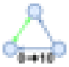
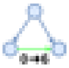
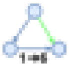
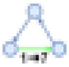
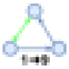
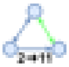
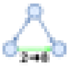
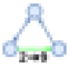
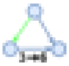
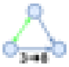
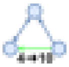
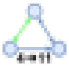
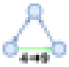
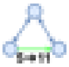
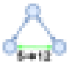
Fig. 4.7: *Distribution of the closing temporal triads in the three socio-economic networks. On the y-axis the frequency of the temporal subgraphs. Each temporal subgraph, along with its index, is reported below its color bar.*

leading to the creation of fully reciprocal triangles are frequent and significant. In short, even from the closing temporal triad standpoint, each network has its own specific profile which depends on the nature of the socio-economic actions it supports.

In the first instance, we look at the distribution $N(g_i)$ of closing temporal triads as reported in Fig. 4.2 and in Fig. 4.7. Overall, the three most frequent closing temporal triads are common to all three socio-economic networks, with slightly different rankings or frequencies. Specifically, all three temporal patterns lead to the formation of the “feed-forward” loop (identifier 6). In this pattern there is a specific hierarchy where a node is an “initiator” — it is only a source of token transfers, a node is a “target” — it is only a destination of transfers - and an “intermediate” node which is both source and destination. In the most frequent temporal triad 3→6 the initiator and the intermediate accounts transferred money to the same account — the target — and, after that, the initiator transfers money to the intermediate one. So, in this case, the target is immediately identified by both the remaining nodes. On the contrary, in 1→6 transfers between the initiator and the target are not immediate at the beginning, rather there is a two-hop connection passing through the intermediate node. Finally, in 0→6 the initiator transfers tokens to the remaining nodes and later the intermediate node interacts with the target. Observing the frequencies of the three most frequent temporal patterns we note that in Steemit and Sarafu patterns are almost equiprobable, while in NFT the gap between 3→6 and the other two temporal subgraphs is more evident. Indeed, in the NFT context, the pattern 3→6 may represent a collector behavior of the initiator which first collects and buys NFTs from a target creator and then collects other NFTs produced by the same creator but bought by the intermediate node, i.e. a third account.

A comparison among the overall profiles of the closing temporal triad distribution reveals an important difference: the frequencies of closing temporal triads excluding the top three in Steeemit and Sarafu are higher compared to the NFT network, where the gap between the top three and the other temporal triads is much more evident. More precisely, in NFT, besides the three most frequent subgraphs, only a few closing temporal triads are notable in terms of frequency: 1→7 — a directed closing loop, 2→10 and 2→8; where the last two are strictly related to the feed-forward loop as triads 8 and 10 are “feed-forward” loops where either the link between the initiator and the intermediate or the link between the intermediate and the target is reciprocated. On the other side, Steemit is characterized by a more varied distribution, where all the remaining temporal triads are more frequent, especially those involving open triads 4 and 5 as starting points (leftmost side of the distribution in Fig. 4.7a), i.e. open triads containing reciprocal links. This characteristic is even more evident in the closing temporal triad distribution for Sarafu (see Fig. 4.7c), where the temporal pattern 5→12, made by reciprocal links only, is among the most frequent items. Even in this case, the cooperative nature of the Sarafu socio-economic network impacts how open triads close, especially when reciprocal links are involved in the pattern.

Table 4.3: *Significance of the possible closing temporal triads for all three socio-economic networks. For each motif, we compute the z-score (z) and report the scores with p-value* < 0.01*, while the rest are not significant (NS). In each cell, the first line reports the count of the pattern, the second one its frequency, and the third line reports the z-score, between parenthesis.*

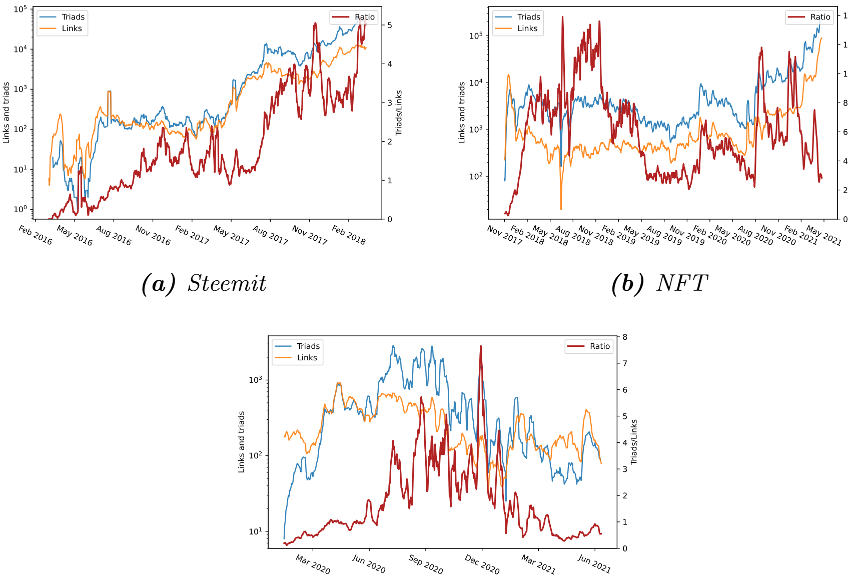
Pattern	Steemit	NFT	Sarafu
	137829 3.01% (NS)	192243 1.48% (-24.10)	14394 5.27% (NS)
	785803 17.16% (-11.44)	2834845 21.79% (-75.09)	29521 10.81% (-4.64)
	710914 15.52% (-24.35)	3079939 23.68% (-59.07)	26072 9.55% (-21.63)
	59399 1.30% (-21.94)	478230 3.68% (-39.07)	12181 4.46% (-19.54)
	138149 3.02% (-25.33)	147872 1.14% (-34.85)	12722 4.66% (-35.63)
	196684 4.30% (32.13)	61866 0.48% (-5.90)	14953 5.48% (NS)
	315517 6.89% (5.48)	533712 4.10% (3.33)	18672 6.84% (NS)
	175873 3.84% (2.74)	192721 1.48% (-5.15)	10297 3.77% (NS)
	853664 18.64% (NS)	3968946 30.51% (156.49)	32630 11.95% (NS)
	123287 2.69% (-8.38)	254909 1.96% (20.98)	14827 5.43% (7.95)
	289478 6.32% (11.80)	779546 5.99% (41.91)	21629 7.92% (3.48)
	187757 4.10% (20.12)	90725 0.70% (27.30)	16260 5.96% (7.10)
	182410 3.98% (6.45)	260942 2.01% (49.41)	10887 3.99% (3.18)
	317065 6.92% (31.38)	113449 0.87% (11.39)	17988 6.59% (20.28)
	105461 2.30% (47.69)	17633 0.14% (3.25)	19968 7.31% (56.86)

Closing temporal triadic motifs. So far we individuated some differences in the frequencies of temporal triads. However, as in the static case, closing temporal triads with high frequency may not be statistically significant. Therefore, we move on to the study of closing temporal triadic motifs, i.e. statistically significant closing temporal triads w.r.t a null model. We compute the z-score (Equation 4.2) for all the possible closing temporal triads and report the values in Table 4.3. Overall, we can observe important differences in the set of closing temporal triadic motifs. An interesting result concerns the statistical significance of the most frequent closing temporal triad 3→6. In fact, in Steemit and Sarafu, it is not statistically significant, i.e. we would find it similarly in a randomized network. Not being statistically significant does not mean it is not an impacting pattern during the evolution of these socio-economic networks, rather it raises some doubts on the willingness of such trait since it may be a consequence of random behavior. On the contrary, the same closing temporal triad is largely over-represented in the NFT network, a further signal that the purchasing strategy of the initiator in “feed-forward” loops has a certain level of intentionality. As for the remaining two most frequent closing temporal triads, they are under-represented in all networks, indicating that these patterns do not result from random behaviors. Furthermore, in Sarafu many closing temporal triads have failed the significance test as motifs, i.e. they occur in a comparable manner in randomized versions of the network. Finally, the analysis of the statistical significance further supports the findings about closing temporal triads involving reciprocal links; in fact, we observe that the closing temporal triads starting from open triad 5 (5→11 and 5→12), tend to be significant and overrepresented in Sarafu and Steemit. This result emphasizes the importance of reciprocal links in the creation of fully or almost fully reciprocal closed triads (identifiers 11 and 12).

In summary, to answer the second research question RQ2, a common growth pattern involves only the formation of “feed-forward” loops, while each network is characterized by specific creation patterns for closed triads.

Measuring triadic closure

Finally, we address RQ3, i.e. we focus on the stability of the triadic closure process as the network grows, and we assess how fast closing temporal triads form. To these aims, we measure a few dynamic aspects of triadic closure by leveraging the temporal information of the edges and computing different temporal metrics for triadic closure. Here, we find that each network has its own specific closure process trend, but all trends are unstable and sometimes connected to



(a) Steemit

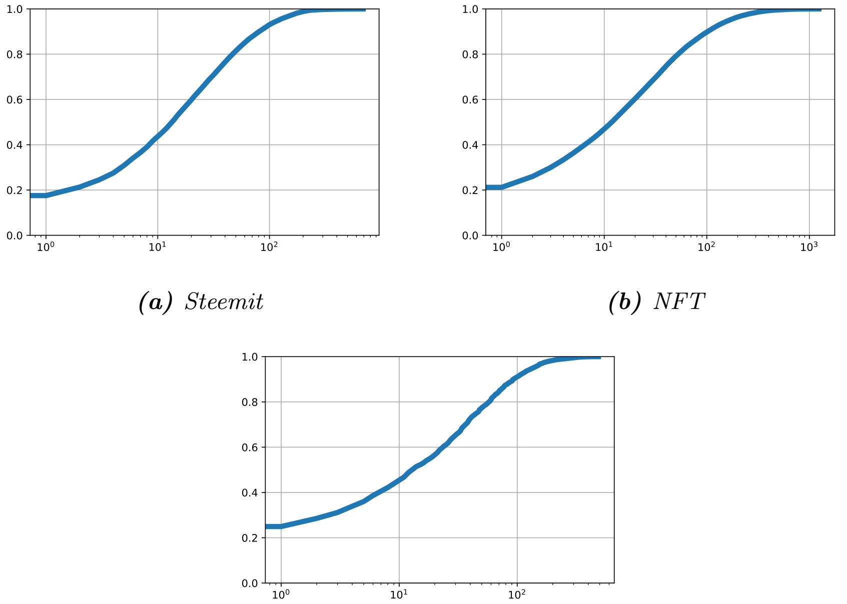
(b) NFT

(c) Sarafu

Fig. 4.8: *Measurements of links and triads. On the x-axis: days. On the left y-axis (log scale): the daily number of new links (orange) and triads (blue) formed during the growth of the three socio-economic networks. Trends have been smoothed by a moving average on a week sliding window. On the right y-axis (linear scale): the daily triad/link ratio between the triads and the links (red). The trend has been smoothed by a moving average.*

external conditions. Moreover, the triadic closure process is fast, i.e. half of the closed triads have formed in ten days. In general, from a dynamic viewpoint, these decentralized socio-economic networks are more unstable, more dynamic, and faster than centralized online social networks.

First, we study the impact of closure focusing on the number of triads that become closed (n_closed_triads), compared to the formation of new links (n_links) and their *ratio* over time. This metric highlights the average contribution of a new link in closing open triads. The obtained measurements are reported in Fig. 4.8, and they can be also confronted with those from previous studies on not-decentralized online social networks [21]. In more detail, in the Steemit network (see Fig. 4.8a), we have 730 days with at least a new



(a) *Steemit*

(b) *NFT*

(c) *Sarafu*

Fig. 4.9: *Triadic closure delay in days. On the y-axis: CDF of triadic closure delays. On the y-axis: number of days of closure before the triad closed.*

link formed, with an average of 1858 links per day and a peak of 23009 new links established on the same day. As for triadic closure, it is worth noting that in a few days, we did not observe any closing temporal triads. In fact, in the very beginning of the decentralized platform — the bootstrap period — is very common that most of the new links have involved new accounts reducing the chance of closing an open triad. However, after the bootstrap period, we observe an increase in the number of new daily triads resulting in an average number of daily closures equal to 6384, with a peak of 137414 on the same day, for a total of 4481692 closing patterns. This leads to an average ratio of 1.88 triad/link and a peak of 8.45 triad/link. Note that while the number of links and triangles are both rising, the ratio is actually growing, indicative that the links forming are actually making the structure more cohesive. In a comparison with not-decentralized online social networks, the average ratio resembles the measurements on the RenRen online social network, but the peak clearly surpasses the mainstream social networks. In fact, the values are similar to the peak values observed in Facebook after the introduction of the friend recommendation system, namely the "People you may know" (PYMK) service [21]. So, in Steemit, especially after the summer of 2017, the average contribution of links towards closing triads is naturally more important than in platforms that had introduced algorithms incentivizing the formation of triads.

As for daily new links and closing triads in the NFT network, shown in Fig. 4.8b, we have 1252 days with at least a new link, with an average of 2389 new daily links and a maximum of 103486 links formed in one day. Every day has at least a triadic closure, with an average of 10389 new daily closing temporal triads and a peak of 288827 on the same day, for a total of 13007578 closures. This leads to a high average ratio of 5.94 triad/link and a peak of 19.30 triad/link. The ratio is actually in a larger range than Steemit, and the average ratio is actually quite large. Over the entire observation period, the trend of the triad/link ratio is characterized by two phases of higher closing activities (from November 2017 to July 2019 and from November 2020 to February 2021) and a central period of low closing activity — from July 2019 to November 2020. This trend is generally different from the Steemit trend, where the triad/link ratio has almost always grown. By comparing these outcomes with other social platforms, the measured values are indeed in line with traditional online social networks, with peak values actually higher than the ones observed in the initial growth of RenRen and Facebook.

Finally, the measurements on Sarafu offer another different trait of the dynamic of the triadic closure process. In Sarafu we have 507 days with at least a new link, a total of 143239 links, an average of 283 new daily links, and a peak of 1370 new daily links created. Only in one day, we did not observe the formation of any triads. We record an average of 540 and a peak of 7328 new daily closing triads, leading to a total of 273001 closures. In Sarafu we observe an average triad/link ratio of 1.73 and a large peak of 15. The average ratio is indeed similar to Steemit, but the peaks are larger and closer to the NFT network. Unlike the previous networks, triadic closure seems to have an important impact in only a portion of the observation period (see Fig. 4.8c): the triad/link ratio started to grow only around July 2020, with the largest spikes occurring during the central period, from September 2020 to January 2021, while in the last period the triad/link ratio has reached a closing activity similar to the initial period: a low and stable average contribution of the new links to closed triads. In Sarafu, the overall trend is strongly connected to conditions external to the decentralized network, indeed, it had huge growth during the pandemic period, given its important role in supporting economic activities during the COVID-19 pandemic [20, 131]. Moreover, when compared to traditional OSNs, the values are still similar, and the peak is actually large, confirming the importance of triadic closure even in Sarafu.

To assess how fast is the triadic closure process in the three decentralized socio-economic networks, we analyze the triadic closure delay to understand if the triadic closure is a relevant factor. In fact, triangle closing speed compared to social networks would be another strong indicator of the importance of triadic closure. In Fig. 4.9, we report the Cumulative Distribution Function — CDF — of triadic closure delays, for the three networks. We can observe an interesting result: the distributions of delay have similar shapes, with a significant amount of triadic closures happening fast. More precisely, we focus on triads that close in less than a day: in Steemit at 18%, in the NFT network at 21%, and in Sarafu at 23%. In a comparison with not-decentralized social networks, the triadic closure process is much faster, in fact, in both Facebook and RenRen those values are actually much lower, in the range of 5% [21]. In particular, in those OSNs, half of the triads close in 25 days, while we find even higher values in these decentralized networks: in Steemit and NFT network 64% of closing triads are closed in less than 25 days, and a similarly high value characterizes Sarafu (61%). In all the networks, we record very fast closures, as most of them are closed in less than 3 months (90 days): respectively, Steemit 91%, NFT 88%, Sarafu 89%. In the centralized counterparts, the values are similar, around 80%.

To answer the third question RQ3; from a dynamic and longitudinal perspective, in decentralized socio-economic networks, the triadic closure has impacted the evolution and the growth of these platforms even more than in traditional and centralized online social platforms. The process is not stable at all, rather each network, as already discussed in the previous sections, is characterized by its own dynamics. However, there is a characteristic common to all these networks: the closure process is very fast, faster than in the centralized online social networks.

4.7 Conclusion

In this Chapter, we analyzed how triadic closure, one of the primary mechanisms underlying the formation of social ties, affects decentralized socio-economic networks, where social and economic interactions are strongly intertwined. We extended the existing methodology for triadic closure studies to generalize with directed networks, making it suitable to cope with the characteristics of decentralized networks, such as directionality a key component in economic transactions. We conducted an analysis of network structure centered on triads, i.e. 3-node subgraphs, and triadic motifs, i.e. statistically significant triads while considering both a static and dynamic viewpoint. The methodology was applied to three distinct decentralized socio-economic networks (Steemit, Sarafu, NFT trades) with varying degrees of influence from social ties. The main takeaways are:

- *From both a static and dynamic perspective, each network has a distinctive profile depending on the nature of the socio-economic activity it facilitates..* From a static viewpoint, the analysis shows that networks, where the *interplay between social and economic traits is stricter*, are characterized by *more reciprocal relationships and triads*, whereas networks where the *interplay is weaker*, such as NFT networks, are characterized by a *predominance of feed-forward loops*. Moreover, although all triadic closure patterns bear significance, rendering them inexplicable through random behavior, we have observed variations among networks regarding the prevalence of both underrepresented and overrepresented close triads. From a temporal perspective, the *distribution of closing temporal triads* serves as an indicative representation of these socio-economic networks. The distributions *exhibit variations among each other*, particularly excluding the three commonly occurring frequent closing temporal triads. For instance, the NFT network is mainly built around patterns leading to feed-forward loops while other patterns are unimportant. In contrast, the distributions of the closing temporal triads in Steemit and Sarafu are more evenly dispersed across all the possible patterns. In particular, the temporal triads that result in the creation of fully reciprocal triangles are frequent and significant.
- *Triadic closure has impacted the evolution and the growth of these platforms even more than in traditional and centralized online social platforms.* The analysis of the stability of the process over time shows how the *triadic closure process is not stable at all*, rather each network is characterized by its own dynamics. The measurement of how fast closing temporal triads form, through the *directed triadic closure delay*, showed how there is a characteristic common to all these networks: the *closure process is very fast, faster than in the centralized online social networks.*

Overall our work presents strong evidence that triadic closure is an important evolutionary mechanism in the selected networks. Our analysis through temporal motifs highlighted similarities and differences across decentralized networks with different levels of social components. And indeed those observations make sense when we consider that the method highlighted both differences and similarities between systems where native cryptocurrencies are used for social-economic purposes and the maintenance of the platform (Steemit and Sarafu), from systems where exchanges of cryptocurrency still have a social component but are also tied to the trade of the NFT tokens, created for specific purposes (NFT market). This highlights the expressivity of the footprints based on temporal motifs. Indeed, our findings suggest that the social component cannot be ignored for a better comprehension of network growth of decentralized socio-economic networks.

Future works include the analysis of other Web3 systems with more or less of a social component. Understanding the growth of other decentralized online social networks not following the Web3 paradigm is also an important open issue. It would also be interesting to analyze trade relationships in other economic networks, to understand the differences in their structure. Moreover, we could leverage user features to study the interplay with triadic closure. The evaluation of other established growth mechanisms would also be an important step toward the comprehension of the growth of these innovative systems.

Part III

The interplay of user behavior and currencies

Chapter 5

Interplay of user activity and currencies in social networks

5.1 Introduction

We are currently witnessing a dramatic moment of crisis and deep renewal of the social media landscape induced by two opposite forces. On the one hand, these platforms are increasingly playing a fundamental role in many aspects of the life of human beings, especially in the new generations who continuously ask for new services. On the other hand, there is a growing awareness that the traditional model of centralized social networks is no longer sustainable and poses crucial challenges that require adequate and rapid solutions to the well-known issues of privacy, content quality, censorship, and data ownership and monetization.

Among the various possible solutions, one of the most promising is blockchain-based online social networks (BOSN), which put themselves forward as social platforms able to overcome all current issues of centralized social networks. Actually, three specific aspects, common to most of the current BOSNs, are: a decentralization based on blockchain technologies that overcome privacy and censorship problems; a token system based on proprietary cryptocurrency used for fostering high-quality content; and a rewarding system for distributing the wealth of the platform giving data monetization back to users and encouraging good practices. Despite having been around for a few years, we are very far from having a full understanding of to what extent the Web3 paradigm solves the issues of traditional architectures and what are, if exist, the other problems they potentially introduce.

The true pivot of BOSN is the introduction of a cryptocurrency that shifts the paradigm of online social networks from being purely social to economic-social: in the traditional approach, users are engaged with social interactions, while economic ones are prerogative of platform ownership; while in BOSN users are got dragged into social-economic actions. Thus, the way to understand the BOSN in-depth passes through the investigation of the relations between the economic and social actions carried out by users and how both relate to the value of the cryptocurrency.

To shed light on this complex network of intertwined layers, we adopt a data-driven approach, by analyzing Steemit, one of the first and most successful BOSNs. By gathering data from the underlying blockchain Steem, we have collected a large longitudinal dataset that contains the main social and financial activities of Steemit users spanning more than three years, along with data external to the Steemit platform: longitudinal data of STEEM value in the cryptocurrency market. From these data we were able to reconstruct the high-resolution evolution of the system to address the main goal of our study: the interplay between users' social and financial activities, resulting in social and economic networks, and the currency price; with a specific focus on the possible effects of the currency price on the network structure. We aimed to answer the following research questions: *RQ1)* What is the interplay between currency and network? *RQ2)* What is the relation between user activity and the reward system? Our analysis based on time series correlation has pointed out a possible influence of the platform cryptocurrency on the evolution of the Steemit social network, i.e. "follow" or link creation actions have been partly driven by the trend of the cryptocurrency (*RQ1*). Higher prices have attracted more users and shifted the mechanisms and the strategies ruling link creation. Strategies and action allocation, especially for the most central nodes, are a further focus of our study. In particular, we highlighted which actions central nodes have mainly chosen to gain the highest cumulative rewards. Here, we observe that central nodes exploit both their high rank in the voting system and the mechanism of the rewarding system to get rewards, i.e. they tend to prefer voting operations to actions for producing content, such as posting and commenting (*RQ2*).

The above findings suggest that the transformation of the actual online social platforms — which in the last years have shaped and are still changing our society — into new paradigms supported by blockchain technologies asking for new perspectives for the study of their evolution. Indeed, economic and financial aspects might play a more decisive role in how people behave in these new platforms, enough to question the relational aspects, typical of the main online social networks.

5.2 Related work

Although the research field about blockchain-based solutions and networks resulting from cryptocurrency transactions has been very active in the last few years ([137, 138, 139, 140] to cite a few studies), blockchain-based social networks (BOSNs) and their specific characteristics are not fully understood, yet. Only recently the availability of tools for querying the underlying blockchain and the increasing interest in Web3 and its related technologies have triggered studies focused on different aspects of these large-scale intertwined complex networks. For example, Li *et al.* [104] released a dataset paper, stressing the potentiality of this network, meanwhile highlighting difficulties in extracting and processing the high volume of data produced by the platform. Other works focus on the characteristics of this innovative type of social network ([10, 105, 106, 4]). User-generated content is useful for text mining tasks [107] and bot detection ([108, 141]). There is also a growing interest in social network structure. Chonan [109] and Kim *et al.* [65] focus on the structure of the Steemit social network and its characteristics. Furthermore, Guidi *et al.* [110] delve into a study of the follower–following graph, and analyze other operations in Steemit [102]. Aside from the relationships among users, Guidi *et al.* [111] studies block producers (witnesses) and highlight their social impact on the platform. Other works are more focused on the economic aspects: Ciriello *et al.* [23] and Thelwall *et al.* [24] analyze the relationship between rewards and content, while Li *et al.* [22] describes and analyzes the networked structures behind the Steemit rewarding system.

Even though BOSNs may provide high and detailed volumes of temporal data, there is still limited work focused on network dynamics and temporal aspects of BOSNs. For instance, Jia *et al.* [112] focus on the diffusion of contents at a mesoscopic scale, while Ba *et al.* [113] has been the first work that has started to tackle the interplay between cryptocurrency and graph evolution. This latter study is extended by this Chapter by taking into account all the social and financial actions and inspecting the allocation strategies of the most rewarded users. Finally, further characterization of the processes and dynamical aspects of Steemit's growth has been addressed in [125].

5.3 Research questions

The related work sections highlight an important gap in the comprehension of network evolution in Blockchain online social networks and its relation with the innovative financial aspects. In particular in Steemit, the strict interplay

among the cryptocurrency market, the network-based strategies to gain more STEEM, and the rewarding mechanism have led to the hypothesis that economic/financial factors, such as the price of the STEEM cryptocurrency, may influence the social network supported by Steemit. In this Chapter, we mainly focus on the validation of this hypothesis and we show some pieces of evidence which are in line with it. Second, we also deal with the strategies to gain rewards. Specifically, we focus on users who have obtained the higher amount of rewards, i.e. the most successful one: do wealthy users mostly prefer financial-oriented actions or do they produce or promote content through social actions? In other words, we focus on these two research questions:
Research question RQ1: What is the interplay between currency and network?
Research question RQ2: What is the relation between user activity and the reward system?

5.4 Data Preprocessing

To carry on our investigation, we rely on the Steem-Hive dataset . More precisely, we focus on two types of data, internal and external to the Steemit platform: *i)* data on social and financial activities performed inside the platform, and *ii)* longitudinal data of STEEM value in the cryptocurrency market. The latter information can be retrieved from [142], a website that reports the daily value of the STEEM currency in US Dollars and other cryptocurrencies. The prices are updated daily, allowing us to collect data for the STEEM price in USD for the entire observation period.

In this Chapter, we study social and financial aspects: hence, we focus on two subsets of user operations: *i)* social and *ii)* financial operations. Social operations include actions that users usually do on traditional social media platforms, such as posting content or votes; while we denote as financial operations those operations designated for rewards and token management. Social actions are stored in three social operations: `comment`, `vote` and `custom json`; while rewards and token related operations are stored in six operations: `claim reward balance`, `transfer`, `transfer to vesting`, `withdraw from vesting`, `delegate vesting shares` and `convert`. A full description of the aforementioned operations is presented in Table 5.1. Details on the data collection process have been discussed previously in section 2.4.

Table 5.1: **List of social and financial operations.** *Each operation is characterized by its name, its type and a full description.*

Operation	Group	Description
`comment`	social	A user publishes content or comment on a post
`vote`	social	User upvotes or downwotes. Users can vote on posts and comments
`custom json`	social	A general-purpose operation designed to add new functionalities without the need for new operations. Social functionalities include: i) **"follow"** to receive updates on what other users are posting, ii) **"unfollow"** to stop following other users, iii) **"mute"** to block users from the feed in case of harassing or unwanted content, and iv) **"resteem/reblog"** to share content of another user to all the followers
`claim reward balance`	financial	User claims reward for creation or curation (amounts in STEEM and Steem Power)
`transfer`	financial	Transfer of the main token STEEM from an account to a "target" account
`transfer to vesting`	financial	"Power up": convert STEEM to Steem Power at the current exchange rate
`withdraw from vesting`	financial	"Power down". the conversion from Steem Power back to STEEM
`delegate vesting shares`	financial	Borrowing Steem Power. The Steem Power is still owned by the original account
`convert`	financial	Conversion from STEEM to SBD

5.5 Methodology

Our first objective is to study whether users' behavior is influenced by the cryptocurrency system, or vice versa, if the financial system is influenced by social activities. So, we first describe the methods to highlight the possible interplay between users' behavior, expressed by the trend of social and financial operations, and the value of the cryptocurrency. Then, we focus on the reward system. Here, we are interested in the preferred strategies put in place by the users to gain rewards. Specifically, we focus on users who are gaining the most from the platform, the so-called *whales*.

Analyzing user behavior and cryptocurrency (*RQ1*)

We deal with the influence of the cryptocurrency market on users' behavior in Steemit by investigating the interplay between the trend of the STEEM value in the market and the social/financial activities carried out by users on the platform. To this aim, we construct the time series of the STEEM daily price and, for each of the nine operations, we also build the operation time series, i.e. the number of daily activities carried out by users. A side-by-side comparison of the obtained time series enables us to highlight evidence of whether and how the STEEM price impacts social and financial activities.

First, we search for potential seasonal patterns by computing the *Autocorrelation Function* (ACF). The ACF measures the linear relationship between lagged values of a time series; the resulting plot — also known as *correlogram* — shows the presence of patterns or long-term trends, and seasonal patterns [143]. Specifically, the ACF is the function of autocorrelation values ρ_k for every lag k, where $\rho_y(k)$ is defined as

$$\rho_y(k) = \frac{\sum\limits_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum\limits_{t=1}^{T} (y_t - \bar{y})^2}$$

If data are trended, values of $\rho_y(k)$ will be large and positive for small lags, as closeness in time will lead to closeness in lag size [143]. So, the trended time series will have ACF with positive values that slowly decrease as the lags increase. If the times series has a seasonal trend, the values of $\rho_y(k)$ will be larger for seasonal lags (at multiples of the seasonal frequency) than for other lags. Both these phenomena can be observed when data have both trends and seasonal patterns.

After focusing on the singular time series, we will shift our attention to the link between users' actions and the cryptocurrency price. To this aim, we measure potential correlations between each operation and STEEM prices. We evaluate the correlation by the *Pearson Coefficient*[144]. Given two time series x and y, we compute the *Pearson Coefficient* $\rho(x, y)$ as :

$$\rho(x, y) = \frac{\sum(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum(x_t - \bar{x})^2}\sqrt{\sum(y_t - \bar{y})^2}}. \tag{5.1}$$

with values near 1 indicating perfect correlation, values near 0 indicating the absence of cross-correlation and values towards -1 indicating perfect anti-correlation.

Finally, we measure potential lead-follow relationships between time series using the *normalized cross-correlation* measure. Given two time series x and y, the normalized cross-correlation measure is similar to the correlation measure: instead of correlating x with y once, we do it multiple times, considering the time series y, but shifted by a series of time lags k. We obtain a series of different correlation values ρ, one for each chosen time lag k. In our work, we consider lags in days. This measure can be expressed as:

$$\rho_{xy}(k) = \frac{\sigma_{xy}(k)}{\sigma_x \sigma_y} = \frac{\sum\limits_{t=k+1}^{T} (x_t - \bar{x})(y_{t-k} - \bar{y})}{\sum\limits_{t=1}^{T} (x_t - \bar{x})(y_t - \bar{y})} \tag{5.2}$$

This calculation produces a set of pairs (lag, correlation value). We can better explore them by analyzing their shape and focusing on the time lags k that show the highest correlation values. If we find high correlation values for a positive time lag, then x leads y; vice versa, if the highest values are for a negative time lag, then we have that time series y is leading x.

Users' behavior and rewards (*RQ1*)

In order to study the relationship between users' behavior and the gained reward, we characterize users in two dimensions. We construct for each user, a profile that summarizes two key aspects: *i*) gained rewards, and *ii*) user activity, i.e. number of actions performed.

As for the first aspect, we look at the total amount of rewards received and select the users who have gained the most from the network. This way, we focus on users who have adopted the most effective behaviors. Operationally, in our analysis, we identify the *hubs*, i.e. the users in the top 10% of the reward distribution for each currency. The choice of considering all the currencies is due to the fact that Steemit users can decide how to receive their rewards, so we may find different behaviors based on currency. In fact, only 27% of the union of all hubs are hubs for the three tokens, while 18% are hubs exclusively for the Steem Power token.

As for user's activity, we look at the different types of actions listed in Table 5.1 and for each user we measure *i*) whether s/he relies more on curation or creation, based on currency, and *ii*) whether s/he relies more on financial or social actions, based on currency. To this aim, each user u is characterized by a triple (s_{cr}, s_{cu}, f), where s_{cr} denotes the overall volume of comments and posts published by u, s_{cu} indicates the number of voting operations made by u

and f corresponds to the total volume of financial operations. From this triple, we can measure whether an individual relies more upon creation or curation actions through the

$$creationindex = \frac{s_{cr}}{s_{cr} + s_{cu}}$$

and the

$$curationindex = \frac{s_{cu}}{s_{cr} + s_{cu}} = 1 - creationindex.$$

Similarly, we can measure whether s/he relies more on social or financial actions by computing the

$$socialindex = \frac{s_{cr} + s_{cu}}{s_{cr} + s_{cu} + f}$$

and the

$$financialindex = \frac{f}{s_{cr} + s_{cu} + f}.$$

For both measurements, higher values mean more reliance on social or financial actions, respectively. We compute these indexes for each user, then through an analysis of their distribution, we can inspect the overall behavior and potential differences between currencies. Finally, by correlation analysis, we analyze the relationships among the dimensions s_{cr}, s_{cu}, f, and the rewards obtained in the three token systems.

5.6 Results

Interplay between users's social/financial actions and STEEM price

In order to answer *RQ1*, our first goal is to study the relationship between the value of STEEM in the market and the social/financial activities of users to find evidence of a possible influence of the price of STEEM cryptocurrency on social actions provided by Steemit. We analyze the time series of all the operations performed by the users of the platform, described by the number of actions per day. Alongside them, we analyze the daily price of STEEM. From the overview of all the time series, it is evident the impact the currency value has on users' actions. The successive quantitative trend and correlation analysis reveal a significant pattern of correlations.

Time series: currency and user actions.

By looking at an overall picture of the time series of all the social/ financial actions and of STEEM price, we get some preliminary qualitative evidence. We displayed in Fig. 5.1 the time series of the number of operations per day carried out by all users of the platform for the main social and financial actions, and the currency value. In the figure, we highlighted — blue vertical lines — important external or internal events that may have affected the network growth and/or the value of the STEEM currency.

