Unraveling the Semantic Evolution of Core Nodes in a Global Contribution Network

Tales Lopes, Victor Ströele, Regina Braga Computer Science Postgraduate Program Federal University of Juiz de Fora Juiz de Fora , Brazil {talessilva, victor.stroele, regina.braga}@ice.ufjf.br Michael Bauer Department of Computer Science University of Western Ontario London, Canada bauer@uwo.ca

Abstract—The analysis of social structures is important in many contexts, especially in Global Software Development, where various developers with diverse skills and knowledge are involved. In this sense, searching for essential members is a valuable task since they are fundamental to the network's evolution. With the main goal of identifying and monitoring these individuals, we propose a temporal analysis approach that explores syntactic and semantic network aspects. We describe experiments based on popular projects on Github that consider a network modeled over consecutive time periods. We also propose an ontology to represent the domain knowledge and explore the network by investigating its semantic context. We provide evidence that our approach can detect individuals considered essential based on their role in the network.

Index Terms—Social networks, Global Software Development, contribution network, network evolution, temporal analysis, essential individuals, semantic analysis, ontology

I. INTRODUCTION

With the explosion of data in the past number of years across many aspects of everyday life, extracting knowledge from raw data volumes has become increasingly challenging. The emergence of different data sources, such as IoT devices, posts on social media, or web messages, has added to the volume of data and increased the challenges [1] in sifting through the data to find relevant information.

Many technological advances and research developments are only possible through the use of efficient methods capable of handling large volumes of data [2]. These can be useful in data mining processes supporting the knowledge acquisition in fields such as healthcare [3], which involves people's lives, or in business [4], where companies competitively seek profit.

A range of operations can be used together or separately to support the mining of data. Machine learning algorithms enable the creation of predictive models with distinct strategies [5]. Complex network analysis methods can identify individuals' roles and describe information flow by exploring features such as topology or centrality [6]. With ontologies, it is possible to discover implicit relationships and structure of semantic models [7]. Additionally, high performance and distributed computing techniques are needed to improve the data processing in combinatorial problems, where complexity grows exponentially according to the input size [8]. IEEE/ACM ASONAM 2020, December 7-10, 2020 978-1-7281-1056-1/20/\$31.00 © 2020 IEEE Social networks are the representation of social relationships, obtained directly or indirectly. Different network analysis approaches can further investigate social structures to detect and describe implicit communities [9]. Understanding communities' particularities allow creating methods to optimize decision-making activities necessary in various human practices [10].

The Global Software Development (GSD) context is driven by the growth of open-source communities in order to meet the demand for more complex software. It is motivated by the existence of advantages, such as reducing software development cycle time or using expertise when needed [11]. Consequently, we have a scenario with many developers working synchronously and asynchronously worldwide on the same project. In this sense, recommendation systems play an indispensable role in GSD by enhancing processes that can aid in identifying specialists or allocating teams [12].

Experts can be defined as the most experienced developers, capable of solving complex tasks to achieve project goals or helping other developers [13]. These are essential people in GSD as they can potentially help in specific situations. However, identifying individuals with particular skills is not easy, especially when several factors need to be considered, such as technical interests, availability, previous experience, or collaborative skills [14]. GitHub is one of the biggest social coding platforms with millions of developers and thousands of open source projects, software artifacts, and workflows. It supports GSD with workflows that operate through the pullbased method [15]. The platform also allows data integration through its RESTful API, which is one of the reasons why it has been the target of recent studies [13].

Considering the challenges related to social structures study, we present a temporal analysis approach to recognizing essential individuals in the network. To achieve our goal, we examined overlapping networks from popular projects on GitHub. We seek to understand projects' evolution by investigating changes in syntactic and semantic aspects of the network. Therefore, we concentrate our analysis on identifying core nodes in the network as well as their contribution activity over time.

We carried out structural analyses on the network and discuss the results focusing on nodes described over defined

intervals. Key points in our analyses include: (i) a search for active individuals (core nodes); (ii) an analysis of their contribution activity on the network; (iii) a refinement of the search to search for individuals considered essential to the project's evolution. Further, we also look for key periods considering nodes' participation in (iv) all periods; (v) sequential periods; and (vi) idle sequential periods.

