2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

SpeculoLab: A Protocol and a Tool for Identity Deception Experimentation in Social Networks

Noora Al Roken, Maryam Al Abdooli, Sumaya Khoory, Hakim Hacid
College of Technological Innovation
Zayed University, Dubai, UAE
(201605954, 201608635, 201614136, hakim.hacid)@zu.ac.ae

***Abstract*—A good understanding of the underlying mechanisms that govern identities on the Web is a key aspect for ensuring the privacy of users but also solving some ethical related problems. This paper proposes SpeculoLab, a platform implementing a strong and well defined experimental protocol for supporting research in the area of multiple identities in the (social) Web. The platform supports an end-to-end control of the experimentation process and, more importantly, allows personalizing and extending every part of the process. SpeculoLab is provided as an open source for the community for further improvements and reinforcement.**

***Index Terms*—Social network analysis, Identity deception, Multiple identity detection, Duplicate record detection**

I. Introduction

Thanks to Web 2.0, exemplified by social networks, users can freely express themselves. The feeling of anonymity encourages users to share and comment about different matters varying from political turmoils to economical trends and latest movies, criticize companies, laws, governments, news, etc. While it is agreed that the possibility of creating multiple accounts on the same (social) platform or different ones brings tremendous values in term of freedom of speech and overcoming some psychological barriers, it may also be misused, for example for bullying and social bots. Thus, being able to identify, when necessary, multiple identities can be the right compromise between full anonymity that is provided currently by (social) platforms and full and strict control of the content. Understanding the way multiple identities function can also help protecting users in their social ecosystem if needed to increase their privacy. Efforts around identities will only increase with time, especially with the related privacy issues that this topic includes.

Extracting multiple identities of a user is a difficult task, necessitates an important mining process, and may become impossible for several reasons. The first reason is the absence of dedicated data that could be used for such objective. Other aspects complicate the problem of multiple identities extraction, including: (i) absence of links between accounts, which is actually even more complicated since users with multiple accounts tend to ensure (intentionally or not) that the accounts are as disjointed as possible. Then, (ii) meta-data is impacted as the explicit information attached to accounts in the case of the same social network tend to be of a very limited use due to the expected objective, i.e., hiding the real user behind the accounts. Finally, (iii) the amount of data (e.g., number of interactions and resources shared) generated on the accounts in different social networks tend to follow a similar behavior, which is the opposite in the case of one social network. As such, data generated may be different in terms of quantity from one identity to another. For example, through her main account, a user shares many interactions with a large number of friends while using other accounts, the interactions are limited with a reduced number of connections.

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 © 2020 IEEE

Several efforts are being operated in the area of multiple identity detection in social networks, e.g., [6] [1] [7] [5] [7] [4] [2]. However, these efforts use their own data (usually not public), their own code, and their own experimentation protocol for performance and quality evaluations. This paper proposes SpeculoLab, a platform implementing a strong and well defined experimental protocol for supporting research in the area of multiple identities in the (social) Web. The platform supports an end-to-end control of the experimentation process and, more importantly, allows personalizing and extending every part of the process. The rest of this paper is organized as follows: Section II discusses the system architecture and its different components. The demonstration scenarios are described in Section III. Finally, we conclude and provide some future work in Section IV.

II. Foundations of SpeculoLab

The main objective of this work is to support research around the area of multiple identities in the (social) Web. This goes through an effort to "normalize" and simplify the evaluation and assessment of contributions in this area, especially with the lack of ground truth data. Such a platform must provide an easy and flexible access to its capabilities, as well as rich expansion possibilities. Furthermore, the platform must provide a clear evaluation protocol (or an abstraction of it) to allow an objective comparison of contributions and, more importantly, reproducible results.

A. System Architecture

The architecture of SpeculoLab, which supports the proposed experimentation protocol, is illustrated in Figure 1. It is divided into five main steps: (i) Data Collection, (ii) Identities