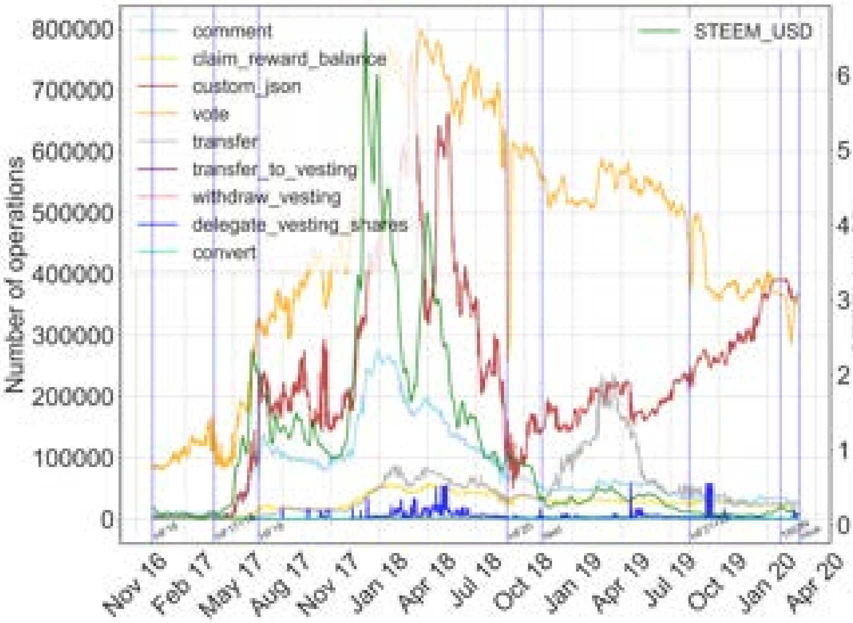


Fig. 5.1: **Social/financial action time series and STEEM price.** *Time plots of the daily volume of social and financial operations along with the STEEM price in USD (green). On x-axis: time in days. On the left y-axis: volume of operations per day. On the right y-axis: STEEM price in USD. The blue vertical lines correspond to important events, like hard forks (HFXX), the crisis announcement by Scott (Ned) — Steemit founder, the selling of the company to TRON Foundation (TRON), and the Hive fork (Hive), which corresponds to the end of the observation period.*

As for the exchange value of the STEEM currency, we observe a few distinct phases: from November 2017 to the second half of December 2017 there is a rapid growth phase, where STEEM reached its maximum quotation; this period is followed by an equally rapid decrease till March 2018, where STEEM bounced back for a short period — April 2018. After this date, we observe a continuously decreasing trend till the end of the observation period. Per se, the STEEM trend has followed the trend of other cryptocurrencies, but specific correlations may emerge if we also consider the trends of the other platform operations. For instance, if we focus on social operations only — which have reached the highest volume of actions — we can see hints of a temporal correlation between social actions and cryptocurrency. In particular, the STEEM value and the volume of `custom json` operation show similar traits, given a time-shift, as already detected in [22] and in [113] on "follow" relationships. For example, a first period of growth of the `custom json` volume (April 2017 — June 2017) corresponded to a higher STEEM price, or, more evidently, the first rapid growth of STEEM corresponded to an equally rapid increase of operations which reached the peak on March 2018; while a bounce similar to what occurred to STEEM has also happened to `custom json` operations on April 2018. And again, a drop in STEEM price hampered the overall activity in the network till the hard fork 20 (HF) and the letter sent to the Steemit community by the founder Ned Scott on 28/11/18, confirming the crisis of the platform [145].

All the above considerations come from a graphical inspection of the trends; in the following, we analyze the time plots more in detail and we perform a quantitative evaluation of correlations between social/financial actions and STEEM price.

Trends in time series.

A preliminary analysis has been conducted on each time series to identify if the above hints of correlation are a consequence of seasonal patterns or trends of the time series itself. In fact, a weak or missing signal of the presence of this kind of pattern would support a search for correlations between the social/financial time series and the STEEM trend. We search for seasonal patterns and trends by computing the auto-correlation function for the time series since the resulting *correlogram* potentially shows the presence of long-term trends and seasonal patterns if exist. A subset of correlograms is reported in Fig. 5.2, while the whole set can be found in Fig. 1 in S1 Text.

The STEEM price correlogram, in Fig 5.2a, is characterized by the lack of repeating peaks, suggesting the absence of seasonal trends. However, the

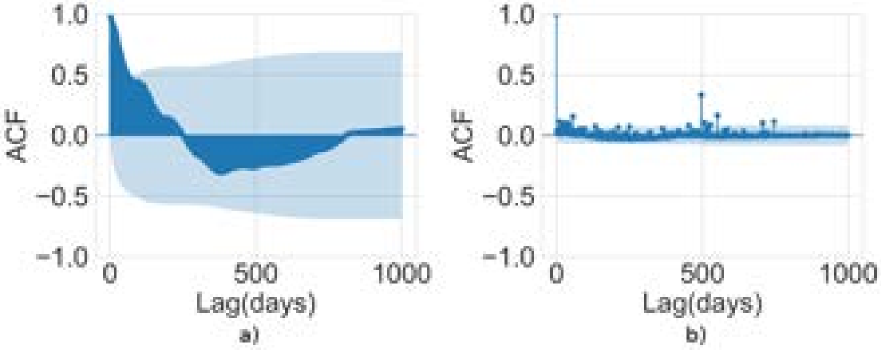


Fig. 5.2: **Autocorrelation Functions (ACFs).** *The autocorrelation function for the a) STEEM price and b) delegation of Steem Power. On the y-axis: the correlation coefficient* $\rho_y(k)$*. On the y-axis: the lag* k *in days. The light cyan area corresponds to the* 95% *confidence interval for the correlation coefficient.*

STEEM price has a short trend since we observe positive values that slowly decrease as lags increase, but only the very first lags are characterized by a positive correlation coefficient which lies outside the confidence intervals, so statistically significant. This trait is common to other actions, showing similar characteristics, except for a few. More precisely, only two actions show different traits, namely two financial actions: lending Steem Power (`delegate vesting shares`) and powering down (`withdraw vesting`). The autocorrelation plot of the former is shown in Fig. 5.2b. Here, the main difference with the STEEM price time series is the rapid drop for small lags, a typical characteristic of time series without a trend. To sum up, all the operation time series do not show seasonal trends, and we only observe short-term correlations. This way, the hints of correlation may be searched by comparing pairs of time series.

Social actions and currency.

Since we are interested in verifying whether financial factors impact how people connect in the social platform, we first look at social actions and their relationship with the STEEM currency. In fact, social actions directly determine or are strictly related to the social graph. The time plots for the social actions — `vote`, `comment` and `custom json` — and STEEM price are displayed in Fig. 5.3, and time plots for each action can be consulted in Fig. 2 in S1 Text.

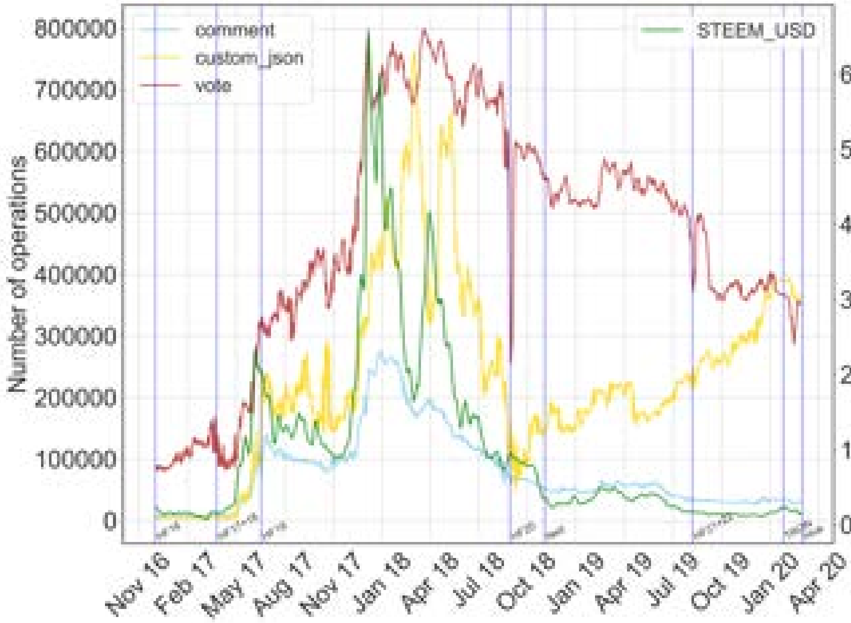


Fig. 5.3: **Daily volume of social actions.** *Time plots for social operations (`vote`, `comment` and `custom json`) and STEEM price value in USD (green). On x-axis: time in days. On the left y-axis: volume of operations per day. On the right y-axis: STEEM price in USD. The blue lines correspond to important events, like hard forks (HFXX), the crisis announcement by Scott (Ned), the selling of the company to TRON Foundation (TRON) and the Hive fork (Hive).*

First, we observe that most social actions drop in volume as time passes. The creation of posts and comments — both included in the `comment` operation — dropped as the currency price falls during the first quarter of 2018. As expected, the incentive to post and comment is weaker as the STEEM price is at the lowest; in fact, a user mainly interested in increasing their rewards by producing high-quality content has to spend a big effort to gain rewards with a low value. From a quantitative standpoint, we find positive correlations between STEEM price and social actions, as reported in Table 5.2. More precisely, STEEM value and posts/comments show a strong positive correlation (0.91). Moreover, by cross-correlation measure, we can also analyze the temporal correlation. In fact, we find an even stronger positive correlation with a maximum cross-correlation of 0.94 associated with a small lag of days, i.e. 15 days.

Table 5.2: **Social actions and cross-correlations with STEEM price.** *The column "Total" reports the overall volume of operations during the observation period. The second column reports the average daily volume. In the last three columns we report the cross-correlation, the maximum cross-correlation and the lag with the highest cross-correlation, respectively.*

	Total	Average	Corr	Max XCorr	Lag (days)
Operation					
comment	93832667	79654	**0.91**	**0.94**	15
vote	546677598	464073	0.53	**0.78**	97
custom json	270860412	229932	0.52	**0.82**	40
custom json (followed)	134608190	114268	0.74	**0.91**	36
custom json (unfollowed)	20179192	17130	0.58	**0.80**	44
custom json (muted)	540182	459	**0.82**	**0.92**	16
custom json (post share)	8267940	7019	**0.87**	**0.93**	11

As for `vote` operations, we notice a similar drop in volume. It would be expected, as there is less content to consume and, especially after the hard fork 19 — HF19 -, the voting power is limited by the amount of Steem Power owned by voters. However, the drop is not as marked as we see in comments, which is reasonable, as votes still require less effort than producing content; in fact, users only need Steem Powers and a click on the post/comment. Therefore the correlation between votes and STEEM price is much lower — 0.53 — than the comment correlation and a moderately positive cross-correlation can only be found with a high lag of more than 90 days (see Table 5.2).

`custom json` operation has a different evolution: in the initial period, till the hard fork HF20, the number of operations behaves more like posts and comments, rising and dropping as STEEM price does. However, after the hard fork H20, we can observe that the number of daily operations started an increasing trend again. In this case, this is a consequence of the fact that among the operations we have not only social actions (follow, share, unfollow, ignore), but also other actions, as well. In fact, new apps and platforms can rely on `custom json` operation to save their data. These operations are also used by other services outside Steemit, other decentralized apps, such as Dtube [1] and SteemMonsters [2].

To understand whether the rise is caused by social actions or other factors, we detailed the `custom json` time series by separately analyzing the daily volume of the specific actions contained in `custom json` records (see Fig. 3, Fig. 4 and Fig. 5 in S1 Text). The key takeaway is that among the operations belonging to `custom json` category, social actions (follow, share, unfollow, ignore) have declined as the other social actions (votes, posts, comments), so the trend after HF20 is mainly driven by new operations performed by decentralized apps operating on the Steem blockchain. Thus, we observe a clear shift in how the blockchain is being used [3].

Given the above observations, we separately measure the cross-correlation between the STEEM price and the social actions in `custom json` operations, and the STEEM price and the other actions in `custom json`. In fact, by isolating the main social actions (follow, share, unfollow, ignore), we obtain much higher values of cross-correlation, with lower day lags, with respect to correlations and lags computed comparing the STEEM price and the overall volume of `custom json` operations, as shown in Table 5.2.

In general, the analysis of the cross-correlations among the STEEM price and the different social actions highlights that cryptocurrency had an impact across all social activities. This represents the first evidence of the possible influence of economic and financial factors on the structure of the social graph supported by blockchain-based online social networks.

Financial actions and currency.

We also focused on the relation between the STEEM price and financial actions in Steemit, since most of them determine an interaction between Steemit users. These actions are less frequent in the network, in terms of daily volume, with respect to social actions but are still notable since they represent an element of novelty in the online social network landscape. Following the above methodological approach, we first look at their daily volume, as reported in Fig. 5.4, where we jointly display their time plots[4]. Alongside these plots, we report volumes and correlation measures in Table 5.3, respectively.

[1] https://d.tube

[2] https://splinterlands.com

[3] HF20 has introduced many changes. Among them, is a revamped system, currently in use, that influences the amount of actions allowed for a user.

[4] Plots for each action can be consulted in the Fig. 2 in S1 Text.

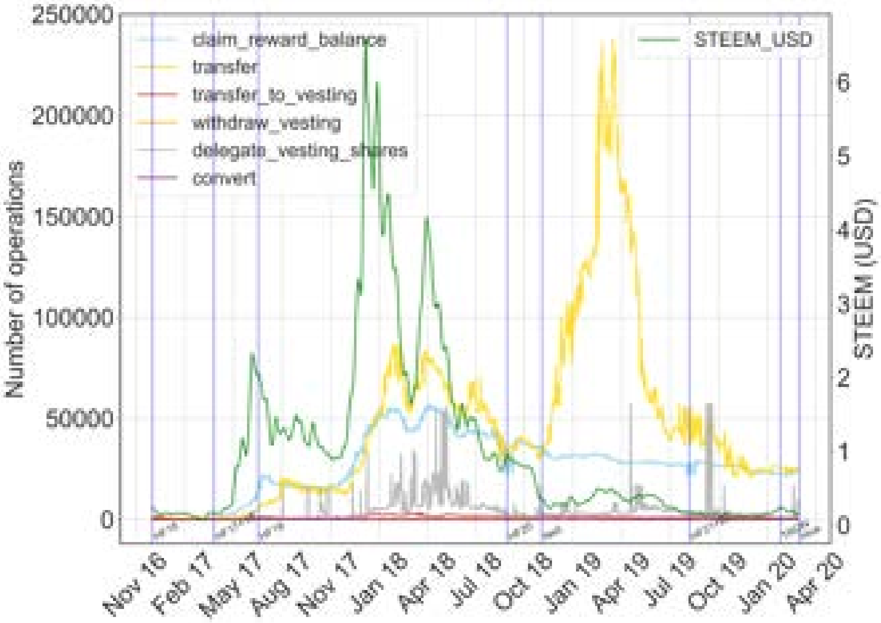


Fig. 5.4: **Daily volume of financial actions.** *Time plots for financial operations and STEEM price value in USD (green line). On the x-axis: time in days. On the left y-axis: daily volume of operations. On the right y-axis: STEEM price in USD. Blue vertical lines correspond to important events, like hard forks (HFXX), the crisis announcement by Scott (Ned), the selling of the company to TRON Foundation (TRON), and the Hive fork (Hive).*

As observed in the case of social actions, we can see that most financial actions dropped in volume as the value of the currency dropped. However, time series have different traits, mainly due to the type of currency involved in the action. For example, reward claims (blue line) are not dropping as steadily, as the reduction of user activity results in less competition. While the amount of conversions to Steem Power (`transfer to vesting` — red line), which is the equivalent of an investment in the platform, drops as the currency becomes less valuable. For these two types of financial actions, we observe a medium positive correlation (0.53 and 0.56), while getting their maximum cross-correlation after about 30 days (0.8 and 0.83), respectively. Whereas other operations seem to have a weak relationship with the STEEM price. For example, we can observe that the power-down operation (withdraw vesting), has spiked in the crisis periods. Similarly, the conversion to SBD (convert) reached its lowest during the period of crisis, as users were looking to move back to STEEM to trade and cut losses or to try and speculate. In these cases, we observe weak correlations and cross-correlations with lag values too high to be related to a possible influence, especially for `convert` operation.

Table 5.3: **Financial actions and cross-correlations with STEEM price.** *The column "Total" reports the overall volume of operations during the observation period. The second column reports the average daily volume. In the last three columns, we report the cross-correlation, the maximum cross-correlation, and the lag with the highest cross-correlation, respectively.*

	Total	Average	Corr	Max XCorr	Lag(days)
operation					
claim reward balance	31609874	29709	0.53	0.80	26
transfer	55033746	46718	0.04	0.83	436
transfer to vesting	1393465	1183	0.56	0.80	39
withdraw vesting	344062	292	0.30	0.60	93
delegate vesting shares	5260366	5039	0.12	0.33	90
convert operation	101308	93	-0.02	0.46	-140

We also obtain a low correlation value for transfers of STEEM, even if its trait is different from the other financial actions: while they seem to rise and fall as the other actions during the first half of the observation period, we notice a spike after the hard fork HF20. While the cross-correlation value is high, the lag is too long — 436 days, indicating a not-informative correlation.

Finally, we observe some spikes in lending of Steem Power (`delegate vesting share`), that are not related to STEEM price: they may be related to other events where Steem Power is critical, such as witness election. In fact, the correlation values are low, suggesting that there could be other factors in play.

To sum up, the correlation values on the overall time period between STEEM price and financial actions are not as strong as in the case of social actions. While visual evidence suggests some effects, it looks like the relationship may be more complex, and deserves further analyses.

Rewards and users: the behaviors of highly rewarded accounts

For *RQ2*, we are interested in users' preferred ways to gain rewards on the platform. Specifically, we analyze the users with the highest rewards, i.e. the hubs or richest nodes, by focusing on

1. whether they rely more on curation or creation based on the type of token used to claim their rewards; and
2. whether they rely more on financial or social actions, based on the type of token

.

Therefore for the study of rewards, we describe a user by

1. *rewards sbd*, i.e. the total amount of rewards in SBD;
2. *rewards steem*, i.e. the total amount of rewards in STEEM; and
3. *rewards sp*, i.e. the total amount of rewards in Steem Power

. While for user activity, we examine:

1. the *creation* activity, i.e. the number of posts and comments;
2. the *curation* activity, i.e. the number of votes;
3. the *social* activity given by the total amount of creation and curation actions; and
4. the *financial* activity, i.e. the total amount of financial actions.

The combination of these variables allows us to characterize the behavior of users, looking for potential differences in users' activity according to the type of currency.

Creation and curation activity

We visualize the values of curation and creation for hubs in Fig. 5.5. Through scatter plots, we can see that there is a skew toward either creation or curation for the hubs. The relation between creation and curation activities has been reported for different currencies, since users can choose how to get their rewards, either as 100% Steem Power or a 50/50 split in Steem Power and one of the liquid currencies STEEM/SBD (see Section "The rewarding mechanism"). The visual analysis shows a different distribution between Steem Power and the other currencies. In the scatter plot representing the hubs for Steem Power (`rewards vests`) — Fig. 5.5c — users are distributed in a slightly different way: we can see that there are more hubs that have high levels of curation actions, and some of them have a very low creation activity. This difference between liquid currency holders and Steem Power holders is consistent with the purpose of the tokens. Indeed, users with high Steem Power have more influence on the curation process and rewards. In fact, as the Steem Power behind the vote influences which posts become more visible and the rewards for curators are proportional to the power to their weight, it becomes more effective for Steem Power owners to curate, instead of spending time and effort creating new content.

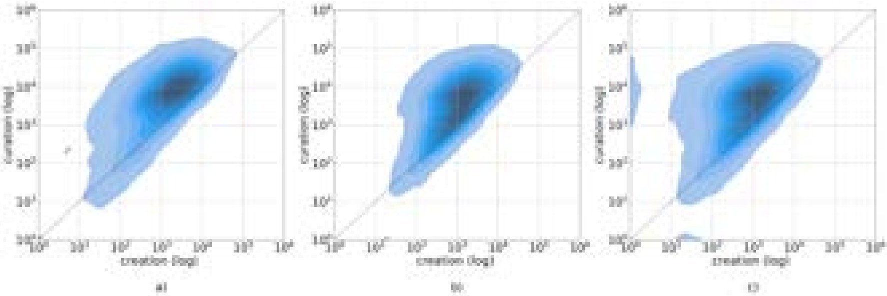


Fig. 5.5: **User's creation and curation, for each currency.** *Scatter plots (obtained by Kerned Density Estimation — KDE) relating curation (number of votes) and creation (number of posts and comments), for each of the three currencies: a) STEEM, b) SBD and 3) Steem Power. On the x-axis: the creation activity. On the y-axis: the curation activity. Darker colored areas correspond to higher density.*

As for the relation between curation and creation activities, we also analyzed the *curationindex* and the *creationindex* as defined in Section Methodology. In Fig. 5.6, we visualize the distribution of the curation index for the hubs[5]. The curation index distribution shows that overall users rely more on curation, in line with the previous observation. We can also see that there are only small differences between currencies.

Social and financial activities

We have also applied the above approach by focusing on the relation between social and financial activities. Indeed, the allocation of social and financial

[5] Since the creation index is complementary to the curation index, we only report the latter.

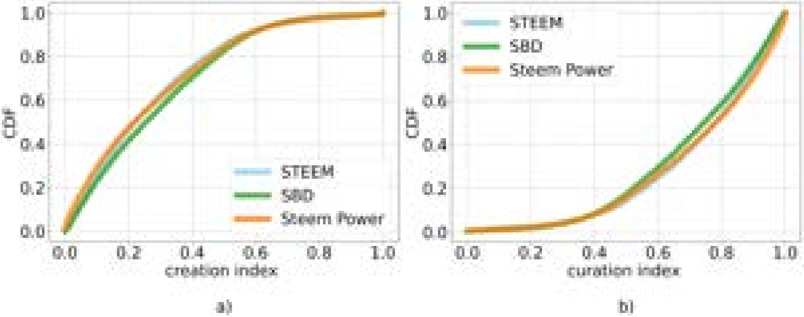


Fig. 5.6: **Curation index distribution.** *Cumulative Distribution Function of the curationindex, for each of the three currencies. On the x-axis: the creation index, which measures how much a user relies more on curation operations. On y-axis: Cumulative Distribution Function — CDF.*

actions represents a further strategy for hubs acting within the rewarding system. In Fig. 5.7 we report the relationship between the total amount of social and financial actions for the hubs separately identified for the three types of token.

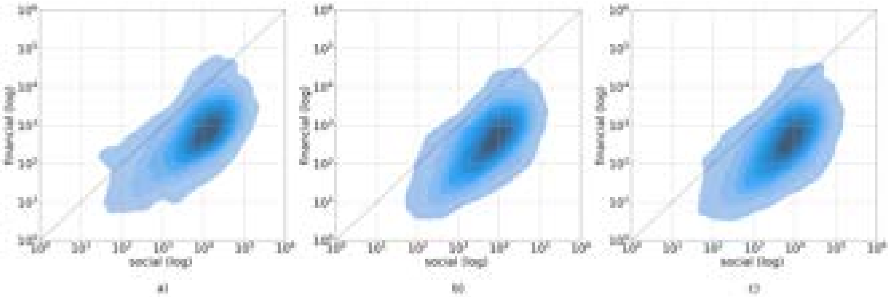


Fig. 5.7: **User's social and financial activity, for each currency.** *KDE scatter plots relating social and financial activities (number of votes) and creation (number of posts and comments)for each of the three currencies. On the x-axis: social actions. On y-axis: financial actions. Darker colored areas correspond to higher density.*

Through scatter plots, we can see if there is a skew towards either one of them. As for the previous case, we look at the differences between currencies. Here the difference is less marked between the currencies. Then, we visualize the distributions of social and financial indexes for the hubs in Fig. 5.8. The distributions of the two indexes show that users mainly rely more on social actions to gain tokens. This observation is in line with the previous one, but we do not observe differences between different currencies.

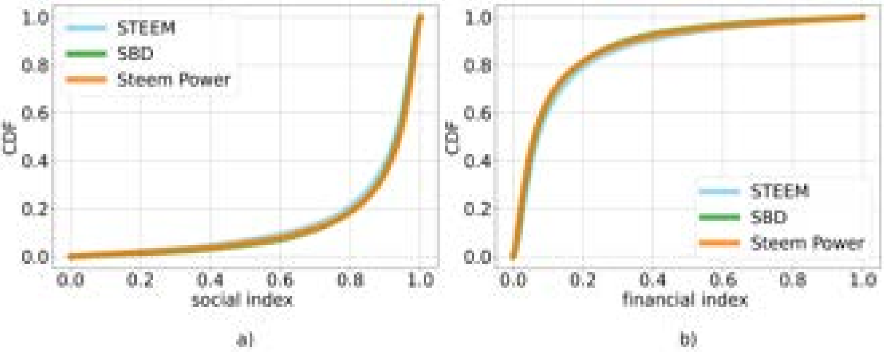


Fig. 5.8: **Social and financial indexes distribution.** *Cumulative Distribution Function — CDF — of a) $socialindex$ and b) $financialindex$, for each currency. On the x-axis: $socialindex$ and $financialindex$, which measure whether a user relies more on social or financial actions, respectively. On y-axis: CDF.*

Correlation Analysis

Finally, we conducted an analysis of the correlation between the rewards and the above four indexes:

1. *creation*,
2. *curation*,
3. *social* and
4. *financial*

. The outcome of the analysis has been reported in Table 5.4, where we take into account the rewards gained through different tokens, separately. In general, we observe low correlation values across the different combinations of indexes and rewards. In fact, we find an absence of correlation with the creation factor, for all tokens. If we observe the curation operations, the correlation is higher than in the previous case, but correlation coefficients are still low. That is in

Table 5.4: **User reward hubs.** *Correlation of currency and indexes.*

	STEEM	SBD	Steem Power
operation creation	0.07	0.08	0.08
curation	0.15	0.22	0.24
social	0.13	0.18	0.20
financial	0.02	0.03	0.11

line with the above observations on the scatter plots, confirming the usage of curation operations to gain rewards by some hubs. We have a similar situation when we consider the correlation values between social and financial actions: correlation values are close to zero, suggesting a lack of linear correlation.

5.7 Discussion and Conclusions

The idea of Web3 revolved around decentralization solutions is becoming one of the most promising responses to the over-centralization of Web 2.0. In Web3 decentralization is mainly reached through different blockchain technologies which aim at supporting nowadays online services and platforms and promoting novel paradigms such as decentralized finance — DeFi — or self-sovereign identity. Online social networks and media, services leading the Web 2.0 landscape, are now moving towards decentralized solutions as highlighted by the rise of many different blockchain-based social networks — BOSNs. These social platforms replicate all the social functionalities that have facilitated online relationships in the past, meanwhile introducing novel mechanisms to overcome issues related to data monetization, content quality, misinformation, and censorship. As for the former two points, blockchain technologies are strongly coupled with rewarding and voting systems that, from one side, allow users to gain rewards from their content and, at the same time, promote the creation of high-quality content. In the BOSN landscape, Steemit is the seminal project and was one of the most widespread platforms. In fact, it includes the most representative features of blockchain-based social networks:

1. contents are evaluated by users through social actions;
2. the rewarding system incentives the production and the promotion of high quality or highly appreciated content; and
3. rewards are paid with an exchangeable cryptocurrency, whose value can fluctuate over time

.

The above features make Steemit, and in general BOSN, a complex cyber-physical system where social, economic, and financial layers are strictly intertwined and influence each other. Indeed, users' social actions also have an economic explicit impact measurable through the amount of gained tokens. This strict interplay among the cryptocurrency market, the network-based strategies to gain more STEEM, and the rewarding mechanism has led to the hypothesis that economic/financial factors, such as the price of the STEEM cryptocurrency, may influence the social network supported by Steemit. In fact, we found evidence of the influence of the STEEM cryptocurrency over users' actions (*RQ1*). More precisely, we found higher values of correlation for social actions, but lower impact on financial actions. Among social actions, the cryptocurrency price strongly correlates with operations — such as "follow" or link creation — which shape the structure of the Steemit social network. So, we can reasonably state that Steemit's social network has been partly shaped and driven by the trend of its cryptocurrency through the rewarding and voting mechanisms the platform has implemented. On the contrary, we do not observe a great influence of user actions on the cryptocurrency value, since correlation lags between STEEM cryptocurrency and actions are always positive. So, users seem to adapt their behavior to the cryptocurrency, whereas it seems that external events have more influence on the cryptocurrency, e.g. the price of Bitcoin.

As we have seen that the cryptocurrency value has an influence over actions, we tried to detect trends or characteristic behavior of users. To this aim, we focused on hubs — the most successful users in terms of rewards — trying to understand their strategies to gain more rewards. We found some differences in users' social behavior (*RQ2*): hubs for Steem Power show differences in the behavior, with higher levels of curation actions w.r.t. the other currencies offered by Steemit. The difference is in line with the purpose of the tokens: it becomes more effective for those who possess more Steem Power to curate, instead of spending time and effort creating new content. However, we did not find significant differences among currencies, when we considered social and financial actions. It is interesting to notice that the purpose of the currency seems to influence user behavior, suggesting that the type of currency should be considered in the analysis.

To sum up, while this Chapter does consider only one platform, it would definitely be interesting to extend the study to more platforms. Indeed, the analysis of rewards focuses on the hubs, but it could be interesting to explore the decisions of other categories, i.e. most influential users on the social side or people mostly involved in data validation — witnesses — and rule-making process — project developers. Despite these limitations, we were able to obtain important insights into the interplay of cryptocurrency and network activity: we showed that external events and cryptocurrency value have a strong impact on users' activity and behavior. These insights lead to other research questions, as there is still limited understanding of the impact of external events on users' activity and social network evolution. The study of network evolution with external events and co-evolution of networks can be improved by focusing on currency-related events, cryptocurrency growth and drop, news or announcements related to currency, in addition to disruptive events like hard forks.

Chapter 6

Cooperative behavior in complementary currencies

6.1 Introduction

The Sustainable Development Goals by the United Nations [75] have incentivized the good use of ICT and emerging technologies in many fields and scenarios. Many systems for sustainable economic development are now relying on a digital form that makes them more accessible and provides access to new functionalities. A very interesting example of such systems is complementary currencies (CCs), i.e. cooperative currency systems that support national economies [26], and studies show that they actually boost local economies [83], address the issues of national currencies [20] or promoting the growth of industries[146]; moreover, they may achieve a positive impact on social sustainability as well, by increasing trust, expanding social networks and fostering social inclusion[27]. One of the most recent interesting uses, which attracted a lot of attention due to recent economic and social shocks, was the use of CCs in the field of humanitarian aid. While there are qualitative studies on the design principles and impact [27, 28, 29] of CCs for humanitarian aid, data-driven, and quantitative investigations are limited and many aspects are still unexplored. For example, cooperative behavior in these systems is a key factor, as CCs are often born out of cooperation among members that face a period of crisis, and communities use them to sustain themselves and support members in need during periods of crisis or instability[27]. Often, CCs have the objective of creating bonds of reciprocity and fostering social integration and inclusion [28], which should lead to increased cooperation. And yet, there are still key aspects of cooperative behavior, that are still unexplored: for instance, understanding how cooperative behavior is affected by other external factors, such as changes over time and geographical location. Another understudied aspect is the role of CCs during a period of crisis like the COVID-19 pandemic. And finally, there is a lack of studies on cooperative behavior. While we have a few works highlighting scenarios where CCs have been useful in times of economic crisis [27], only a few cover CC activity in the pandemic period: a few examples are studies on a Polish CC [86], or in Brazil [85] and Kenya [20].

In this Chapter, we study different aspects of cooperative behavior by focusing on group accounts to understand currency movements and cooperation patterns. We also examine how cooperative behavior is influenced by different external factors such as the COVID-19 pandemic and how geographical location can influence cooperation patterns. As a case study, we focus on the Kenyan CC Sarafu [31] to investigate these aspects. Sarafu is a noteworthy example of a CC that went digital to address several needs, become more accessible, and improve the system with new features. Sarafu has some very interesting characteristics: *i)* it is one of the first blockchain-based CC projects, that, like other Blockchain for Good projects, relies on blockchain technology for transaction processing, *ii)* it is a CC that was relied upon by Red Cross Kenya to successfully deliver humanitarian aid during the COVID-19 pandemic [20], and *iii)* Sarafu further enhances the cooperation of organized groups of individuals, by implementing a special type of account, namely *group account*: this type of account is handled by a group of users to save money and help members in need. Group accounts are an innovative feature, unique to this CC system, that makes Sarafu the best case study for the analysis of cooperation. We conduct our analysis on a dataset of currency transactions [30] during the COVID-19 pandemic. We analyze monetary flows in the transactions network, to monitor the following aspects: *RQ1)* the impact of cooperation groups and how it changes over time as we consider different pandemic situations and restrictions, *RQ2)* how cooperation groups allocate and redistribute resources, considering their *business type*s (such as "food", "farming", etc.), *RQ3)* the impact of geographical location in cooperative behavior, and *RQ4)* the interplay between the geographical location and how users or cooperation groups allocate and redistribute resources.

To answer our research questions, we model currency transactions as a temporal network, that is able to represent the economic ties between users. In addition to transaction networks, we rely on Sankey diagrams to study monetary flows between users and their consumption profiles [147] based on user information, i.e. types of accounts, their *business type*s, or geographical location.

Our analysis has highlighted some interesting findings. First, group accounts have a crucial role, as they are few (0.38%) and yet handle a significant amount of transactions (36%); moreover, their importance even increases over time, as the amount of money spent by these accounts increases significantly over the observation period (*RQ1*). Second, we also found that the allocation of resources by cooperation groups changes the observation period, as we observed variations over the categories of products of interest (*RQ2*). Third, we observed that while cooperation is important across different geographic locations, not all areas relied immediately on group accounts (*RQ3*). Fourth, we found an interesting interplay between geographic areas and the allocation of resources: geographical areas are characterized by their own categories of interest, with urban and periurban areas showing some similarities; and in some areas, the spending/funding of group accounts is much more significant compared to other categories (*RQ4*).