Finally, we propose an ontology for further investigations, assuming a semantic approach to model and explore data. The ontology seeks to support the analysis process by answering the "whys" of peculiarities observed during the structural investigations.

To guide our study, we focus on two main Research Questions (RQ):

- (*RQ1*) Who are the members considered essential to the network? We present a temporal analysis approach to identify active members (cores nodes) and further search for individuals deemed crucial for the projects. We investigated several overlapping structures corresponding to consecutive time stamps, aiming to track core nodes' behavior in the network and understand the project's evolution over time.
- (*RQ2*) How does the network evolves considering its semantic aspects? We propose an ontology representing the project's semantic context. It seeks to provide a better domain understanding through the semantic enrichment of the initial results obtained in the structural analysis. We expect that the ontology will bring forward further network knowledge, inferring information not possible to be obtained by conventional analysis.

The paper organization is as follows: Section II presents work related to temporal analysis in social networks as well as the identification and evolution of core individuals. Section III introduces the proposed temporal analysis method, including explaining each step involved in the approach. Section IV seeks to answer the research question (RQ) by evaluating and analyzing the obtained results. Finally, Section V presents conclusions and future works.

All codes and data used in this paper can be accessed at¹.

II. RELATED WORK

Social networks can be formally defined as a set of members (nodes) bound by one or more types of connections (edges), in which patterns formed by relationships create existing social conditions [6]. Through the modeling of these networks, it is possible to perform a wide range of social investigations, including the evolutionary analysis of the behavior of groups and communities. At the same time, they also play a crucial role in critical situations such as the propagation of emergency information among people [5].

A significant factor in the analysis process is to consider temporal information, which gives an idea about the changes being happened in the network based on any time interval. An event can be defined as an occurrence with enough force to

¹https://github.com/Talessil/Temporal-Analysis

create an observable change in a social context [23]. Thus, different works address distinct temporal-based analyses to have a better understanding of the network.

In evolving social networks with fixed nodes, temporal changes are explored using algorithms that predict the creation and removal of links [21]. Thus, anomalous events, such as traffic accidents, can be detected when nearby edges behave abnormally over several consecutive periods [22]. Machine learning algorithms are also employed to detect event occurrences through structural changes analysis [20]. However, instead of monitoring changes in the entire structure, some community-based approaches monitor the evolution from analyzing network partitions [23].

The core nodes in social networks are considered fundamental individuals and have greater relevance than the average member [16]. In specific communities, they consist of enthusiastic individuals engaged in connecting distinct groups [5] and collaborating with other members [13]. There are many definitions of these people, e.g., the most participative members [17], those who answer others' members' questions [18], or the ones who encourage other members to participate by proposing assignments or discussion topics [19].

The evolution of a social network can be described by analyzing some factors over time, i.e., shared activities, members associations, the similarity between individuals' attributes, and the closure of network cycles [25]. Besides, a temporal analysis results over a multidisciplinary developer network highlights that overlapped nodes might indicate individuals who can collaborate on different technologies simultaneously [13]. Therefore, monitoring and studying active members' progression is advantageous, as they are fundamental elements for the healthy working of the network [26].

Finally, few works aim to find appropriate individuals to help with GSD issues. They specifically recommend individuals considered experts to assist in code reviewing based on their historical contributions [24] [27]. Also, some strategies are used to analyze semantically constructed graphs [28] [31]

However, what are the differences between our work and previous ones? None of the studies addressed proposes a temporal analysis method that seeks to find and examine essential individuals considering the analysis of overlapping network structures. Furthermore, our work stands out for employing syntactic and semantic analysis together over the temporal partitions to explore the network evolution. To the best of our knowledge, this is the first work to propose an ontology to assist in searching for essential individuals exploring semantic aspects of overlapping networks in a GSD context.

In summary, we can say that we propose a novel approach that seeks to support social network analysis employing complementary data analysis techniques in a GSD context. We focus on temporal analysis in order to understand the network evolution and recognize essential individuals for the network.

Therefore, the aspects that make our approach different from previous studies are:

• We proposed a network analysis method that seeks to find essential individuals in a GSD context, considering

overlapping temporal structures.