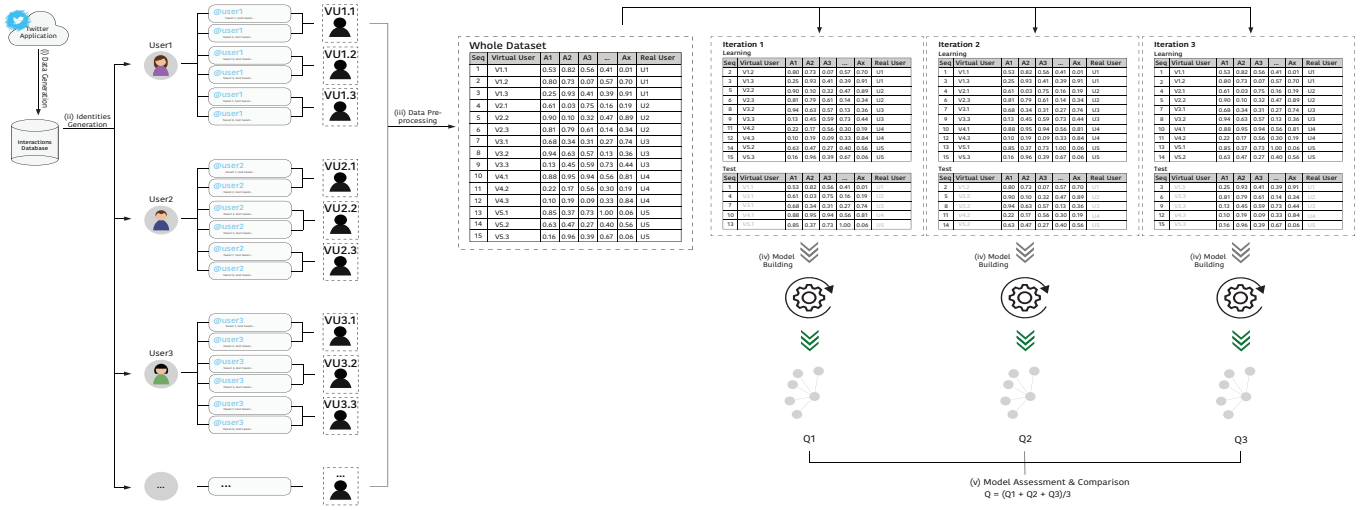

Fig. 1: Overview of the main components of the architecture

Generation, (iii) Data Pre-Processing, (iv) Model Building, and (v) Model Assessment and Comparison. First, in the *Data Collection* step, the social interactions can be either collected (e.g., from Twitter by providing the necessary access keys) or just loaded by the user into the system's database. The fetched data is directly stored in a database and in both cases the storage is user-centric, i.e., the main entity of interest is user. This component is run once to collect the interactions of users. The second step, *Identities Generation*, is responsible for generating virtual accounts that are associated then with real-users. Depending on an input configuration (detailed in Section II-C), this step generates different virtual users with their associated interactions. This data is then fed into the *Data Pre-Processing* step.

The *Pre-Processing* step takes the generated virtual users and their interactions to extract descriptive features for each generated virtual user. These features can be classified into six categories: source, textual, emotion, social, mistakes, and time. The output of this component is a matrix that associates a row with each virtual user (i.e., real users end-up with multiple rows). The *Model Building* step exploits the extracted features to build classification models based on different learning algorithms. SpeculoLab comes with two embeded algorithms: k-D-Tree [8] and Cover Tree [9]. Finally, the last step is that of *Model Assessment and Comparison*, where the built models get evaluated and compared to other models.

B. Identities Generation

This is a key step in the overall protocol. The used data-sets in this context are user-centeric containing interactions posted by different accounts, that we call *Real Users*. These users are denoted as $R = \{r_1, r_2, ..., r_p\}$, with $|R| = p$. For every real user, $r_i \in R$, m *Virtual identities*, denoted $V = \{v_{(1,1)}, ..., v_{(1,m)}, v_{(2,1)}, ..., v_{(2,m)}, ..., v_{(p,1)}, ..., v_{(p,m)}\}$ are associated. $v_{i,j} \in V$ is the j^{th} virtual identity of the i^{th} real user. Note that m can be the same, or different, for the considered real users. This depends on the objective as this will be discussed later in the experiments. Also, while m can be randomly high, we argue that a user can not control a large number of identities in social networks, at least not at the same time. It is recommended to limit this value to 10to remain reasonable and have a good balance between realistic situations and expected scientific outcomes.

Each data-set contains n observations representing users (real or virtual) which are injected in the analysis process. Finally, this is also related to a technical constraints wherein by increasing m to higher values, the volume of interactions that would be associated with each virtual user can be extremely low, and consequently of a limited use for extracting valuable outcomes. In fact, data gathering from social networks is generally limited and necessarily constrain the study. SpeculoLab provides indications about the impact of the chosen parameters on the considered interactions and virtual users. This allows a better consideration of the situations that would make some sense experimenting.

Virtual identities are always related to real users in the different considered data-sets. In addition to always keeping the pair $(v_{i,j}, r_i)$, the naming of the identities follows a simple, yet efficient scheme. Technically, virtual users are named by adding a sequence number to the name of the corresponding real user. For instance, assume a real user account *@xyz*. When $m = 3$ for example, we automatically generate three virtual users: *@xyz.1*, *@xyz.2*, and *@xyz.3*. To handle the variety of constraints, we introduce the concept of *Experiment Configuration*.

