Proceedings of the 10th Annual World Conference of the
Society for Industrial and Systems Engineering,
2021 SISE Virtual Conference
September 23-24, 2021

Enhancing Social Community Analysis on Twitter Using Dynamic Object-Oriented and Network Analysis

M. J. L. Caballes, A. Kazeem, and L. Bronner

Morgan State University
1700 E Cold Spring Ln, Baltimore, MD 21251, USA

Corresponding author's Email: macab1@morgan.edu

Abstract: Social Network Analysis (SNA) plays a significant role in modeling complex systems, where community detection (CD) is the fundamental tool used in finding and obtaining groups within complex communities. Community detection is used for analyzing complex social systems represented by graph, which has the organizational and functional characteristics of the underlying network. In recent years, many community detection algorithms have been proposed in both research and academia to unveil the structural properties and dynamic behaviors of social networks. Several available algorithms in the research community recently show the importance of networks' structural properties and dynamic behaviors. However, only a few peer-reviewed journal articles focus on Twitter as the primary social media topic for SNA. In this research, Twitter is used as the social network community for analysis. Twitter is recognized as one of the prominent social networks globally. It has 198 million users with an average of 500 million tweets per day. Thus, making it a perfect platform to conduct a CD analysis. The primary goal of this research is to develop conceptual, use case, static and dynamic object-oriented and network models to develop a solution for improving the community detection problem. These models were developed to show how different actors interact and behave in a Twitter community. Normally, the results of an object-oriented analysis and modeling effort is a static model. However, this research uses the sequence diagram, one of the artifacts of object-oriented analysis, to interface to a dynamic object-oriented analysis and modeling process. This work extends the object-oriented analysis solution. The tools used to support this research are the Python object-oriented programming language and the Pajek social networking analysis tool. The models developed using these tools were utilized to detect malicious and verified Twitter users. In summary, this research performed and applied systems engineering technology through a series of static and dynamic models where data was collected from Twitter's archives. This data was imported to these models to extract keywords and quickly detect the unverified profiles.

Keywords: Community Detection (CD), Dynamic Models, Social Network, Social Network Analysis (SNA), Static Models, Static Models, System Engineering

1. Introduction

Currently, the usage of social networks is growing exponentially daily, where people react to different events and interact with each other. Social Network Analysis (SNA) operates of analyzing and evaluating social structures using networks and graph theory. Moreover, it is attracting more interest in the scientific community (Abdelsadek et al., 2018). However, it becomes more difficult to analyze data generated by the social networks due to their complexity, which hides the underlying patterns. A collection of interacting entities ends up in advanced systems with hidden properties in several domains, like biology, technology, and economy. These systems are modeled as graphs, where the nodes and edges represent the entities and relationships between them. Additionally, a numerical value can be assigned to the binary edges (Silva et al., 2017). Furthermore, weighted graphs represent the values that can be expressed in either strength or the quantity of two actors. In this context, an efficient network analysis should consider the edge weight in the devised approach.

In this research, we are trying to improve community detection by creating an object-oriented approach to show the overall process in both administrative and technical aspects. One of the widely used social media networks, Twitter, was the central platform of this research. Even with the ease of navigation of Twitter, it is still considered as the primary contributor of unreliable data amongst its competitors. Due to its functionality of collecting data from users and rapid dissemination of information, the downside is that it enables the widespread use of “fake news,” i.e., low-quality news with intentionally false information, which mitigates fake users who are spreading it regularly. Spreading fake news extensively has the potential to change the viewpoint and opinions of people (Alom et al., 2018). Thus, leaving highly negative impacts on individuals and society. Therefore, fake account detection on social media has recently become emerging research that is attracting tremendous