- Our approach seeks to investigate the network evolution by monitoring the contribution of key nodes in the network over time.
- We also introduced an Ontology to further explore the network, providing implicit semantic information, and assisting in the understanding of initial results from structural analysis.

III. TEMPORAL ANALYSIS APPROACH

In this section, we describe the proposed temporal analysis. An overview of the approach is presented in Fig. 1 and illustrates the main steps covered by this work. The extraction, storing, and modeling of data compose the pre-processing stage. Different strategies can be used individually or together to achieve the desired results in the data analysis stage; it includes centrality metrics, machine learning algorithms, semantic models, or temporal approaches.

As discussed in Section II, there are many ways to handle temporal elements as a new dimension in the model or considering specific periods. The graph-modeled data set is split into multiple temporal partitions considering the selected periods. The periods are modeled as sub-networks that are analyzed considering different analysis methods. There is a cyclic process of data investigation between different analysis techniques and the temporal module, e.g., the temporal partitions can increase the ontology and then be additionally detailed to generate new enriched networks. Finally, we show some instances of the results compatible with the case study related to the GSD context.



Fig. 1. Temporal Analysis Approach Overview.

A. Contribution Network

For this work, we consider a contribution network formed from popular projects on GitHub. The network structure aims to illustrate explicit and implicit connections between contributors who share technical knowledge. A contribution relationship is created when someone creates a review comment on someone else's pull request. Other relationships could be used to model the network, as discussion comments, however, they are not directly related to technical knowledge. This model was adopted following the best way to represent a knowledgeable contribution with the available data. In addition to just a textual message, review comments also contain pieces of code, which can be seen as a kind of contribution that involves technical knowledge.

All data used in this research were obtained through the GitHub RESTful API. The data represents a GSD context, and it is used to evaluate our methods in a real-world scenario. The database consists of some of the most popular projects on Github based on the community size and its contribution level [29]: Node.js, Kubernetes, and Symfony.

Distinct interactions can be established depending on the social aspects we want to observe. Seeking to build a contribution network, we propose a directed graph model G = (V, E), illustrated in Fig. 2, to represent the relationships between developers. The connections are inferred from a directional bipartite graph to a bidirectional graph. Considering a set of developers $V = \{v_0, v_1, ..., v_n\}$, when any individual v_i creates a review comment on a pull request of another individual v_j , an implicit relationship $e_{ij} = (v_i, v_j, w_{ij})$ is created. The weight w_{ij} value represents the total number of review comments from one person on all other person's pull requests.



Fig. 2. Contribution Network Modeling.

B. Core Nodes Identification

According to the contribution model, we are most interested in the number of distinct individuals a contributor engages with rather than the number of contributions to the same person. We can define core individuals as those who contribute to a number of individuals higher than a specified threshold of θ . Aiming to look for core nodes in a contribution network, we use the NetSCAN clustering algorithm as a network analysis technique.

NetSCAN [16] is a density-based clustering approach. It was developed to find clusters in social networks and has three input parameters, *epsilon*, *minPnts*, and a third optional parameter, *radius*, which allows a further search within a node's neighborhood. In NetSCAN, *minPnts* and *epsilon* are associated with the minimum number of out-connections and the minimum weight that a node must have to become a core node. NetSCAN was considered appropriate for the identification of core individuals in the proposed network.

Therefore, to identify core members through the clustering process, the parameters epsilon = 1 and minPnts = 4 were defined according to [30]. The *epsilon* value corresponds to the out-degree mode value above zero, while the out-degree average value was used as *minPnts*. The third parameter *radius* is not considered in this first analysis.

C. Temporal Partitioning

All data obtained correspond to the period from 2010 to mid-2018. However, in this first moment, we opted to work with the most recent periods, from January 1st, 2017, to June 30th, 2018. Information about both the first release (FR) and the stable release (SR) of the projects are described below.

- Symfony: FR: 22 October 2005, SR: 2020-03-27
- Kubernetes: FR: 7 June 2014, SR: March 25, 2020
- Node.js: FR: May 27, 2009, SR: April 14, 2020

To model the projects considering the network evolution, we took snapshots from the dataset and modeled them as contribution structures split by month. Our approach seeks to generate overlapping graph structures in which network aspects can be monitored over subsequent periods.

