iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: VI Month of publication: June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35911

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

Social Networks for Threat Perception and Analysis

Pragati Dnyaneshwar Bharsakle[1], Dr. M. A. Pund[2]
[1, 2]Computer Science & Engineering, Prof. Ram Meghe Institute Of Technology & Research Badnera Amravati

Abstract: In the current era of massive knowledge, high volumes of valuable knowledge is simply collected and generated. Social networks square measure samples of generating sources of those huge knowledge. Users in these social networks square measure usually coupled by some interdependency like friendly relationship. As these huge social networks continue to grow, there square measure things during which Associate in Nursing individual user needs to seek out common teams of friends so he will suggest a similar teams to alternative users. Many users of social Network are not aware about the number of security risks in networks such as identity theft, privacy violations, sexual harassment etc,. Recent studies says that most of the social network users expose their personal information like their date of birth, email address, phone number, relationship status. If this type of data reached to the wrong person, then person used that information to harm the users. If the children are users of social network, then these risks become serious. In this paper we present an alternative data analytic solution by using pattern matching solution.

Keywords: Big data; social network; pattern matching algorithm; threats; social network analysis

I. INTRODUCTION

Nowadays, high volumes of valuable knowledge are often simply collected and generated from completely different sources like social networks. Social networks usually fabricated from social entities that ar coupled by some specific forms of reciprocality as an example, a Facebook user will produce a private profile, add different Facebook users as friends, and exchange messages.

He can also join common-interest user groups and categorize his friends into different customized lists The number of common friends may vary from one Facebook user to another. Moreover, a Facebook user can also subscribe or follow public postings of some other Facebook users without the need of adding them as friends. Furthermore, the "like" button allows users to show their admiration of content such as status updates, comments, Photos, and advertisements.

In recent years, the number of users in these social networking sites such as Facebook has grown rapidly. Embedded in these big data in social networks is valuable knowledge. This leads to social network analysis or big data analytics on social networks, which aims to computationally facilitate social studies and human-social dynamics in these networks, as well as to design and use information and communication technologies for handling with social context.

Over the past few years, several data mining algorithms and techniques [2], [12] have been proposed. Many of them are applicable to mine social networks In addition, there are also recent works for friend mining and friend recommendation. Examples of the former include the mining of influential friends [5] and strong friends [7]. samples of the latter embrace a propagation framework that collectively targets user interests and predicts friendships [13] and a semantic-based friend recommendation system known as FriendBook [9]. In this dissertation, an alternative data analytic solution is proposed. This dissertation used pattern matching algorithm for the discovery of popular groups of friends and thus recommendations of these friend groups.

II. LITERATURE SURVEY

Fan Jiang, Carson K. Leung , and Adam G. M. Pazdor[1] evaluated our data analytic solution BigPFM both analytically and empirically. The analytical results show that BigPFM is scalable with the amount of accessed and/or stored social network data. The experimental results with the Stanford Netowrk Analysis Project (SNAP) ego-Facebook datasets from the Stanford Large Network Dataset Collection (http://snap.stanford.edu/data/) shows the efficiency and practicality of BigPFM in representing friendships as graphs, discovering groups frequently connected friends, and forming rules for friend recommendation.

Michael Fire, Roy Goldschmidt, and Yuval Elovici [2] present a thorough review of the different security and privacy risks that threaten the well-being of OSN users in general and children in particular. In addition, they present an overview of existing solutions that can provide better protection, security, and privacy for OSN users. they also offer simple-to-implement recommendations for OSN users which can improve their security and privacy when using these platforms.

RizaAktunc and Ismail HakkiToroslu, MertOzer, and HasanDavulcu[5] extended their implementation to define the community structure in a dynamic rather than static way. they made use of the past calculation results of the SLM algorithm in order to calculate the current network community structure. This usage is the main extension and contribution to the SLM algorithm.

