

# Cross-Platform Identification of Anonymous Identical Users in Multiple Social Media Networks

P.Hemanth Kumar Reddy<sup>1</sup>, A.Mallikarjuna<sup>2</sup>, S.Ramakrishna<sup>3</sup>

<sup>1</sup> PG Scholar, Department of Computer Science, S.V University, Tirupati – INDIA

<sup>2</sup> Teaching Assistant, Department of Computer Science, S.V University, Tirupati -INDIA

<sup>3</sup> Professor, Department of Computer Science, S.V University, Tirupati- INDIA

**Abstract-** The last few years have witnessed the emergence and evolution of a vibrant research stream on a large variety of online social media network (SMN) platforms. Recognizing anonymous, yet identical users among multiple SMNs is still an intractable problem. Clearly, cross-platform exploration may help solve many problems in social computing in both theory and applications. Since public profiles can be duplicated and easily impersonated by users with different purposes, most current user identification resolutions, which mainly focus on text mining of users' public profiles, are fragile. Some studies have attempted to match users based on the location and timing of user content as well as writing style. However, the locations are sparse in the majority of SMNs, and writing style is difficult to discern from the short sentences of leading SMNs such as Sina Microblog and Twitter. Moreover, since online SMNs are quite symmetric, existing user identification schemes based on network structure are not effective. The real-world friend cycle is highly individual and virtually no two users share a congruent friend cycle. Therefore, it is more accurate to use a friendship structure to analyze cross-platform SMNs. Since identical users tend to set up partial similar friendship structures in different SMNs, we proposed the Friend Relationship-Based User Identification (FRUI) algorithm. FRUI calculates a match degree for all candidate User Matched Pairs (UMPs), and only UMPs with top ranks are considered as identical users. We also developed two propositions to improve the efficiency of the algorithm. Results of extensive experiments demonstrate that FRUI performs much better than current network structure-based algorithms.

## CROSS-PLATFORM

cross-platform software (multi-platform, or platform independent software) is computer software that is implemented on multiple computing platforms. Cross-platform software may be divided into two types; one requires individual building or compilation for each

platform that it supports, and the other one can be directly run on any platform without special preparation, e.g., software written in an interpreted language or pre-compiled portable bytecode for which the interpreters or run-time packages are common or standard components of all platforms.

## EXISTING SYSTEMS

Existing algorithms FRUI chooses candidate matching pairs from currently known identical users rather than unmapped ones. This operation reduces computational complexity, since only a very small portion of unmapped users are involved in each iteration. Moreover, since only mapped users are exploited, our solution is scalable and can be easily extended to online user identification applications. In contrast with current algorithms FRUI requires no control parameters. The main question in the above scenario is the overlap of the users' friends. To address this issue, we discuss the overlap of SMNs, including node and edge overlap, below. Node overlap. Many studies have verified that numerous users are overlapped in different SMNs. Nearly all cross-platform user identification studies mention node overlap, because it is the fundamental assumption to solve this issue. Early in 2007, 64% of Facebook users had MySpace accounts.

## PROPOSED SYSTEMS

Proposing a novel Friend Relationship-based User Identification (FRUI) algorithm. In our analysis of crossplatform SMNs, we deeply mined friend relationships and network structures. In the real world, people tend to have mostly the same friends in different SMNs, or the friend cycle is highly individual. The more matches in two unmapped

users' known friends, the higher the probability that they belong to the same individual in the real world. Based on this fact, we proposed the FRUI algorithm. A preprocessor is designed to acquire as many Prior UMPs as possible. Currently, there is no common approach available to obtain UMPs between two SMNs. Specified methods must be formulated according to given SMNs. Although no unified process is suitable for the Preprocessor, some algorithms can be adopted according to the application, e.g., email address, screen name, URL, etc. Edge overlap. Until very recently, no statistical studies quantified relationship overlap in two SMNs. However, some studies noted that these relationships overlap to a certain extent. NS which identifies users purely through networks in ground-truth datasets, proved that users have similar relationships in Twitter and Flickr. Paridhi also found that users tend to connect with a segment of the same people across SMNs, and introduced network structure to improve the accuracy of user identification between Twitter and Facebook.