D. Ontology

Ontologies are semantic technologies used to model the domain knowledge with primitives such as classes, data properties, object properties, property chains, and inference rules (SWRL rules). We implemented the ontology in OWL (Ontology Web Language) using the Protegé tool. The ontology seeks to explore further the network analysis with semantic investigations. These can come as issues that arise during the structural investigation stage or new interrogations to complement the obtained results.

The proposed ontology model can be seen in Fig. 3, where an index is also presented to help identify the different classes in which data objects can be associated.



Fig. 3. Ontology Model.

We defined two main classes. The *input classes* (yellow) are classes instantiated directly from the data source. The classes **Developer**, **Period**, and **Project** have a straightforward definition, whereas the class **Keywork** represents the occurrence of tags in review comments, and the class **KeywordWeight** corresponds with the total of times a tag is associated with an individual.

The *Classes derived from taxonomy* (blue) are predefined in the ontology according to concepts that describe the domain knowledge. Instances are associated with these classes through inference methods. **Topics** are the characterization of project issues, represented by tags groups. Some topics may be more specific and described as a **SpecificTopic**. In the same way, we have the **TopicWeight** and the **SpecificTopicWeight** representing, respectively, the relevance of the topic or specific topic to an individual.

Moreover, object properties represent relationships between classes. The dotted lines in the model represent connections only obtained by using ontological rules. In addition to classes and objects properties, the data properties Name, Date, and Weight correlate classes and objects instances in the ontology. It is worth mentioning that we do not represent all possible associations in our model, focusing on the most relevant for our analysis.

IV. RESULTS

We considered two sets of experiments. We first carried out a structural analysis to understand network contributions and answer RQ1. We conducted some investigations to provide insights regarding the evolution of core nodes. Next, we investigated RQ2 using semantic analysis to explore implicit data knowledge. The ontology was developed to represent the knowledge domain and explore relevant points that appeared during the structural analysis.

A. Structural Investigation

In order to perform the temporal analysis, we run the NetSCAN in each period represented by overlapping structures to find the core nodes. Thus, all the analyzes below refer mainly to the core nodes (CN) evolution over time.

Table I exhibits quantitative data that describes the clustering of the network in each period. The number of nodes, edges, clusters, and cores is exposed to support the next analysis.

Period	Nodes	Edges	C*	Cores
2017-1	436	987 9		66
2017-2	494	1117	6	62
2017-3	3 489 1016		8	66
2017-4	477	1034	10	67
2017-5	547	1299	7	70
2017-6	551	1137	6	62
2017-7	522	1110	10	63
2017-8	595	1337	8	70
2017-9	557	1186	5	63
2017-10	704	1519	4	73
2017-11	624	1367	9	85
2017-12	489	873	11	39
2018-1	530	1043	6	59
2018-2	573	1168	12	76
2018-3	497	1032	9	59
2018-4	528	1020	8	60
2018-5	493	1012	11	60
2018-6	481	926	11	44

 TABLE I

 The clustering of the overlapping structures

First, for each period, we analyzed each CN's total contribution and contribution degree. Fig. 4 contain the graphics that show (A) the number of CN contributing and (B) the degree of each CN, corresponding to each period. Correspondingly, in Fig. 5, we display a box plot of the CN's degree.

In Fig. 4-A, it is possible to observe a pattern regarding the number of CN collaborating over time. However, there is a

particular scenario between 2017-10 and 2017-12, where we see the number of CN increasing up to the upper bound and then decreasing to the lower bound. Moreover, although 2017-11 has the most substantial number of CN contributing, Fig. 4-B shows that 2017-10 presents a larger number of specific individuals contributing more intensely (lighter colors), while 2017-11 has less intense contributions (darker colors).

Also, Fig. 5 shows that 2017-10 presents the CN with the largest contributions, including the individuals with the most significant contributions to the network evolution. We can also see particular individuals, considered outliers, contributing more and more up to 2017-10. After that period, these peak values decreases. This scenario may indicate specific demands by some projects, although it is not possible to confirm a contribution pattern concerning that period.