attention. Developing an object-oriented approach creates a more vivid and more precise visualization for immediate understanding of the main scope of the given problem and how to formulate the necessary actions to be made. Once finished, data will be collected on the Twitter API and used to create both static and dynamic networks, but at the same time, using Python to filter and detect the differences between a real and fake user. Although many social media users are legitimate, social media users can also be malicious. There are several cases where users are not even humans but bots. Bots generate automated accounts created by users to generate messages and topics to steer discussions and promote distinct schemes or commodities on social media (Golovchenko et al., 2020). The ease of generating new social media accounts contributes to malicious user accounts, such as social bots, robot users, and trolls. Social robots refer to social networks controlled by computer algorithms to automatically generate content and interact with humans (or other robot users) on social media. Social bots can become malicious entities specifically designed to cause harm, such as manipulating and spreading fake news on social media. Figure 1 shows the impacts of fake accounts on social media. Fake account users are not new. Instead, it has changed over time from newsprint to radio/television and, recently, online news and social media (Milovanović et al., 2019). We denote "traditional fake users" as the fake users' problem before social media affected its production and dissemination. In the diagram, different users generate their made-up persona with different information, which will result in three profiles – cloned, compromised, and bot profile. These profiles will spread disinformation and end up in chaos.

The objectives of this research are: (1) develop a conceptual model defining a solution for improving community detection methods, (2) develop a use case model defining the requirements for solving the community detection problem, (3) develop an object-oriented model that shows the transformation of a static to a dynamic system through nodes, links, attributes, and relationships, and (4) utilize the Pajek networking and Python programming tools to model a dynamic network solution with an object-oriented approach.

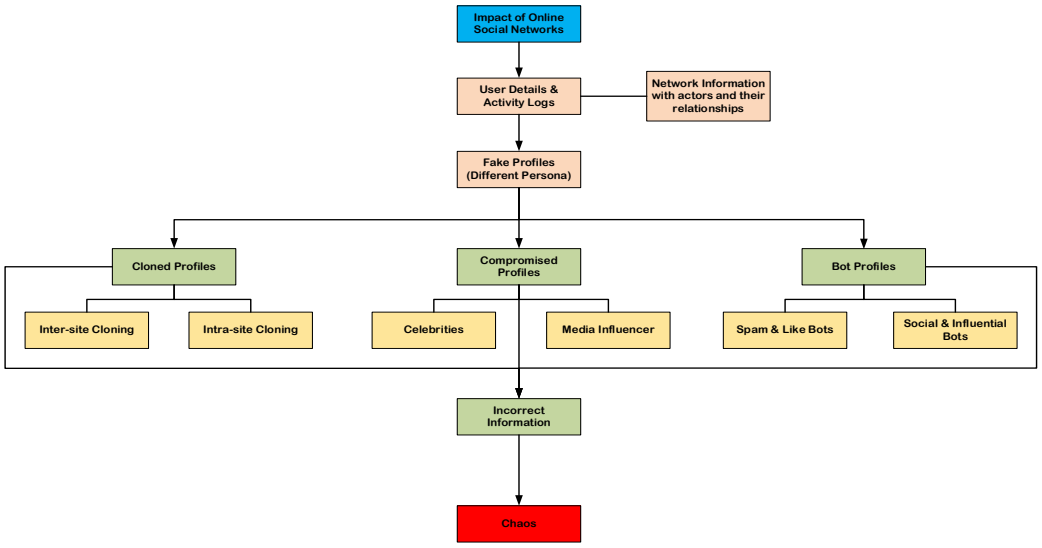

Figure 1. Impact of Fake Accounts in Social-Media

2. Methodology

Figure 2 illustrates the methodology used in this research, from defining the problem, creating a high-level systems diagram, conceptual models, use case models, and object-oriented models, where the created models are part of Phase-1. The structure of figure 2 empowers the whole research effort since it clarifies each classification of the steps in the methodology and their condition. In addition, following this guideline enhances many applications, including robust commenting and note-taking features that help users or the researcher interact with each other and still be on the same track. After the object-oriented analysis is complete, the data collected from the Twitter API will input for modeling using Python Programming and Pajek Network Analysis.