#### ADVANTAGES

Advances in SMN services, more SMNs allow users to bind their accounts with other major SMNs. In this case, priori knowledge can be obtained with bound information. For example, PaPa and ChangBa, two major mobile applications (apps) in China, encourage users to link their Sina Microblog accounts for commercial interests, bridging their websites with the largest microblog service in China. Twitter provides an attribute, called a URL, for user self-identification. Preprocessors can directly use URLs to match a Twitter account to Facebook or other SMN accounts. When no extra information except the network structure can be employed, the seed identification approach in NS and the de-anonymization attacks in are alternatives for the Preprocessor.

#### IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

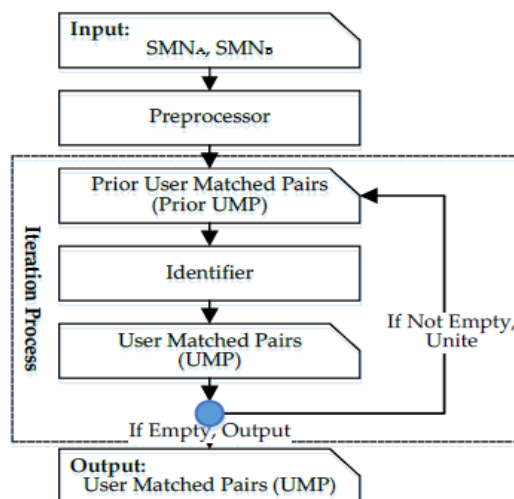
#### MODULES

In this project we have following four modules .

- i).Cross-PlatForm In SMN's
- ii).Anonymous Identical User
- iii).Friends And Relation

Cross-PlatForm In SMN's:-

SMN connections fall into two categories: single-following connections and mutual-following connections. Singlefollowing connections are also called following relationships or following links. If user A follows user B, then user A and user B have a following relationship (single-way fans in which one knows the other, but not vice versa). Following relationships are common in microblogging SMNs, such as Twitter and Sina Microblog. Likewise, mutual-following connections are called friend relationships. In microblogging SMNs, a friend relationship refers to the mutual following relationships between two users. In our analysis of crossplatform SMNs, we deeply mined friend relationships and network structures. In the real world, people tend to have mostly the same friends in different SMNs, or the friend cycle is highly individual. The more matches in two unmapped users' known friends, the higher the probability that they belong to the same individual in the real world. Based on this fact, we proposed the FRUI algorithm.



#### Anonymous Identical User:-

Anonymous is a loosely associated international network of activist and hacktivist entities. A website nominally associated with the group describes it as "an Internet gathering" with "a very loose and decentralized command structure that operates on ideas rather than directives". The group became known for a series of well-publicized publicity stunts and distributed denial-of-service attacks on government, religious, and corporate websites. Although no unified process is suitable for the Preprocessor, some algorithms can be adopted according to the application, e.g., email address, screen name, URL, etc. An email address appears to be a unique feature for each account, and can be used to collect Priori UMPs. Node overlap. Many studies have verified that numerous users are overlapped in different SMNs. Nearly all cross-platform user identification studies mention node overlap, because it is the fundamental assumption to solve this issue. The identifier finds UMPs using connections among users and Priori UMPs. As noted above, a match degree for each candidate UMP should be calculated in advance. NS formulates the match degree using in- and out-degrees in directed networks.