The Chapter is organized as follows. In Section 6.3 we introduce the main research questions we focus on. In Section 6.4 we describe the Sarafu dataset and its preprocessing. The approach for modeling, extracting, and analyzing the transaction networks and their projections is presented in Section 6.5. Section 6.6 reports the main findings on the role of group accounts in supporting cooperation, the changes in the usage of Sarafu during the pandemic period, and the impact of geographical location on cooperative behavior. Finally, Section 6.7 concludes the Chapter, pointing out possible future works.

6.2 Related work

Sarafu and its impact are described in a few works in the literature. The GE Foundation provided an anonymized dataset for researchers [30], that covers a year and a half of user transactions. Mattsson *et al.* [31] have released a dataset paper, providing important context and background on Sarafu. The dataset has been used to study the program's success: Ussher *et al.* [20] presented an accurate description of CCs, the Sarafu project history, and an analysis of the dataset. Mqamelo [148] investigated the impact on people's welfare and local economic engagement, while Mattsson *et al.* [130] proposed an analysis modeling the entire dataset through a static network structure: their analysis highlights that money circulation is highly modular, geographically localized and occurring among users with diverse jobs. Clark *et al.* [84] rely on user information to perform simulations of the performance of the economic system using network-based complex systems model of subpopulation interactions. In our previous work [131], we conducted a preliminary analysis focused on cooperation behavior, where we highlighted the presence of cooperation patterns and the importance of group accounts. In this Chapter, we focus on the analysis of cooperative behavior by leveraging the geographic information available. More precisely, we focus on:

- how cooperative behavior impacts the allocation and redistribution of resources;
- the impact of geographical location on cooperative behavior; and
- the interplay between the geographical location and the allocation and redistribution of resources

.

6.3 Research questions

In Kenya, persons in need would frequently turn to informal saving organizations known as *chamas*[1] for assistance [20]. *Chamas* are savings groups usually composed of 15–30 people, often defined by a neighborhood, a shared occupation, or friendship and family ties [31]. Group members gather regularly at a fixed time of the day to pool their savings together and discuss the possibility of loans to other fellow members [149]. Essentially, it is a saving and lending scheme with no or small interest rate [150]. To facilitate the actions of these cooperation groups, the Sarafu system implements a particular type of account called *group account*. These *group accounts* were given to *chamas*, allowing them to save and lend Sarafu tokens like they would for the standard currency. Therefore, *group accounts* are the most crucial part of the analysis: the higher the amount of currency managed by group accounts in Sarafu, the higher the amount of group saving and lending, and consequently, we have higher the cooperation. As a result, *group accounts* enable an effective examination of cooperation patterns since they support and highlight cooperative behavior that could not be properly evaluated in other CC systems.

The essential feature of cooperative behavior that we investigate is *how cooperation behavior is impacted by a crisis*, such as the COVID-19 pandemic, and to what extent cooperative behavior is influenced by other factors, such as geographic location. We can summarize the key aspects that we aim to investigate through the following research questions:

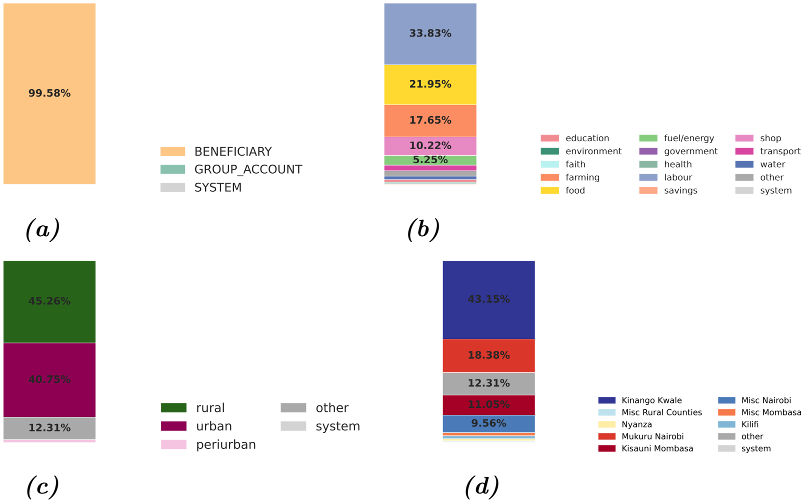
[1] “Chama” is the Kiswahili word for “group”

Research question 1 (RQ1): To what extent are cooperation groups used as a supporting tool for Sarafu participants? To what extent do the COVID-19 pandemic and the pandemic mitigation strategies impact the importance of cooperation groups?
Research question 2 (RQ2): How do cooperation groups allocate and redistribute resources? Does the allocation of resources by cooperation groups change over time?
Research question 3 (RQ3): What is the role of geographical location on the redistribution of resources? How does the geographical area impact cooperation groups?
Research question 4 (RQ4): Is there any interplay between the behavior of users and cooperation groups and the geographical location?

6.4 Data Preprocessing

To answer our research question, we rely on the Sarafu dataset presented in Section section 2.4. For the analysis, we leverage both transaction data and user attributes, with the latter being especially important as they enable us to distinguish the cooperative accounts as well as the geographical characteristics.
Data preparation. It is worth noting that, since we are interested in transactions involving actual users and group accounts, we opted to exclusively investigate transactions where at least the source or the target are accounts of the *beneficiary* and *group account* categories. We consider all the available transactions, except for the last 5 days of January 2020, since they are characterized by a set of preliminary transactions that served to migrate pre-existing accounts from the prior system [31]. Furthermore, because a few accounts contained inconsistent information, a preprocessing step was necessary. For example, only group accounts should have *business type* set to *savings* according to the information in [30]. However, in our study, we observed certain beneficiary accounts were set to *savings*, which should not have been the case. In the analysis, we do not take this subset of inconsistent accounts into consideration. Moreover, there are some group accounts associated with *business type* values other than *savings*. We opted to set their *business type* to *savings*. Similarly, we made sure that all the accounts used by GE staff (*Token Agent*, *Vendor*, *Admin*) have their *held role* set to *SYSTEM* and all their attributes (*business type*, *area name*, *area type*) to *system* as well. In the end, we consider 54807 users and 919930 transactions.
Users' attributes distribution. Fig. 6.1 depicts the distribution of the user attributes. As shown in Fig. 6.1a, the majority of users are standard accounts

(*beneficiary*, 99.5%). In terms of *business type* (see Fig. 6.1b), a large fraction of users (88.75%) has one of the following five *business types*: labour (33.8%), food (21.2%), farming (17.6%), shop (10.2%) and fuel/energy (5.2%). In Table 2.2 we reported the description provided by [30] for each possible *business type* value. In terms of geographic information, the majority of users are separated into rural (45.3%) and urban areas (40.7%), as shown in Fig. 6.1c. When we consider the area names (Fig. 6.1d), the rural region *Kinango Kwale* is at the top, followed by some urban areas *Mukuru Nairobi*, *Kisauni Mombasa*, *Misc Nairobi*. It is to be noted that area types and names are assigned by the GE staff after a standardization process derived from user-provided names [31].



(a) *(b)* *(c)* *(d)*

Fig. 6.1: *Distribution for the main user attributes, in order: a) held role, the account type, b) business type, user's economic activity, c) area type, and d) area name, which are derived from the location provided by the user.*

6.5 Methodology

Modeling. In general, transactions can be modeled as a set of tuples $I = \{(u, v, t, a)\}$ where u and v are users that traded tokens: user u transferred to user v an amount a of Sarafu tokens at time t. Transactions over a time interval $[t_0, t_1]$ can be modeled as a temporal network [99]. Therefore, given the interval $[t_0, t_1]$, the set I can be transformed into a weighted directed graph $\mathcal{G}_{[t_0,t_1]} = (V, E, X, W)$, namely a transaction network, where:

- V is the set of users,
- E is a set of directed weighted links $(u, v) \in E$, two users are linked if they performed at least a trade in the time interval $[t_0, t_1]$,
- X is a $|V| \times f$ matrix of user attributes, where f is the number of available attributes,
- W is a weight matrix representing the flow of money. In fact the weight $w \in W$ of an edge $e = (u, v) \in \mathcal{G}_{[t_0,t_1]}$ is the sum of the amounts sent from u to v during the time interval $[t_0, t_1]$.

Defining a sequence of these *transaction networks*, we may investigate changes in network structure over time [36] as well as total monetary flow in different time intervals.

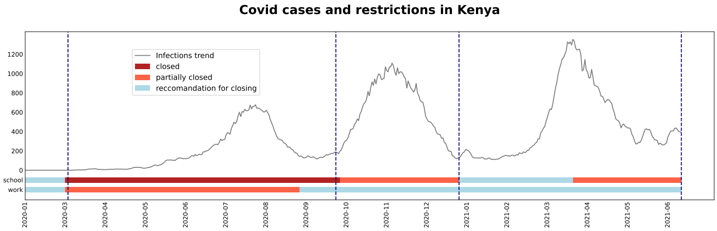


Fig. 6.2: *COVID-19 cases and restrictions in Kenya. As the number of cases (blue line) varies over time, we can observe different restrictions over time (closed, partially closed, recommended closing) for both school and work, during the pandemic period. The figure from [131] is a reworking of data published by Reuters COVID-19 Tracker at [151].*

Analysis. To answer our research questions, in addition to transaction networks, we also rely on Sankey diagrams: Sankey diagrams are an effective visualization tool for many different types of flows such as material, traffic, water, and money [152]. Given a transaction network, we can derive different types of Sankey diagrams that enable the analysis of monetary flows. The construction can be performed by aggregating currency values on incoming and outgoing edges, while we consider user attributes. Therefore, through Sankey representation, we can perform various analyses, as nodes can represent different user attributes — i.e. the types of accounts or the *business types*, or the

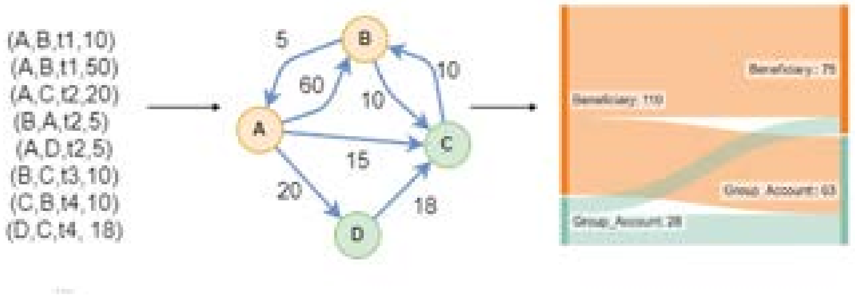


Fig. 6.3: *An example outlining the proposed methodology. Starting from the transactions, in format (sender, receiver, amount, timestamp), we filter them on the timestamps to obtain a subset for the time period of interest. Then, we construct the transaction network. Relying on the weights and attributes of the transaction network, we can aggregate to construct the Sankey diagrams. In the example of the transaction network, nodes are colored according to the type, while the weights on links correspond to the amount of tokens flowing from the source to the destination.*

user location — while the directed links indicate the cumulative flows between sources and targets.

A recap of the proposed methodology is shown in Fig. 6.3. We first build a transaction network out of the complete dataset. Then, we integrate the dataset with additional contextual information about COVID-19 cases and restriction policies by the Kenyan government, as illustrated in Fig. 6.2. Using such information, we divided transactions into four time periods, based on the different restriction policies in effect. As a result, we can apply the aforementioned methodology to construct four transaction networks, one for each period. Then, we analyze the transaction networks and understand the differences between different periods.

To answer RQ1 we need to comprehend the importance of group accounts, so we analyze Sankey diagrams with nodes representing the “role" of the account,*beneficiary*,*group account* or *system*. Then, we assess the importance of cooperation in the pandemic scenario using group accounts and we analyze changes over time.

In order to answer RQ2 we need to understand the categories of users that are involved in exchange group accounts: we focus on the group accounts' spending behavior by looking at the categories group accounts are spending on; and we analyze funding, by observing the categories of users who send money to group accounts. We observe the flows both from a static and over-time perspective to obtain a deeper understanding of how COVID-19 cases and restriction policies have influenced users' and cooperation groups' behavior.

For RQ3, we assess the impact of geographic location on user behavior as well as the possible impact on cooperation. Therefore, we first analyze the flows of money across geographic regions using Sankey diagrams that take into account the nodes' geographic information, i.e. their *area name*. Then, we concentrate on cooperation groups, using Sankey diagrams centered on group accounts to study both spending and funding behavior, i.e. which geographic regions get money from group accounts and which give money to group accounts, respectively. We observe both the static and over-time flows, using the geographical information of the *area name* for both users and group accounts. Moreover, we leverage group accounts' geographical area information and users' *business type* to describe the categories of funding and spending in each geographical area. We generate multiple Sankey diagrams, that provide an effective overview of the differences across geographical areas. The same methodology, applied over time, will allow the observation of spending and funding behavior changes in each geographic area.

Finally, we address RQ4. We deal with any potential relationship or interplay between the behavior of a user/cooperation group and their geographic location. In other words, we want to verify if users in a specific category, such as "food," behave differently based on the geographical location, and if such behavior changes over time and vice-versa we would like to see if in a given geographic area, users prioritize different categories and whether their priorities vary over time. A crucial aspect must be kept in mind when studying changes over time: certain changes might simply be due to an increase or reduction in the number of users in a specific category or geographical location. Therefore to highlight changes that are not simply a byproduct of the distribution of users we must account for the changes in the population the so-called *population drift* [153] or *population turnover* [154]. This is an important issue known in data science and computational social science literature: when studying behavioral drift, i.e. changes in how people are using a system, we should always monitor population drift as well as system drift, i.e. changes in the system itself. We highlight changes over time relying on stacked area plots: these plots dedicate a colored area to describe the variation of different time series, allowing us to visualize changes over time, but at the same time they allow the comparison of different data without overlapping. We study different quantities based on the category or geographical area of users: we focus on spending (the total amount spent by users), funding (the amount received by users), or the number of active users. Therefore, for a given geographical area we can plot the categorical variation, an area plot that separates quantities based on the user category. By comparing the *categorical variation* of each geographical area, we highlight potential differences or characteristics of a given area. Vice-versa, we analyze and compare each category through its *geographical variation*. These plots also are suitable to highlight how cooperation groups are affected: we only need to focus on the category of group accounts (*savings*) to the other user categories. The same methodology is used to compare funding, i.e. the money that is sent to the category or area. The methodology allows us to monitor population drift, as we also keep track of the number of active users, allowing us to exclude variations due to population drift. In addition, we make sure to consider system changes based on the time periods observed, accounting for the system drift when we make our observations.

6.6 Results

In this and the next sections, we have applied the methodology discussed above to the Sarafu dataset, which is modeled as a sequence of transaction networks whose characteristics are displayed in Table 6.1.

Table 6.1: *Transactions and transaction network statistics over the entire dataset and in different periods. The periods are selected based on changes in the mitigation policies and restrictions adopted during the pandemic period (see Fig. 6.2).*

Start	End	Active users	Edges	Transactions
2020-02-01	2020-03-15	4218	10449	14486
2020-03-15	2020-10-01	39410	162226	411191
2020-10-01	2021-01-01	41472	91155	182013
2021-01-01	2021-06-16	47928	131000	306855

Transaction volume had grown substantially over time, with a notable increase in active users in the second period when the pandemic reached Kenya and the Red Cross made an effort to promote Sarafu to respond to the crisis [20]. We see an interesting difference when we consider only the *standard* transactions between beneficiary and group accounts as we did in our previous work [131] and as shown in table 6.2. We observe that the quantity of unique

Table 6.2: *Transactions and transaction network statistics in different periods, but considering only standard transactions between beneficiary and group accounts, in the same periods as in the previous Table 6.1.*

Start	End	Active users	Edges	Transactions
2020-02-01	2020-03-15	3802	7325	10744
2020-03-15	2020-10-01	28070	96266	251594
2020-10-01	2021-01-01	7030	22872	63262
2021-01-01	2021-06-16	13960	35225	85026

active users of beneficiary and group accounts diminishes in subsequent time periods, but still with a large number of active users in the last time period. However, as we consider all the transactions in the dataset, we can observe that a bigger portion of users can be considered active, as they were involved in at least one system-related action in subsequent time periods.

Impact of cooperation

As indicated in Section 6.3, our first research topic focuses on the role of group accounts in money flows. Fig. 6.4 portrays the Sankey diagram of money transfers constructed using the entire dataset. Due to the distinct nature of group accounts (which account for 0.42% of all users only), the percentage of money flows involving them is significant (36%). This result clearly emphasizes

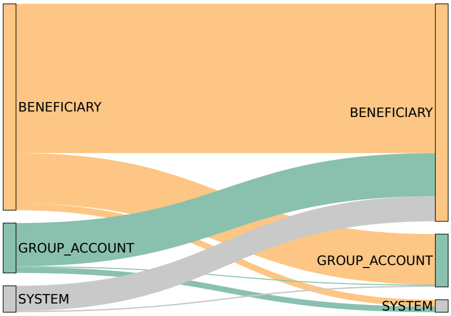
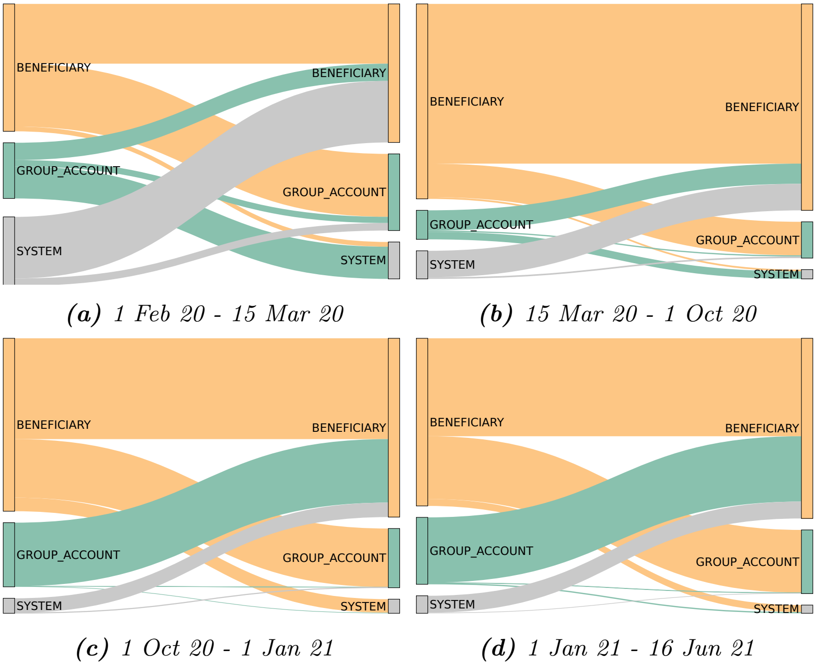


Fig. 6.4: *Study of the importance of group accounts: Sankey diagram of monetary flows from group accounts to beneficiary accounts and vice-versa*

the importance of group accounts, which are few and yet handle over one-third of all currency transactions.



(a) *1 Feb 20 - 15 Mar 20*

(b) *15 Mar 20 - 1 Oct 20*

(c) *1 Oct 20 - 1 Jan 21*

(d) *1 Jan 21 - 16 Jun 21*

Fig. 6.5: *The impact of group accounts over time, as measured by monetary flows. We display the monetary flows from group accounts to beneficiary accounts, and vice-versa, for each period.*

We proceed with the study of the role of group accounts and their spending behaviors, by identifying potential changes during different pandemic phases. Fig. 6.5 shows the money flows grouped by the held roles we consider: beneficiary and group accounts. It is clear from the Sankey diagrams that the impact of group accounts changes over time. The first observation concerns the rise of flows from group accounts beginning in the third period: although the percentage of flows from group accounts to *beneficiary* users, in the first two periods, is on average 7%, it rises to 25% in the last two periods. As noted in [31], group accounts are able to exchange tokens for Kenyan Shillings, the importance of the functionality is observable through the flow from *group ac-*

counts to *system* accounts. A second interesting observation can be made by observing the period characterized by the most stringent mitigation policies. In fact, the second period corresponds to the first wave of COVID-19 cases, and it is also the most different period, because of its outlier percentage of transactions among beneficiary accounts. In this situation, the complete closure of schools and the partial closure of workplaces — both of which impose significant restrictions on mobility and sociality — may have encouraged private and direct transfers of money, bypassing the use of group accounts. On the same note, if we observe the number of group accounts in Table 6.3, we can see how more group accounts are established. In the remaining periods, the flows within beneficiary accounts remain almost stable (from 40% to 39% of transactions), whereas the percentage of operations from group accounts to *beneficiary* users grows (from 8% to 25%). Finally, the pre-pandemic period is the only one where beneficiary accounts exchange money with other beneficiary accounts and group accounts in balanced percentages. In fact, they only differ by 0.36% while in the other periods, the difference is consistently greater than 13%.

Table 6.3: *Number of group accounts over time, at the end of each period. The periods are selected based on changes in the mitigation policies and restrictions adopted during the pandemic period.*

Group accounts for each area	15 Mar. 2020	1 Oct. 2020	1 Jan. 2021	16 Jun. 2021
Kilifi	1	1	3	5
Kinango Kwale	56	73	73	78
Misc Mombasa	2	2	2	3
Misc Nairobi	8	9	9	9
Mukuru Nairobi	4	45	45	48
Nyanza	0	3	3	5
Kisauni Mombasa	0	0	0	61
Turkana	0	0	0	1
other	0	0	0	1
Total group accounts	71	133	136	211

Therefore, we can conclude that a) group accounts are few and yet handle a significant volume of currency; and b) their importance increases over time.

Cooperation groups funding and spending

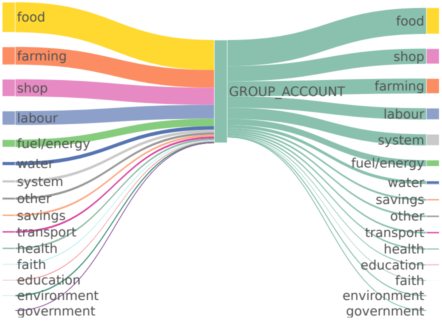


Fig. 6.6: *The importance and behavior of group accounts. Through a double Sankey diagram, we highlight group account funding and spending behavior. For funding, we show the categories of users that send money to group accounts, while for the spending behavior, we look at the categories of receiving users.*

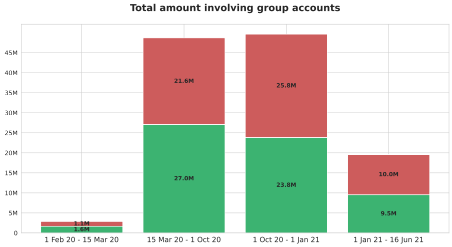
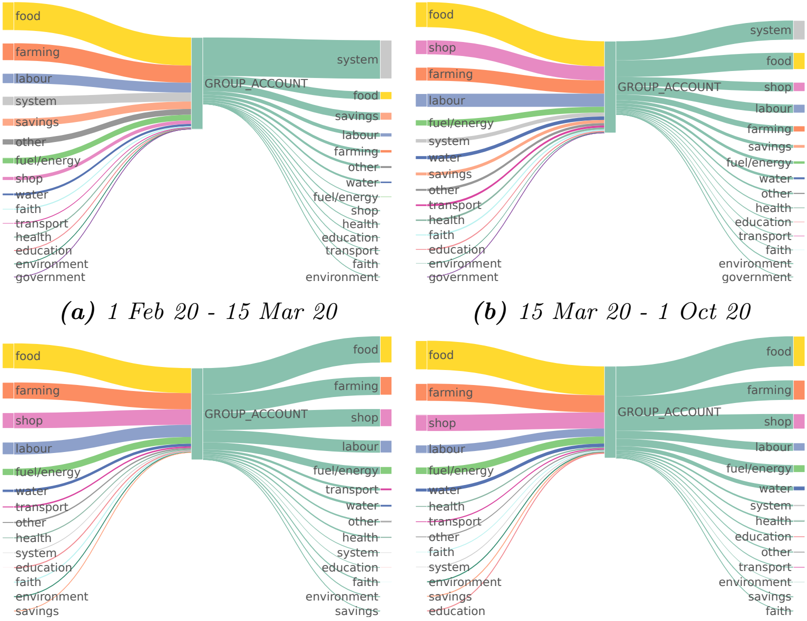


Fig. 6.7: *The total amount of money handled by group accounts for each period. For each period, the stacked barplot shows both incoming and outgoing money for group accounts. The amounts consider the transactions involving users, group accounts, and system accounts.*

Moving on to the next research question, we proceed with the study of group accounts and their spending behaviors. To get a deeper understanding



(a) *1 Feb 20 - 15 Mar 20*

(b) *15 Mar 20 - 1 Oct 20*

(c) *1 Oct 20 - 1 Jan 21*

(d) *1 Jan 21 - 16 Jun 21*

Fig. 6.8: *Group account funding and spending behavior, over time. For each time period, we have a double Sankey diagram, showing both funding and spending monetary flows. For funding, we show the categories of users that send money to group accounts, while for the spending behavior, we look at the categories of receiving users. Below each figure, we report the time interval.*

of the money flows from and to group accounts, we rely on a double Sankey diagram (see Fig. 6.6, where flows are grouped by *business type* of the *beneficiary node*. Fig. 6.6 shows that the most prevalent categories remain stable: the first four (food, farming, shop, and labour) account for 70% of the incoming operations to group accounts and 75% of the outgoing ones. Note that the ranking of the top categories is different from the general ranking over the whole dataset, depicted in Fig. 6.1b, allowing us to exclude that the ranking is just a byproduct of the distribution of users in the dataset. Indeed, when the *business types* are ranked not by frequency but by the percentage of flows (the relative amount of money involved in the transactions grouped by categories) we can observe that: *food* and *shop* categories gain importance (first and second place, respectively) in both directions while the, *labour* is less important (fourth position instead of first).

Fig. 6.7 shows the total amount of money of all transactions transferred to and from group accounts throughout each period. In addition to the ratio of incoming to outgoing amounts, the magnitude of money spent has risen over time. In fact, the central periods have a substantially higher total than the other ones.

We can further investigate the spending behavior of beneficiary and group accounts throughout specific pandemic periods through the Sankey diagrams shown in Fig. 6.8. At first sight, it is noticeable that the incoming and outgoing relative amounts vary over time. Initially, there is a propensity to store money on group accounts, which spend only a small percentage of the income (the outgoing total is only 49% of the incoming total). Over time, the percentage of outgoing over incoming amount grows so much that in the third period, the outgoing amount is actually higher than the incoming. Another interesting observation from Fig. 6.8 concerns the order of the categories. First, the *saving* category presents an anomalous behavior: we observe a great flow in the first period (even if in the general distribution shown in Fig. 6.1b this category is just the third last), in the successive periods it loses some positions and then becomes even less frequent. On the contrary, the *shop* category begins at a very low ranking position (after the first six) but moves up in the top three from the second period. With the exception of the *savings* and *system* categories, the top six slots of the ranking are always taken by the first eight categories in the overall distribution. Furthermore, the *food* category is always on the top, with a large lead from the second one. It is also worth noting that the categories generally keep the same position with incoming and outgoing transactions.

So, we can conclude that: a) spending behavior is not just a byproduct of the distribution of users; and b) the allocation of resources by cooperation groups changes over time, as users adjust to the Covid-19 pandemic, mitigation policies, and changes in the Sarafu system.

Geographical location and cooperation groups

We also focused on the role of geographic information in cooperative behavior. As a tool, we rely on double Sankey diagrams where we observe the flows to and from group accounts, grouped by the geographic area of the *beneficiary* users. Moreover, we also consider the geographical area of the group accounts for a more expressive representation of the flows.

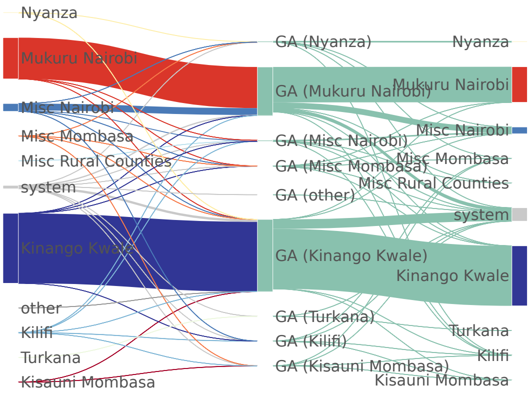
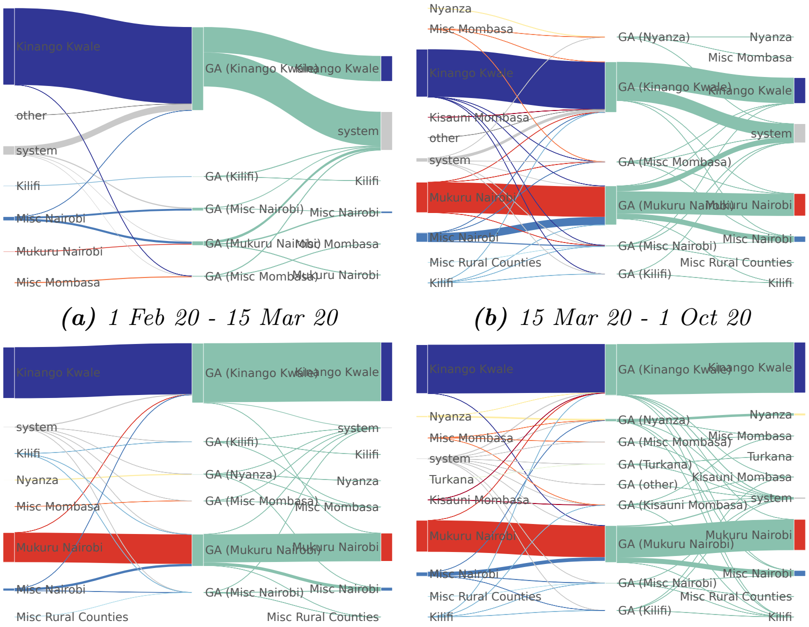


Fig. 6.9: *Monetary flows across different geographic regions. Through a double Sankey diagram, we highlight group account funding and spending in different geographic regions. For funding, we show the area type of users that send money to group accounts, while for the spending behavior, we look at the area type of receiving users.*

We obtain the Sankey diagram in Fig. 6.9 when we consider the flows based on the *area name* of the *beneficiary node*. We can observe a money flow to group accounts from all areas. While there are flows among different geographical areas, most of the circulation is local, in line with the observation in Mattson *et al.* [130]. We can observe that the biggest flows involve the top 2 areas in terms of overall users, i.e. *Kinango Kwale* and *Mukuru Nairobi* (as we noticed in the overall distribution in Fig. 6.1). However, the ingoing and outgoing flows of *Kisauni Mombasa* are less than those from *Misc Nairobi*, even though the former has fewer users.

We also analyzed how flows change over time as displayed in Fig. 6.10. We can see in the first time period, for *Mukuru Nairobi*, one of the main urban areas, there is almost no flow towards and no money received from group accounts; instead, in other areas, there is a reliance on group accounts right from the starting period. However, this trait changes during the second period, when cooperation groups increase their spending: the area of *Mukuru Nairobi* receives a comparable amount of money to the top one *Kinango Kwale*. However, in the last two periods, the gap between these areas increases again: while the flow to *Mukuru Nairobi* is similar, we observe an increase in the flow to the *Kinago Kwale* area. These changes in behavior for *Mukuru Nairobi* may be a direct consequence of the important growth coinciding with the effort



(a) *1 Feb 20 - 15 Mar 20*

(b) *15 Mar 20 - 1 Oct 20*

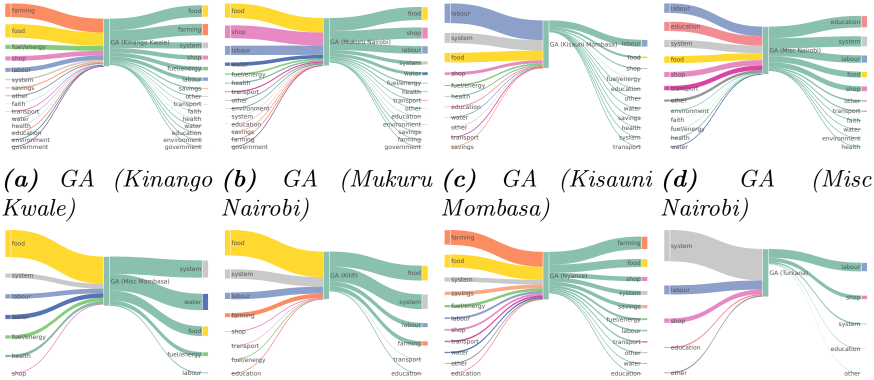
(c) *1 Oct 20 - 1 Jan 21*

(d) *1 Jan 21 - 16 Jun 21*

Fig. 6.10: *Monetary flows across different geographic regions. Through a double Sankey diagram, we highlight group account funding and spending in different geographic regions. For funding, we show the area type of users that send money to group accounts, while for the spending behavior, we look at the area type of receiving users. Below each figure, we report the time interval.*

by *Red Cross Kenya* to provide aid during the pandemic: while it shows the importance of Sarafu, it is not simply a change of behavior or use caused by the pandemic. Similarly, the flows seem to highlight the effects of the policy changes in the third and fourth periods, when users were incentivized to spend more by the GE Foundation [20].