Furthermore, despite the higher number of contributors between 2017-10 and 2017-11, the average degree value does not change as much over the periods, even in 2017-12, with the lowest number of CN contributing. Consequently, we sought to investigate in the next subsection the semantic aspects related to the evolution of the network that could justify the intense variation concerning the number of individual contributions in specific periods.



Fig. 4. Total of Contribution per Period.

The graphics in Fig. 6 show (A) the number of CN related to the total number of months they have contributed, and (B) the periods in which each individual is considered a core node. It is possible to notice many individuals participating in just a few months, most of them contributing in one month, while others seem to have specific periods of contribution, such as 2, 3, or 4 months. For more extended periods of contribution, the number of individuals contributing seems to stabilize.

Hence, it is possible to notice individuals with more relevance among those identified as CN. Some developers contributed every month and possibly have a more significant role on the network. Therefore, since it is possible to locate different contribution profiles, the next subsection also address the semantic characterization of individuals who contributed during all periods.



Fig. 5. Boxplots of Contribution Degree per Period.



Fig. 6. Individuals Related to Contribution Periods.

Nevertheless, we further explore distinct contribution profiles in Fig. 7. Similarly to Fig. 6, Fig. 7 contains graphics showing the number of CN related to (A) the total of sequential contribution periods, (B) the maximal sequential contribution period, (C) the total of sequential idle periods, and (D) the maximal sequential idle period. It is worth explaining that (A) and (C) represent the totality of cumulative periods, e.g., a developer who had contributed for a period and contributed again after some idle time, it counts in both periods.

Both Fig. 7-A and Fig. 7-B give us the idea of many developers contributing in short periods and few developers contributing in numerous sequential periods. However, in Fig. 7-C and Fig. 7-D, it is possible to detect some absent developers over periods of 12 and 13 months between periods of contribution. Therefore, we can further extend the discussion in the semantic step to characterize individuals who make specific contributions and have a contribution history signalized by periods of inactivity.

Finally, it is possible to answer RQ1. Through overlapped structural analysis of the network, it was possible to identify CN in all periods, as well as the individuals considered essential to the network due to their high contribution degree throughout all the analyzed periods. Yet, some results could



Fig. 7. Graphics of Accumulated Contributions.

not be fully explained by the structural investigations alone. Consequently, we highlight three Secondary Research Question (SQR) that should guide our semantic analysis and help to answer RQ2.

- (SRQ1) What are the semantic aspects related to the network evolution that justify the strong contrast in the number of contributions of some individuals in specific periods?
- (SRQ2) What characterizes the profiles of different individuals, especially those who contributed during all periods?
- (SRQ3) What characterizes individuals who have contributed during specific periods, considering unique contributions and contributions between idle periods?

B. Semantic Analysis

In this step, we use the proposed ontology to represent the domain knowledge and employ semantic analyzes in the network.

For that purpose, we defined a taxonomy from the study of each project's keywords (tags) in order to characterize the projects' issues. Table II shows the total number of tags and the number of different contributors to each project.

TABLE II TAGS AND CONTRIBUTORS PER PROJECT

Project	Tags	Contributors
Kubernetes	9322	119
Nodejs	4734	43
Symfony	1423	27

The definition of each tag was obtained from the project's repositories analysis. As a result, we have identified some main classifications (topics) in which the tags are related: **Status**, **Kind**, **Technology**, **Support**, **Size** and **Priority**.

Accordingly, our ontology can associate keywords with topics and specific topics of projects' issues, defined through the taxonomy. In the same way, individuals can be associated with different topics and specific topics by relating to tags that are bonded to multiple topics and specific topics, e.g., the keywords *bug*, *kind/bug*, and *errors* are associated with the topic **Status**, and with the specific topic **Corrective Maintenance**. Fig. 8, 9, and 10 present graphics that correspond to each project's tags' evolution and show trend lines related to the main tags' distribution over time.



Fig. 8. Tags Distribution: Symfony



Fig. 9. Tags Distribution: Kubernetes



Fig. 10. Tags Distribution: Nodejs/Node

In the projects Symfony and Kubernetes, mostly tags are related to the topic **Kind** through the tags *bug*, *deprecation*, and *feature*. In this case, it was possible to infer that these projects have the majority of the efforts related to the specific topics **corrective maintenance**, **evolutionary maintenance**, and **development of new features**.