Alessandro Epasto, Silvio Lattanzi, VahabMirrokni[6] focussed on analyzing the structural properties of ego-networks. their findings are quite interesting, ego-networks are easily clusterable and the user-defined circles are somehow similar to the cluster retrieved by classic clustering algorithms. Toward this end, we also developed an efficient technique to cluster all the ego-networks in a graph in parallel efficiently. Finally, they develop a new feature for friend suggestion, the ego-network friendship score, and prove theoretically and experimentally that our new feature outperforms the most well-known features

JurgenHolsch and Michael Grossniklaus[7] presented how the relational algebra can be extended with a general Expand operator in order to enable the transformation-based enumeration of different traversal patterns for finding a pattern in a graph G. The input and output of algebra operators are so-called graph relations. The tuples of a graph relation represent subgraphs of G matching a given pattern. Additionally, they introduced a set of equivalence rules and gave proof for their correctness. In the context of Neo4j, they demonstrated how these equivalences can be used to algebraically transform Cypher query at the logical level. Finally, they illustrated the potential performance benefits of this approach by comparing the database hits of the query evaluation plan found to Neo4j to one resulting from applying our equivalence rules.

Irene Teinemaa [8]have studied the problem of predicting which users in a communication network will either increase or decrease their activity after a given period of time and beyond a given change threshold. In contrast to traditional approaches to this problem, which focus on making predictions for individuals, they have approached the problem at the level of communities. Specifically, they applied two representative community detection algorithms: a global modularity-based community detection method (Louvain) and a local-first method that produces denser communities (HDemon). Furthermore, they used single users, ego networks and random groups as the baselines for comparison.

Zhibo Wang [9]presented the design and implementation of Friendbook, a semantic-based friend recommendation system for social networks. Different from the friend recommendation mechanisms relying on social graphs in existing social networking services, Friendbook extracted lifestyles from user-centric data collected from sensors on the smartphone and recommended.

Reynold S. Xin, Joseph E. Gonzalez, Michael J. Franklin, Ion Stoica [11] e have presented GraphX, an interactive graph computation engine that combines the advantages of graph-parallel systems and data-parallel systems. It provides a programming abstraction called Resident Distributed Graphs (RDGs) that significantly simplifies graph loading, construction, transformation, and computations.

Based on RDGs, we implement Pregel and PowerGraph APIs in 20 lines of code. GraphX's internal data representation uses a vertex cut partitioning scheme that minimizes the movement of data during graph computation. GraphX will be open-sourced as part of the Berkeley Data Analytics Stack. We have identified a number of possible future research directions. First, they will continue to work on improving the performance of GraphX. Second, their tabular representation of the vertex-cut partitioning is a natural fit for relational databases and we would like to implement the GraphX abstraction on top of a distributed relational database. A thorough study of performance characteristics between relational databases, Spark, and PowerGraph will help them understand the trade-off in more general systems. Last but not least, they plan to investigate more declarative interfaces that can be compiled down into GraphX programs to further simplify graph computation.

Effectively modeling interest and friendship and accordingly recommending services and/or suggesting friends are fundamental to all social network services. In the paper of Shuang Hong Yang, Bo Long, A. J. Smola, N. Sadagopan, Z. Zheng, and H. Zha, [13] have shown that the interest and friendship information is highly relevant and mutually helpful. they established a joint friendship-interest propagation model that leverages both pieces of evidence to address both tasks in one unified framework. The FIP model bridges collaborative filtering in recommendation systems and random walk in social network analysis with a coupled latent factor model. They conducted extensive experiments to benchmark different variants of FIP in the Yahoo! Pulse social networking system. Two directions of future research appear attractive: The FIP model offers a latent factor for each user that captures both interest and friendship information.

They plan to leverage such deeper user profiles to detect interest communities (i.e. grouping users according to interest with user-friendship in mind) and to identify the macro-behavior (i.e. the global effect as a result of individual actions) of each interest group. They also plan to investigate the underlying mechanism of how the interactions between users impact individual decision-making in the context of social networks.

III.EXISTING SYSTEM

A. Pattern Matching Algorithm

In applied science, pattern matching is that the act of checking a given sequence of tokens for the presence of the constituents of some pattern. In distinction to pattern recognition, the match typically needs to be exact: "either it'll or won't be a match." The patterns typically have the shape of either sequences or tree structures. Uses of pattern matching embody outputting the locations (if any) of a pattern inside a token sequence, to output some part of the matched pattern, and to substitute the matching pattern with another token sequence (i.e., search and replace). Sequence patterns (e.g., a text string) ar usually represented exploitation regular expressions and matched exploitation techniques like backtracking. Tree patterns ar employed in some programming languages as a general tool to method information supported its structure, e.g. C#, F#, Haskell, ML, Rust, Scala, Swift and also the symbolic arithmetic language Mathematica have special syntax for expressing tree patterns and a language construct for conditional execution and worth retrieval supported it. For simplicity and potency reasons, these tree patterns lack some options that ar accessible in regular expressions. usually it's doable to allow various patterns that ar tried one by one, that yields a robust conditional programming construct. Pattern matching generally includes support for guards.