Then, we leverage geographical area information associated with the group account and users' *business type* to describe funding and spending in each geographical area. The different Sankey diagrams, reported in Fig. 6.11, provide an effective overview of the differences across geographical areas. Overall, we can see how geographical areas tend to have different priorities. Except for the food category, which can be usually found among the top categories, the categories of interest vary between areas. In terms of overall flow, we can see how group



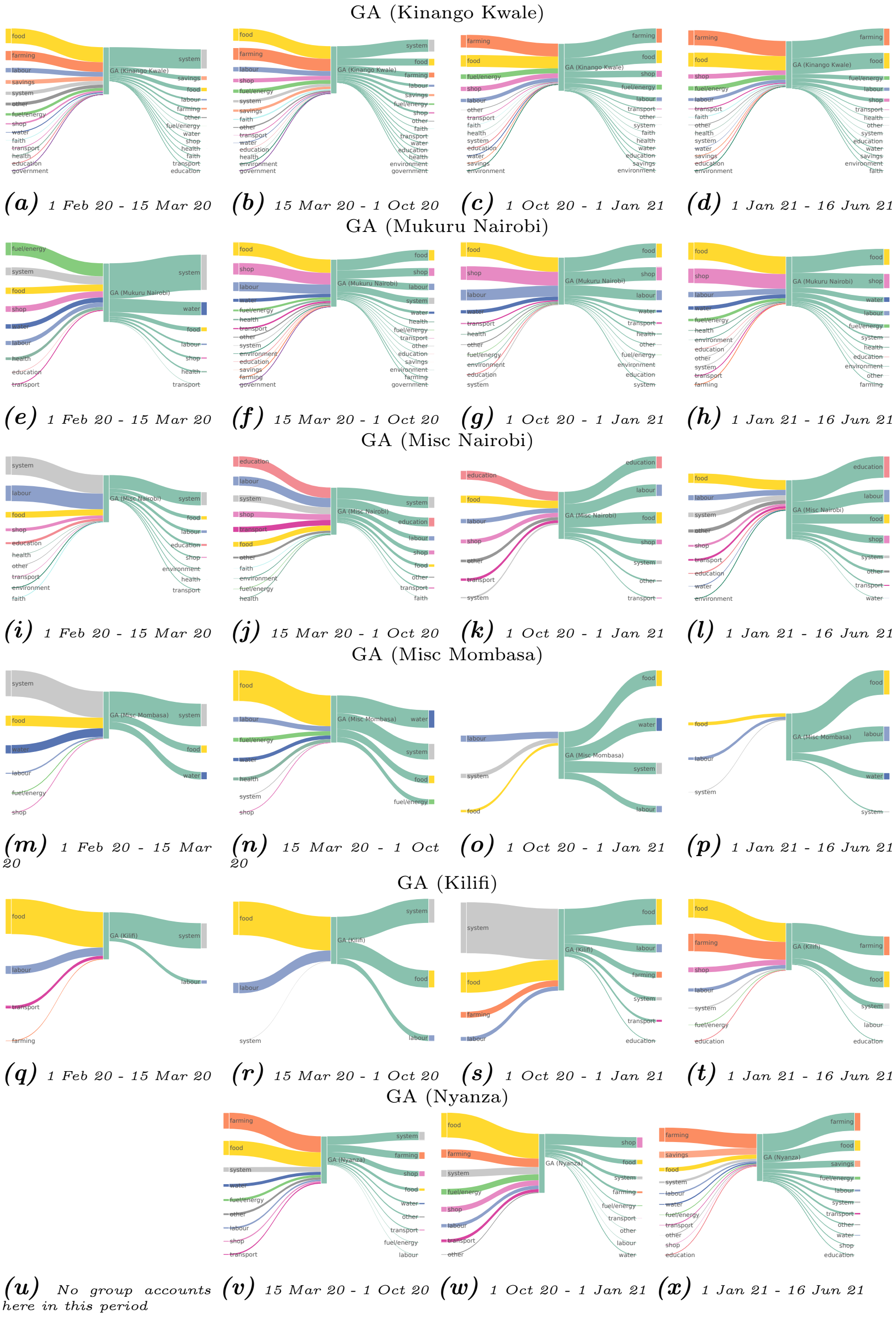
(a) *GA (Kinango Kwale)* ***(b)*** *GA (Mukuru Nairobi)* ***(c)*** *GA (Kisauni Mombasa)* ***(d)*** *GA (Misc Nairobi)*

(e) *GA (Misc Mombasa)* ***(f)*** *GA (Kilifi)* ***(g)*** *GA (Nyanza)* ***(h)*** *GA (Turkana)*

Fig. 6.11: *Monetary flows from group accounts to beneficiary accounts, and vice-versa, for each period, leveraging geographical information (area name) of group accounts.*

accounts in more established areas such as *Kinango Kwale*, *Mukuru Nairobi* have spent most of their tokens, while in smaller or growing areas, there was a tendency to accumulate tokens. An example is the *Turkana* area, established only in the last period, where we can see significant funding flows from *system* accounts, as new users join the platform. Another interesting insight is how the same categories are much more important in certain areas. For instance, in the area of *Misc Nairobi*, users, whose occupation is in *education*, are quite important in both funding and spending. Similarly, in the area *Misc Mombasa*, there are fewer categories and most of the tokens are spent on *water* users.

Finally, the same methodology can be applied over time, to provide additional insights. Fig. 6.12 supports an analysis of how spending and funding behavior changes over time in each geographic area. The subdivision over time highlights the differences in this area in the earlier period, where there was a bigger influx of money from the system, as users registered and obtained various bonuses for being active [31]. Similarly in funding, we can see a flow from *group accounts* to *system* accounts, as groups were relying on the exchange functionality. Similarly, we can observe how most of the areas tend to save money in the first period and increase their spending attitude as time progresses.



GA (Kinango Kwale)

(a) 1 Feb 20 - 15 Mar 20 *(b)* 15 Mar 20 - 1 Oct 20 *(c)* 1 Oct 20 - 1 Jan 21 *(d)* 1 Jan 21 - 16 Jun 21

GA (Mukuru Nairobi)

(e) 1 Feb 20 - 15 Mar 20 *(f)* 15 Mar 20 - 1 Oct 20 *(g)* 1 Oct 20 - 1 Jan 21 *(h)* 1 Jan 21 - 16 Jun 21

GA (Misc Nairobi)

(i) 1 Feb 20 - 15 Mar 20 *(j)* 15 Mar 20 - 1 Oct 20 *(k)* 1 Oct 20 - 1 Jan 21 *(l)* 1 Jan 21 - 16 Jun 21

GA (Misc Mombasa)

(m) 1 Feb 20 - 15 Mar 20 *(n)* 15 Mar 20 - 1 Oct 20 *(o)* 1 Oct 20 - 1 Jan 21 *(p)* 1 Jan 21 - 16 Jun 21

GA (Kilifi)

(q) 1 Feb 20 - 15 Mar 20 *(r)* 15 Mar 20 - 1 Oct 20 *(s)* 1 Oct 20 - 1 Jan 21 *(t)* 1 Jan 21 - 16 Jun 21

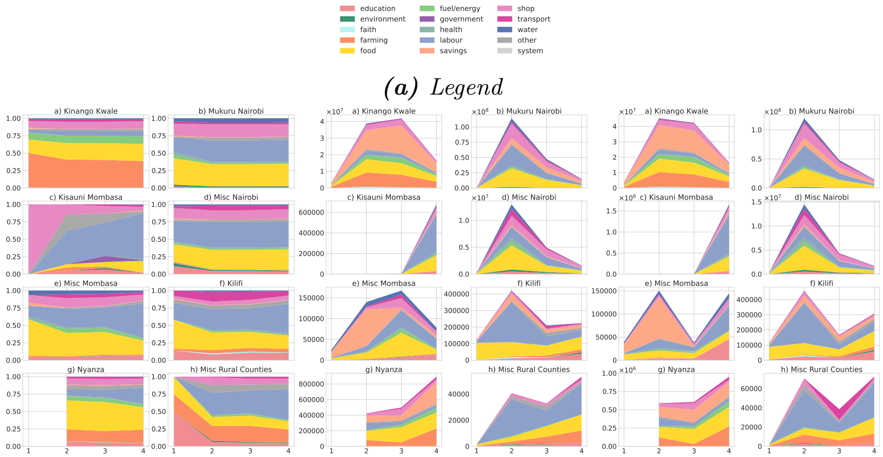
GA (Nyanza)

(u) No group accounts here in this period *(v)* 15 Mar 20 - 1 Oct 20 *(w)* 1 Oct 20 - 1 Jan 21 *(x)* 1 Jan 21 - 16 Jun 21

Fig. 6.12: *Monetary flows from group accounts to beneficiary ones, and vice-versa, for each period, using geographical information (area name) of group accounts.*

In conclusion, we notice that: a) both urban and rural areas rely on cooperation groups, and b) geographical areas are characterized by their own different behavior, with their priorities changing over time.

Interplay of funding and spending behavior with geographical locations

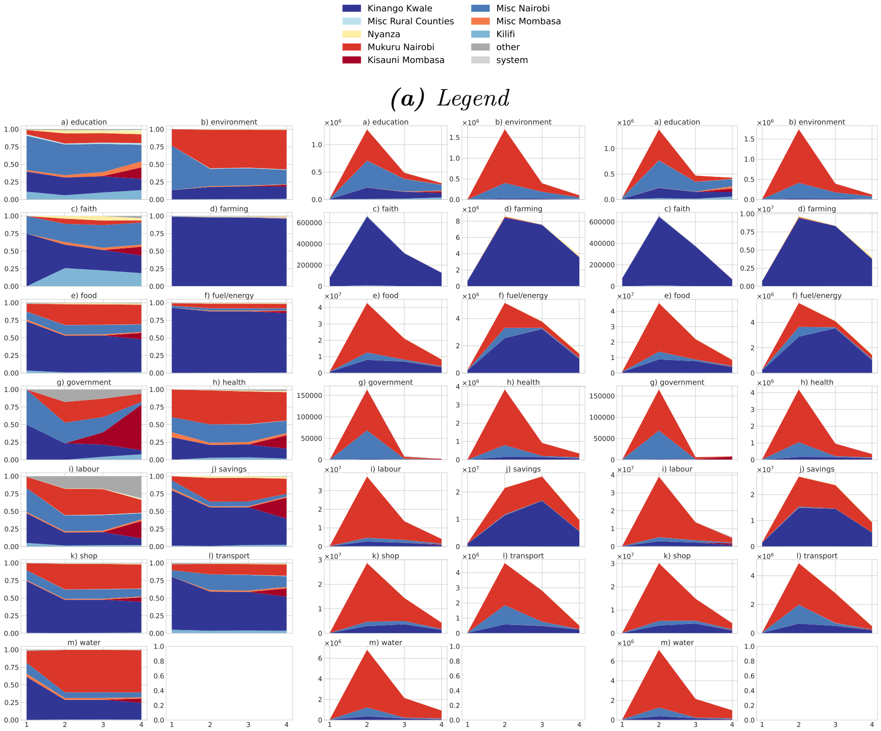


(a) *Legend*

(b) *Variation of active users* *(c)* *Variation of spending behavior* *(d)* *Variation of funding behavior*

Fig. 6.13: *Variation of a) active users b) spending behavior c) funding behavior in each of the areas, separated through category information. Colored areas represent categories. Please note that the Turkana area was omitted, as the project started only in the last time period.*

In this section, to answer RQ4, we analyze the interplay between categories and geographic areas and whether there is an impact on cooperation groups. We start our analysis from the plot for each area of its *categorical variation*, i.e. an area plot that separates quantities based on the user category. We monitor different important quantities: *i)* the number of active users, *ii)* spending (the total amount spent by users), and *iii)* funding (the amount received by users). Through the use of stacked area plots, we can visualize the variation over time for each area separated by category, as well as compare the overall volume changes, as also described in Section 6.5. While keeping track of



(a) *Legend*

(b) *Variation of active users*

(c) *Variation of spending behavior*

(d) *Variation of funding behavior*

Fig. 6.14: *Variation of a) active users, b) spending behavior, and c) funding behavior in each category group by geographical area. Colored areas represent the geographical areas.*

the distribution of active users, we are able to account for the problem of population drift [153] or population turnover[154], i.e. the changes in the population using Sarafu, as the system grows: we are able to identify whether we are observing an actual change in behavior or if it is more likely a byproduct of the user population changing. We represent the active users distribution in Fig. 6.13b (active users count normalized per time period, so we obtain a percentage/distribution), the users' spending behavior in Fig. 6.13c and users' funding behavior in Fig. 6.13d. We can observe that every area type has a very different profile. Focusing on the distribution of user categories in Fig. 6.13b, we can notice that the frequency of categories is not the same for all areas. As expected, in the rural area *Kinango Kwale*, the most frequent are *farming*, *food*, *shop*, *fuel/energy*. We can see that the *Nyanza* province is similar, with food more present; whereas the *Misc Rural Countries* has an important presence of education and more *labour* nodes. The *Turkana* Area has no variation values since the project started only in the last period. The urban areas have different distributions. Starting from the most populous one, *Mukuru Nairobi*, we can see that food and shop are still largely present. But, we have almost no *farming* and less *fuel/energy*, as well as higher *labour*. The area of *Misc Nairobi* is similar, while the two *Mombasa* areas show a few differences: *Misc Mombasa* shows the presence of more *labour* nodes, while *Kisauni Mombasa* has a bigger *shop* component, more *labour*, *government* and *other*. The peri-urban area (*Kilifi*) is also quite similar to the urban ones. So, even though every area shows some characteristic traits, we also find some similarities and common characteristics. Furthermore, when we observe the variation in spending (Fig. 6.13c) and funding behavior (Fig. 6.13d), we can see that the area plots tend to be quite different for each geographic area. Here, we only discuss the outcomes in the spending plots, since spending and funding plots are pretty similar, except for small variations in the sizes of the areas. Most of the geographic areas experienced a peak of spending/funding in the second period — the policies are stricter and most of them relied more on Sarafu during this period — and we observe a decline for some areas in the third period — policies have become less strict. Finally, areas tend to differ in the fourth period, with a few still decreasing, while most are showing growth. Some of those variations coincide with the variation of overall users, while others seem actual changes in behavior that happen independently from the number of users. For example, in the rural area of *Kinango Kwale*, most of the spending occurs by users of the categories *farming food* and *savings*, and their spending rises significantly in the second period. While the rise in *food* category is in line with the rise in the number of *food* users. For *farming*, and especially *savings* (the category of group accounts), this is not the case: the number of *savings* accounts remains just a small fraction, and *farming* users are actually dropping even though their spending volume rises. Similarly in the successive periods, we observe drops in spending and funding but they do not correspond to significant swings in the distribution. Similarly, when we look at the main urban area of *Mukuru Nairobi*, we can observe the growth of *food*, *labour*, and *shop* categories, but it does not coincide with a change in the distribution of users for those categories: there are different variations based on the geographic area that the distribution of user categories in that area cannot only explain. When it comes to cooperation groups, the split by geographic area shows that in some areas the spending/funding of group accounts is very significant compared to other categories: for example in the rural *Kinango Kwale*, *Nyanza*, and in the urban *Misc Mombasa*, the area of spending for *savings* covers a huge portion of the overall area plots, while in other areas it is not as huge, at least in comparison to the rest of the categories. Indeed, cooperation, while always present, is also dependent on the geographic area.

Finally, we analyze for each category its *geographical variation*: an area plot that separates quantities based on the users' geographic location. In this case, the stacked area plots visualize the variation over time for each category, with the measurements separated by category. Each area plot is focused on changes in either i) the number of active users ii) spending (the total amount spent by users), iii) funding (the amount received by users), separated by the users' geographical information, i.e. the attributes *area type* or area name. We present the users' distribution in Fig. 6.14b (number of users normalized per time period to obtain a percentage/distribution), the spending behavior in Fig. 6.14c and funding behavior in Fig. 6.14d. From Fig. 6.14b we can see that every category has different user distributions. Most noticeable is that farming and fuel/energy are mostly present only in one area. The others tend to be more distributed across regions. When we consider the spending variations in Fig. 6.14c, we can observe that in most categories, spending volume is dominated by the urban area *Mukuru Nairobi* and the periurban *Kilifi*. However, in some categories *faith*, *farming*, *fuel/energy*, and *savings*, *Kinango Kwale* is predominant: the total amount of flow surpasses the dedicated flow in the other areas by a large margin. But while for *farming* and *fuel/energy* is sort of in line with the changes in the overall distribution, for *faith* and *savings* it is not. In fact, it is interesting how faith nodes play such a huge role in one area only. Finally, in terms of cooperation groups (*savings*), we can see that spending and funding grow in all areas, in line with the previous observations.

According to these observations, we can conclude that a) geographical areas are each characterized by their own profile, with urban and periurban areas showing more similarities, b) in some areas the spending/funding of group accounts is much more significant compared to other categories, c) categories also have different profiles, and certain categories are only important in a subset of geographical areas, and d) cooperation groups maintain their importance in every area.

6.7 Conclusion

Our findings on group accounts suggest that this sort of account or similar mechanisms that promote cooperation could be useful for other humanitarian or community development projects: with this methodology, we could analyze currency flows to detect cooperation and coordination, and when absent, consider how to promote it. Moreover, similar cooperation enhancers could have an important role in other social development projects, and in general, in any setting where there is a strong need to foster cooperation for reaching social good. Finally, it would be interesting to understand if group accounts could be a catalyst of cooperation in other systems or scenarios: if so, the introduction of similar “institutional" cooperation accounts could be an effective solution for systems where there is a strong need to foster cooperation, a key factor in reaching social good and other sustainable development goals.

In addition, the proposed methodology could be used for the analysis of other currency systems, to analyze changes over time as well as to detect potential issues or anomalies. We have shown how our methodology effectively highlights the impact of external events as well as the effects of policies and organizational intervention. A similar study applied to other CC systems could provide invaluable information to administrators, policymakers, or even the government to leverage CCs, especially in times of crisis. In general, it can help to detect the strengths and weaknesses of a CC system, and how they should intervene. For example, the methodology proposed for the analysis of user behavior in terms of funding and spending categories could help define which users should be engaged for discussing issues, implementing changes, or evaluating the system’s performance. Moreover, we have shown how leveraging geographical information can distinguish the needs and priorities of a community: understanding the needs of people would allow better delivery of humanitarian aid. In fact, recognizing inequalities across should be important for effective management and decisions — making locality-based policies and incentives.

Overall, the resulting information from data-driven quantitative studies with this methodology could be especially beneficial in decision-making processes for current and new humanitarian aid initiatives, as well as currency systems in general.

Part IV

Modeling and prediction of user migration

Chapter 7

Modeling and predicting user migration

7.1 Introduction

Online Social Media (OSM) have become an important part of the life of more than half of the World's population[1], and nowadays they are among the most used web applications. People use social media for many purposes, including sharing their personal information, keeping in touch with friends and family, gathering information about the latest events in the world, and more. The current OSM landscape is characterized by competition to get larger audiences, the introduction of novel and disruptive services leading to the death of the oldest ones, and massive customer migrations that continuously reshape the social web scenario. Users often tend to migrate, i.e. move to different social media platforms due to specific events, such as the emergence of new platforms or changes to previous platforms. Thanks to the emergence of technologies related to Web3, decentralization through blockchain dominates the landscape of new OSM platforms, proposing creative solutions to the well-known problems of OSM, and introducing innovative key aspects. In this context, Blockchain Online Social Networks (BOSNs) have been proposed and are still being raised. In BOSNs, blockchain technology enables the possibility to redistribute the wealth generated by their users by means of a reward granted to the users that helps the platform grow. These rewarding systems are usually based on the attention economy and/or token economy [66, 155]. Several new BOSNs are proposed, motivated with the common trait of decentralizing control [156, 157], adopting different strategies, such as encouraging a constant

[1] https://wearesocial.com/digital-2021

social and economic dedication or rewarding the creation of pieces of content with outstanding quality.

Due to the lively competition among OSM platforms, *user migration*, is manifesting in these scenarios. There are numerous causes of the user migration phenomenon, including the ethics of the company offering the social service, or the sheer quality of the service offered. User migration affects both centralized and decentralized Social Media, and this aspect is related not only to social services but also to the infrastructure of Social Media. In the scenario of BOSNs, the phenomenon of user migration can be observed and measured with a high temporal resolution when a new BOSN is generated after a fork event.

As concerns user migration, the literature proposes several works on this topic. However, none of these works is focused on the study of the evolution of the subgraphs of users induced by migration. Most importantly, none of them considers the peculiar characteristics of the user migration that manifests after a fork event of a BOSN. In such a system, migration can be studied, with some advantages thanks to blockchain technology, which represents an invaluable and unprecedented source of reliable longitudinal data.

The contribution of this Chapter is to deal with the evolution of BOSNs from the perspective of user migration among platforms. Specifically, we focus on the impact of a shocking event — a hard fork leading to a user migration — on the structural properties of the social and economic networks supported by the blockchains, and to what extent social and economic structural features can be predictive of the choice of a single user migration. In practice, we want to understand *RQ1)* What is the impact of fork events on the social and financial networks? *RQ2)* Is user migration predictable through network structure? *RQ3)* Is a social or financial structure more important for prediction? To investigate these issues, we propose a framework to model the user migration process that is general and therefore applicable to any process of this type. It is based on a representation through an attributed temporal multidigraph, which allows us to measure the effects of the fork on the evolution of the social and economic networks derived from the underlying blockchains. Furthermore, we deal with the prediction of migrating users by exploiting some user characteristics — individual and structural — or activities. As a case study, we apply our framework to the social blockchain Steem, used by Steemit, a leading BOSNs, and the blockchain Hive, introduced after a fork event happened on Steem. To the best of our knowledge, this is the first study on the fork of a blockchain and the corresponding migration of the users of the services relying on it. Furthermore, it is the first work that deals with the prediction of which users will migrate to the new platform right at the time of the fork.

It shows that even with only information on the network structure, without including textual or context data such as the trend of the cryptocurrency, it is possible to predict user migration. Finally, it shows how a multilayer approach improves the performance of the predictors compared to settings that consider the different types of interaction separately.

The Chapter is organized as follows. Section 7.2 presents the related work most relevant to our problem. Section 7.4 describes how we model the activity of a social blockchain in a fork scenario. Section 7.5 describes the preprocessing steps applied to the Steem-Hive dataset , used in this Chapter. Section 7.6 presents our results concerning the difference in the structural evolution of the interaction networks supported by the two blockchains; and the feasibility of predicting which users are willing to migrate after a fork. Finally, Section 7.7 concludes the Chapter, pointing out possible future works.

7.2 Related work

In this Section, we only deal with the literature on user migration as it is the focus of the Chapter. Users often tend to migrate, i.e. move to different social media platforms. Among the main reasons, there is the emergence of new platforms, with novel interesting features. But we often find scenarios where users decide to leave social media due to changes introduced in the platform such as moderation or rule variations. In other cases, conflicts or disagreements in the community lead to the migration of groups of users.

One of the earlier data-driven studies on user migration is by Kumar *et al.* [33], which analyzes migration patterns across multiple platforms. Accounts across different social media platforms are matched by relying on self-published accounts or usernames in Blogcatalog. The study shows the presence of different migration patterns in terms of attention. A reference point in user migration studies is by Newell *et al.* [32] which focuses on permanent migration of activity. They also examine cross-platform migration, by matching accounts between Reddit and Reddit alternatives with an algorithmic approach. They, then, divide users into migrants (those who move all their activities to another platform and remain there), tourists (those who change platforms only temporarily), and dual citizens (active in both). It is a macroscopic analysis of user activity that relies on user surveys to understand user motivations. Other works study user migrations between communities in the same platform. Senaweera *et al.* [34] construct a weighted network that treats a subset of Facebook groups as vertices, while weighted edges represent the number of user migrations among them, showing the presence of non-random migration patterns. Whereas Davies *et al.* [35] studies user migration between COVID-19-related subreddits, by analyzing migration both at the microscale (attention migration, shift of activity from post to post) and macroscale (shift of activity of entire groups). They show the presence of migration through the aggregation of activity values, too.

7.3 Research questions

None of these works is focused on the study of the evolution of the subgraphs of users induced by migration. While some works try to study motivations, none of them try to predict user migration. And most importantly, none of them is looking at user migration in a BOSN, which happens after a fork event. In such a system, user migration can be studied, with some advantages. First, a fork event effectively generates two platforms, allowing the study of cross-platform migration. Moreover, unlike the other scenarios, account matching is trivial, as user accounts are duplicated. Finally, blockchain technology, at the basis of BOSNs, represents an invaluable and unprecedented source of reliable longitudinal data. Therefore, we can tackle the following research questions:
Research question RQ1: What is the impact of fork events on social and financial networks?
Research question RQ2: Is user migration predictable through network structure?
Research question RQ3: Is a social or financial structure more important for prediction?

7.4 Modeling BOSNs, fork and migration

Blockchain Online Social Networks offer their users a rich set of actions and functions to support different kinds of interaction, namely *interaction actions.* Interaction actions — such as comments, likes, reacting, and following — generate different types of relationships among users. In some interactions, the relationship between two users is explicit, that is the case of a user A following B; while some others are implicit, such as a user A who likes or leaves a comment on a post made by B. Moreover, all the interaction actions happen at a precise point in time. Finally, besides the functionalities of traditional OSM, BOSNs provide interaction actions not merely "social", but rather economic or financial. In fact, users can share cryptocurrency tokens by asset transfer actions. A more detailed example will be presented in the case study, in Section 7.5.

In general, interaction actions can be modeled as a set of tuples $I = \{(u, v, t, r)\}$ where u and v are users, who explicitly or implicitly interact through an action of type r at time t. We leverage the temporal information associated with each tuple in I to build a sequence of directed multigraphs [158]. Specifically, due to the different types of relation expressed by r, we consider an evolving edge-labeled multidigraph $\mathcal{G}$ represented by a sequence $< G_1, ..., G_T >$ where each $G_t = (V_t, E_t, R, w_t)$ is a weighted edge-labeled multidigraph, and T is the maximum timestamp in I [159]. Each graph of the sequence is defined by the following elements:

- V_t: the set of users u which belong to at least one interaction action in I which has occurred before or at the timestamp t;
- E_t: the set of triple (u, v, r) with $u, v \in V_t$ and $r \in R$, which represents a specific type of action taking value on the set R of actions offered by the blockchain;
- $w_t : E_t \rightarrow \mathbb{R}$: a weighting function which, given the triple (u, v, r), returns the number of interaction actions of type r involving u and v and occurring before or at the timestamp t.

Finally, it is worth noting that throughout our analysis, we focus only on additive interaction actions, i.e. actions which can only increase the state of a multidigraph. For example, the “follow” action is additive as once a directed link is added to the graph, it cannot be removed unless we also consider the dual operation “unfollow”. This way, in our setting the number of nodes, edges and the values returned by w_t always increase, up to the last timestamp T.

Given the above representation, modeling user migration is quite straightforward. As depicted in Fig. 7.1, both the original blockchain — Steem in our case study — and the new one — Hive — result in two distinct evolution multidigraphs: $\mathcal{G}^S$ and $\mathcal{G}^H$, respectively, with a common ancestor representing the multidigraph at fork time t_F. Despite the modeling, the construction of the sequences of multidigraphs is more challenging, since we may cope with two scenarios:

- *internal user migration*: the set of migrant users remains on the same platform but they move to a different “place” in the platform, e.g. a migration from subreddit A to subreddit B in Reddit, or a change of group in Facebook. In this case, the identification of the migrant users is immediate, since they maintain the same identity (username or user ID).

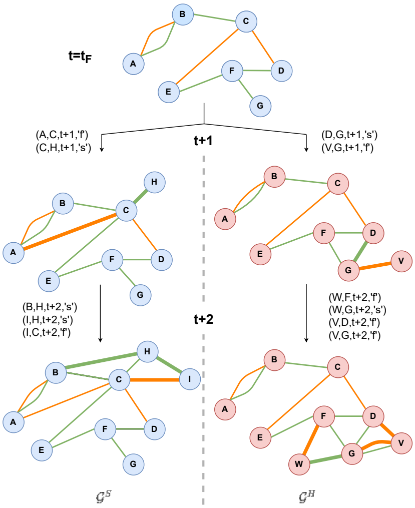


Fig. 7.1: *Example of construction of $\mathcal{G}^S$ and $\mathcal{G}^H$ before and after the blockchain fork. The multidigraph on top represents the state of the network at fork time t_F. Then, we report the bifurcation, and the two sequences $\mathcal{G}^S$ and $\mathcal{G}^H$ evolve independently. Alongside the arrows, we display the interaction actions, occurring during a time window, which generate the links in the corresponding multidigraph. Social links are shown in green and financial links in orange. Bold links indicate the newly added interactions. The sequence on the left describes the evolution of the original blockchain — Steem, while the sequence on the right is related to Hive.*

- *across-platform user migration*: users migrate to a different platform. In this scenario, it is difficult to identify the migrants — especially in the case of game-changing events, like a fork — due to the lack of explicit signals, such as account deletion or migration communication. In these cases, profile-matching or entity-linkage techniques may be applied to connect accounts on different platforms to the same identity.

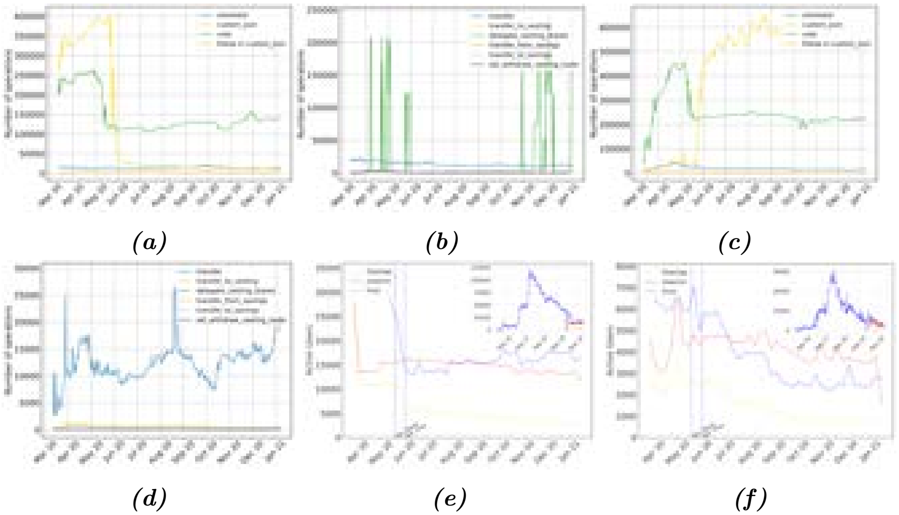
In BOSNs, the user migration due to a fork is part of the second scenario, but with a crucial difference: after the fork, the blockchain supporting the original BOSN is completely copied, so that just after the fork both platforms have the same set of users. In this case, and in particular, in our case study, profile-matching techniques are not required, since the profiles related to the same identity are explicitly linked, i.e. they are cloned. However, the issue related to the identification of the migrants still persists, as the accounts of migrants are still in the blockchain supporting the original platform, as well as the users who remain on the original platform are also in the new platform.

To identify migrants we exploit the activity of users on both platforms. Specifically, a user u migrates from platform S to H after a fork occurring at t_F, if after t_F s/he does at least one action on H; while a user u remains on the original platform if s/he keeps performing actions on the platform S and no actions on H after the fork event. We call *migrant* the first type of user, and *resident* the latter. It is to note that in the remainder of the Chapter a third category — *inactive users*, i.e. people who are inactive or have abandoned both platforms — has been only considered in the feature construction for the prediction task. The above categorization of the users is at the basis of the construction of the node sets $V^S_{t_F}$ and V^H_1, we detail in Section 7.5.

Finally, the above representation and modeling methodology is applicable not only to other blockchain forks but also to other user migration processes whose data are known. In fact, the construction of the sequence of multidigraphs only requires the set of tuples I. In the absence of blockchain data, data availability, and profile matching are the obstacles and put a limit on the applicability of the representation. Specifically, how to collect high-resolution temporal data from the old and the new platform and to carry out account matching are the main issues to be faced when applying the proposed model and methodology to out-of-blockchain contexts.

7.5 Data Preprocessing

We use the Steem-Hive dataset , presented in section 2.4. We first present an overview of the data from Steem and Hive after the fork, from Fig. 7.2a to Fig. 7.2d. In detail, as shown in Fig. 7.2a, in Steem, we observe a stable or even increasing trend for `vote` and `custom json` operations during the first two months after the fork, but at the beginning of June 2020, two abrupt changes in the volume of operations occurred. On Steem, we observe a steep decrease of the `custom json` operation, while in Hive (Fig. 7.2c) we observe the opposite, with an increase in the volume of `custom json` operations. Specifically, on Steem, the volume dropped by 10X in a week (from 350K to 30K operations), while on Hive the volume rose by the same factor. Moreover, after this abrupt increase the overall volume of `custom json` operations on Hive has reached higher values w.r.t. Steem's volumes before the drop of June. As for `vote`, the trend in Hive is similar to Steem on the whole observation period, with a sudden decrease in the volume at the beginning of June 2020. The `vote` trend is different in the bootstrap phase of Hive, where `vote` operations have continuously increased until May 2020. Moreover, it is to note that after the drop in June, the volume of `vote` operations in Hive is double the volume in Steem. Conversely, the volume of `comment` operations and `follow` operations are quite stable on both blockchains and are marginally influenced by June's events.



(a) (b) (c) (d) (e) (f)

Fig. 7.2: *From (a) to (d) daily volume of interaction actions in Steem and Hive blockchains after the fork, grouped by category. In order: (a) the daily volume of the social operations on Steem, (b) the daily volume of financial operations on Steem, (c) the daily volume of social operations in Hive, and (d) the daily volume of financial operations in Hive. In (e) and (f): number of unique users in Steem, Hive and their overlap, i.e. active users in both platforms. In particular: (e) unique users performing social actions, (f) unique users for financial actions. In the inset, unique users over the entire observation period, from 2016 to 2021.*

As for financial actions, in Fig. 7.2b and Fig. 7.2d we report the daily volume of each operation belonging to the financial group. Each blockchain is characterized by a specific financial action. In particular, in Steem `delegate vesting shares` operations reach the highest daily volumes and are characterized by an unstable trend with a few spikes in the first (April to June 2020) and last (November 2020 to January 2021) months. Such a trait might indicate anomalous behaviors in the voting operations since, through `delegate` `vesting shares`, users can “borrow” their voting power to other accounts. On the contrary, in Hive, we do not observe spikes in `delegate vesting shares`, and the `transfer` operations are the most common ones. In the case of `transfer` operations, the average volume in Steem and Hive is comparable, i.e. from 10K to 15K daily `transfer` operations. The remaining operations are quite marginal on both blockchains and have stable trends.