Fig. 8 shows a soft drop in the trend lines related to the project Symfony, possibly indicating a project stabilization. Node.js goes in the opposite direction (Fig. 10), with an increase in the tags associated with the specific topic **evolutive maintenance**, being indicative of the project's growth.

It is possible to notice an increasing trend in developing new features in Kubernetes (Fig. 9), especially in periods of significant contributions, such as 2017-10 and 2017-11. The trend lines also show that the number of tests has dropped slightly over the periods, which may be one of the reasons for the increase in the number of bugs in the project.

The strong contrast in the number of contributions in different periods and by specific developers can be justified by the project's demands related to particular topics. For instance, Node.js required experts to deal with new features and meet the project needs between 2017-10 and 2017-11. Similarly, Kubernetes also saw an increase in weight on various topics and specific topics from 2017-10 to 2017-11. Thus, SRQ1 is answered.

Furthermore, we can see a considerable variation in the quantity and purpose of the tags in each project. Some projects are associated with higher stability, while others demand more development effort. In this sense, we conducted a semantic analysis to describe the individuals considered essential for the network, as well as their specialties and main technologies adopted in each project.

With the proposal of characterizing the contributor's profile, seven individuals were selected and further described. Table III exhibits a summary of some authors' contributions considering the main topics they are related to, such as **kind**, **technology**, and **contribution size** (lines of code).

TABLE III DISTINCT CONTRIBUTOR PROFILES

Core ID	Period	Project	Kind	Technology	Size
730123	all	Kubernetes	bug, clean	sig/api-machinery	size/L,size/M
980082	all	Kubernetes	bug	sig/api-machinery, sigh/auth	size/M, size/XS, size/L
439929	all	Nodejs	lib/src, errors	C++,HTTP2	-
2512748	all	Nodejs	test	C++	-
243674	all	Symfony	bug	DependencyInjection, Frameworkbundle	-
826111	2017-3,2017-4	Kubernetes	Feature	sig/node,sig/storage	size/L
610090	2017-7,2017-10,2018-2	Symfony	Bug,Deprecation,Feature	TwigBridge,HttpKernel	-

All projects have collaborators who contributed throughout all the periods analyzed. They work to resolve bugs, perform buildings, and have a large number of approved issues. In general, they operate with more than one project technology, although there is always one technology that stands out. We also observed that these individuals have different contribution profiles, characterized by the topics and specific topics of their contributions. To illustrate these aspects, we present graphics of individuals *980082* (Fig. 11) and *730123* (Fig. 12), who contributed to the project Kubernetes.



Fig. 11. Tags Distribution: Contributor 980082



Fig. 12. Tags Distribution: Contributor 730123

Comparing these graphics with the Kubernetes graphic (Fig. 9), we see that the individual *980082* has a contribution pattern matching to issues related to bugs in the project, indicating

that the author collaborates according to the project's demand. Inversely, 730123 contributed little in periods of high demand, as in 2018-3. However, it has great relevance in contributions related to the specific topic **cleanup**, specifically in periods where the weight of this topic is more elevated.

Besides, individuals who contributed during all the periods to the Node.js and Symfony projects are related to specific topics concerning the projects' principal technologies and status, i.e., **C** ++ **programming language** in Node.js or **corrective maintenance** in Symfony.

Answering SRQ2, we can characterize the essential individuals as those who contributed during all periods, working according to the project's demand. It also includes some individuals who make more isolated contributions having a more significant role in specific periods, probably addressing specific topics.

Several individuals also contributed at only specific periods, such as the individuals 826111 and 610090, described in Table III. Often, individuals that contribute in a few periods are the ones that work with smaller codes, while the most substantial contributions are made by individuals who work on projects for longer.

Hence, in order to answer SRQ3, we can say that contributors who work in specific periods are characterized by working with particular technologies in the projects. In contrast, more frequent contributors work in a more embracing way, aiding in current project needs such as specific bug fixing. Likewise, individuals who work at particular moments in the projects tend to contribute to more general issues, with low priority. Some projects even recommend this behavior so that new members can become more familiar with the project's demands.