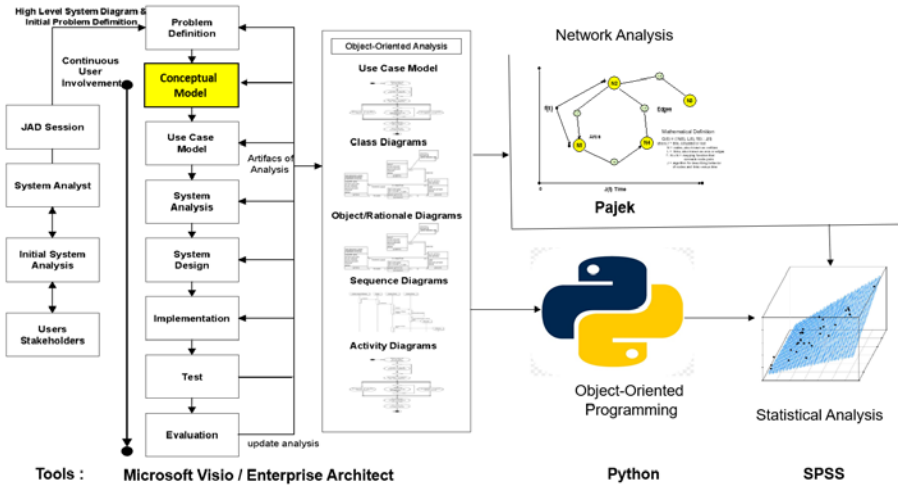

Figure 2. Methodology: Integrated Systems Development Life Cycle (ISDLC) (Bronner, 2021)

Figures 3 illustrates the conceptual community detection model in the Twitter social media network. Users will log in to Twitter to either post a tweet or update their status. Both user and activity log data will be recorded in the Twitter database, accessed, and downloaded through Twitter's API. Once the necessary information is collected, it will be saved on the local storage to be extracted later and filter the users that were flagged in both federal and state data as potential fake accounts. Furthermore, the data will then be used to create static and dynamic networks to show how the data changes over time. Data will also be used in Python logarithm to filter hashtags and specify the given event and address.

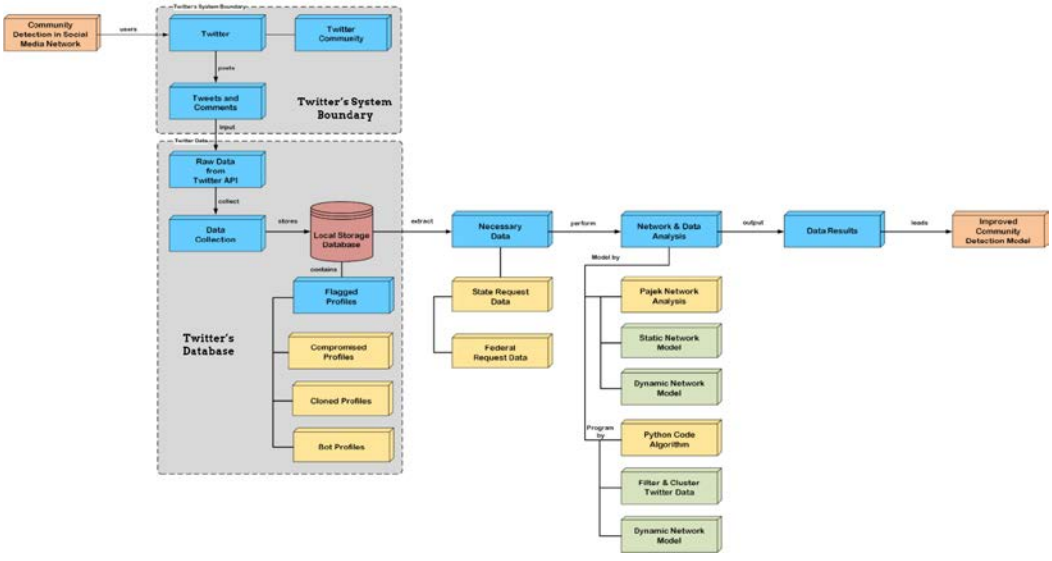

Figure 3. Conceptual Model of Community Detection in Twitter's Social Media Network

3. Results and Discussions

3.1. Use Case Diagrams

Implementing a Unified Modeling Language (UML) for this problem is a must to help researchers understand how a user will interact with the system designed and engineered. Like many diagrams and layouts, it is best to keep the details as minimal as possible. Figure 4 shows the in-depth look for each element of the system through UML usage. Furthermore, each element provides a high-level overview of how each use case, actor, and system are related. Twitter's use case diagram and its Social Media Network categorizes into four major parts, which are (1) Twitter API, (2) Community Detection, (3) Twitter Data Evaluation, and (3) Persona Evaluation. Furthermore, figure 5 elaborates the model through the use case scenarios to provide a high-level view of the system and convey the requirements in laypeople's terms for the stakeholders and users. Additionally, the use case model and its scenarios provide a complete functional and technical view of the created system.