Construction of the evolving multidigraphs:

We process the blockchain data presented in section 2.4, so as to cast the sequence of interaction actions returned by the blockchain into the representation framework described in Section 7.4. In particular, given an interaction action (u, v, t, r), we consider the timestamp associated with the block containing the interaction operation of type r as the time t of the interaction. Hence, we can build each multidigraph G_i of the evolving multidigraph by selecting a one-month temporal window between two consecutive graphs G_i and G_{i+1}. Specifically, we aligned each evolving graph G_i to the 20th day of each month, at 2:00 PM. This allowed us to start the first snapshot post-fork for both sequences exactly at fork time, for a better comparison of network characteristics. Finally, we selected and grouped the interactions based on categorization defined in Table 2.1, so that r takes values on the set $\{social, financial\}$.

As also displayed in Fig. 7.1, the construction of $\mathcal{G}$ proceeds incrementally. Given $G_i \in \mathcal{G}$, we define G_{i+1} by first setting $G_{i+1} = G_i$. Then, we iterate over the interaction actions (u, v, t, r) such that $i < t \leq i + 1$. If the labeled edge (u, v, r) is not in G_{i+1}, we insert it and assign to it a weight equal to 1; otherwise, if $(u, v, r) \in G_{i+1}$, we only increment by one its weight.

As a final note, `custom json` is an operation that can be used for multiple functionalities, and in the analysis we only considered one of the actions assignable to this operation, i.e. the “follow” action.

7.6 Results

In this Section, we report the main findings on the structural effects of the fork on the Steem and Hive social and financial networks (*RQ1*), by taking into account the entire Steem-Hive dataset , from 2016 to 2021, resulting in 48 pre-fork and 9 post-fork temporal snapshots. Then, we deal with the problem of predicting whether or not a user will migrate after the fork, within a machine learning framework.

Structural effects of the fork

The March 20th fork represents a game-changing event in the history of Steem and Hive due to both its exceptional nature and the way it happened, i.e. a reaction of part of the Steem users towards some design choices and hostile behaviors in the original blockchain. In this Chapter, we deal with the impact of this important event on the interaction networks generated by the interaction actions in Steem and Hive, taking separately into account social and financial relationships. Through the representation of the evolution of the blockchains described in Section 7.4, we aim to identify to what extent the fork event has made the Steem and Hive interaction networks different.

Evolving graphs (all users)

We first analyze the evolving interaction multidigraphs $\mathcal{G}^H$ for Hive and $\mathcal{G}^S$ for Steem by inspecting different structural properties on each element — a multidigraph — of the sequence $\mathcal{G}^{(\cdot)}$. We analyze them at regular time steps, by following the construction methodology presented in Section 7.5. Therefore, we have 48 snapshots — multidigraphs — describing Steem before the fork date — *pre-fork*, while for the snapshots after the fork, we rely on data from Steem and Hive, thus obtaining 9 snapshots after the fork — *post-fork* — for both platforms. In Table 7.1, we show a summary of the network properties measured pre-fork and post-fork. Since our focus is the comparison between Hive and Steem interaction networks, we focus on the properties in the post-fork period, reported in the last two columns of the table. For each platform, we report the average and standard deviation of each property, both on the financial and social networks, separately.

Starting from diameter measures, we can observe similar values, with Hive showing only a slightly smaller diameter. This may suggest the fork had no shrinkage effects on the diameters of both social networks. Similarly, Hive has a bigger largest connected component in both social and financial networks. Other properties computed, such as average clustering coefficient, reciprocity, and degree assortativity are similar across both platforms. The values of degree assortativity suggest a lack of degree assortativity and reciprocity values are also low with respect to other measurements on major online social networks [160]. As for reciprocity, we also observe a further decrease from 0.22 to 0.19, which suggests the creation, on both blockchains, of many non-reciprocal links after the fork. In fact, by construction, the sequences of multidigraphs after the fork keep the information of the previous snapshots, so even a small variation of the indices might indicate a significant change in the structure.

Table 7.1: *Network statistics. Statistics are computed on the evolving multidigraphs every 30 days.*

	Steem (Pre-fork)		Steem (Post-fork)		Hive (Post-fork)	
Metrics	Social	Financial	Social	Financial	Social	Financial
Density (x10^{-4})	17.81 ± 58.0983	0.72 ± 1.3204	1.15 ± 0.0324	0.02 ± 0.0055	1.17 ± 0.0069	0.03 ± 0.0004
Diameter	6.06 ± 1.2784	10.06 ± 11.2655	5.89 ± 0.3333	9.00 ± 0.0000	5.67 ± 0.7071	9.00 ± 0.0000
Degree Assortativity	-0.09 ± 0.0362	-0.13 ± 0.0566	-0.06 ± 0.0002	-0.09 ± 0.0030	-0.06 ± 0.0001	-0.10 ± 0.0008
Reciprocity	0.22 ± 0.0302	0.15 ± 0.0454	0.19 ± 0.0003	0.18 ± 0.0048	0.19 ± 0.0002	0.18 ± 0.0003
Average Local Clustering	0.38 ± 0.0382	0.39 ± 0.0340	0.37 ± 0.0031	0.40 ± 0.0037	0.37 ± 0.0035	0.41 ± 0.0035
Perc Largest Component	58.88 ± 5.2383	17.42 ± 8.1788	57.74 ± 0.5297	12.23 ± 1.5715	58.15 ± 0.1312	15.00 ± 0.0184

Active users

Finally, we compare Steem and Hive in terms of active users. We measured the number of active users, in different time periods, in both Steem and Hive. We also retrieve the intersection of user activities in both platforms, to get a grasp of the overall overlap. We show the obtained information in Fig. 7.2e for social interactions and in Fig. 7.2f for financial ones. We can see an overall drop in active users, in both social and financial networks. However, there was an already decreasing trend in the number of users, as we can see from the inset of figures, that cover the entire period. The trend continues on both Hive and Steem. Specifically, on the social side, we observe that Steem still has a higher number of active users. We can also note that the overlap — the yellow line in the figures — also drops quickly. Over time, users stop being active on both platforms, deciding where to focus their efforts. We note a few differences in the financial side. First, the number of active users on the financial side of Hive surpasses Steem. Also, while we still have a drop in overlap, the drop is slower than the one we observe for social actions.

Active users induced subgraph

In addition to the generated evolving networks, we also study in more detail the behavior of active users in the period before the fork. In our set of users of interest, we include users active before the fork (3 months before), while including new users that would appear in the following nine months, namely

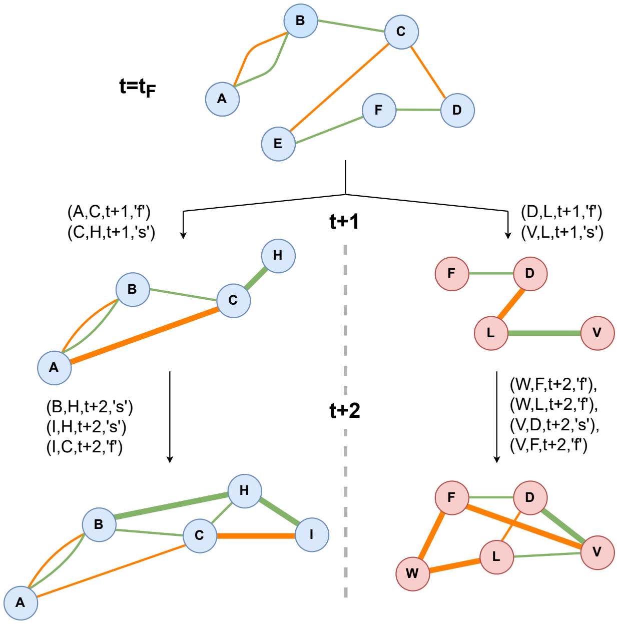


Fig. 7.3: *Construction of the sequence of multidigraphs for active users. The multidigraph on top corresponds to the subgraph induced by the set of active nodes on the top graph in Fig. 7.1 — the node G is inactive. Then, for each blockchain, we only maintain resident and migrant nodes, respectively. This multidigraph represents the starting point of the building procedure depicted in Fig. 7.1. Node E will be active after $t + 2$.*

the set U. The obtained set of users is then monitored throughout the period after the fork, by extracting the subgraph induced by the set of selected users in each snapshot of the sequence. More specifically, as summarized in Fig. 7.3, we identified the subgraph induced by U on $G_{t_F} \in \mathcal{G}^S$: it represents the starting point for the construction of the evolution sequences for active users. In the case of Steem, we only keep resident nodes and their links from the induced subgraph and proceed with the procedure described in Section 7.4. In the case of Hive, from the induced subgraph we only consider migrant nodes and apply the same procedure to Hive data only. In Table 7.2 we show the network properties for the induced sequences. Compared to the previous networks, we can see that Hive still has lower diameter values. Hive also exhibits bigger largest component, in both financial and social networks. We see slightly higher values of reciprocity, but they are still far from reciprocity values typical of online social networks. Finally, the degree assortativity is not significant in the subgraphs as well.

Table 7.2: *Network statistics on the induced subgraphs of active graphs. The multidigraphs were induced by considering users active three months before the fork and those active after the fork. Statistics are measured on each snapshot of the evolving multidigraphs.*

	Steem (Post-fork)		Hive (Post-fork)	
	Social	Financial	Social	Financial
Metrics				
Density (x10^{-4})	42.20 ± 8.7483	6.16 ± 0.1424	46.21 ± 2.8060	5.80 ± 0.3213
Diameter	7.00 ± 1.5811	8.11 ± 1.4530	5.78 ± 0.6667	7.33 ± 2.0000
Degree Assortativity	-0.07 ± 0.0013	-0.20 ± 0.0008	-0.08 ± 0.0018	-0.20 ± 0.0009
Reciprocity	0.25 ± 0.0005	0.32 ± 0.0038	0.25 ± 0.0031	0.32 ± 0.0023
Average Local Clustering	0.39 ± 0.0045	0.40 ± 0.0057	0.40 ± 0.0052	0.41 ± 0.0043
Perc Largest Component	89.22 ± 5.4459	88.60 ± 1.9379	92.17 ± 0.7496	86.23 ± 2.2581

User migration prediction

As shown by the above results, the fork-based user migration has been a relevant event that has involved a substantial amount of users. So, for each user, we would like to understand if their choice to adopt a new platform could be explained or even predicted by some user's characteristics or activity; and, in that case, which are the early signals indicating that s/he will move to a new platform (*RQ2*).

This problem can indeed be formulated as a machine learning task, specifically a binary node classification task.

Definition 13 (User migration prediction task). *Given the graph G_t and considering the successive timestamps t', where $t' > t$, we define the user migration prediction task as the prediction of a node migration in one of the successive time steps.*

The objective is to predict the two classes (Migrant or Resident) based on several user/node features. The assumption is that user features, at the network structure level, could be predictive of a future user migration. Note that features can be extracted from both layers of the evolving multidigraph: the financial and the social layers; thus obtaining two additional scenarios.

We can define the first case as a *financial user migration prediction task*, whereas for social actions only, we can define a *social user migration prediction task*.

Definition 14 (Financial user migration prediction task). *Given a graph G_t and considering the successive timestamps t', where $t' > t$, we define the financial user migration prediction task as the prediction of a node migration, on the financial layer, in one of the successive time steps.*

Definition 15 (Social user migration prediction task). *Given a graph G_t and considering the successive timestamps t', where $t' > t$, we define social user migration prediction task as the prediction of a node migration, on the social network layer, in one of the successive time steps.*

As in the first task, for both tasks, we predict the label Migrant or Resident based on the user/node features, extracted on the financial or social layers, respectively.

Features and labels

The features we considered are the most common node-level features utilized in many network-based prediction tasks, and that encode information about a node and its neighborhood. Specifically, for each user in G_t, we compute in-degree and out-degree, weighted in-degree, Pagerank, neighborhood average degree, and local clustering coefficient. Alongside the structural information, we also include information on the status of nodes in the neighborhood. We define two additional features:

- Percentage of inactive neighbors: the number of neighbors whose status is inactive at time t, divided by the total number of neighbors.
- Percentage of resident neighbors: the number of neighbors whose status is resident at time t divided by the total number of neighbors.

These features can be computed on both the financial and social layers. Given the defined features, the objective is to predict a potential migration in the future. The labels for the two classes are Migrant and Resident.

Experimental Setting

Our prediction context is the migration from Steem to Hive. Hence, for the following experiments, we focus on the Steem evolving multidigraph of active users and its financial and social subgraphs. More precisely, we select the snapshot at fork time, t_F = 2020/03/20, at 2:00 PM. Then, we obtain the labels describing the future cases, Migrant or Resident. However, the two classes observed are imbalanced. In the social layer, there is a more severe imbalance, as residents are 3/4x more than migrants (66.9 %, 33.1%). While in the monetary layer, the two categories are closer, there are more migrants (56.1%) than residents (43.9%).

The main options to deal with sample imbalance consist of undersampling, so discarding examples from the most numerous classes, or oversampling, which generates new examples starting from the existing minority class. One of the pivotal advantages of oversampling is that we would not discard any of the available data. Among the many oversampling techniques, the most used is SMOTE [161]. Oversampling allows us to balance the example for both classes.

We perform experiments in a 5-fold cross-validation setting. For each fold, we apply oversampling on the training portion of the fold. Note that oversampling is applied only to the training portion of the data. Then, we train a model and compute a set of evaluation metrics. The metrics are averaged over the five folds. For the evaluation, we compute the main evaluation metrics for classification tasks: weighted F1, accuracy, precision, recall, and AUC. The metrics are computed on the testing portion of each fold and then averaged. For the classification task, we rely on standard machine learning methods: Logistic Regression, Random Forest, Support Vector Machine with linear kernel, and a Gradient Boosting classifier.

Results

In this Section, we are dealing with three migration prediction tasks: social user migration, financial user migration, and user migration.

The experimental results for the social user migration prediction task 15 are presented in Table 7.3. As we can see, the structural features, together with the simple information on the activity of the neighbors, are able to provide a prediction on the migration of a node, even if the performances are modest across the different models. Among them, we observe that Random Forest and Gradient Boosting are leading the tested models in F1, Accuracy, with Gradient Boosting performing better in terms of precision, while RF shows a better recall. The other two models tested lag behind in terms of performance, with lower scores across the board.

Similar results can be observed for the financial user migration prediction task. In Table 7.4, we report the obtained evaluation metrics. Overall, we can see better performances for all models. Indeed, as in the previous experiment,

Table 7.3: Social migration prediction. Features were computed on the Steem multidigraph, limiting on edges with type " social". Metrics (Weighted F1, Accuracy, Precision, Recall, AUC) are the average over a 5-fold cross-validation.

Models	F1	Acc.	Prec.	Rec.	AUC
Random Forest	0.66	0.65	0.78	0.70	0.61
Logistic Regression	0.58	0.56	0.79	0.51	0.59
Linear SVM	0.58	0.56	0.79	0.52	0.59
Gradient Boosting	0.62	0.60	0.79	0.59	0.61

we can see that Random Forest and Logistic Regression are performing better than the other models. We can infer that financial information may be more informative for the prediction of future user activities. We may hypothesize a possible explanation for that: as detailed in Section 7.5, the 51% attack has been conducted by gaining a large amount of voting power, and the reaction to the attack acted in the same direction. Since the voting power is strictly related to financial operations, such as exchanging assets for shares and borrowing shares to gain more rights to vote, the structure of the resulting financial interaction networks has been influenced by the dynamics leading to the hard fork, and the resulting migration of one of the factions.

Table 7.4: Financial user migration prediction task. Features computed on the Steem multidigraph, limiting on edges with type " financial". Metrics (Weighted F1, Accuracy, Precision, Recall, AUC) are the average over a 5-fold cross-validation.

Models	F1	Acc.	Prec.	Rec.	AUC
Random Forest	0.69	0.69	0.78	0.78	0.63
Logistic Regression	0.60	0.59	0.75	0.62	0.57
Linear SVM	0.61	0.59	0.75	0.63	0.57
Gradient Boosting	0.65	0.64	0.79	0.67	0.62

In Table 7.5 we show the results for the user migration prediction task. In this task, we are combining features from both the social and financial layers, fully leveraging both the evolving graphs. The concatenated features provide additional information for the prediction of user migration. Overall, we can see an improvement in the metrics all across the board. Specifically, the models that were performing the best improved their performances over the previous migration prediction tasks. In addition, we can observe that the additional information aids the models that were not performing well, like SVM and Logistic regression, that see an improvement over all the metrics. The obtained results suggest the need for modeling more layers to fully understand user behavior. So, to answer *RQ2*, structural information can be predictive of user migration.

Table 7.5: *User migration prediction. The features are a concatenation of those computed on the Steem financial network and Steem social network, respectively. Metrics (Weighted F1, Accuracy, Precision, Recall, AUC) are the average over a 5-fold cross-validation.*

Models	F1	Acc.	Prec.	Rec.	AUC
Random Forest	0.71	0.71	0.72	0.78	0.71
Logistic Regression	0.66	0.66	0.67	0.77	0.65
Linear SVM	0.66	0.67	0.66	0.78	0.66
Gradient Boosting	0.70	0.70	0.70	0.78	0.69

Finally, we perform a feature importance analysis to highlight the most predictive features (*RQ3*), and, in our specific temporal setting, to identify the early signals of willingness to migrate. The features along with their importance ranked in descending order are displayed in Fig. 7.4. The most important features are related to both social and financial layers. Specifically, the clustering coefficient in the social layer and the neighbor degree in both social and financial ones are among the most important features. The analysis confirms the importance of taking into account information derived from both types of interaction action for the user migration prediction task.

7.7 Conclusion and future works

In this Chapter, the topic of user migration among OSM has been addressed. User migration has been a relevant process in the past with the migration of users from one platform to another, a new and more interesting one. But it might become more and more massive with the crisis of traditional platforms and the emergence of new social media paradigms, among which the most

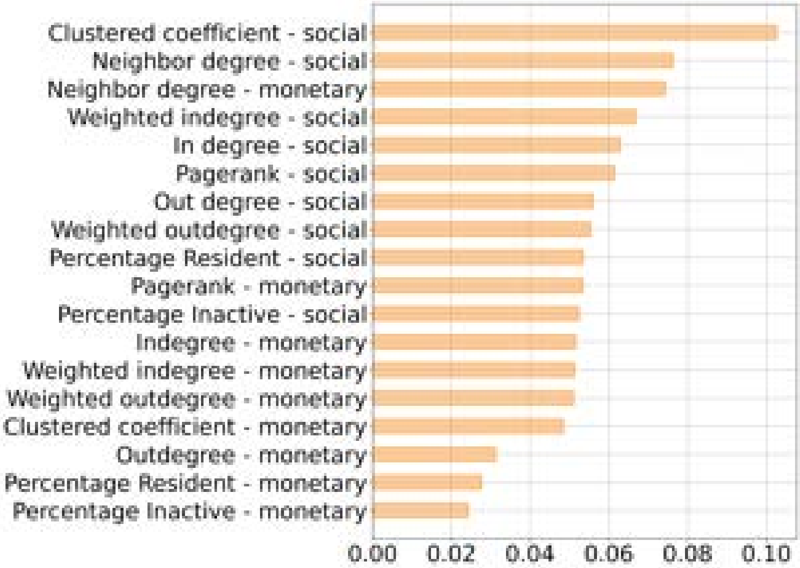


Fig. 7.4: *Feature importance for the best performing model, i.e. Random Forest, on the user migration prediction task. Importance values are based on the mean accumulation of the impurity decrease within each tree of the Random Forest.*

interesting are blockchain social media with their promises to be able to overcome the many well-known issues of traditional OSM. Also, BOSNs have issues as they are not yet mature platforms, often subject to internal changes that can lead to blockchain forks and related user migration between the overlying services.

Despite the importance of these processes, research on user migration in general, and on blockchain forks in particular, is still at an early stage. Among the many obstacles to research on this topic, there is certainly the difficulty in collecting representative datasets, as they must be longitudinal and need user matching. In this sense, BOSNs represent an invaluable source of data in this field.

To the best of our knowledge, this is the first study on blockchain fork and user migration in BOSN. It contributes to a general user migration model applicable to other BOSNs; it shows that it is possible to predict user migration even on the basis of the network structure only, as in the Steem-Hive case study. The methodology, the tools, and the results herein provided are applicable in the case of a possible hard fork, but they do not offer practical solutions to prevent a hard fork. In fact, platform administrators, if a hard fork is a very likely event, should look at both social interactions and economic transactions to identify the set of users who likely will abandon the old platform to join the new blockchain. To this aim, our findings about prediction have highlighted that, in a stratified context where social and economic relationships are mixed together, both dimensions are important in describing and forecasting users' behaviors during and after a shocking event in the network. Actually, we have focused on the proprieties of the networks to predict user migration, however, a further step would be to understand the motivations that lead a user to migrate or not. To this aim, an integration of the features extracted from the textual content produced by users with the structural features might highlight the reasons for the migration.

We hope that this Chapter will pave the way for other studies on blockchain fork and user migration in order to better understand these so important, but still largely unknown processes. Besides user migration, the representation for the blockchain data modeling might be applied to a few phenomena characterizing the Web3, for instance, the trading networks generated by NFT (not-fungible token) exchanges or other kinds of social and financial interaction mediated or fueled by Dapps, such as games or thematic social networks.

Chapter 8

Predicting user migration with Graph Neural Networks

8.1 Introduction

Despite an increasing number of studies [33, 32, 34, 35, 36], user migration remains an understudied topic, particularly in BOSM platforms. One primary gap in the literature is the lack of methodologies for accurately predicting user migration, especially when there is a scarcity of user information or features. Existing methods that rely on interaction graphs built from user interactions show promise in addressing this challenge. Despite the graph-based representation of the task graph neural networks have not been yet applied to the user migration prediction, even though they have achieved state-of-the-art results in many machine learning tasks on graphs. This is an important research gap to be addressed in this context as GNNs do not require any feature engineering step on the interaction graph and they have shown their prediction power even without contextual information on users. Furthermore, user migration, as many other learning tasks on graphs, is often characterized by class imbalance, i.e. the target class sizes available in the dataset differ by a substantial margin [38], which can negatively impact the performance of machine learning models. The most used techniques primarily operate on the data level, aiming to modify the distribution of training data instead of altering the machine learning model itself. Typically, these methods utilize sampling-based approaches to tackle the issue of class imbalance. Existing methods often focus on oversampling-like strategies, differing in the methodologies for generating features, structures, or labels for the creation of artificial minority data instances [162]. However, in scenarios where there are sufficient data samples available for each class, the generation of new synthetic data is not preferable, as sampling may introduce bias. Interestingly, to the best of our knowledge, none of the current approaches address undersampling techniques. These aspects led us to the following research questions (RQs): *RQ1)* Are graph neural networks a suitable method for user migration prediction? *RQ2)* Can we improve performance in cases of severe class imbalance with a balancing method following an undersampling approach?

To fill these research gaps, we focused on predicting the phenomenon of user migration in the context of Blockchain Online Social Media (BOSM) platforms. Specifically, we design a machine learning pipeline to verify the effectiveness of graph neural networks for user migration prediction, where we model the data as a directed temporal multilayer graph describing social and monetary interactions among users to predict user behavior as a classification task. We also designed a data-level balancing technique following an undersampling approach, comparing the results within the same pipeline. To evaluate our methodology, we gathered data from the ecosystem of social platforms based on the Steem blockchain, whose main member is Steemit, and Hive, the blockchain originating from a hard fork of the Steem blockchain on March 20, 2020.

Our methodology for the selection of the best model and the proposed balancing approach have highlighted some interesting findings. Graph neural networks are an effective method to predict user migration in blockchain-based online social networks: the GNN model is able to leverage graph structure on the graph of monetary interactions, even with moderate data unbalance; however, the GNN model struggles on the graph of social interactions that is characterized by severe data imbalance (*RQ1*). However, after applying our proposed data-level balancing approach that produces a more balanced training set, graph neural networks show good predictive power even on severely imbalanced data (*RQ2*).

The Chapter is organized as follows. Section 8.2 provides a brief introduction to blockchain online social media and machine learning on graphs. In Section 8.3 we introduce the main research questions we focus on. The methodology for modeling interaction data, performing user migration prediction, and the proposed balancing method is presented in Section 8.4. Section 8.5 reports the main findings on the effectiveness of graph neural networks and the impact of applying a balancing approach. Finally, Section 8.6 concludes the Chapter, pointing out possible future works.

8.2 Related work

Machine learning on graphs

In the last decade, there has been a growing interest in developing machine learning techniques tailored for graphs to solve various tasks, such as node classification, link prediction, and graph generation. In this context, traditional approaches adopted a manual feature generation approach in order to get a vector of statistics for each node, that could later be fed into traditional learning models. However, these approaches are often time-consuming and inflexible as they cannot be adapted to the learning process. More recent approaches however rely on the concept of graph representation learning, i.e. encoding the structural information of nodes into a low-dimensional latent space. In the field of graph representation learning, graph neural networks (GNNs) have emerged as the state-of-the-art approach in many different tasks, such as node classification [37], link prediction [163], community detection [164] and graph classification [165]. GNNs were designed to perform predictions by exploiting both topology and graph attributes by redefining basic deep learning operations, such as convolution, for graph-structured data. The concept has been formalized as the *message passing framework* [166]: the convolution on graphs can be performed by aggregating the values of each node's features along with its neighboring nodes' features. One of the earliest examples is the Graph Convolutional Network (GCN) model proposed by [167]. Given a graph $G = (V, A, X)$ such that V is the set of vertexes, X is the node feature matrix, and A the adjacency matrix, at each layer k the embedding h of a node i is updated with the following computation:

$$h_i^{(k+1)} = \sigma \left(\sum_{j \in N(i)} \frac{1}{\sqrt{\widetilde{D}_{ii} \widetilde{D}_{jj}}} h_j^{(k)} W^{(k+1)} \right) \tag{8.1}$$

where $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$ corresponds to the degree of i, computed on A_{ij} the adjacency matrix with self-loops added. The aggregation is order-invariant, (examples of such functions are average or summation). The number of layers of a GNN defines the number of hops up to which a node will receive information. Starting from these, we have seen the proposal of many architectures such as GAT [168], graph autoencoders [169], GraphSAGE [37], and many more, to cover different tasks and types of graph data.

Class imbalanced learning on graphs

A classification problem is considered imbalanced when the target class sizes of a dataset differ relatively by a substantial margin [38]. There are several examples of real problems that are affected by this phenomenon, such as fraud detection, disease diagnosis, anomaly detection, and sentiment analysis. An imbalanced data sample can have a negative impact on the predictive performance of the model, especially for the minority classes. This is because the model has fewer opportunities to learn the characteristics of the samples within the minority classes, which can lead to poor generalization skills when applied to unseen testing data. Finally, a class imbalance can cause the model to be biased towards the majority classes, resulting in a tendency to predict the class with the larger number of instances. Class imbalance remains a challenging problem in machine learning, but there exist techniques and strategies that can be employed to mitigate its negative effects. Current approaches can be divided into two categories [162]: *i) data-level* methods, which modify the distribution of training data, and *ii) algorithm-level* methods, which modify learning algorithms. Acting at the data level is the most flexible approach as it allows the use of already available models. Data-level methods try to address the imbalance through sampling-based approaches [38]. Methods usually rely on under-sampling approaches to select a subset of instances from the majority classes or over-sampling approaches to create additional instances of the minority classes or even a mix of both (hybrid sampling). All those techniques have been designed on point-based data and have limitations when it comes to learning on graphs [170]: while in traditional cases it is just a matter of considering more or less independent data points, in graphs is more complicated, as removing nodes/edges will automatically modify the graph structure, and this can create issues during the model training, especially during the message-passing process in GNN models. On the other hand, adding a node requires managing both the node attributes and connectivity. As a result, some proposals have been made to address class-imbalanced learning on graphs, acting at both the data level and the algorithmic level. Current data-level methods are focused on oversampling-like approaches and they differ in their approach to generating features, structures, or labels for synthetically created minority data instances [162]. However, in cases where there are sufficient data samples for each class, generating new artificial data is not desirable, as it could introduce bias in the dataset. And yet, to the best of our knowledge, none of the current works addresses approaches following the undersampling approach.

8.3 Research questions

The problem we address in this Chapter is user migration prediction, which has received limited attention in the context of Web3 platforms. While developing a machine learning pipeline to predict whether a user will migrate, stay on the original platform, on both, or become inactive, this Chapter will answer two main research questions:

Research question 1 (RQ1): Are graph neural networks a suitable method for user migration prediction?

Research question 2 (RQ2): Can we improve performance in cases of severe class imbalance with a balancing method following an undersampling approach?

By answering these questions, we aim to contribute to the development of effective techniques for predicting user migration in Web3 platforms, which could have implications for improving user experience and enhancing platform design and management.

8.4 Methodology

Our objective is to leverage user interaction data to predict future user migration decisions. We utilize a similar setting to the one proposed in a previous work [36], where user migration is treated as a machine-learning task on graphs, using only the network structure of the graph to perform predictions, while user behavior is encoded in classes, allowing us to handle user migration as a multiclass node classification problem. In this section, we define the machine learning pipeline, that will be used to perform the user migration prediction task. Our proposed pipeline is presented in Fig. 8.1.

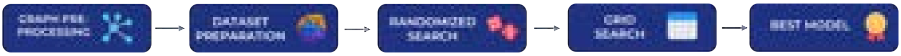


Fig. 8.1: *The proposed methodology to solve node classification tasks.*

In the following, we describe the methodology adopted in each step, which will allow us to leverage interaction data as input for machine learning models, to verify the effectiveness of graph neural networks in the setting of a user migration prediction task, as well as to address the class imbalance in datasets.

Modeling user interactions and user decisions: graphs and labels

User interactions can be modeled as a set of tuples $I = (u, v, t, r)$, where u and v are users, who explicitly or implicitly interact at time t through an action of type r. As we are interested in the graph structure before the fork, we can consider the interactions before t_{Fork} (March 20th, 2020, 2:00 PM), which we denote as $I_{t_{Fork}}$. From this subset of interactions, we are able to construct a temporal directed multilayer graph [158], that we denote as $\mathcal{G} = \{\mathcal{G}^r_{t_{Fork}} \forall r\}$, where each element $\mathcal{G}^r_{t_{Fork}}$ is a layer of the multilayer graph. More precisely for each interaction type r, a layer of the graph can be seen as a temporal weighted graph $\mathcal{G}^r_{t_{Fork}} = (V^r_{t_{Fork}}, E^r_{t_{Fork}})$ that stores the interactions of type r that happened up to t_{Fork}. Each edge $(u, v, t, c) \in E^r_{t_{Fork}}$ encodes the operations from node u to node v, described by the counter c and timestamp t. Specifically, the counter c keeps track of the number of operations within the directed pair of nodes, while the timestamp t corresponds to the time of the first operation from u to v. While the obtained graphs could be used to perform prediction on all users, they may have not been active before the fork, therefore it is important to filter users that stopped using the platform before the fork event. We define a set U of users of interest, in which we consider only users active before the fork while including new users that would appear in the following time period. Similarly to what has been done in [36], a user u belongs to the set U (therefore *active*) if it performed at least one operation in the 3 months before the fork event. In this way, we are able to extract $G_{t_{fork}}$, i.e. the subgraph of $\mathcal{G}_{t_{fork}}$ induced by the set U of active nodes. If we consider the set of $r \in \{monetary(m), social(s)\}$, we can denote the layer graphs $G^m_{t_{Fork}}$ and $G^s_{t_{Fork}}$, representing monetary interactions and social interactions respectively, that will be leveraged to predict behavior after the fork. We then need to process interaction data to encode user behavior after the fork, in a way that can be learned by machine learning models. This means defining labels for each node based on the user activity after the fork. If we observe the interactions that happened after the fork event involving a user u, we can consider 4 possible cases:

- *resident*: a user active only on the original platform (Steemit)
- *migrant*: a user active only on the new platform (Hive Blog)
- *co-active*: a user that performs actions on both platforms
- *inactive*: a user that stops using both platforms

These cases are defined at the end of an observation period, after the fork event, considering the activity up to the last interaction in the available data. So each user a is assigned to one of the four labels after observing the interactions

$I = (u, v, t, r)$ where $u = a$ and with $t > t_{Fork}$. The assigned label (*resident*, *migration*, *co-active*, or inactive) is defined as the *migration decision* l of user u.

Leveraging graph neural networks: model training and best model selection

The first step is the selection of the architecture for our machine learning model: we selected the GCN architecture from [167]. We implemented a similar version, as represented in Fig. 8.2. First, the input features and the adjacency matrix are then leveraged by two graph convolutional layers that create node embeddings. Finally, a linear transformation layer uses the embeddings generated by the GNN, to return a vector with a dimension equal to the number of target classes of the task. Then we obtain a vector representing a probability distribution on the target classes by applying to the output of the previous layer z a softmax function $\sigma(z)$.