B. Steps of Pattern Matching Algorithm

Main steps in algorithm is:

1) Look at current character, and possible current state in FSM.
2) If the state in FSM is a choice node, that means the current character might correspond to either of the choices- so put them on the queue as possibilities for current character.
3) If the state in the FSM contains a character matching the current character, then the next character should match the next state in the FSM- so put that next state on the (end of) queue.

C. System Design

Fig. 1 shows the architecture diagram of Existing System. The working and functionality of each component is given in this section. The system is divided into two parts i.e. front end and back end.

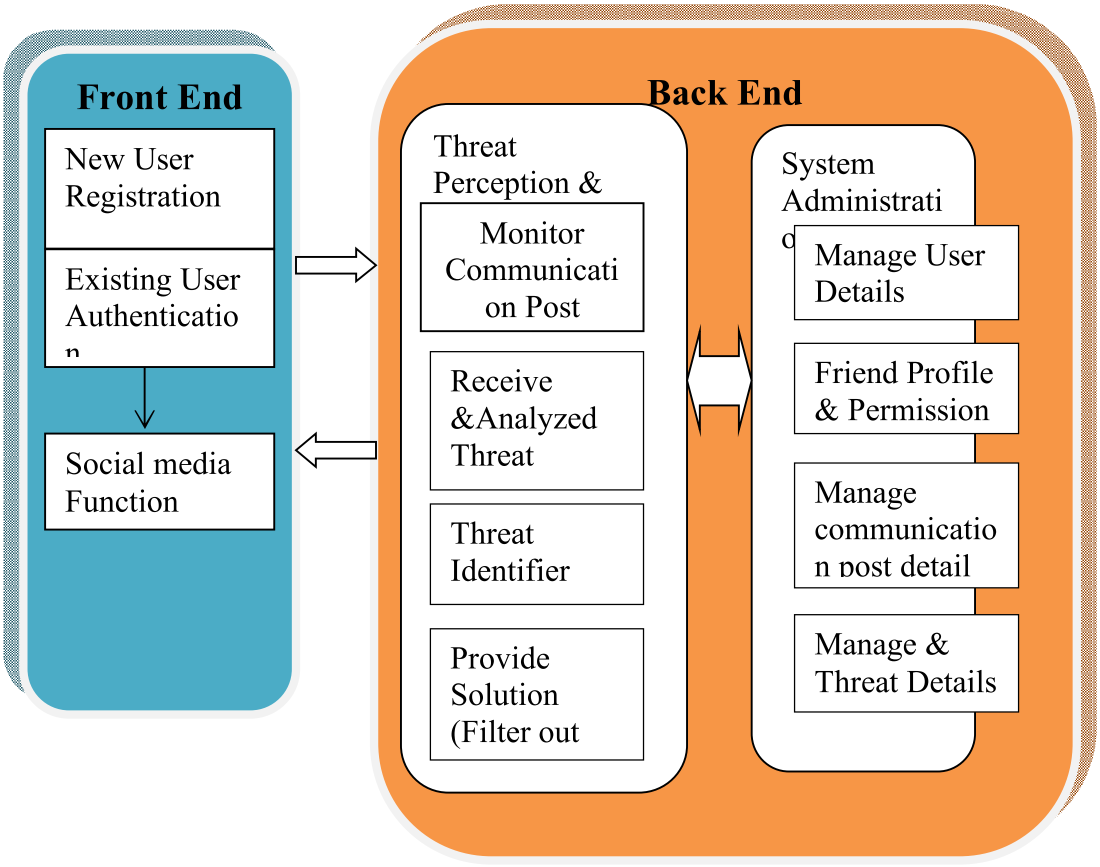

Fig. 1 System Architecture

1) *Front End:* This includes what the user sees, such as text, images, and the rest of the UI, along with any actions that an application performs within the user's browser.

a) *New User Registration:* If there is new user on social media platform then he/she has to do registration first by filling registration form.

b) *Existing User Authentication:* Once registration is done, users have to do login. Login process is called as Authentication.

c) *Social Media Function:* After authentication process users are allow for performing social media functions like see user profile, edit profile, share messages with friends, send/ accept friend request and post photos and so on.