Figure 4. Twitter Social Network Use Case Diagram

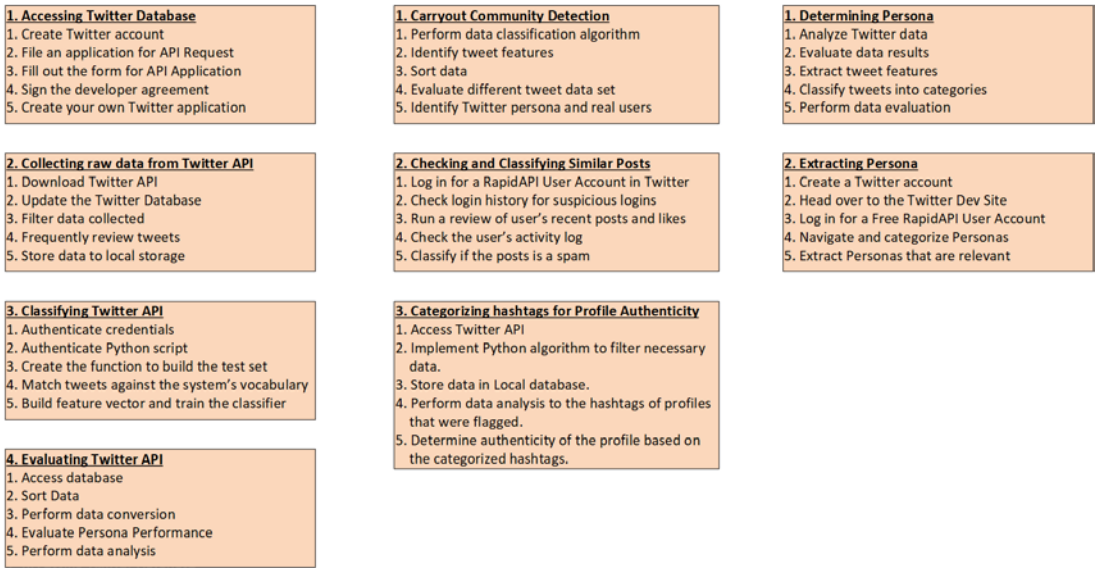

Figure 5. Twitter Social Network Use Case Scenarios

3.2. Object-Oriented Diagram (OOD)

Figure 6 illustrates the Object-Oriented Model of Collecting Raw Data from Twitter API. After logging in, both user and activity logs will be recorded through the Twitter database through its API to update their status. Afterward, the administrator accesses the Twitter API, granting the researcher's request to access the data. The researcher will download the necessary information, such as the user's tweets and usage of hashtags. Once finished, the researcher's local database will store the collected data and filter specific tweets and hashtags from users flagged as fake accounts.