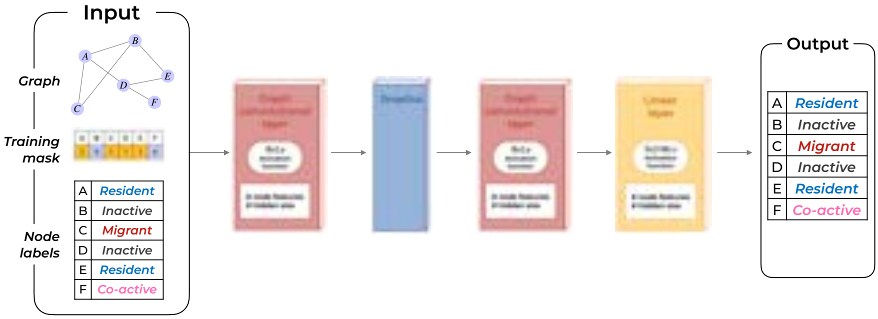
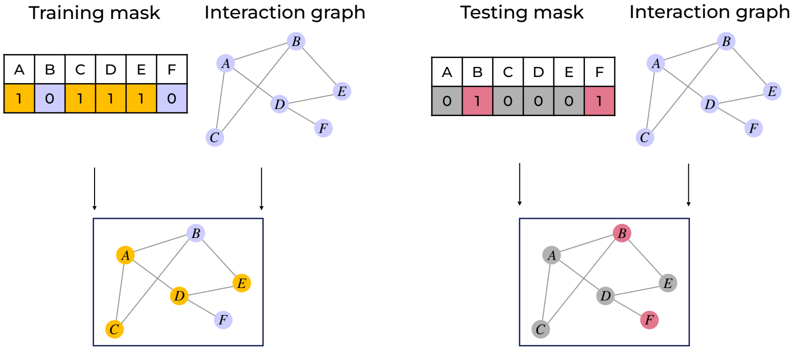


Fig. 8.2: *Representation of selected graph neural network architecture. The selected architecture is inspired by the classical GCN architecture by Kipf [167].*

The selected graph neural network model needs to be trained, i.e. its weights need to be adjusted so that it can learn to predict the right classes. When the ground truth is available, GNNs can be trained in a supervised setting. For node classification tasks, supervised learning requires the so-called train-test split [171]. While in traditional machine learning tasks, the split requires the separation into two sets of training samples, when dealing with graphs, the split is not as straightforward: for graph neural networks, the training and test sets are defined as the creation of masks $M_1 \in \mathbb{R}^n$, like in Fig. 8.3.



Training mask

A	B	C	D	E	F
1	0	1	1	1	0

Testing mask

A	B	C	D	E	F
0	1	0	0	0	1

(b)

Fig. 8.3: *Supervised training example. On the left side, the training mask is defined, while on the right side, an example of the corresponding test mask. Training and message passing are performed using the complete graph structure, but the loss function is computed only for training nodes. During testing, message passing is performed over the entire graph, but evaluation is conducted on test nodes.*

As in traditional supervised learning frameworks, the objective is to make the model output as close as possible to the ground truth values. This is done by adjusting model parameters through the data learning process to minimize a loss function. A common choice for classification problems is the cross-entropy loss function [167]. Alongside the model parameters, the selection of the best model configuration for the task requires the optimization or tuning of hyper-parameters, i.e. parameters that can not be estimated from data learning and must be set before training an ML model because they define the model architecture [172]. Testing all the possible combinations of hyperparameters from the grid of possible parameters — *grid search* — can be a computationally demanding and time-consuming phase as there are many hyperparameters in GNN models, leading to a huge number of combinations to verify. For this reason, a popular strategy is to perform a *random search* [173]: only a subset, of possible hyper-parameter combinations, is chosen at random and tested. in this Chapter, we combine the two approaches. After the first exploratory step is conducted with a random search, the best configurations are used to refine the candidate configurations, so that we can reduce the number of combinations before performing a full grid search.

Dealing with class imbalance: a new undersampling based approach.

Formally, in a multiclass supervised learning task, there are m classes in total, $\{C_1, ...C_m\}$, and $|C_i|$ is the size of the i-th class, referring to the number of samples belonging to that class. Here, we introduce an under-sampling technique to balance the distribution of the target variable at the data level. Formally we balance the target variable as follows: we choose a percentage p, and compute the number of samples $n = min_i|C_i| * p$ to get the number of samples per class to include in the training set. To build a balanced training set, we perform under-sampling of each class C_i: we consider a random subset of cardinality n of samples, creating a uniform distribution. This leads to a reduced training set size, but each target class is equally represented. In Fig. 8.4, we report a toy example with two classes. The selected method can be applied seamlessly in the pipeline we described previously in Fig. 8.1.

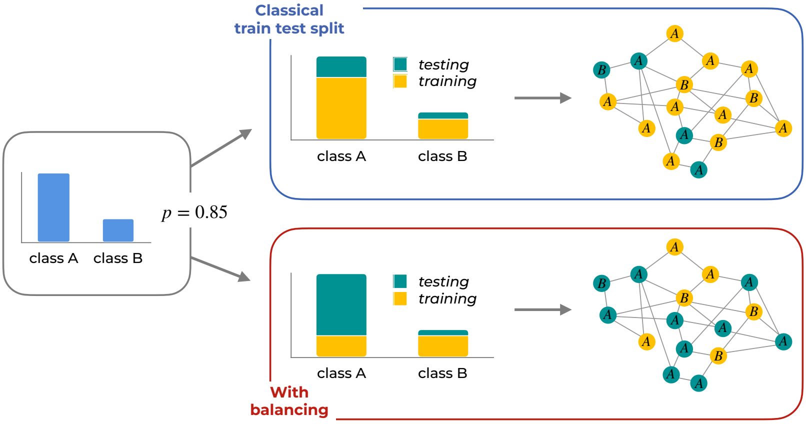


Fig. 8.4: *Train-test split with unbalanced classed. A visual example of an imbalanced dataset with 2 classes (A, B). On the top half, a representation of a classical 85/15 train-test split: in this case, the training set presents more examples of class A (9) than class B (3). On the lower half, we illustrate our proposed approach: we select 85% of the minority class B as training data, and the same number of examples is kept for the other classes. The obtained training set will present the same number of training nodes for each class (3).*

Experimental setting

In this Chapter, for both RQ1 and RQ2, we are interested in evaluating the performance of graph neural networks in the task of user migration. Performance can be evaluated with different *evaluation metrics*. We selected some of the most used metrics for multiclass classification problems, *accuracy* and *F1* [162]. Both metrics are computed from the evaluation of true positives (TP) and true negatives (TN) that represent the number of accurate classifications of positive and negative samples, while false positives (FP) and false negatives (FN) indicate the number of incorrect classifications of positive and negative samples. The $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ represents the relationship between observations correctly predicted and total observations. While the $F1 = \frac{2*TP}{2*TP+FP+FN}$, represents the average of $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$. However, in multiclass classification, with C classes and N samples, F1 can be adjusted to account for each class size, leading to the *weighted F1* $= \sum_{i=1}^{C} w_i * F1(C_i)$, as the weighted average of class-wise F1 scores, where for each class C_i, we have a weight $w_i = \frac{|C_i|}{N}$. The metrics are evaluated both on training and test sets: to obtain more robust results. It is common in the literature to consider the average performance over multiple random seeds for each combination, therefore we report the average over 3 random seeds as done in [174]. Through the selected metrics, we compare the predictive performance of graph neural networks to two baseline classifiers: the *Uniform Baseline classifier* that generates predictions uniformly at random (hence it will make a correct prediction in around 1/4 of the cases) and the *Most Frequent Baseline classifier*, which predicts always the most frequent class observed in the training set.

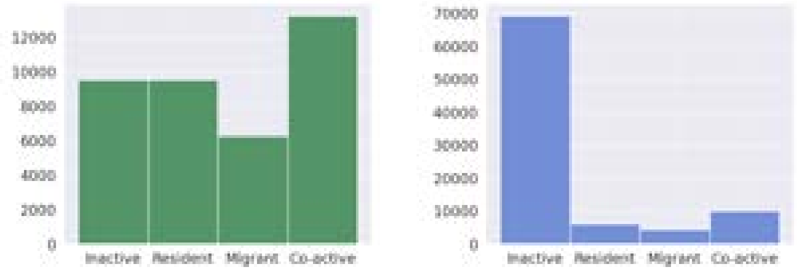
For RQ1, data is separated into training and test sets through a random train test split, with 70% of the nodes as a training set, and 30% of the nodes as a test set. Whereas we answer RQ2 by applying our under-sampling technique for balancing, generating various training and test sets, with different sizes. In this Chapter, for both RQ1 and RQ2, we are interested only in the impact of network structure. Therefore, node attributes from the dataset are not considered in the prediction: a constant attribute (equal to 1) is associated with each node. The weight update over the training in this Chapter is done by Adam optimizer [175].

8.5 Results

In this section, we present the graph and labels obtained by applying the graph preprocessing methodology shown earlier. Then we show how we apply the proposed methodology to answer our research question.

Graph and labels

Applying the proposed methodology on the Steem-Hive dataset , we obtain a multilayer graph $G_{t_{fork}}$. Note that $G_{t_{fork}}$ is the active users' subgraph, i.e. the subgraph induced by the set of active users on the two layers $r \in \{monetary(m), social(s)\}$. The monetary layer graphs $G^{m}_{t_{Fork}}$ contains $38,566$ nodes connected by $949,046$ edges. While the social layer graph $G^{s}_{t_{Fork}}$ has $90,055$ nodes and $42,556,877$ edges, Overall, the social layer has more active users and links: this is consistent with the selected operations; in fact, social operations are far more common than monetary transactions. For these users, we encoded their behavior in the 4 possible classes whose frequencies are shown in Fig. 8.5a for the monetary interactions and in Fig. 8.5b for social interactions. We can observe how the distribution of labels is not balanced: in the monetary layer, there is a slight skew in the number of *co-active* users, and the minority class is composed of *migrant* users. Whereas the social layer is severely imbalanced as the majority of users become *inactive* after the fork event.



(a) *Decisions on monetary layer* ***(b)*** *Decisions on social layer*

Fig. 8.5: *The distribution of the generated labels encoding the user migration decision, in the two layers a) monetary and b) social respectively.*

Predicting user migration

We now investigate whether graph neural networks are a suitable method for user migration prediction (RQ1) by applying the methodology presented in Section 8.4 on our dataset. We first train our models for prediction on the graph $G^{m}_{t_{Fork}}$ representing monetary interactions before the fork. In Table 8.1 we show the obtained results on the monetary layer. The trained GNN model surpasses both Baseline classifiers by a significant margin, both in terms of accuracy and weighted F1. These results indicate that the model can learn by exploiting only the topology derived from monetary interactions. We then perform the prediction task on the *social* layer $G^{s}_{t_{Fork}}$ that represents social interactions before the fork. In Table 8.2, we show the evaluation results. The gap between the trained model and the baseline classifiers is not as large. The most frequent baseline classifier (the one that predicts the most frequent class observed in training) obtains an accuracy score similar to the best GNN model, while the Uniform Baseline lags severely behind. When we consider the weighted F1 scores we observe a similar trend: the Baseline performs similarly to the GNN model. As the accuracy scores coincide with the percentage of the most frequent class for both the baseline and the GNN model, we investigated the predictive behavior of the best model; we discovered that after a few epochs, begins to predict always the same class, the most frequent class in the training set. In the case of a severely imbalanced dataset, the graph neural network model struggles in the prediction of less frequent classes, it acts similarly to the baseline classifier. In general, we can say that the GNN has learned from the input data, making it a suitable model for solving the problem on the monetary layer. While prediction in more imbalanced settings, like in the social graph requires addressing the class imbalance problem.

Table 8.1: *Accuracy and weighted F1 (mean and standard deviation over 3 random seeds [174]) obtained by the Baseline classifiers and the best GNN model on the monetary graph* $G^{m}_{t_{Fork}}$.

Model	Train accuracy	Test accuracy	Train weighted F1	Test weighted F1
Baseline most-frequent	0.346 ± 0.001	0.336 ± 0.003	0.178 ± 0.0011	0.169 ± 0.002
Baseline uniform	0.249 ± 0.001	0.249 ± 0.004	0.253 ± 0.001	0.2515 ± 0.001
Best model	**0.426 ± 0.003**	**0.424 ± 0.006**	**0.381 ± 0.002**	**0.379 ± 0.003**

Table 8.2: *Accuracy and weighted F1 (mean and standard deviation over 3 random seeds [174]) obtained by the Baseline classifiers and the best GNN model on the social graph* $G^{s}_{t_{Fork}}$ *.*

Model	Train accuracy	Test accuracy	Train weighted F1	Test weighted F1
Baseline most-frequent	**0.770 ± 0.001**	**0.770 ± 0.002**	**0.671 ± 0.001**	**0.670 ± 0.004**
Baseline uniform	0.250 ± 0.001	0.250 ± 0.001	0.319 ± 0.001	0.318 ± 0.001
Best model	**0.770 ± 0.001**	**0.770 ± 0.002**	**0.671 ± 0.001**	**0.670 ± 0.004**

Dealing with class imbalance

In the following, we now analyze how we can deal with class imbalance (RQ2) by applying the methodology presented in Section 8.4. We compare the best GNN model obtained on the two layers and compare the best model using the balancing approach.

We first make the comparison on the monetary layer: in Table 8.3 we report the evaluation metrics for both the approaches. The models that learned from the balanced graph and those that learned from the original graph have roughly the same performance. This is expected as the target variable is not very imbalanced in the monetary layer. Moreover, the fact that the performances are similar is an additional positive factor: the model trained on the balanced train set, actually learns from fewer examples, and yet does not lose in performance. In fact, we actually observe the opposite effect, with slight improvements overall. We then present the evaluation results obtained on the social layer: where target labels are more imbalanced, In Table 8.4. We can see the impact of the balancing technique we proposed. First, we verified that the model that learns from the balanced set actually returns as a prediction not just the most frequent class, but other classes as well. The model that learns on the balanced dataset exhibits a drop in both accuracy and weighted F1 over the training sets, however, the performance over the test sets is high, especially in terms of weighted F1.

Overall the balancing technique constitutes an improvement for the GNN model. In more balanced datasets, we are able to obtain good performance but training on fewer data, while on the more imbalanced datasets, it improves the learning phase for the model, which learns to better predict minority classes.

Table 8.3: *Accuracy and weighted F1 (mean and standard deviation over 3 random seeds [174]) obtained by the best GNN model trained on the imbalanced training set and the best model trained on the balanced training set, on the monetary graph* $G^m_{t_{Fork}}$ *.*

Model	**Train accuracy**	**Test accuracy**	**Train weighted F1**	**Test weighted F1**
Best model imbalanced	0.426 ± 0.003	0.424 ± 0.006	0.381 ± 0.002	0.379 ± 0.003
Best model balanced	**0.427 ± 0.001**	**0.424 ± 0.001**	**0.386 ± 0.007**	**0.382 ± 0.007**

Table 8.4: *Accuracy and weighted F1 (mean and standard deviation over 3 random seeds [174]) obtained by the best GNN model trained on the imbalanced training set and the best model trained on the balanced training set, on the social graph* $G^s_{t_{Fork}}$*.*

Model	**Train accuracy**	**Test accuracy**	**Train weighted F1**	**Test weighted F1**
Best model imbalanced	**0.770 ± 0.001**	**0.770 ± 0.002**	**0.671 ± 0.001**	0.670 ± 0.004
Best model balanced	0.403 ± 0.002	0.725 ± 0.006	0.359 ± 0.003	**0.788 ± 0.004**

8.6 Conclusion

In this Chapter, we addressed the problem of user migration prediction, focusing on some understudied aspects like the effectiveness of graph neural networks as a prediction method, as well as addressing the class imbalanced learning problem typically observed in classification tasks, and in blockchain-based systems. Our findings show that graph neural networks are an effective method to predict user migration in blockchain-based online social networks as our methodology, modeling user interaction data into multilayer temporal graphs suitable for graph neural network modeling, leads to a model able to leverage the graph of monetary interactions but struggling on the severely imbalanced social layer. However, after applying our proposed data-level balancing approach that produces a more balanced training set, graph neural networks show increased predictive power even on severely imbalanced data. The obtained performances are an important result since they highlight the predictive power of graph structure, without the need for manual feature engineering. Moreover, the trained models perform well even with a lack of node features, something that is typical of blockchain-based systems. Future research will look into the applications, as user migration is not limited to online social networks: leaving for another social, leaving for another crypto or other Dapp. The proposed methodology could lead to significant improvement in other prediction tasks typical of online social networks or blockchain-based systems such as fraud detection, and bot detection. Future additional works could focus on developing other balancing strategies.

Part V

Machine learning on multilayer graphs

Chapter 9

Simplifying graph structure for graph neural networks

9.1 Introduction

The graph analysis and mining research field has raised in popularity in the last two decades, thanks to the ability of graphs to model a wide range of real-life phenomena from physical [176] to biological [95] and social systems [177], from scientific [97] to financial data [98, 178], transportation routes [96], and many others [6]. In this regard, the multilayer graph model [92] is widely used as a powerful tool to represent the organization and relationships of complex systems covering many different domains. Multilayer graphs are designed to provide a more realistic representation of the different and heterogeneous relationships that may characterize an entity in the graph-structured system, using the rich data available from complex systems [40].

However, collecting a wide set of different relationships among a large set of entities can easily result in a significant amount of noise (e.g., incomplete, imprecise, or redundant information) caused by the choice regarding which entities and relations should be included in the data. Single-layer graphs already have this issue, known as the boundary specification problem [179], which is exacerbated in multilayer ones. While for the simplification of single-layer graphs, several machine learning techniques have already been proposed in the literature and have proved to be effective [42, 180], for multilayer graphs there are only unsupervised heuristics of preprocessing [40], while cutting-edge techniques such as graph neural networks have not yet been exploited. Furthermore, work on multilayer graph neural networks [181, 39] demonstrated how crucial it is to conceive approaches specially tailored for these complex structures, i.e., to produce embeddings that convey the rich information present in the input graph. As a matter of fact, the direct application of single-layer approaches to multilayer graphs is not trivial: while a single-layer approach could be applied on each layer separately, the important interplay among the various layers would be lost.

Based on these facts, we would like to understand: *RQ1*) What is the impact of graph simplification performed on multilayer graphs? *RQ2*) How does graph simplification influence the structure of multilayer graphs?

In this work, we propose the MultilAyer gRaph simplificAtion (MARA) framework, a GNN-based framework designed to simplify multilayer graphs based on the downstream task. MARA generates node embeddings for a specific task by training end-to-end two main components: i) an edge simplification module and ii) a (multilayer) graph neural network. We tested MARA under node classification on real-world multilayer graphs from different domains. Experimental results show the effectiveness of the proposed approach: MARA dramatically reduces the dimension of the input graph not only maintaining but also improving the performance (*RQ1*). In addition, MARA provides different approaches allowing us to provide insights on the most effective simplification strategy depending on the domain of the downstream task. In fact, with MARA, we enable simplification approaches that leverage single-layer simplification techniques on multilayer graphs but we also extend existing methods to work directly on multilayer graphs. Thus, MARA can select the appropriate simplification approach depending on the task. Moreover, we observe that deep learning driven simplification with MARA can influence and enhance important graph properties, such as label assortativity (*RQ2*): as the selection of task-irrelevant edges is refined during the training, MARA is guided in the selection of the most important properties to preserve or enhance. To our knowledge, MARA represents the first GNN-based simplification framework for multilayer graphs.

Due to the wide range of data that can be modeled as a multilayer graph, the proposed framework can have a large application room covering different fields like biology, physics, and health/medical analysis, where increased robustness is needed to address noise from data acquisition. Furthermore, data quality, computational performances, and information visualization are also crucial aspects of any process dealing with massive amounts of graph-structured data, such as social media mining, communication, biological, transportation, and financial systems.

9.2 Related work

In this section, we discuss related work regarding the use of graph neural networks for the analysis of multilayer graphs, and graph simplification approaches based on deep learning. Background information on the multilayer graph model adopted in this section can be found in Chapter 2. The notations used in this Chapter are summarized in Table 9.4.

Graph neural networks for multilayer graphs.

Deep learning tasks are more challenging on these graphs because of the presence of intra-layer and inter-layer relations, different layer characteristics, as well as node features. There have been some attempts to design methods and frameworks for deep learning for multilayer graphs. State-of-the-art results have been obtained by the framework presented in [39]. The framework reformulates the propagation rule of the GNN component (i.e. GCN or GAT) to aggregate topological neighborhood information from different layers. While in GCN, aggregation involves a node's features and its neighbors' features, in the ML-GCN the aggregation is performed with both its neighbors in that layer (dubbed within-layer neighborhood, denoted as $\mathit{\Gamma}(i,l)$) and on its neighbors located in other layers where the entity occurs (referred to as outside-layer neighborhood, denoted as $\mathit{\Psi}(i,l)$). More formally:

$$h_{(i,l)}^{(k+1)} = \sigma\left(\sum_{(j,m)\in\mathit{\Gamma}(i,l)\cup\mathit{\Psi}(i,l)} \frac{1}{\sqrt{\widetilde{D}_{ii}\widetilde{D}_{jj}}} h_{(j,m)}^{(k)} W^{(k+1)}\right) \tag{9.1}$$

Where $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}^{sup}$ where A_{ij}^{sup} is the supra-adjacency matrix with self-loops added.

Deep learning for graph simplification.

Graph simplification consists of removing uninformative or redundant edges while keeping almost all information of the input graph [41]. While there are many works on simplification [182], only a few are focused on simplification for deep learning on graphs. Dropedge [41] simplifies the graph for a GNN model (e.g. GCN, GAT) by randomly removing a fraction of the edges from the input graph during the training phase. The evaluation of Dropedge shows that even a random removal can lead to an improvement in performance across different tasks, such as node classification and link prediction. In NeuralSparse [42], the simplification process is done through the deep neural network: during the training phase, the deep neural network learns a simplification strategy that favors downstream tasks. In the testing phase, that neural network is used to select the edges to remove from the input graph, based on the learned strategy. Other works rely on similar principles. In AdaptiveGCN [43] simplification process is led by a deep neural network like in NeuralSparse, but a simplification step is performed before each graph convolution step. In PTDnet [183] additional constraints on the simplification process are introduced, encouraging the removal of more edges or prioritizing the simplification of edges connecting different node clusters. Other works such as [184] and [180] have designed frameworks for simplification with reinforcement learning. While there are several works on single-layer graph simplification, there is a lack of work relying on deep learning for the simplification of multilayer graphs.

9.3 Research questions

From the literature, it becomes clear that graph simplification has many advantages, such as the limitation of overfitting, that can lead to better performance and it also limits the effects of over-smoothing, thus allowing for deeper models [41]. But while there are several works on single-layer graph simplification, there is a lack of works relying on deep learning for the simplification of multilayer graphs, mainly because the applications of single-layer methods in the multilayer case are not straightforward. Given the benefits of graph simplification and the usefulness of the multilayer graph model, it is very important to fill this research gap. Therefore, in this work, we face the problem of understanding how we can apply the current deep learning based approaches designed for single-layer graphs to multilayer graphs. Among various aspects, we would like to see how graph simplification methods influence prediction performances, compared to single-layer cases. Moreover, we would like to deepen our understanding of the simplification process, especially when the methods can tune their selection strategy. These aspects can be summarized in the following research questions:
Research question RQ1: What is the impact of graph simplification performed on multilayer graphs?
Research question RQ2: How does graph simplification influence the structure of multilayer graphs?

9.4 The MARA framework

In order to answer our research questions, in this work, we want to provide a framework for the simplification of multilayer graphs and evaluate the impact of a simplification approach on a machine learning task. We want to evaluate the impact of graph simplification approaches on a typical machine learning task, i.e., node classification. In this section, we formally present the problem and the framework.

Problem definition.

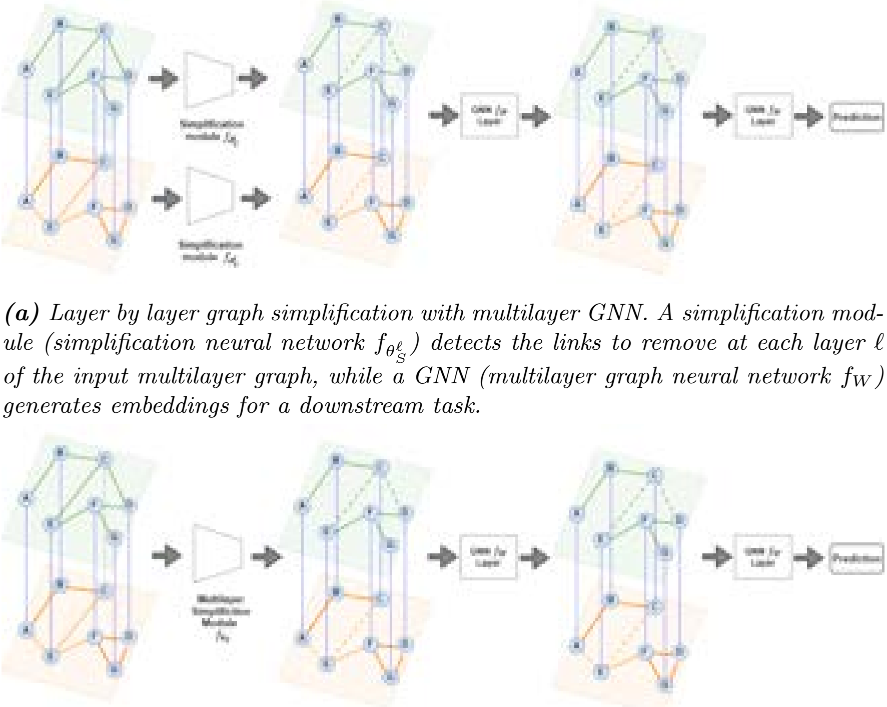
The graph simplification problem on single-layer graphs can be defined as follows: given a graph $G(V, E, X_E, X_V)$, where V is a set of n nodes, $E \subset V \times V$ is the set of edges; X_V is a set of node attributes, X_E is a set of edge attributes. simplification tries to obtain a subgraph of G, that would be $G' = G(V', E', X_E, X_V)$, where $V' \subset V \vee E' \subset E$ i.e the number of nodes and/or edges is reduced. Similarly, on a multilayer graph, simplification can be defined as the problem of obtaining a graph $f_{\theta_S}(\mathcal{G}_\mathcal{L}) = \mathcal{G}_\mathcal{L}{}' = (\mathcal{V}_\mathcal{L}{}', \mathcal{E}_\mathcal{L}{}', \mathcal{V}', \mathcal{L}')$ so that the number of nodes and/or edges is reduced. Formally, we are looking for a simplified multilayer graph $\mathcal{G}_\mathcal{L}{}'$. such that the following disjunction of conditions holds: $| \mathcal{V} |<| \mathcal{V}' | \vee | \mathcal{L} |<| \mathcal{L}' | \vee | \mathcal{V}_\mathcal{L} |<| \mathcal{V}_\mathcal{L}{}' | \vee | \mathcal{E}_\mathcal{L} |<| \mathcal{E}_\mathcal{L}{}' |$. In the following, we'll present the framework to compute the simplified multilayer graph.

Simplification approaches.

In order to perform graph simplification on a multilayer graph, we propose two approaches: i) Layer by layer graph simplification and ii) Multilayer graph simplification. We now present the two concepts behind them.

Layer by layer graph simplification. To perform graph simplification on a multilayer graph by exploiting methods for single-layer graphs, we can use a layer-by-layer approach. In the layer-by-layer simplification, methods are applied to each layer before recomposing the supra-adjacency matrix: cross-layer links are not involved. We can define a layer graph as $G[\ell]$ where every edge connects nodes in the same layer ℓ. Therefore, at each layer ℓ a simplification neural network $f_{\theta_S^\ell}$ detects noisy links over the layer-graph $G[\ell]$, generating a new version of the graph that we can define as $G[\ell]'$. The simplified graphs are used to update A'^{sup}, which will be used to train the graph neural network.

It's important to note that simplification can be applied at a different *stage*s of the process: we can simplify *once* or before *each* graph convolutional layer.


Simplification module $f_{\theta_S^1}$
Simplification module $f_{\theta_S^2}$
GNN f_W Layer
GNN f_W Layer
Prediction
Multilayer Simplification Module f_θ
GNN f_W Layer
GNN f_W Layer
Prediction

(a) Layer by layer graph simplification with multilayer GNN. A simplification module (simplification neural network $f_{\theta^\ell_S}$) detects the links to remove at each layer ℓ of the input multilayer graph, while a GNN (multilayer graph neural network f_W) generates embeddings for a downstream task.

(b) Multilayer graph simplification with multilayer GNN. A multilayer simplification module (simplification neural network f_θ) detects the links to remove by taking into account the whole input multilayer graph, while a GNN (multilayer graph neural network f_W) is used to generate node embeddings for a downstream task.

Fig. 9.1: *Overview of the proposed approaches for multilayer graph simplification: (a) layer-by-layer and (b) multilayer. Note that the difference between the two approaches lies in the simplification process, while the use of the GNN is the same.*

In the first case, a simplification module detects noisy elements while a graph neural network model is used to generate node embeddings for a downstream task. Here, simplification occurs only *once*, so that the graph is the same at each GNN layer. In the other case, at each GNN layer, a simplification module detects noisy elements while a graph neural network model generates node embeddings for a downstream task. The simplification is performed multiple times so that before *each* GNN layer, we are working on different versions of the graph.

MARA allows training with both simplification stages. The training phase for layer-by-layer graph simplification is summarized in algorithm 1.

Multilayer graph simplification. To define a simplification methodology conceived explicitly for a multilayer graph, able to properly take into account the complex structure of such models, we propose to use a simplification neural network f_θ that detects noisy edges and a graph neural network f_W to generate node embeddings for a downstream task (cf. Fig. 9.1b). The key difference with respect to the single-layer counterpart is that the simplification module is unique, and acts directly on the supra-adjacency matrix A^{sup} to generate the simplified A'^{sup}. Working directly on the supra-adjacency matrix also has an additional advantage: the simplification module can remove noisy or redundant cross-layer links as well. Even in the multilayer simplification case, simplification can be applied at different *stage*s: we can simplify *once* (i.e., the graph is the same at each GNN layer) or before *each* graph convolutional layer (i.e., the simplification is performed multiple times, so that each GNN layer works on a different version of the graph). The training phase for multilayer graph simplification is summarized in algorithm 2.

9.5 Experimental evaluation

Data.

For the experimental evaluation, we selected datasets from different domains showing different structural characteristics, summarized in Table 9.1. All of them correspond to multilayer graphs with associated real-world node features, a characteristic that can be leveraged by a simplification module to guide its decision process. The ***um-econ*** and ***um-socioeco*** [36] multilayer graphs are derived from the Steem-Hive dataset , describing the interactions in Steemit [185]. In these graphs nodes are users, and layers are interactions of different types. User features are graph-based metrics. User labels describe their migration to another social media platform, called Hive (4 cases: inactive, stay, leave, active on both). In *um-econ* is a subgraph composed of 2 layers of economic interactions, while *um-socioeco* considers interaction on 4 layers, 2 social and 2 economic. Note that the graphs used in this Chapter are a subgraph, where the nodes considered are only those active on every selected layer. ***IMDb-mlh*** [186] is a multilayer graph constructed from the IMDb movie database, where nodes are movies, and two movies are connected

Algorithm 1 Training algorithm for layer-by-layer simplification with ML GNN

```
Input: training graph $G(V, E, X_E, X_V)$, $\mathcal{L}$ multilayer graph layers, simplification
neural network $f_{\theta_S}$, simplification stage stage, number of GNN hidden layers $K$
Output: Embeddings for downstream task
if stage = "once" then                                        ▷ Simplify graph just once
    for layer $\ell \in 1...\mathcal{L}$ do
        $A'_\ell \leftarrow f_{\theta^\ell_S}(A_\ell)$ simplification function applied on $G[\ell]$
    end for
    $A'^{sup} \leftarrow$ Combine $A'_\ell$ in supra adjacency matrix
end if
for $k = 1...K$ do
    if stage = "each" then                                    ▷ Different graph every time
        for layer $\ell \in 1...\mathcal{L}$ do
            $A'_\ell \leftarrow f_{\theta^\ell_S}(A_\ell)$ simplification function applied on $G[\ell]$
        end for
        $A'^{sup} \leftarrow$ Combine $A'_\ell$ in supra adjacency matrix
    end if
    $H^k \leftarrow f^{(k-1)}_W(H^{(k-1)}, A'^{sup})$                  ▷ hidden representations update
end for
Backpropagation to update $\theta_W, \theta^\ell_S$
```

Algorithm 2 Training algorithm for multilayer simplification with ML GNN

```
Input: training graph $G(V, E, X_E, X_V)$, $\mathcal{L}$ multilayer graph layers, simplification
neural network $f_{\theta_S}$, simplification stage stage, number of GNN hidden layers $K$
Output: Embeddings for downstream task
if stage= "once" then                                         ▷ Simplify graph just once
    $A'^{sup} \leftarrow f_{\theta_S}(A^{sup})$                                    ▷ Simplify
end if
for $k = 1...K$ do
    if stage = "each" then                                    ▷ Different graph every time
        $A'^{sup} \leftarrow f_{\theta_S}(A^{sup})$                                ▷ Simplify
    end if
    $H^k \leftarrow f^{(k-1)}_W(H^{(k-1)}, A'^{sup})$                  ▷ hidden representations update
end for
Backpropagation to update $\theta_W, \theta_S$
```

if they share either an actor or a director. Movie features encode text from the plots, while the labels describe the movie type (action, comedy, drama). Finally ***Koumbia 2*** and ***Koumbia 5*** [187, 39] are multilayer graphs extracted from a time series of Sentinel-2[1] optical satellite images, covering the agricultural landscape of Koumbia in Burkina Faso. Nodes represent segments of the satellite image, and labels correspond to either crop (cultivated areas) or no-crop (uncultivated areas, such as forests) segments. Layers correspond to functional classes (e.g., temporal radiometric profiles). The network includes inter-layer edges and real-world attributes, corresponding to a time series of radiometric statistics for each segment. The graphs are generated with the geo2net framework[2], which allows the production of multilayer graphs from satellite images with an arbitrary number of layers: in this work, we consider 2 and 5 layers.

Table 9.1: *Summary of structural characteristics of the graph datasets: type of the graph, number of layers (L), number of nodes ($|V|$), number of edges ($|E|$), density (mean/SD) over the layers (d), and number of classes (C)*

dataset	L	$|V|$	$|E|$	d	C
imdb-mlh	2	5614	23208	0.0007 ± 0.0000	3
um-econ	2	15414	224855	0.0018 ± 0.0012	4
um-socioeco	4	18212	1199863	0.0138 ± 0.0118	4
Koumbia 2	2	4492	18783	0.0010 ± 0.0001	2
Koumbia 5	5	11230	91938	0.0010 ± 0.0002	2

Experimental setting.