#### Friends And Relation:-

The friend relationship requires confirmation by the two users, and is much more reliable and consistent in SMNs. Thus, it can reduce the noise introduced by a discretionary single-following relationship. Making use of the friend relationship in undirected networks, JLA defines the match degree as, For any two SMNs, SMNA and SMNB can be considered as mirrors of the real world. Suppose that people set up random friendships in the real world; then the probability of a friendship between any two persons is  $p$  ( $0 < p < 1$ ), and for any friendship,  $s_a$  ( $0 < s_a < 1$ ) and  $s_b$  ( $0 < s_b < 1$ ) are probabilities that it exists in SMNA and SMNB, respectively. Therefore, the probabilities that a friendship exists in SMNA and SMNB are  $ps_a$  and  $ps_b$ , respectively. we use ground truth datasets to evaluate the user identification resolution. In order to verify FRUI in different types of SMNs, we collected data from two heterogeneous SMNs: Sina Microblog and RenRen. The Sina Microblog dataset was captured from the Sina Microblog search page, while the RenRen dataset was directly obtained from its Open API. As shown in the Sina Microblog dataset

consisted of 1.17 million users and 1.9 million friend relationships, and each user had an average of 3.2 friends. The RenRen dataset was comprised of 5.5 million nodes and 14.6 million edges, and each user had an average of 5.3 friends. Therefore, the RenRen dataset was much denser than Sina Microblog's.

### ALGORITHMS

#### FRUI (Friends and Relation User Identifier):-

In the implementation, the Identifier first calculates matrix  $R$  using Proposition 1 and initializes the match degree. Then it iterates and identifies UMPs using function  $g$  until no UMP can be identified. In each iteration, once the UMPs are identified, the items are removed from the Candidate UMP list, and  $R$  is recalculated based on Proposition 2. The process is summarized in Algorithm 1. Suppose that there are  $s$  Valid Priori UMPs in any iteration. Lines 4-11 in Algorithm 1 remove the identified UMPs and update the maximum match degree, and the time complexity costs  $O(s) + O(\min(v_A, v_B)) = O(\min(v_A, v_B))$ , where  $v_A$  and  $v_B$  denote the numbers of the users in SMNA and SMNB, respectively. Lines 12-19 update the Candidate UMP list and the maximum match degree using Propositions 1 and 2.

#### FRUI algorithm:-

Input: SMNA, SMNB, Priori UMPs: PUMPs

Output: Identified UMPs: UMPs

1: function FRUI(SMNA, SMNB, PUMPs)

2:  $T = \{\}$ ,  $R = \text{dict}()$ ,  $S = \text{PUMPs}$ ,  $L = []$ ,  $\max = 0$ ,  $FA = []$ ,  $FB = []$

3: while  $S$  is not empty do

4: Add  $S$  to  $T$

5: if  $\max > 0$  do

6: Remove  $S$  from  $L[\max]$

7: while  $L[\max]$  is empty

8:  $\max = \max - 1$

9: if  $\max == 0$  do

10: return UMPs

11: Remove UMPs with mapped UE from  $L[\max]$

12: foreach  $UMPA \sim B(i, j)$  in  $S$  do

13: foreach  $UEA_a$  in the unmapped neighbors of  $UEA_i$  do

14:  $FA[i] = FA[i] + 1$

15: foreach  $UEA_b$  in the unmapped neighbors of  $UEA_j$  do

16:  $R[UMPA \sim B(a, b)] += 1$ ,  $FB[j] = FB[j] + 1$

```

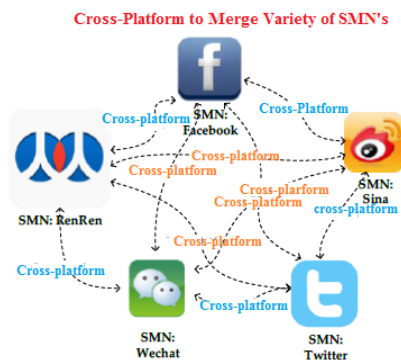
17: Add  $UMPA \sim B(a, b)$  to  $L[R[UMPA \sim B(a, b)]]$ 
18: if  $R[UMPA \sim B(a, b)] > \max$  do
19:  $\max = R[UMPA \sim B(a, b)]$ 
20:  $m = \max$ ,  $S = \{ \}$ 
21: while  $S$  is empty do
22: Remove UMPs with mapped UE from  $L[\max]$ 
23:  $C = L[m]$ ,  $m = m - 1$ ,  $n = 0$ 
24:  $S = \{ \text{un-Controversial UMPs in } C \}$ 
25: while  $S$  is empty do
26:  $n = n + 1$ ,  $I = \{ \text{UMPs with top } n \text{ } M_{ij} \text{ in } C \text{ using (5)} \}$ 
27:  $S = \{ \text{un-Controversial UMPs in } I \}$ 
28: if  $I = C$  do
29: break ;