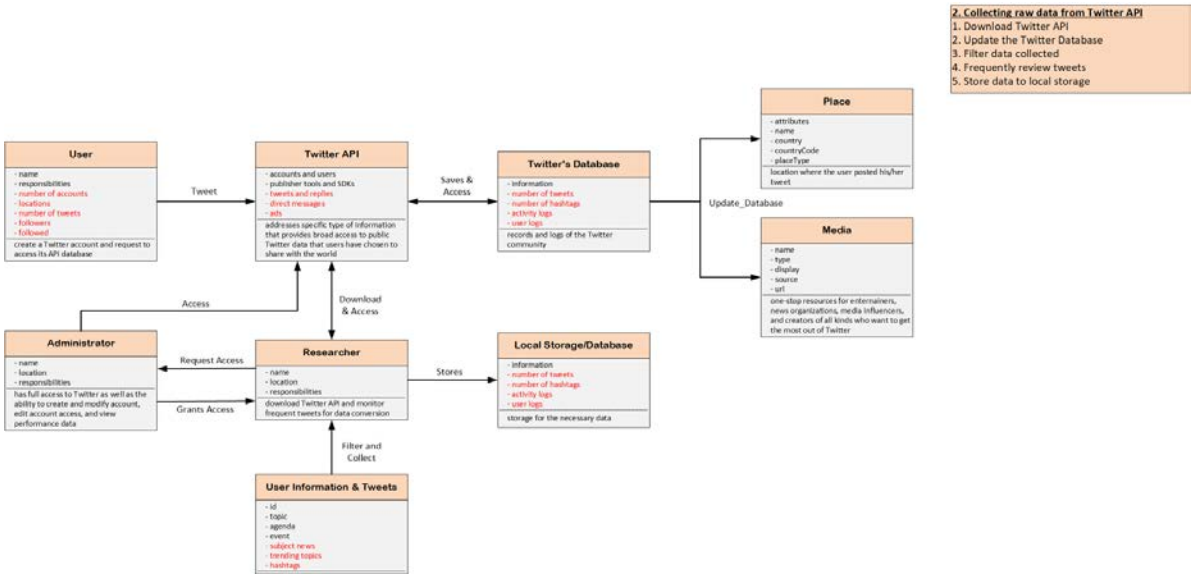

Figure 6. Object-Oriented Model of Collecting Raw Data from Twitter API

3.3. Sequence Diagram (SD)

This project created a series of sequence diagrams (SD) to show how Twitter's operation works. SDs are time-focused, and they show the order of the interaction visually by using the vertical axis of the diagram to represent not actual time, but relative time and messages sent. Figure 7 illustrates a dynamic SD showing the steps to collect the Twitter API data and store it in the local database. Before everything else, the researcher must request permission from Twitter's administrator to access the user's information in the API and download it. Once the researcher collects and stores the necessary data, the Twitter administrator will terminate the researcher's access to continue collecting data from the Twitter API. In addition, the red words in the sequence diagram are dynamic, where it continuously changes either in numbers, places, or events, through time. The boxes below the sequence diagram represent the methods used to power the Python dynamic object-oriented model.