In this work, we focus on node classification tasks, i.e., we learn the embeddings required to predict the label associated with each node in the graph. As GNN for MARA we select the GCN, but note that other GNNs could be employed. As a baseline, we consider a multilayer GNN without simplification (***GNN***). As previously discussed, MARA is flexible and can be equipped with different simplification strategies as well. In this work, we selected i) DropEdge [41] (MARA(DE)), a single-layer graph simplification method that randomly removes edges with probability p, and ii) NeuralSparse [42] (MARA(NS)), which is able to leverage node features to select a subset of edges to keep (a subgraph-based selection process is performed where for each node only k of its neighbors are kept and their connecting edges). Note that both approaches were originally designed for single-layer simplification, hence for this work, we implemented

[1] https://sentinel.esa.int/web/sentinel/missions/sentinel-2

[2] https://gitlab.irstea.fr/raffaele.gaetano/geo2net

extended versions in order to perform multilayer (*multi*) and layer-by-layer (*l-b-l*) simplification (cf. Section 9.4). Moreover, each implementation can be applied at different *stages*: we can simplify *once* or before *each* graph convolution layer (cf. Section 9.4). For MARA(DE), we test different drop rate probabilities $p = \{0.1, 0.3, 0.5, 0.7\}$, while for MARA(NS), we test different $k = \{5, 10.15\}$, with τ varying during training as in [42]. We perform all the experiments with a transductive learning setting like in [39]. In a transductive setting, all node attributes and topological information can be used for training, while only a subset of labels is visible to the GNN model. All models were trained using the Adam optimization algorithm [175] with full batch training [167], L2 weight regularization set to 0.0005. For each graph and method, the average accuracy was computed over $N = 3$ independent runs, where each run corresponded to a different train-validation-test split, with 25% of training entities as previously done in [39] and the rest split in validation (25%) and test entities (50%). The combination of hyperparameters with the best average validation metric is selected, and we report the final test metric. Since we are working on a huge number of possible combinations, we rely on early stopping, training for 250 epochs with 10 epochs of patience (reloading the best model). As an evaluation metric, we select AUC (Area under the ROC Curve) evaluated like in [42], because it is well suited for datasets showing unbalanced label distribution, such as *Imdb*, *um-econ* and *um-socioeco*.

9.6 Results

Framework evaluation.

We first focus on our research question RQ1 by focusing on the performance of the simplification methods. Table 9.2 reports the average AUC scores on the test set. We can observe how MARA generally improves upon the GNN baseline, and always corresponds to the best performances. Note that MARA(NS) almost consistently outperforms MARA(DE), demonstrating the importance of exploiting node features for the simplification task. The only exception is represented by *imdb-mlh*, where features information improves the performance, but the MARA(DE) variant obtains even better performance. Additional insights can be obtained by comparing the multilayer (*multi*) vs layer-by-layer (*l-b-l*) and the *once* vs *each* approaches. Regarding MARA(DE), we note that *multi* tends to be more effective on 2-layer graphs (i.e., *um-econ*, *um-socioeco* and *Koumbia-2*) while *l-b-l* seems to be more effective in presence of a greater number of layers. Note also that, with the (DE) variant, simplifying *once* tends to be the winning choice. This is consistent with the stochastic nature of this approach, i.e., repeating a random process at each layer may negatively impact the result. As concerns MARA(NS), *l-b-l* tends to be the best choice in most cases: it may be because the NeuralSparse simplification is based around a single-layer notion of a node's neighborhood. Devising an advanced strategy to properly take into account the multilayer neighborhood is left as future work. In terms of when to simplify (stage), for the task-aware (NS) variant, we can see that simplifying *once* brings better results for datasets showing an unbalanced distribution of the labels (i.e., *um-econ*, *um-socioeco* and *imdb-mlh*), while simplifying before *each* convolution layer seems the best approach for the more balanced *Koumbia* graphs.

Overall, MARA leads to significant performance improvements, while the variety of proposed approaches allows MARA to find the most suitable simplification approach for tasks of different domains.

Table 9.2: *AUC (mean and standard deviation over 3 random seeds [174]) obtained by the baseline and MARA.*

		data	um-econ	um-socioeco	imdb-mlh	Koumbia 2	Koumbia 5
model	simp	stage					
GNN	-	-	0.7420 ± 0.0022	0.6939 ± 0.0234	0.8035 ± 0.0218	0.9056 ± 0.0049	0.9237 ± 0.0033
MARA (DE)	multi	once	0.7451 ± 0.0128	0.6936 ± 0.0279	**0.8135 ± 0.0351**	0.9068 ± 0.0007	0.9228 ± 0.0041
		each	0.7487 ± 0.0150	0.6905 ± 0.0233	0.8122 ± 0.0324	0.9042 ± 0.0075	0.9246 ± 0.0069
	l-b-l	once	0.7407 ± 0.0083	0.6939 ± 0.0234	0.8005 ± 0.0253	0.9059 ± 0.0059	0.9252 ± 0.0063
		each	0.7418 ± 0.0102	0.6988 ± 0.0130	0.8079 ± 0.0280	0.9022 ± 0.0045	0.9238 ± 0.0051
MARA (NS)	multi	once	**0.7522 ± 0.0084**	0.6924 ± 0.0208	0.8011 ± 0.0299	0.9023 ± 0.0042	0.9223 ± 0.0138
		each	0.7458 ± 0.0107	0.6817 ± 0.0347	0.7987 ± 0.0257	0.9080 ± 0.0023	0.9244 ± 0.0093
	l-b-l	once	0.7438 ± 0.0113	**0.7199 ± 0.0099**	0.8077 ± 0.0260	0.9087 ± 0.0045	0.9205 ± 0.0022
		each	0.7457 ± 0.0008	0.7076 ± 0.0423	0.8046 ± 0.0249	**0.9103 ± 0.0052**	**0.9281 ± 0.0067**

Analysis of simplified graphs.

In this section, we discuss how the simplification impacts the structural characteristics of the multilayer graphs, providing an answer to research question RQ2. For each dataset, we compare structural characteristics before and after the simplification with MARA is performed. We show results for one of the prediction sub-tasks, user migration prediction on *um-econ* (Table 9.3) while the others can be consulted in Section 9.8. It can be noted how the impact of MARA(NS) can be different on each layer of a specific graph, while the action of MARA(DE) seems to be more uniform over a given graph. Once again, this is consistent with the fact that one approach leverages node features while the

Table 9.3: *Statistics for each graph layer before and after the simplification on um-econ dataset.*

	ℓ	Intra edges	Label assortativity	Transitivity	Indegree mean	Indegree max	Outdegree mean	Outdegree max
MARA (NS)		174381.00	0.08	0.01	23.63	3610.00	23.63	6021.00
	L0	6207.00	0.35	0.02	1.17	328.00	1.02	3.00
		(-96.44%)	(+320.57%)	(+86.70%)	(-95.06%)	(-90.91%)	(-95.69%)	(-99.95%)
		35060.00	0.27	0.00	5.55	937.00	5.55	4769.00
	L1	5038.00	0.62	0.02	0.87	145.00	1.02	3.00
		(-85.63%)	(+127.57%)	(+2947.23%)	(-84.39%)	(-84.53%)	(-81.71%)	(-99.94%)
MARA (DE)		174381.00	0.08	0.01	23.63	3610.00	23.63	6021.00
	L0	121999.00	0.08	0.01	16.53	2552.00	16.53	4230.00
		(-30.04%)	(+2.17%)	(-31.27%)	(-30.03%)	(-29.31%)	(-30.03%)	(-29.75%)
		35060.00	0.27	0.00	5.55	937.00	5.55	4769.00
	L1	24578.00	0.27	0.00	3.89	642.00	3.89	3349.00
		(-29.90%)	(-1.44%)	(-31.29%)	(-29.87%)	(-31.48%)	(-29.89%)	(-29.78%)

other is a random approach. The clearest impact is observed on the number of intra-edges, MARA drastically reduces the number of edges while still improving the performance: this is extremely important as the computation cost of graph convolution is linear in the number of graph edges [167], making a reduced number edges an ideal property. In addition, some interesting observations can be drawn about label assortativity, i.e., the similarity of connections in the graph with respect to node labels (high label assortativity means that a node is more likely to connect with a node with the same label). We can see how MARA(NS) tends to increase label assortativity across layers: this makes sense as MARA(NS) can leverage node features, so it would be able to preserve the connections between similar nodes. Such behavior cannot be replicated by the random procedure behind MARA(DE). Similarly, as regards transitivity (i.e., the fraction of all possible triangles present in a graph), we can observe a general decrease, since the number of triangles is necessarily reduced as we remove edges. However, on layers with lower transitivity (< 0.1), only MARA(NS) increases transitivity values: this can be observed in *um-econ* and *um-socioeco* (Supporting Information).

The relevance of training jointly simplification and graph neural network is, therefore, the most important observation: during the training, MARA(NS) improves its capacity to recognize edges that are unrelated to the task at hand, allowing it to determine which graph characteristics are most crucial to maintain or enhance. Additionally, with both variants, MARA demonstrates the capability of significantly reducing the number of edges while improving or at least keeping performance.

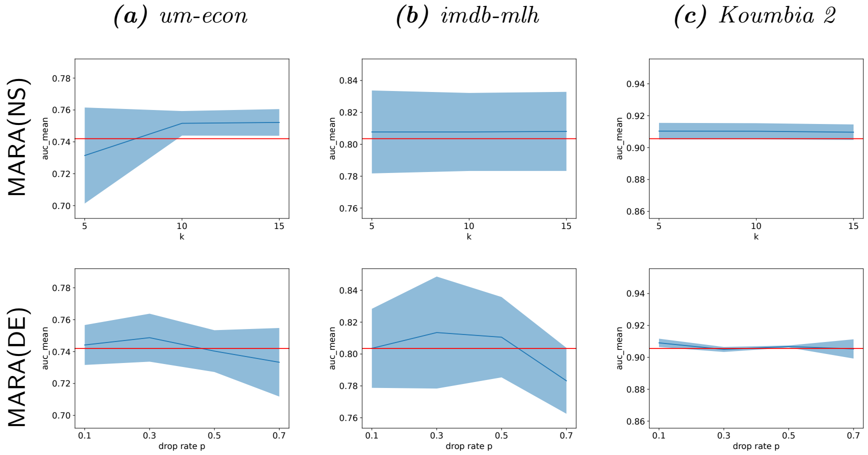

(a) um-econ *(b) imdb-mlh* *(c) Koumbia 2*

Fig. 9.2: *Sensitivity analysis based on AUC, for the 2-layer graphs. We compare the AUC for the baseline (in red), with the AUC (average, standard deviation) for the best simplification method (in blue). The remaining ones can be found in Section 9.8.*

Hyperparameters sensitivity analysis.

As a last analysis step, we study the impact of varying the main hyperparameters, i.e., the drop rate p for MARA(DE) and k for MARA(NS). While this is not directly related to our main research questions, it is an important step to further validate the results and conclusions. In Figure 9.2 we report a sensitivity analysis for p and k, where the other hyperparameters are set to the best-performing combination. We can see that for *um-econ*, low k values lead to lower performance. Similarly, a high drop rate of p seems to lead to worse performance. For *imdb-mlh*, we can draw similar observations for p (i.e., a high drop worsens the performance), while variations of k seem to have a minor impact on the process. Similarly, the impact of k is minor also for *Koumbia-2*. In this case, the impact of p seems to be reduced too.

Overall, the takeaway is that both MARA(NS) and MARA(DE) are not very sensitive to variations of their respective main hyperparameters k and p.

This makes their use more solid and easy, by making hyperparameter tuning relatively unimportant.

9.7 Conclusions

The findings presented in this Chapter show the significance of the proposed framework: MARA leads to significant performance improvements, by selecting the best available simplification strategies. These advances in performance are even more noteworthy when we take into account that MARA achieves them while drastically reducing the number of edges. Most importantly, MARA shows the importance of training jointly for the simplification of graphs for node classification tasks: as the ability to identify task-irrelevant edges increases, MARA is guided in discovering the most important graph properties to preserve or enhance.

Future research will focus on analyzing how multilayer simplification can be beneficial for a variety of tasks, including link prediction (removing unimportant or "spam" links to improve prediction performance), clustering (removing redundant links should improve boundaries between clusters, thus improving cluster quality), and graph classification (removing noisy links should help in the identification of similar graphs). Finally, additional future works will focus on the interaction between graph properties and downstream tasks to support multilayer simplification. A better understanding of graph properties can be beneficial in the development of simplification algorithms and overall it could lead to a better understanding of complex systems in different domains.

9.8 Supporting tables and images

Notation table

Table 9.4: *Summary of notations used in the Chapter and their description.*

Notations	**Description**
$\mathcal{G}_\mathcal{L}$	Multilayer graph
$\mathcal{V}$	Set of N entities (e.g., users)
$\mathcal{L}, \ell, L_l$	Set of layers, number of layers, l-th layer
$\mathcal{V}_\mathcal{L}$	Set of nodes in $\mathcal{G}_\mathcal{L}$
$\mathcal{E}_\mathcal{L}$	Set of edges $\mathcal{G}_\mathcal{L}$
A, A_ℓ	Adjacency matrix in G, Adjacency matrix of the l-th layer of $\mathcal{G}_\mathcal{L}$
A^{sup}	Supra-adjacency matrix
$\widetilde{A}, \widetilde{A}^{sup}$	Adjacency matrix and supra-adjacency matrix with self loops
v_i, i	Index i of a node $V_i \in \mathcal{V}_\mathcal{L}$
$\Gamma(i)$	Neighborhood of node V_i
$\Gamma(i, l)$	Within-layer neighborhood of node V_i
$\Psi(i, l)$	Outside-layer neighborhood of node V_i
X, X_l	Attribute (input feature) matrix, resp. in the l-th layer of $\mathcal{G}_\mathcal{L}$
$x, x_{(i,l)}$	Attribute (input feature) vector for node v_i, resp. node v_i in the l-th layer of $\mathcal{G}_\mathcal{L}$
f	Number of attributes (input features)
E	Edge attribute matrix
f_E	Number of edge attributes
$\mathcal{G}_{(\mathcal{L},\mathcal{X},\mathcal{E})}$	Attributed multilayer graph
d	Size of the embedding
Z, Z_l	Embedding (output feature) matrix, resp. in the l-th layer of $\mathcal{G}_\mathcal{L}$
$z_i, z_{(i,l)}$	Embedding (output feature) vector for node v_i, resp. node v_i in the l-th layer of $\mathcal{G}_\mathcal{L}$
W, W^k	Weight matrix of a generic, resp. weights of l-th GNN layers
$f_W, f_W^{(k)}$	GNN module, GNN at the k-th GNN layers
K, k	Number of GNN layers, index of a layer of the GNN
$H^{(k+1)} = f_W^{(k)}(H^{(k)}, A)$	A GNN layer computation
$f_{\theta_S}, f_{\theta_S}^k$	simplification neural network and its parameters, resp. simplification neural network for a certain GNN layer
h_i	Hidden layer vector for node v_i
$h_{(i,l)}^{(k)}$	Hidden layer vector at the k-th layer of the GNN for entity v_i in layer L_l of $\mathcal{G}_\mathcal{L}$
$Y, \hat{Y}$	Ground truth, predictions

Analysis of simplified graphs — datasets not included in the Chapter

Table 9.5: Statistics for each graph layer before and after the simplification on imdb-mlh dataset.

	ℓ	Intra edges	Label assortativity	Transitivity	Indegree mean	Indegree max	Outdegree mean	Outdegree max
MARA (NS)	L0	6121.00	0.70	0.40	4.27	79.00	4.27	79.00
		2818.00	0.87	0.29	3.09	42.00	3.09	40.00
		(-53.96%)	(+23.27%)	(-28.36%)	(-27.55%)	(-46.84%)	(-27.55%)	(-49.37%)
	L1	5355.00	0.72	0.38	4.00	69.00	4.00	69.00
		2816.00	0.90	0.00	3.09	42.00	3.09	38.00
		(-47.41%)	(+24.60%)	(-100.00%)	(-22.63%)	(-39.13%)	(-22.63%)	(-44.93%)
MARA (DE)	L0	6121.00	0.70	0.40	4.27	79.00	4.27	79.00
		4277.00	0.71	0.26	3.00	55.00	2.98	53.00
		(-30.13%)	(+1.44%)	(-34.14%)	(-29.80%)	(-30.38%)	(-30.27%)	(-32.91%)
	L1	5355.00	0.72	0.38	4.00	69.00	4.00	69.00
		3749.00	0.73	0.26	2.79	46.00	2.81	49.00
		(-29.99%)	(+0.47%)	(-29.83%)	(-30.22%)	(-33.33%)	(-29.71%)	(-28.99%)

Table 9.6: Statistics for each graph layer before and after the simplification on Koumbia 2 dataset.

	ℓ	Intra edges	Label assortativity	Transitivity	Indeg mean	Indegree max	Outdegree mean	Outdegree max
MARA (NS)	L0	5724.00	0.72	0.16	4.39	20.00	4.39	24.00
		2254.00	0.90	0.00	2.85	11.00	2.85	11.00
		(-60.62%)	(+24.26%)	(-100.00%)	(-35.18%)	(-45.00%)	(-35.18%)	(-54.17%)
	L1	4779.00	0.79	0.20	3.97	25.00	3.97	27.00
		2253.00	0.91	0.00	2.85	22.00	2.85	20.00
		(-52.86%)	(+15.07%)	(-100.00%)	(-28.32%)	(-12.00%)	(-28.32%)	(-25.93%)
MARA (DE)	L0	5724.00	0.72	0.16	4.39	20.00	4.39	24.00
		2909.00	0.72	0.08	2.21	13.00	2.22	15.00
		(-49.18%)	(-0.67%)	(-52.59%)	(-49.64%)	(-35.00%)	(-49.55%)	(-37.50%)
	L1	4779.00	0.79	0.20	3.97	25.00	3.97	27.00
		2356.00	0.79	0.09	1.97	13.00	1.97	17.00
		(-50.70%)	(+0.24%)	(-53.31%)	(-50.41%)	(-48.00%)	(-50.50%)	(-37.04%)

Table 9.7: *Statistics for each graph layer before and after the simplification on um-socioeco dataset.*

		Intra edges	Label Assortativity	Transitivity	Indegree mean	Indegree max	Outdegree mean	Outdegree max
MARA (NS)	L0	579352.00	0.06	0.20	130.25	1875.00	130.25	2990.00
		4624.00	0.14	0.00	4.02	31.00	4.02	5.00
		(-99.20%)	(+146.19%)	(-100.00%)	(-96.92%)	(-98.35%)	(-96.92%)	(-99.83%)
	L1	476439.00	0.05	0.11	107.64	1759.00	107.64	4262.00
		4652.00	0.15	0.01	4.02	34.00	4.02	6.00
		(-99.02%)	(+181.36%)	(-95.09%)	(-96.26%)	(-98.07%)	(-96.26%)	(-99.86%)
	L2	74580.00	0.12	0.01	19.38	2543.00	19.38	3753.00
		4603.00	0.44	0.03	4.01	331.00	4.01	5.00
		(-93.83%)	(+277.44%)	(+287.20%)	(-79.30%)	(-86.98%)	(-79.30%)	(-99.87%)
	L3	14856.00	0.39	0.01	6.26	276.00	6.26	701.00
		4586.00	0.73	0.00	4.01	66.00	4.01	5.00
		(-69.13%)	(+85.09%)	(-100.00%)	(-36.02%)	(-76.09%)	(-36.02%)	(-99.29%)
MARA (DE)	L0	579352.00	0.06	0.20	130.25	1875.00	130.25	2990.00
		492450.00	0.06	0.17	111.16	1601.00	111.16	2543.00
		(-15.00%)	(+0.84%)	(-15.26%)	(-14.65%)	(-14.61%)	(-14.65%)	(-14.95%)
	L1	476439.00	0.05	0.11	107.64	1759.00	107.64	4262.00
		404974.00	0.05	0.09	91.95	1493.00	91.95	3623.00
		(-15.00%)	(+0.57%)	(-15.12%)	(-14.58%)	(-15.12%)	(-14.58%)	(-14.99%)
	L2	74580.00	0.12	0.01	19.38	2543.00	19.38	3753.00
		63393.00	0.12	0.01	16.92	2173.00	16.92	3201.00
		(-15.00%)	(-0.89%)	(-14.71%)	(-12.68%)	(-14.55%)	(-12.68%)	(-14.71%)
	L3	14856.00	0.39	0.01	6.26	276.00	6.26	701.00
		12628.00	0.39	0.01	5.77	229.00	5.77	604.00
		(-15.00%)	(+0.24%)	(-14.64%)	(-7.81%)	(-17.03%)	(-7.81%)	(-13.84%)

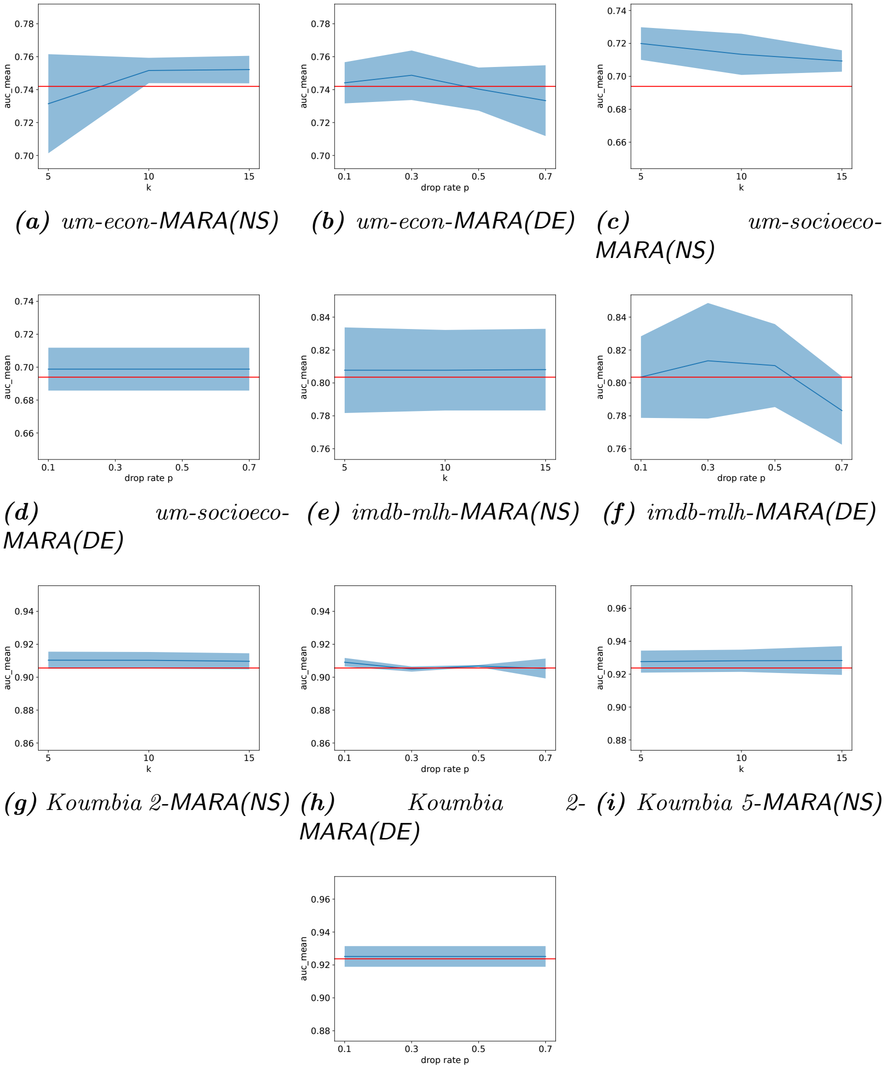
Table 9.8: Statistics for each graph layer before and after the simplification on Koumbia 5 dataset.

		Intra edges	Label Assortativity	Transitivity	Indegree mean	Indegree max	Outdegree mean	Outdegree max
MARA (NS)	L0	4157.00	0.84	0.25	7.15	33.00	7.15	38.00
		2252.00	0.95	0.00	6.30	28.00	6.30	28.00
		(-45.83%)	(+12.85%)	(-100.00%)	(-11.87%)	(-15.15%)	(-11.87%)	(-26.32%)
	L1	5752.00	0.70	0.22	9.03	39.00	9.03	36.00
		2249.00	0.90	0.00	7.47	27.00	7.47	25.00
		(-60.90%)	(+28.24%)	(-100.00%)	(-17.27%)	(-30.77%)	(-17.27%)	(-30.56%)
	L2	4951.00	0.70	0.22	8.64	47.00	8.64	53.00
		2252.00	0.88	0.00	7.44	41.00	7.44	41.00
		(-54.51%)	(+25.94%)	(-100.00%)	(-13.91%)	(-12.77%)	(-13.91%)	(-22.64%)
	L3	3635.00	0.98	0.24	6.82	39.00	6.82	42.00
		2252.00	1.00	0.00	6.21	37.00	6.21	36.00
		(-38.05%)	(+1.36%)	(-100.00%)	(-9.02%)	(-5.13%)	(-9.02%)	(-14.29%)
	L4	5605.00	0.68	0.20	9.29	48.00	9.29	50.00
		2266.00	0.87	0.04	7.80	41.00	7.80	41.00
		(-59.57%)	(+28.12%)	(-79.11%)	(-16.00%)	(-14.58%)	(-16.00%)	(-18.00%)
MARA (DE)	L0	4157.00	0.84	0.25	7.15	33.00	7.15	38.00
		3534.00	0.85	0.20	6.87	33.00	6.87	37.00
		(-14.99%)	(+0.65%)	(-18.32%)	(-3.88%)	(-)	(-3.88%)	(-2.63%)
	L1	5752.00	0.70	0.22	9.03	39.00	9.03	36.00
		4890.00	0.70	0.19	8.65	36.00	8.65	35.00
		(-14.99%)	(+0.28%)	(-16.06%)	(-4.25%)	(-7.69%)	(-4.25%)	(-2.78%)
	L2	4951.00	0.70	0.22	8.64	47.00	8.64	53.00
		4209.00	0.69	0.19	8.31	46.00	8.31	52.00
		(-14.99%)	(-1.12%)	(-13.45%)	(-3.82%)	(-2.13%)	(-3.82%)	(-1.89%)
	L3	3635.00	0.98	0.24	6.82	39.00	6.82	42.00
		3090.00	0.98	0.20	6.58	39.00	6.58	40.00
		(-14.99%)	(+0.18%)	(-14.39%)	(-3.56%)	(-)	(-3.56%)	(-4.76%)
	L4	5605.00	0.68	0.20	9.29	48.00	9.29	50.00
		4765.00	0.68	0.17	8.92	47.00	8.92	48.00
		(-14.99%)	(-0.38%)	(-14.34%)	(-4.03%)	(-2.08%)	(-4.03%)	(-4.00%)

Hyperparameter tuning — parameter space

```
{'datasets': [('um-econ', 'features'),
  ('um-socioeco', 'features'),
  ('imdb-mlh', 'features'),
  ('Koumbia_2', 'features'),
  ('Koumbia_5', 'features')],
 'architecture': ['multi'],
 'architecture_simp': ['multi', 'single'],
 'model': ['gcn', 'gcn-de', 'gcn-ns'],
 'gnn_level': [True, False],
 'drop_rate_p': [0.1, 0.3, 0.5, 0.7],
 'k': [5, 10, 15],
 'tau': [0.001],
 'standardize': [True],
 'feat-variability': ['fixed'],
 'split': ['25 50 25'],
 'plots': [True],
 'early-stop': [True],
 'fastmode': [True],
 'gpu': [1],
 'run': [1],
 'debugging': [False],
 'dropout': [0.3],
 'hidden': [16, 32],
 'lr': [0.002],
 'num-layers': [2],
 'ns_num_hidden': [32],
 'epochs': [250],
 'patience': [10]}
```

Hyperparameters sensitivity analysis — all datasets



(a) *um-econ-MARA(NS)* ***(b)*** *um-econ-MARA(DE)* ***(c)*** *um-socioeco-MARA(NS)*

(d) *um-socioeco-MARA(DE)* ***(e)*** *imdb-mlh-MARA(NS)* ***(f)*** *imdb-mlh-MARA(DE)*

(g) *Koumbia 2-MARA(NS)* ***(h)*** *Koumbia 2-MARA(DE)* ***(i)*** *Koumbia 5-MARA(NS)*

(j) *Koumbia 5-MARA(DE)*

Fig. 9.3: *Hyperparameters sensitivity analysis — all datasets*

Chapter 10

Conclusions and future works

In this thesis, we have addressed a series of open problems concerning the comprehension of the novel Web3 paradigm. Recalling the main topics presented in the introduction, the main contributions of the works presented in this thesis can be categorized as follows:

Modeling Web3. In Part I we focused on an extensive background in Web3 platforms and details on the datasets retrieved for our works before delving into methodologies to model Web3 data. A high level of dynamicity characterizes the field as new building blocks in terms of consensus mechanisms, tokens, and smart contracts are proposed as well as new platforms. From the analysis we conducted on some of the main applications of Web3 in different fields, we observed how Web3 developers have a series of important design choices, that determine the characteristics of the platform. The key takeaway is that every platform may have some of these features, not all of them, so a one-size-fits-all modeling may bring unnecessary levels of complexity. Throughout this thesis, we observed how a network-based approach provides the right amount of flexibility in this regard, as the choice of the model can be adjusted to the dataset and the problem investigated. What is currently missing are more datasets on the applications of Web3 that go beyond the purely financial field, which is caused by the complexity of collecting, storing, and analyzing such large-scale data. During our work, we contributed by collecting the Steem-Hive dataset , covering the field of online social media. However, collecting datasets from other domains would allow us to deepen their characteristics and compare them with the available ones.

Network evolution dynamics. In Part II we focused on the evolution of Web3 systems. The temporal component is extremely important, especially given the fast-changing nature of the new platforms, therefore we focused on dynamical aspects and mechanisms. An interesting result is how some of the most important mechanisms observed in traditional systems, including the ones from Web 2.0, are still observable in the new platforms, although with some differences. For example, bursty behavior characterizes Web3 social networks like traditional ones, but when we consider different interaction types we see how each process has different parameters. Similarly, we saw how triadic closure is an important mechanism in Web3 as well, but triadic closure happens much faster compared to Web 2.0. Therefore, we now have supporting evidence that established properties and mechanisms found in Web 2.0 can be observed in Web3. Therefore, our work on the topic could be extended and deepened in various directions.

The interplay of currency and user behavior. In Part III we analyzed the interplay between users and the cryptocurrency or reward systems. The presence of this disruptive element is a crucial difference with Web 2.0, adding an additional level of complexity but also providing an opportunity to deepen our understanding of human behavior. From our analysis conducted in chapter 5, we confirmed the hypothesis that there is an interplay between those aspects, with the economic dimensions influencing user activity levels, particularly on actions that shape the structure of social networks. We also observed in chapter 6 how we can monitor the activity levels over time, something that is important as user behavior can quickly change over time. There is still a severe lack of studies focusing on the interplay of social activity and economic dimensions, especially from a temporal standpoint. Moreover, we also observed how this interplay could have unexpected effects: for example in Steemit, when studying the reward mechanism, we saw how the most successful users seem to prefer actions to the promotion of content rather than the creation of high-quality content, exploiting the reward distribution mechanisms implemented by the platform. Similarly, in Sarafu, we observe how not only the pandemic situation, but some system design choices influenced user behavior. Given the importance of cryptocurrency-based systems and reward mechanisms, it is critical that their characteristics do not remain understudied in the field. We plan to analyze more platforms to quantify the importance of the currency in other Web3 platforms, with the hope of clarifying their positive and negative impact.

Modeling and prediction of user migration. In Part IV, we focused on the study of user migration across platforms. While previous works had shown the usefulness of a network-based approach in traditional platforms, we now have a model suited for Web3 platform characteristics ranging from the heterogeneity of possible interactions to the movement across platforms. Moreover, we assessed the predictability of user migration showing how network structure can be leveraged to predict users' decisions, and that in a stratified context

where social and economic relationships are mixed, both dimensions are important in describing and forecasting users' behaviors. What is missing in the field is potentially extending the methodology to provide some explanations of user behavior, either based on their activity or their content production.
Machine learning on multilayer graphs. In Part V we shifted our attention to deep learning methods. Given the previous results, it becomes evident that representing through more complex models that are able to describe different interaction types, like multilayer networks is of paramount importance. Approaches for deep learning of multilayer graphs are still developing, but our works show how these models can be applied effectively for Web3-related tasks. Therefore, we plan to explore new avenues and tasks like link prediction, anomaly detection, and especially in settings where cross-chain behavior is present.

In summary, from our analysis, we saw how the Web3 landscape reveals numerous promising avenues, with significant open problems that remain unexplored. The introduction of the economic facet adds a challenging level of complexity to analyses when compared with conventional Web 2.0 platforms. However, this complexity simultaneously presents a rich and exciting opportunity for in-depth exploration. The examination of novel Web3 applications not only sparks intriguing research questions but also prompts a reevaluation of established open problems, providing a platform to assess the efficacy of current models and theories or to construct more comprehensive and innovative ones. Moreover, we have seen how the influence of Web3 extends well beyond the realm of data analysis, permeating into diverse and dynamic fields such as behavioral science, machine learning, economics, sociology, and beyond. Web3 is characterized by a profound and multidisciplinary impact, presenting a dynamic and stimulating landscape for researchers and practitioners to explore and contribute to various domains.