2) *Back End:* It is divided into two parts i.e. threat perception and analysis and system Administration.

a) *Threat Perception and Analysis:* While using social networking if an attacker threatens the user then fallowing modules perform their function.

- Monitor Communication Post: To check whether there is threat occurs or not, monitor the communication post.
- Receive and analysed threat: If threat occurs in system, then our system received that threat and investigate it.
- Threat identifier: Threat identifier recognizes that what type of threat it is.
- Provide Solution: According to type of threat our system provide a solution.

b) *System Administration:* Function of system administration is to monitor the whole system. Some functions of system administration is given below:

- Manage User Detail: Admin of the system can manage user details like manage user profile, personal information and so on.
- Friend profile and permission: System admin can manage friend profiles of user and their permission.
- Manage Communication post Detail: Admin can manage the post detail to save user from attackers.
- Manage Threat details: To overcome the problem of threat admin manages the threat details like, monitor the post, analysis of threats, identify what type of threat it is, and provide solution according to threats.

IV. CONCLUSIONS

We proposed a big data analytic solution that conducts big social network mining and analysis via popular friendship mining. It helps social network users to discover groups of frequently connected users from big social networks. Evaluation results show the efficiency and practicality of mining big social networks, discovering popular users, and recommending friends. When it comes to security and privacy risks that threaten the well-being of OSN users in general. We present solutions that can provide better protection, security, and privacy for OSN users that means when any type of attack occurs, the system alerts the user.

V. ACKNOWLEDGMENT

It is our proud privilege to release the feelings of our gratitude to every person who helped us directly or indirectly to conduct this research work. we express our heart full indebtness and owe a deep sense of gratitude to Prof. Sangram Dande for their sincere guidance and inspiration for completing this paper.

REFERENCES

[1] Fan Jiang, Carson K. Leung , and Adam G. M. Pazdor, "Big Data Mining of Social Networks for Friend Recommendation," in Proceedings of the ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), August 18-21, 2016,pg 921-922.
[2] Michael Fire, Member, Roy Goldschmidt, and Yuval Elovici, Member, "Online Social Networks: Threats and Solutions,"VOL. 11, NO. 4, DECEMBER 2012
[3] https://en.wikipedia.org/wiki/Social_media_mining
[4] https://www.adweek.com/digital/5-social-media-threats/
[5] R. Aktunc, I. H. Toroslu, M. Ozer, and H. Davulcu, "A dynamic modularity based community detection algorithm for large-scale networks: DSLM," in Proc. IEEE/ACM ASONAM-FAB 2015, pp. 1177–1183.
[6] A. Epasto, S. Lattanzi, V. S. Mirrokni, I. Sebe, A. Taei, and S. Verma, "Ego-net community mining applied to friend suggestion," PVLDB, 9(4), pp. 324–335, Dec. 2015.
[7] J. Hölsch and M. Grossniklaus, "An algebra and equivalences to transform graph patterns in Neo4j," in Proc. EDBT/ICDT Workshops 2016. http://ceur-ws.org/Vol-1558/paper24.pdf
[8] https://ithandbook.ffiec.gov/it-booklets/information-security/iii-security-operations/iiia-threat-identification-and-assessment.aspx
[9] https://www.cloudflare.com/learning/serverless/glossary/client-side-vs-server-side/

[10] C. K. Leung, F. Jiang, A. G. M. Pazdor, and A. M. Peddle, “Parallel social network mining for interesting ‘following’ patterns,” Concurrency and Computation: Practice & Experience, 2016. DOI:10.1002/cpe.3773.

[11] C. K. Leung, S. K. Tanbeer, and J. J. Cameron, “Interactive discovery of influential friends from social networks,” Social Network Analysis and Mining, 4(1), art. 154, Dec. 2014.

[12] C. K. Leung, S. K. Tanbeer, A. Cuzzocrea, P. Braun, and R. K. MacKinnon, “Interactive mining of diverse social entities,” KES Journal, 20(2), pp. 97–111, May 2016.

[13] S. K. Tanbeer, C. K. Leung, and J. J. Cameron, “Interactive mining of strong friends from social networks and its applications in ecommerce,” Journal of Organizational Computing and Electronic Commerce, 24(2-3), pp. 157–173, 2014.