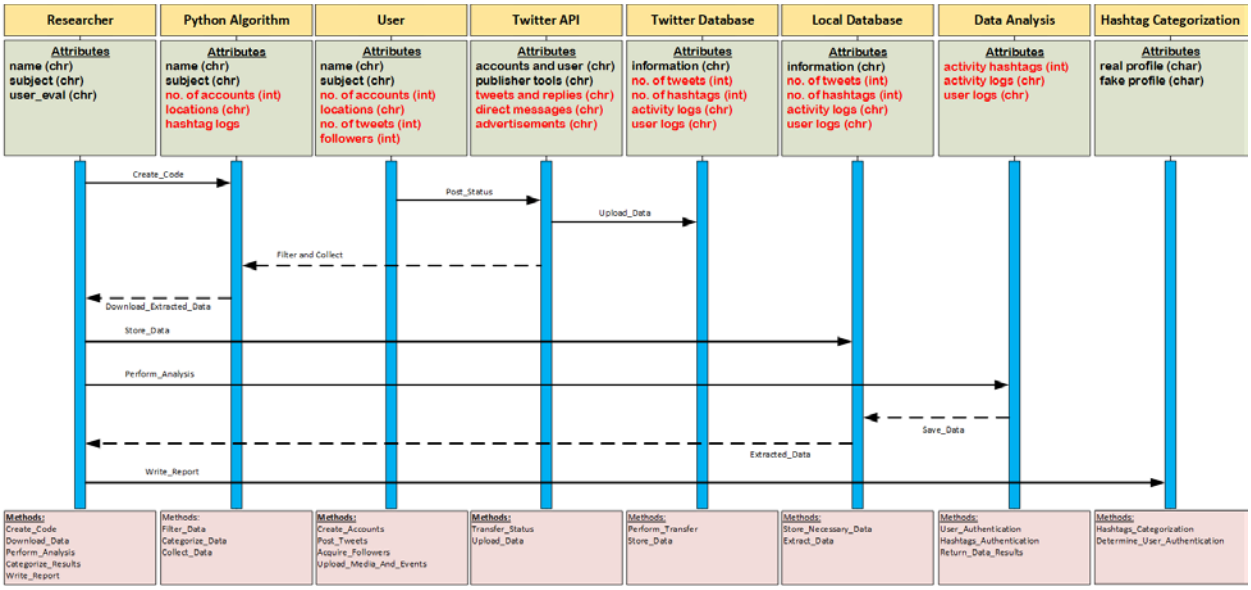

Figure 7. Sequence Diagram for Collecting Raw Data from Twitter API

3.4. Pajek Networking Analysis

In this research, the data extracted was from Twitter's API in 2019 - 2020. The data contains both state and federal reports based upon the number of users flagged as accounts. There are instances where there is a considerable gap of users captured between the state and federal due to the difference of resources in technology – software and systems. Additionally, in 2020, the state that contributed fake accounts based on the federal report was the District of Columbia, which had 658, followed by New York with 293 flagged accounts. However, based on the state report, the highest data they captured was in New York with 121 flagged users, followed by Florida with only 58. We used Python 3.8.2 and VSC (Visual Studio Code) as the IDE to write the code for this research and Twitter's Tweepy library. Figure 8 and 9 illustrates a network diagram using the Pajek Networking Analysis tool. The circles in the diagram represent the states on the U.S. East Coast. Each state has a different and unique size based on the given number of flagged users. Pajek's vector property serves as storage properties of vertices measured in different scales: flagged users in state and federal requests. While on the other hand, Pajek's partitioning function is responsible for changing the different colors of every state. The bigger the circle means that the state has more flagged fake users caught in both state and federal than in its neighboring states. The three leading states with the fakest accounts are New York, Florida, and Virginia, based on the network model.