References

1. L. La Cava, S. Greco, A. Tagarelli, Understanding the growth of the fediverse through the lens of mastodon, Applied Network Science 6 (1) (2021) 1–35.
2. H. Bin Zia, A. Raman, I. Castro, I. Hassan Anaobi, E. De Cristofaro, N. Sastry, G. Tyson, Toxicity in the decentralized web and the potential for model sharing, Proceedings of the ACM on Measurement and Analysis of Computing Systems 6 (2) (2022) 1–25.
3. S. Voshmgir, Token economy: How the Web3 reinvents the internet, Vol. 2, Token Kitchen, 2020.
4. B. Guidi, An overview of blockchain online social media from the technical point of view, Applied Sciences 11 (2021) 9880. doi:10.3390/app11219880.
5. A.-L. Barabási, Network science, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371 (1987) (2013) 20120375.
6. L. D. F. Costa, J. Oliveira, Osvaldo, G. Travieso, F. A. Rodrigues, P. Villas Boas, L. Antiqueira, M. P. Viana, L. Correa Rocha, Analyzing and modeling real-world phenomena with complex networks: a survey of applications, Advances in Physics 60 (3) (2011) 329–412.
7. M. Coscia, The atlas for the aspiring network scientist (2021). arXiv:2101.00863.
8. M. Karsai, H.-H. Jo, K. Kaski, et al., Bursty human dynamics, Springer, New York, NY, 2018.
9. S. Gaito, M. Zignani, G. P. Rossi, A. Sala, X. Zhao, H. Zheng, B. Y. Zhao, On the bursty evolution of online social networks, in: Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research, ACM, New York, NY, 2012, pp. 1–8.
10. B. Guidi, When blockchain meets online social networks, Pervasive and Mobile Computing 62 (2020) 101131.
11. D. Easley, J. Kleinberg, et al., Networks, crowds, and markets, Vol. 8, Cambridge university press Cambridge, 2010.
12. D. Romero, J. Kleinberg, The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter, Proceedings

of the International AAAI Conference on Web and Social Media 4 (1) (2010) 138–145. doi:10.1609/icwsm.v4i1.14015.
URL https://ojs.aaai.org/index.php/ICWSM/article/view/14015

13. G. Bianconi, R. K. Darst, J. Iacovacci, S. Fortunato, Triadic closure as a basic generating mechanism of communities in complex networks, Physical Review E 90 (4) (2014) 042806.
14. H. Huang, J. Tang, L. Liu, J. Luo, X. Fu, Triadic closure pattern analysis and prediction in social networks, IEEE Transactions on Knowledge and Data Engineering 27 (12) (2015) 3374–3389. doi:10.1109/TKDE.2015.2453956.
15. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, Network motifs: simple building blocks of complex networks, Science 298 (5594) (2002) 824–827.
16. L. Kovanen, M. Karsai, K. Kaski, J. Kertész, J. Saramäki, Temporal motifs in time-dependent networks, Journal of Statistical Mechanics: Theory and Experiment 2011 (11) (2011) P11005. doi:10.1088/1742-5468/2011/11/p11005.
URL http://dx.doi.org/10.1088/1742-5468/2011/11/P11005
17. A. Paranjape, A. R. Benson, J. Leskovec, Motifs in temporal networks, CoRR abs/1612.09259 (2016). arXiv:1612.09259.
URL http://arxiv.org/abs/1612.09259
18. C. T. Ba, M. Zignani, S. Gaito, The role of cryptocurrency in the dynamics of blockchain-based social networks: the case of steemit, PloS One (2022).
19. M. Nadini, L. Alessandretti, F. Di Giacinto, M. Martino, L. M. Aiello, A. Baronchelli, Mapping the nft revolution: market trends, trade networks, and visual features, Scientific reports 11 (1) (2021) 1–11.
20. L. Ussher, L. Ebert, G. M. Gómez, W. O. Ruddick, Complementary currencies for humanitarian aid, Journal of Risk and Financial Management 14 (11) (2021) 557.
21. M. Zignani, S. Gaito, G. P. Rossi, X. Zhao, H. Zheng, B. Zhao, Link and triadic closure delay: Temporal metrics for social network dynamics, Proceedings of the International AAAI Conference on Web and Social Media 8 (1) (2014) 564–573. doi:10.1609/icwsm.v8i1.14507.
URL https://ojs.aaai.org/index.php/ICWSM/article/view/14507
22. C. Li, B. Palanisamy, Incentivized blockchain-based social media platforms: A case study of steemit, in: Proceedings of the 10th ACM Conference on Web Science, WebSci19, 2019, pp. 145–154.
23. R. Zhang, J. Park, R. Ciriello, The differential effects of cryptocurrency incentives in blockchain social networks, in: Conference: SIGBPS2019 - Pre-ICIS Workshop on Blockchain and Smart ContractAt: Munich, Germany, 2019, pp. 1–5.
24. M. Thelwall, Can social news websites pay for content and curation? the steemit cryptocurrency model, Journal of Information Science 44 (2018) 736–751.
25. R. Adams, B. Kewell, G. Parry, Blockchain for Good? Digital Ledger Technology and Sustainable Development Goals, Springer International Publishing,

Cham, 2018, Ch. 1, pp. 127–140. doi:10.1007/978-3-319-67122-2_7.
URL https://doi.org/10.1007/978-3-319-67122-2_7

26. L. Doria, L. Fantacci, Evaluating complementary currencies: from the assessment of multiple social qualities to the discovery of a unique monetary sociality, Quality & Quantity 52 (2018) 1291–1314.
27. A. Michel, M. Hudon, Community currencies and sustainable development: A systematic review, Ecological economics 116 (2015) 160–171.
28. M. Fare, P. O. Ahmed, Complementary currency systems and their ability to support economic and social changes, Development and Change 48 (5) (2017) 847–872.
29. T. Criscione, E. Guterman, S. Avanzo, J. Linares, Community currency systems: Basic income, credit clearing, and reserve-backed. models and design principles, Tech. rep., FRIBIS Discussion Paper Series (2022).
30. W. O. Ruddick, Sarafu Community Inclusion Currency, 2020-2021 uk data service reshare (Aug. 2021). doi:10.5255/UKDA-SN-855142.
URL https://reshare.ukdataservice.ac.uk/855142/
31. C. E. Mattsson, T. Criscione, W. O. Ruddick, Sarafu community inclusion currency 2020–2021, Scientific data 9 (1) (2022) 1–13.
32. E. Newell, D. Jurgens, H. Saleem, H. Vala, J. Sassine, C. Armstrong, D. Ruths, User migration in online social networks: A case study on reddit during a period of community unrest, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 10, 2016, pp. 279–288.
33. S. Kumar, R. Zafarani, H. Liu, Understanding user migration patterns in social media, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 25, 2011, pp. 1204–1209.
34. M. Senaweera, R. Dissanayake, N. Chamindi, A. Shyamalal, C. Elvitigala, S. Horawalavithana, P. Wijesekara, K. Gunawardana, M. Wickramasinghe, C. Keppitiyagama, A weighted network analysis of user migrations in a social network, in: 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), IEEE, 2018, pp. 357–362.
35. C. Davies, J. Ashford, L. Espinosa-Anke, A. Preece, L. Turner, R. Whitaker, M. Srivatsa, D. Felmlee, Multi-scale user migration on reddit, in: Workshop on Cyber Social Threats at the 15th International AAAI Conference on Web and Social Media (ICWSM 2021), AAAI, AAAI, 2021, pp. 1–9.
36. C. T. Ba, A. Michienzi, B. Guidi, M. Zignani, L. Ricci, S. Gaito, Fork-based user migration in blockchain online social media, in: Proceedings of the 14th ACM conference on web science, WebSci '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 174–184. doi:10.1145/3501247.3531597.
URL https://doi.org/10.1145/3501247.3531597
37. W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Advances in neural information processing systems 30 (2017).
38. H. Kaur, H. S. Pannu, A. K. Malhi, A systematic review on imbalanced data challenges in machine learning: Applications and solutions, ACM Computing Surveys (CSUR) 52 (4) (2019) 1–36.

39. L. Zangari, R. Interdonato, A. Calió, A. Tagarelli, Graph convolutional and attention models for entity classification in multilayer networks, Applied Network Science 6 (1) (2021) 1–36.
40. R. Interdonato, M. Magnani, D. Perna, A. Tagarelli, D. Vega, Multilayer network simplification: approaches, models and methods, Comput. Sci. Rev. 36 (2020) 100246.
41. Y. Rong, W. Huang, T. Xu, J. Huang, Dropedge: Towards deep graph convolutional networks on node classification, in: ICLR, 2020, pp. 1–18.
42. C. Zheng, B. Zong, W. Cheng, D. Song, J. Ni, W. Yu, H. Chen, W. Wang, Robust graph representation learning via neural sparsification, in: H. D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, Vol. 119 of Proceedings of Machine Learning Research, PMLR, 2020, pp. 11458–11468.
    URL https://proceedings.mlr.press/v119/zheng20d.html
43. D. Li, T. Yang, L. Du, Z. He, L. Jiang, Adaptivegcn: Efficient gcn through adaptively sparsifying graphs, Proceedings of the 30th ACM International Conference on Information & Knowledge Management (2021).
44. M. Di Pierro, What is the blockchain?, Computing in Science & Engineering 19 (5) (2017) 92–95.
45. P. Zhang, D. C. Schmidt, J. White, A. Dubey, Consensus mechanisms and information security technologies, Advances in Computers 115 (2019) 181–209.
46. S. Aggarwal, N. Kumar, Cryptographic consensus mechanisms, in: Advances in Computers, Vol. 121, Elsevier, 2021, pp. 211–226.
47. Ethereum development documentation (2023).
    URL https://ethereum.org
48. B. Lashkari, P. Musilek, A comprehensive review of blockchain consensus mechanisms, IEEE Access 9 (2021) 43620–43652.
49. S. Sayeed, H. Marco-Gisbert, Assessing blockchain consensus and security mechanisms against the 51% attack, Applied Sciences 9 (9) (2019) 1788.
50. S. Nakamoto, et al., Bitcoin: A peer-to-peer electronic cash system, Decentralized Business Review (2008) 21260.
51. S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, S. Savage, A fistful of bitcoins: characterizing payments among men with no names, in: Proceedings of the 2013 conference on Internet measurement conference, 2013, pp. 127–140.
52. Steemit BluePaper (2020).
    URL https://steem.com/steem-bluepaper.pdf
53. V. Buterin, et al., A next-generation smart contract and decentralized application platform, white paper 3 (37) (2014).
54. E. Kapengut, B. Mizrach, An event study of the ethereum transition to proof-of-stake, Commodities 2 (2) (2023) 96–110.
55. The Merge (2023).
    URL https://ethereum.org

56. V. Buterin, What proof of stake is and why it matters, Bitcoin Magazine 26 (2013).
57. D. Larimer, Dpos consensus algorithm—the missing whitepaper, steemit, 2018 (2018).
58. C. Li, B. Palanisamy, R. Xu, L. Duan, Cross-consensus measurement of individual-level decentralization in blockchains, in: 2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), IEEE, 2023, pp. 45–50.
59. W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, D. I. Kim, A survey on consensus mechanisms and mining strategy management in blockchain networks, Ieee Access 7 (2019) 22328–22370.
60. History and Forks of Ethereum (2023).
URL `https://ethereum.org`
61. A. M. Antonopoulos, G. Wood, Mastering ethereum: building smart contracts and dapps, O'reilly Media, 2018.
62. W. Entriken, D. Shirley, J. Evans, N. Sachs, EIP-721: Non-Fungible Token Standard (Jan. 2018).
URL `https://eips.ethereum.org/EIPS/eip-721`
63. P. Freni, E. Ferro, G. Ceci, Fixing social media with the blockchain, in: Proceedings of the 6th EAI international conference on smart objects and technologies for social good, 2020, pp. 175–180.
64. L. Liu, W. Zhang, C. Han, A survey for the application of blockchain technology in the media, Peer-to-Peer Networking and Applications 14 (5) (2021) 3143–3165.
65. M. S. Kim, J. Y. Chung, Sustainable growth and token economy design: The case of steemit, Sustainability 11 (1) (2019) 167.
66. T. H. Davenport, J. C. Beck, The attention economy, Ubiquity 2001 (May) (May 2001). `doi:10.1145/375348.376626`.
URL `https://doi.org/10.1145/375348.376626`
67. Steemit, Steemit Whitepaper, Online (2020).
URL `https://steem.com/steem-whitepaper.pdf`
68. B. Dale, Justin Sun Bought Steemit. Steem Moved to Limit His Power (Feb. 2020).
URL `https://www.coindesk.com/tech/2020/02/24/justin-sun-bought-steemit-steem-moved-to-limit-his-power/`
69. G. Smith, Steemit for Sale: Popular Crypto Blogging Platform Sold to Tron, Community Reacts (Feb. 2020).
URL `https://news.bitcoin.com/steemit-for-sale-tron/`
70. Steem, Steem Consensus Witness Statement: Code Updated (Feb. 2020).
URL `https://steemit.com/steem/@softfork222/soft-fork-222`
71. B. Dale, Steem Community Mobilizes Popular Vote in Battle With Justin Sun (Mar. 2020).

URL https://www.coindesk.com/tech/2020/03/03/steem-community-mobilizes-popular-vote-in-battle-with-justin-sun/

72. B. Dale, Steem Community Plans Hostile Hard Fork to Flee Justin Sun's Steemit (Mar. 2020).
URL https://www.coindesk.com/tech/2020/03/17/steem-community-plans-hostile-hard-fork-to-flee-justin-suns-steemit/
73. Hive Hard Fork is Successful, STEEM Crashes Back to Earth (Mar. 2020).
URL https://cointelegraph.com/news/hive-hard-fork-is-successful-steem-crashes-back-to-earth
74. J. D. Sachs, From millennium development goals to sustainable development goals, The lancet 379 (9832) (2012) 2206–2211.
75. B. X. Lee, F. Kjaerulf, S. Turner, L. Cohen, P. D. Donnelly, R. Muggah, R. Davis, A. Realini, B. Kieselbach, L. S. MacGregor, et al., Transforming our world: implementing the 2030 agenda through sustainable development goal indicators, Journal of public health policy 37 (2016) 13–31.
76. N. Deepa, Q.-V. Pham, D. C. Nguyen, S. Bhattacharya, B. Prabadevi, T. R. Gadekallu, P. K. R. Maddikunta, F. Fang, P. N. Pathirana, A survey on blockchain for big data: approaches, opportunities, and future directions, Future Generation Computer Systems (2022).
77. B. Tomlinson, J. Boberg, J. Cranefield, D. Johnstone, M. Luczak-Rösch, D. J. Patterson, S. Kapoor, Analyzing the sustainability of 28 'blockchain for good' projects via affordances and constraints, Information Technology for Development 27 (2021) 439–469.
78. A. Zwitter, M. Boisse-Despiaux, Blockchain for humanitarian action and development aid, Journal of International Humanitarian Action 3 (2018) 1–7.
79. D. Dreer, A. Weeger, Block by block–how immutable ledgers drive sustainability efforts: An organizing literature review on the capabilities of the blockchain technology in supply chains, in: Proceedings of the 2022 ACM Conference on Information Technology for Social Good, 2022, pp. 269–275.
80. B. Guidi, L. Ricci, R. Nyffenegger, R. Ribback, Helios cj app: The decentralization of the citizen journalism, in: Proceedings of the Conference on Information Technology for Social Good, GoodIT '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 31–36. doi:10.1145/3462203.3475901.
URL https://doi.org/10.1145/3462203.3475901
81. A. Rugeviciute, A. Mehrpouya, Blockchain, a panacea for development accountability? a study of the barriers and enablers for blockchain's adoption by development aid organizations, Frontiers Blockchain 2 (2019) 15.
82. P. Dumitriu, Blockchain applications in the united nations system: Towards a state of readiness, United Nations, Report of the Joint Inspection Unit (2020).
83. D. Reppas, G. Muschert, The potential for community and complementary currencies (ccs) to enhance human aspects of economic exchange, Digithum (2019).
84. A. Clark, A. Mihailov, M. Zargham, Complex systems modeling of community inclusion currencies, Computational Economics (2023) 1–36.

85. L. Gonzalez, A. K. Cernev, M. H. d. Araujo, E. H. Diniz, Moedas complementares digitais e políticas públicas durante a crise da covid-19, Revista de Administração Pública 54 (4) (2020) 1146–1160. doi:10.1590/0034-761220200234.
URL https://doi.org/10.1590/0034-761220200234

86. N. Stepnicka, G. Zimon, D. Brzozowiec, The complementary currency zielony in poland and its importance for the development of local economy entities during the covid-19 pandemic lockdown, Sustainability (2021).

87. Grassroots Economics (2023).
URL https://www.grassrootseconomics.org/pages/about-us

88. A. C. An, P. T. X. Diem, L. T. T. Lan, T. V. Toi, L. D. Q. Binh, Building a product origins tracking system based on blockchain and poa consensus protocol, 2019 International Conference on Advanced Computing and Applications (ACOMP) (2019) 27–33.

89. M. La Morgia, A. Mei, A. M. Mongardini, E. N. Nemmi, Nft wash trading in the ethereum blockchain, arXiv preprint arXiv:2212.01225 (2022).

90. W. Chan, A. Olmsted, Ethereum transaction graph analysis, in: 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), 2017, pp. 498–500.

91. C. D. T. Barros, M. R. F. Mendonça, A. B. Vieira, A. Ziviani, A survey on embedding dynamic graphs, ACM Comput. Surv. 55 (1) (nov 2021). doi:10.1145/3483595.
URL https://doi.org/10.1145/3483595

92. M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, M. A. Porter, Multilayer networks, Journal of complex networks 2 (3) (2014) 203–271.

93. M. Magnani, O. Hanteer, R. Interdonato, L. Rossi, A. Tagarelli, Community detection in multiplex networks, arXiv preprint arXiv:1910.07646 (2019).

94. E. Almaas, B. Kovacs, T. Vicsek, Z. N. Oltvai, A.-L. Barabási, Global organization of metabolic fluxes in the bacterium escherichia coli, Nature 427 (6977) (2004) 839–843.

95. W. Thompson, P. Brantefors, P. Fransson, From static to temporal network theory: Applications to functional brain connectivity, Network Neuroscience 1 (2017) 1–56. doi:10.1162/NETN_a_00011.

96. J. L. Curzel, R. Lüders, K. V. O. Fonseca, M. Rosa, Temporal performance analysis of bus transportation using link streams, Mathematical Problems in Engineering (2019).

97. R. Pastor-Satorras, C. Castellano, P. Van Mieghem, A. Vespignani, Epidemic processes in complex networks, Rev. Mod. Phys. 87 (2015) 925–979. doi:10.1103/RevModPhys.87.925.
URL https://link.aps.org/doi/10.1103/RevModPhys.87.925

98. L. Zhao, G.-J. Wang, M. Wang, W. Bao, W. Li, H. E. Stanley, Stock market as temporal network, Physica A: Statistical Mechanics and its Applications 506 (2018) 1104–1112. doi:10.1016/j.physa.2018.05.039.
URL http://dx.doi.org/10.1016/j.physa.2018.05.039

99. P. Holme, J. Saramäki, Temporal network theory, Vol. 2, Springer, New York City, NY, 2019.
100. S. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, P. Poupart, K. Borgwardt, Representation learning for dynamic graphs: A survey, J. Mach. Learn. Res. 21 (2020) 70:1–70:73.
101. S. developer documentation, Broadcast Ops (2021).
URL https://developers.steem.io/apidefinitions/broadcast-ops
102. B. Guidi, A. Michienzi, L. Ricci, Steem blockchain: Mining the inner structure of the graph, IEEE Access 8 (11 2020). doi:10.1109/ACCESS.2020.3038550.
103. H. D. Documentation, API Docs - API Definitions (2021).
URL https://developers.hive.io/apidefinitions/
104. C. Li, B. Palanisamy, R. Xu, J. Xu, J. Wang, Steemops: Extracting and analyzing key operations in steemit blockchain-based social media platform, Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy (2021).
105. R. Ciriello, R. Beck, J. Thatcher, The paradoxical effects of blockchain technology on social networking practices, in: Proceedings of the Thirty Ninth International Conference on Information Systems, AIS, 2018, pp. 1–19.
106. A. Kiayias, B. Livshits, A. M. Mosteiro, O. Litos, A puff of steem: Security analysis of decentralized content curation, ArXiv abs/1810.01719 (2019).
107. K. Kapanova, B. Guidi, A. Michienzi, K. Koidl, Evaluating posts on the steemit blockchain: Analysis on topics based on textual cues, in: Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good, EAI, 2020, pp. 1–6.
108. T.-H. Kim, H. min Shin, H. Hwang, S. Jeong, Posting bot detection on blockchain-based social media platform using machine learning techniques, ArXiv abs/2008.12471 (2020).
109. U. W. Chohan, The concept and criticisms of steemit, CBRI Working Papers: Notes on the 21st Century, Available at SSRN: http://dx.doi.org/10.2139/ssrn.3129410 (2018).
110. B. Guidi, A. Michienzi, L. Ricci, A graph-based socioeconomic analysis of steemit, IEEE Transactions on Computational Social Systems PP (2020) 1–12. doi:10.1109/TCSS.2020.3042745.
111. B. Guidi, A. Michienzi, L. Ricci, Analysis of witnesses in the steem blockchain, Mobile Networks and Applications (2021) 1–12.
112. P. Jia, C. Yin, Research on the characteristics of community network information transmission in blockchain environment, in: IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Vol. 1, IEEE, New York, NY, 2019, pp. 2296–2300.
113. C. T. Ba, M. Zignani, S. Gaito, G. P. Rossi, The effect of cryptocurrency price on a blockchain-based social network, in: R. M. Benito, C. Cherifi, H. Cherifi, E. Moro, L. M. Rocha, M. Sales-Pardo (Eds.), Complex Networks & Their Applications IX, Springer International Publishing, Cham, 2021, pp. 581–592.

114. P. Holme, Modern temporal network theory: a colloquium, The European Physical Journal B 88 (9) (2015) 1–30.
115. Hive, Hive WhitePaper, Online (2020).
URL `https://hive.io/whitepaper.pdf`
116. A. Clauset, C. R. Shalizi, M. E. Newman, Power-law distributions in empirical data, SIAM review 51 (4) (2009) 661–703.
117. K.-I. Goh, A.-L. Barabási, Burstiness and memory in complex systems, EPL (Europhysics Letters) 81 (4) (2008) 48002.
118. E.-K. Kim, H.-H. Jo, Measuring burstiness for finite event sequences, Physical Review E 94 (3) (2016) 032311.
119. J. Alstott, E. Bullmore, D. Plenz, powerlaw: a python package for analysis of heavy-tailed distributions, PloS one 9 (1) (2014) e85777.
120. E. Wong, B. Baur, S. Quader, C.-H. Huang, Biological network motif detection: principles and practice, Briefings in bioinformatics 13 (2) (2012) 202–215.
121. Y. Hulovatyy, H. Chen, T. Milenković, Exploring the structure and function of temporal networks with dynamic graphlets, Bioinformatics 31 (12) (2015) i171–i180.
122. S. Purohit, L. B. Holder, G. Chin, Item: Independent temporal motifs to summarize and compare temporal networks, ArXiv abs/2002.08312 (2020).
123. L. Gauvin, M. Génois, M. Karsai, M. Kivelä, T. Takaguchi, E. Valdano, C. L. Vestergaard, Randomized reference models for temporal networks, arXiv preprint arXiv:1806.04032 (2018).
124. C. T. Ba, M. Zignani, S. Gaito, The role of groups in a user migration across blockchain-based online social media, in: 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), IEEE, 2022, pp. 291–296.
125. C. T. Ba, M. Zignani, S. Gaito, Social and rewarding microscopical dynamics in blockchain-based online social networks, in: Proceedings of the Conference on Information Technology for Social Good, GoodIT '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 127–132. `doi:10.1145/3462203.3475913`.
URL `https://doi.org/10.1145/3462203.3475913`
126. H. Tang, J. Ni, Y. Zhang, Identification and evolutionary analysis of user collusion behavior in blockchain online social medias, IEEE Transactions on Computational Social Systems (2022).
127. A. Galdeman, M. Zignani, S. Gaito, Disentangling the growth of blockchain-based networks by graph evolution rule mining, in: 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2022, pp. 1–10.
128. A. Kapoor, D. Guhathakurta, M. Mathur, R. Yadav, M. Gupta, P. Kumaraguru, Tweetboost: Influence of social media on nft valuation, in: Companion Proceedings of the Web Conference 2022, 2022, pp. 621–629.
129. M. Franceschet, Hits hits art, Blockchain: Research and Applications 2 (4) (2021) 100038.

130. C. E. Mattsson, T. Criscione, F. W. Takes, Circulation of a digital community currency, arXiv preprint arXiv:2207.08941 (2022).
131. C. T. Ba, A. Galdeman, M. Zignani, S. Gaito, Temporal analysis of cooperative behaviour in a blockchain for humanitarian aid during the covid-19 pandemic, in: Proceedings of the 2022 ACM Conference on Information Technology for Social Good, GoodIT '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 292–299. doi:10.1145/3524458.3547245.
URL https://doi.org/10.1145/3524458.3547245
132. P. L. Erdős, I. Miklós, Z. Toroczkai, A simple havel-hakimi type algorithm to realize graphical degree sequences of directed graphs, arXiv preprint arXiv:0905.4913 (2009).
133. V. Batagelj, A. Mrvar, A subquadratic triad census algorithm for large sparse networks with small maximum degree, Social networks 23 (3) (2001) 237–243.
134. Ö. N. Yaveroğlu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. Karapandza, A. Stojmirovic, N. Pržulj, Revealing the hidden language of complex networks, Scientific reports 4 (1) (2014) 4547.
135. R. Interdonato, J. Bourgoin, Q. Grislain, M. Zignani, S. Gaito, M. Giger, The parable of arable land: Characterizing large scale land acquisitions through network analysis, Plos one 15 (10) (2020) e0240051.
136. O. M. Granados, A. Vargas, The geometry of suspicious money laundering activities in financial networks, EPJ Data Science 11 (1) (2022) 6.
137. D. D. F. Maesa, A. Marino, L. Ricci, Data-driven analysis of bitcoin properties: exploiting the users graph, International Journal of Data Science and Analytics 6 (1) (2018) 63–80.
138. D. D. F. Maesa, A. Marino, L. Ricci, The bow tie structure of the bitcoin users graph, Applied Network Science 4 (1) (2019) 56.
139. Q. Ji, E. Bouri, R. Gupta, D. Roubaud, Network causality structures among bitcoin and other financial assets: A directed acyclic graph approach, The Quarterly Review of Economics and Finance 70 (2018) 203–213.
140. N. Gensollen, M. Latapy, Do you trade with your friends or become friends with your trading partners? a case study in the g1 cryptocurrency., Applied Network Science 5 (1) (2020) NA–NA.
141. B. Guidi, A. Michienzi, Users and bots behaviour analysis in blockchain social media, in: 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), IEEE, 2020, pp. 1–8.
142. Steem (STEEM) price, charts, market cap, and other metrics (2020).
URL https://coinmarketcap.com/currencies/steem/
143. R. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice, 3rd Edition, OTexts, 2019.
URL https://Otexts.com/fpp3/
144. D. Freedman, R. Pisani, R. Purves, Statistics (international student edition), Pisani, R. Purves, 4th edn. WW Norton & Company, New York (2007).
145. N. Scott, Steemit Update (Nov. 2018).
URL https://steemit.com/steem/@ned/2fajh9-steemit-update

146. G. Iosifidis, Y. Charette, E. M. Airoldi, G. Littera, L. Tassiulas, N. A. Christakis, Cyclic motifs in the sardex monetary network, Nature Human Behaviour 2 (11) (2018) 822–829.
147. A. Galdeman, C. T. Ba, M. Zignani, C. Quadri, S. Gaito, City consumption profile: a city perspective on the spending behavior of citizens, Applied Network Science 6 (1) (2021) 1–20.
148. R. Mqamelo, Community currencies as crisis response: Results from a randomized control trial in kenya, Frontiers in Blockchain (2021) 44.
149. E. Barinaga, A route to commons-based democratic monies? embedding the governance of money in traditional communal institutions, Frontiers in Blockchain 3 (2020) 575851.
150. How 'chamas' and mutual credit are changing Africa (Jun. 2021).
URL https://www.lowimpact.org/how-chamas-mutual-credit-changing-africa/
151. Reuters, Kenya: the latest coronavirus counts, charts and maps, Reuters (Jul. 2022).
URL https://www.reuters.com/graphics/world-coronavirus-tracker-and-maps/countries-and-territories/kenya/
152. P. Riehmann, M. Hanfler, B. Froehlich, Interactive sankey diagrams, in: IEEE Symposium on Information Visualization (InfoVis 05), IEEE Computer Society, Los Alamitos, CA, USA, 2005, pp. 233,234,235,236,237,238,239,240. doi:10.1109/INFVIS.2005.1532152.
URL https://doi.ieeecomputersociety.org/10.1109/INFVIS.2005.1532152
153. M. J. Salganik, Bit by bit: Social research in the digital age, Princeton University Press, 2019.
154. I. Waller, A. Anderson, Quantifying social organization and political polarization in online platforms, Nature 600 (7888) (2021) 264–268.
155. A. Kazdin, R. Bootzin, The token economy: An evaluative review, Journal of Applied Behavior Analysis - J APPL BEHAV ANAL 5 (1972) 343–372. doi:10.1901/jaba.1972.5-343.
156. B. Guidi, When blockchain meets online social networks, Pervasive and Mobile Computing 62 (2020) 101131.
157. T. Poongodi, R. Sujatha, D. Sumathi, P. Suresh, B. Balamurugan, Blockchain in social networking, Cryptocurrencies and Blockchain Technology Applications (2020) 55–76.
158. P. Holme, J. Saramäki, Temporal networks, Physics Reports 519 (3) (2012) 97–125, temporal Networks.
159. M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, M. A. Porter, Multilayer networks, Journal of Complex Networks 2 (3) (2014) 203–271.
160. S. A. Myers, A. Sharma, P. Gupta, J. Lin, Information network or social network?: the structure of the twitter follow graph, in: Proceedings of the 23rd International Conference on World Wide Web, WWW '14, ACM, 2014, pp. 493–498.

161. N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.
162. Y. Ma, Y. Tian, N. Moniz, N. V. Chawla, Class-imbalanced learning on graphs: A survey, arXiv preprint arXiv:2304.04300 (2023).
163. M. Zhang, Y. Chen, Link prediction based on graph neural networks, Advances in neural information processing systems 31 (2018).
164. J. You, R. Ying, J. Leskovec, Position-aware graph neural networks, in: International conference on machine learning, PMLR, 2019, pp. 7134–7143.
165. Z. Zhang, J. Bu, M. Ester, J. Zhang, C. Yao, Z. Yu, C. Wang, Hierarchical graph pooling with structure learning, arXiv preprint arXiv:1911.05954 abs/1911.05954 (2019).
166. J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl, Neural message passing for quantum chemistry, in: International conference on machine learning, PMLR, 2017, pp. 1263–1272.
167. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).
URL https://openreview.net/forum?id=SJU4ayYgl
168. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, arXiv preprint arXiv:1710.10903 (2017).
169. T. N. Kipf, M. Welling, Variational graph auto-encoders, arXiv preprint arXiv:1611.07308 (2016).
170. T. Zhao, X. Zhang, S. Wang, Graphsmote: Imbalanced node classification on graphs with graph neural networks, in: Proceedings of the 14th ACM international conference on web search and data mining, 2021, pp. 833–841.
171. I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.
172. L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, Neurocomputing 415 (2020) 295–316.
173. J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization., Journal of machine learning research 13 (2) (2012).
174. J. You, T. Du, J. Leskovec, Roland: graph learning framework for dynamic graphs, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 2358–2366.
175. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 abs/1412.6980 (2014).
176. A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, C. Zhou, Synchronization in complex networks, Physics reports 469 (3) (2008) 93–153.
177. M. Nekovee, Y. Moreno, G. Bianconi, M. Marsili, Theory of rumour spreading in complex social networks, Physica A: Statistical Mechanics and its Applications 374 (1) (2007) 457–470.
178. S. Battiston, G. Caldarelli, M. D'Errico, The financial system as a nexus of interconnected networks, in: Interconnected networks, Springer, 2016, pp. 195–229.

179. M. E. Dickison, M. Magnani, L. Rossi, Multilayer social networks, Cambridge University Press, 2016.

180. R. Wickman, X. Zhang, W. Li, Sparrl: Graph sparsification via deep reinforcement learning, ArXiv abs/2112.01565 (2021).

181. U. S. Shanthamallu, J. J. Thiagarajan, H. Song, A. Spanias, Gramme: Semisupervised learning using multilayered graph attention models, IEEE Trans. Neural Networks Learn. Syst. 31 (10) (2020) 3977–3988. doi:10.1109/TNNLS.2019.2948797.
URL https://doi.org/10.1109/TNNLS.2019.2948797

182. Y. Liu, T. Safavi, A. Dighe, D. Koutra, Graph summarization methods and applications: A survey, ACM computing surveys (CSUR) 51 (3) (2018) 1–34.

183. D. Luo, W. Cheng, W. Yu, B. Zong, J. Ni, H. Chen, X. Zhang, Learning to drop: Robust graph neural network via topological denoising, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, pp. 779–787.

184. L. Wang, W. Yu, W. Wang, W. Cheng, W. Zhang, H. Zha, X. feng He, H. Chen, Learning robust representations with graph denoising policy network, 2019 IEEE International Conference on Data Mining (ICDM) (2019) 1378–1383.

185. C. T. Ba, M. Zignani, S. Gaito, The role of cryptocurrency in the dynamics of blockchain-based social networks: The case of steemit, PloS one 17 (6) (2022) e0267612.

186. L. Martirano, L. Zangari, A. Tagarelli, Co-mlhan: contrastive learning for multilayer heterogeneous attributed networks, Applied Network Science 7 (1) (2022) 1–44.

187. R. Interdonato, R. Gaetano, D. L. Seen, M. Roche, G. Scarpa, Extracting multilayer networks from sentinel-2 satellite image time series, Network Science 8 (S1) (2020) S26–S42.

Acknowledgments