```

#### Clustering:-

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem.

#### ARCHITECTURE DIAGRAMS



#### CONCLUSION

This study addressed the problem of user identification across SMN platforms and offered an

innovative solution. As a key aspect of SMN, network structure is of paramount importance and helps resolve de-anonymization user identification tasks. Therefore, we proposed a uniform network structure-based user identification solution. We also developed a novel friend relationship-based algorithm called FRUI. To improve the efficiency of FRUI, we de-scribed two propositions and addressed the complexity. Finally, we verified our algorithm in both synthetic networks and ground-truth networks. Results of our empirical experiments reveal that network structure can accomplish important user identification work. Our FRUI algorithm is simple, yet efficient, and performed much better than NS, the existing state-of-art network structure-based user identification solution. In scenarios when raw text data is sparse, incomplete, or hard to obtain due to privacy settings, FRUI is extremely suitable for cross-platform tasks. Results of our empirical experiments reveal that network structure can accomplish important user identification work. Our FRUI algorithm is simple, yet efficient, and performed much better than NS, the existing state-of-art network structure-based user identification solution. In scenarios when raw text data is sparse, incomplete, or hard to obtain due to privacy settings, FRUI is extremely suitable for cross-platform tasks.

#### ACKNOWLEDGEMENTS

The authors thank the three anonymous reviewers for providing constructive and generous feedback.

#### REFERENCES

- [1] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework," *J. Netw. Comput. Appl.*, vol. 67, pp. 147–165, May 2016.
- [2] M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of Internet of Things and cloud computing," *J. Netw. Comput. Appl.*, vol. 67, pp. 99–117, May 2016.
- [3] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Generat. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

- [4] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in Proc. 10th Int. Conf. Wireless Algorithms, Syst., Appl. (WASA), Qufu, China, 2015, pp. 685–695.
- [5] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for Cloud of Things," in Proc. IEEE Int. Conf. Future Internet Things Cloud (FiCloud), Barcelona, Spain, Aug. 2014, pp. 464–470.
- [6] V. Medina and J. M. García, "A survey of migration mechanisms of virtual machines," ACM Comput. Surv., vol. 46, no. 3, 2014, Art. no. 30.
- [7] A. Shribman and B. Hudzia, "Pre-copy and post-copy VM live migration for memory intensive applications," in Euro-Par 2012: Parallel Processing Workshops (Lecture Notes in Computer Science). New York, NY, USA: Springer, 2012, pp. 539–547.
- [8] U. Deshpande, Y. You, D. Chan, N. Bila, and K. Gopalan, "Fast server deprovisioning through scatter-gather live migration of virtual machines," in Proc. 7th IEEE Int. Conf. Cloud Comput. (CLOUD), Anchorage, AK, USA, Jun./Jul. 2014, pp. 376–383.
- [9] C. Jo, E. Gustafsson, J. Son, and B. Egger, "Efficient live migration of virtual machines using shared storage," in Proc. 9th ACM SIGPLAN/SIGOPS Int. Conf. Virtual Execution Environ., Houston, TX, USA, 2013, pp. 41–50.
- [10] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations," IEEE Commun. Mag., vol. 50, no. 9, pp. 34–40, Sep. 2012.