Furthermore, we also analyzed the intersection of CN's contribution to different projects and observed that the vast majority (96%) contributed to only one project. Therefore, assuming the three projects used in this study and the periods analyzed, we can say that individuals considered essential to the network always work on a particular project. The other 4% worked mainly on a single project and, at some point, made specific contributions to another project.

Finally, the three SRQ proposed in the previous subsection were satisfied. Consequently, we can answer RQ2 by affirming that the proposed ontology allowed us to explore the network domain, providing a structure with a higher semantic weight, and bringing us further knowledge about the projects' evolution and essential individuals in the network.

V. CONCLUSIONS AND FUTURE WORKS

In social networks, individuals considered essential are those who have greater relevance than the average member. They consist of the most participative members, engaged in connecting distinct groups, and collaborating with other members. They can generally be seen as experts: the most experienced developers who can help in specific situations by solving complex tasks to achieve project goals. Therefore, identifying these individuals is valuable, especially in Global Software Development (GSD) contexts. In order to find and monitor the evolution of essential individuals in social networks, we proposed a temporal analysis approach. Our method was able to perform investigation considering the study of overlapping structures in subsequent periods. Furthermore, it proved to be able to explore temporal changes by investigating syntactic and semantic network aspects.

To achieve this goal, we proposed a contribution network using some popular GitHub projects. Initial findings pointed out that it was possible to identify core nodes in all periods, as well as the individuals considered essential for the network evolution due to their high contribution degree. However, these analyses were not sufficient to fully understand the role these individuals play in the network. Consequently, we also proposed an ontology representing the domain knowledge and enriching the results with implicit semantic understanding.

Evidence was provided that the proposed ontology allows us to explore the network domain further, providing a structure with a higher semantic weight. It was also able to bring knowledge about the projects' evolution and detect individuals considered essential due to their role in meeting specific project demands.

As future work, we intend to advance the ontology to work on the recommendation of specialists and teams of specialists with complementary skills. Furthermore, in this work, we focused on analyzing a GSD context since it is a domain of high development integration with numerous individuals, projects, and artifacts to be explored. Therefore, in future works, we also plan to extend our work to explore other contexts, such as the academic or scientific domain.

Acknowledgment: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and UFJF.

REFERENCES

- Bello-Orgaz, Gema, Jason J. Jung, and David Camacho. "Social big data: Recent achievements and new challenges." Information Fusion 28 (2016): 45-59.
- [2] Rao, Arni SR Srinivasa, and Jose A. Vazquez. "Identification of COVID-19 Can be Quicker through Artificial Intelligence framework using a Mobile Phone-Based Survey in the Populations when Cities/Towns Are Under Quarantine." Infection Control Hospital Epidemiology (2020): 1-18.
- [3] Wang, Yichuan, LeeAnn Kung, and Terry Anthony Byrd. "Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations." Technological Forecasting and Social Change 126 (2018): 3-13.
- [4] Raguseo, Elisabetta. "Big data technologies: An empirical investigation on their adoption, benefits and risks for companies." International Journal of Information Management 38.1 (2018): 187-195.
- [5] Kim, Jooho, and Makarand Hastak. "Social network analysis: Characteristics of online social networks after a disaster." International Journal of Information Management 38.1 (2018): 86-96.
- [6] Wasserman, Stanley, and Katherine Faust. Social network analysis: Methods and applications. Vol. 8. Cambridge university press, 1994.
- [7] Hoehndorf, Robert, Georgios V. Gkoutos, and Paul N. Schofield. "Datamining with ontologies." Data mining techniques for the life sciences. Humana Press, New York, NY, 2016. 385-397.
- [8] Lopes, T., Meháut, J.-F., Ströele, V., Dantas, M., and Braga, R. (2020). A parallel graph partitioning approach to enhance community detection in social networks. In 2020 IEEE 25th Symposium on Computers and Communications (ISCC). IEEE.