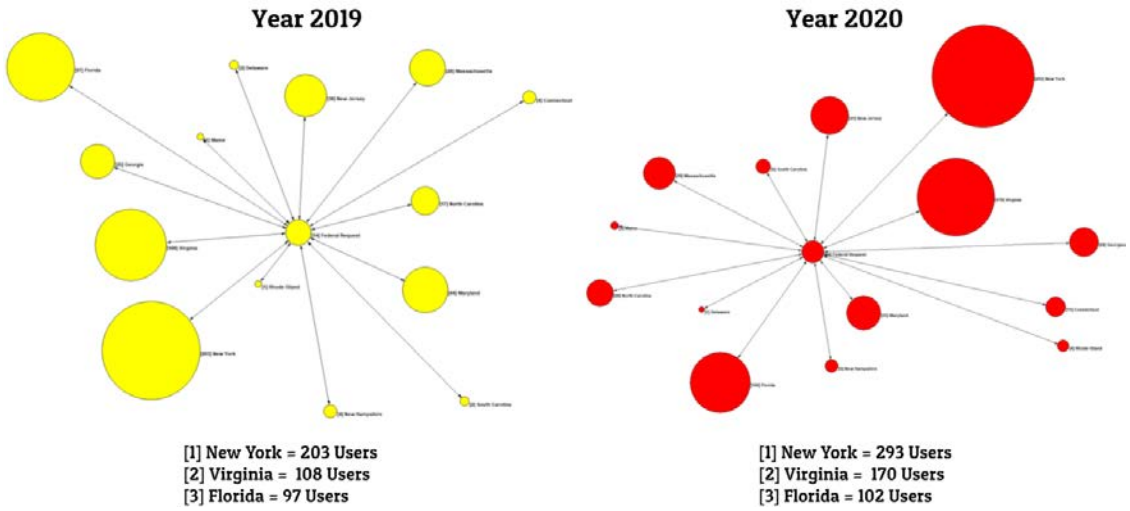

Figure 8. Dynamic Model of Federal Request of the Year 2019 and 2020

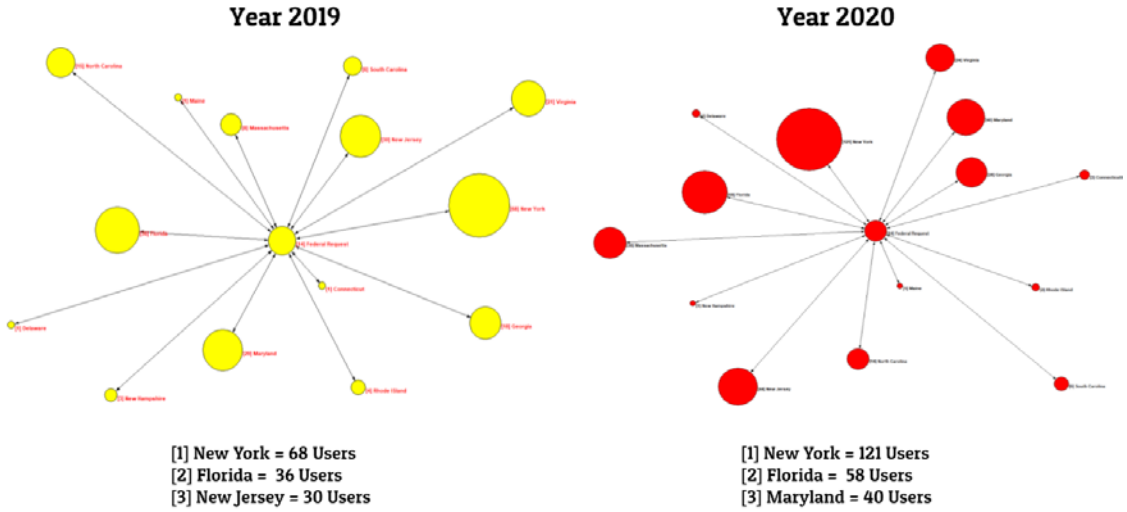

Figure 9. Dynamic Model of State/Local Request of the Year 2019 and 2020

4. Conclusion

In conclusion, this research was able to achieve the stated objectives which are to create object-oriented models and use cases to show the impact of fake accounts in Twitter and how to access and download data through their API. Through this research, the following things were developed: (1) a conceptual model defining a solution for improving community detection method; (2) a use case model defining the requirements for solving the community detection problem; (3) an object-oriented model that shows the transformation of a static to a dynamic system through nodes, links, attributes, and relationship; and (4) A dynamic network model that shows the movement of objects as time goes by. Furthermore, the dynamic network of federal and state requests of flagged profiles from the year 2019 to 2020 shows that there is an imminent increase in fake users that were captured or labeled as flagged accounts. In 2019, there were 203 fake users found in New York, 108 in Virginia, and 97 in Florida. However, it rose to almost 5% in 2020, where New York now has 293, 170 users in Virginia and 102 in Florida.

ISBN: 97819384962-1-9

5. References

Abdelsadek, Y., Chelghoum, K., Herrmann, F., Kacem, I., & Otjacques, B. (2018). Community extraction and visualization in social networks applied to Twitter. *Information Sciences, 424*, 204-223.

Alom, Z., Carminati, B., & Ferrari, E. (2018, August). Detecting spam accounts on twitter. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 1191-1198). IEEE.

Bronner, L. (2021). Integrated System Development Life Cycle for Solving Complex Problems. *Journal of the International Council on Systems Engineering*, Currently Under Peer Review.

Golovchenko, Y., Buntain, C., Eady, G., Brown, M. A., & Tucker, J. A. (2020). Cross-platform state propaganda: Russian trolls on twitter and youtube during the 2016 US presidential election. *The International Journal of Press/Politics, 25*(3), 357-389.

Milovanović, S., Bogdanović, Z., Labus, A., Barać, D., & Despotović-Zrakić, M. (2019). An approach to identify user preferences based on social network analysis. *Future Generation Computer Systems, 93*, 121-129.

Silva, W., Santana, Á., Lobato, F., & Pinheiro, M. (2017, August). A methodology for community detection on Twitter. In *Proceedings of the International Conference on Web Intelligence* (pp. 1006- 1009).