- [9] Ströele, Victor, Geraldo ZimbrãO, and Jano M. Souza. "Group and link analysis of multi-relational scientific social networks." Journal of Systems and Software 86.7 (2013): 1819-1830.
- [10] Sievi-Korte, Outi, Sarah Beecham, and Ita Richardson. "Challenges and recommended practices for software architecting in global software development." Information and Software Technology 106 (2019): 234-253.
- [11] Bosch, Jan, and Petra Bosch-Sijtsema. "From integration to composition: On the impact of software product lines, global development and ecosystems." Journal of Systems and Software 83.1 (2010): 67-76.
- [12] Pereira, Thaís Alves Burity, et al. "A recommendation framework for allocating global software teams in software product line projects." Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering. 2010.
- [13] Horta, Vitor, et al. "Collaboration Analysis in Global Software Development." 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, 2019.
- [14] Mahmood, Sajjad, et al. "Key factors that influence task allocation in global software development.' Information and Software Technology 91 (2017): 102-122.
- [15] Gousios, Georgios, Martin Pinzger, and Arie van Deursen. "An exploratory study of the pull-based software development model." Proceedings of the 36th International Conference on Software Engineering. 2014.
- [16] Horta, Vitor, et al. "Analyzing scientific context of researchers and communities by using complex network and semantic technologies." Future Generation Computer Systems 89 (2018): 584-605.
- [17] Probst, Gilbert, and Stefano Borzillo. "Why communities of practice succeed and why they fail." European Management Journal 26.5 (2008): 335-347.
- [18] Liu, Xiaoyong, W. Bruce Croft, and Matthew Koll. "Finding experts in community-based question-answering services." Proceedings of the 14th ACM international conference on Information and knowledge management. 2005.
- [19] Bourhis, Anne, Line Dubé, and Réal Jacob. "The success of virtual communities of practice: The leadership factor." The Electronic Journal of Knowledge Management 3.1 (2005): 23-34.
- [20] Aldhaheri, Abdulrahman, and Jeongkyu Lee. "Event detection on large social media using temporal analysis." IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2017.
- [21] Wang, Wen-Qiang, Qian-Ming Zhang, and Tao Zhou. "Evaluating network models: A likelihood analysis." EPL (Europhysics Letters) 98.2 (2012): 28004.
- [22] Mongiovi, Misael, et al. "Netspot: Spotting significant anomalous regions on dynamic networks." Proceedings of the 2013 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2013.
- [23] Bidoki, Neda Hajiakhoond, et al. "Predicting Social Network Evolution from Community Data Partitions." International Conference on Social Computing, Behavioral-Cultural Modeling Prediction and Behavior Representation in Modeling and Simulation. Springer. 2019.
- [24] Yu, Yue, et al. "Who should review this pull-request: Reviewer recommendation to expedite crowd collaboration." 2014 21st Asia-Pacific Software Engineering Conference. Vol. 1. IEEE, 2014.
- [25] Kossinets, Gueorgi, and Duncan J. Watts. "Empirical analysis of an evolving social network." science 311.5757 (2006): 88-90.
 [26] Rios, Sebastian A., et al. "Semantically enhanced network analysis for
- [26] Rios, Sebastian A., et al. "Semantically enhanced network analysis for influencer identification in online social networks." Neurocomputing 326 (2019): 71-81.
- [27] Zanjani, Motahareh Bahrami, Huzefa Kagdi, and Christian Bird. "Automatically recommending peer reviewers in modern code review." IEEE Transactions on Software Engineering 42.6 (2015): 530-543.
- [28] Bader, Brett W., Richard A. Harshman, and Tamara G. Kolda. "Temporal analysis of semantic graphs using ASALSAN." Seventh IEEE international conference on data mining (ICDM 2007). IEEE, 2007.
- [29] Sheoran, Jyoti, et al. "Understanding" watchers" on GitHub." Proceedings of the 11th working conference on mining software repositories. 2014.
- [30] Lopes, T., Ströele, V., Braga, R., David, J. M. N., and Campos, F. (2020). "Diversity-based Expert Recommendation in Collaborative System Development.", in press.
- [31] Lee, Y.J. and Park, J.Y., 2019. Emerging gender issues in Korean online media: A temporal semantic network analysis approach. Journal of Contemporary Eastern Asia, 18(2), pp.118-141.