C. Experiment Configuration

An *experiment configuration* encapsulates the different parameters used to generate the data (users, virtual users, interactions, etc.) that are used for further pre-processing, and

ultimately analysis. Virtual users are generated from real users. Once the virtual users are created, for each $v_{i,j} \in V$ is associated a set of interactions that are inherited from the parent real user. We then define an *Experiment Configuration* as a tuple, given by Equation 1:

$$c_i = (n, m, q, k, s, \phi) \tag{1}$$

In Equation 1, n, is the number of real identities involved in the considered experiment, and m, is the number of virtual accounts associated with each real user. The number of interactions associated with each virtual user denoted q where k, is the size of the neighborhood. This can be seen as in the sense of $k - nn$ where we consider the k closest neighbors, following a distance measure, to make a decision. For performance and interpretation reasons, this parameter is expressed as a percentage. So, when $k = 5\%$, this means that 5% of the nearest neighbors to the target identity are used for verification. The amount of sampling is denoted with s. By sampling we refer to the consideration of the whole sequence of messages or parts of it, i.e., with holes in the sequence. In the experiments, $s \in \{1, 1/2, 1/3, 1/4, 1/5\}$ wherein $s = 1$ expresses the case where all the data is considered in a full sequence, while $s = 1/2$ expresses the situation where interactions of a virtual identity are sampled by considering only one interaction out of two for the analysis. The same reasoning is followed for the rest of the possible values. Finally, $\phi \in \{All, Textual, Social, Source, Emotion, Time, Mistakes\}$, captures the features space that is used to perform the analysis.

Equation 2 illustrates a data-set with a configuration of $5,000$ involved identities with 5 virtual identities are associated with each, and each virtual identity must have 100 interactions. These interactions are in full sequence, i.e., no holes. During the learning/test process, 5% of the closest identities are used to compute the quality, i.e., if the other identities are found or not.

$$c_1 = (5000, 5, 100, 5, 1, All) \tag{2}$$

The set of all experiment configurations is denoted by $C \in \{c_1, c_2, ..., c_p\}$. We make use of several configurations each capturing a specific aspect of the multiple identity problem. Finally, once the interactions are associated with each virtual user, its feature vector, composed of ≈ 500 features, is then extracted. In addition, the real user as well as virtual user are attached to each vector.

D. Model building and quality assessment

At this stage, the data is ready for analysis. With such data, we have handled a supervised learning problem where the predictive features are calculated on each virtual user while the class represents the real user, i.e., two virtual users originating for the same real user represent an occurrence of multiple identities. As noticed previously, this context brings in an extra layer of difficulty: (i) there is a very large number of classes, (ii) the classes are not highly represented, and (iii) classes may vary in terms of number of associated observations. While this constitutes a challenging context, it corresponds to real cases in social networks.

Having said that, we follow a classical learning/test data generation: (i) the learning subset is used to build the models and (ii) the test subset is used to evaluate the model. As for the proportion between learning and test data-sets, it mainly depends on the number of virtual identities associated with each real user. For instance, when each real user has 2 associated virtual users, the learning data-set contains 50% of the observations and the test data-set contains the other 50% of the observations, giving us two iterations. However, in the case where users have, for example, 3 virtual identities, the learning data contains $\approx 67\%$ and the test data-set contains $\approx 33\%$. This leaves us naturally with 3 iterations. To have a holistic view on the performance, and exclude the effect of chance, SpeculoLab operates several rounds on the same data-set. In fact, each observation must participate at least once in the test data-set but usually present many times in the learning set (i.e., a cross validation principle).

Finally, SpeculoLab comes with neighborhood-based techniques that allow increasing/decreasing the size of the neighborhood in the process of searching for multiple identities. Thus, if two virtual identities, derived from the same real user, are in the suspicious set for a target identity, these are considered to be a match and are duplicate of the target identity. Thus, we make use of the precision as a quality measure.

E. Storage and Computation

It is expected that SpeculoLab handles large datasets and performs an important amount of computation. In fact, after the identities generation, features have to be calculated for each user and his/her interactions w.r.t. the considered experiment configurations. For example, if a user chooses to generate configurations for users with a number of virtual identities varying from 2 to 10 (with a step of 1), each having interactions like, e.g., 50, 100, 150, and 200, the raw data is then dynamic and changing for identities. This forces the system to recalculate all the representation space in each configuration, for each virtual user.

To handle this part, SpeculoLab integrates (i) a noSQL database and (ii) a distributed processing tool. In fact, SpeculoLab integrates *Apache Cassandra*[1] for data storage, which allows us to handle larger sets of data in an efficient manner. It should be noted here that SpeculoLab generates a new database for each experience (set of experiment configurations) in order to (1) ensure a clean working environment and (2) increase the performance of the system. Further to the storage of the data, SpeculoLab integrates a *Apache Spark*[2] to perform the computation of the features. This is critical as this is the most demanding phase due to the dynamics of the configurations, impacting the number of recalculations that need to be performed for each experiment configuration.

[1] http://cassandra.apache.org/

[2] https://spark.apache.org/

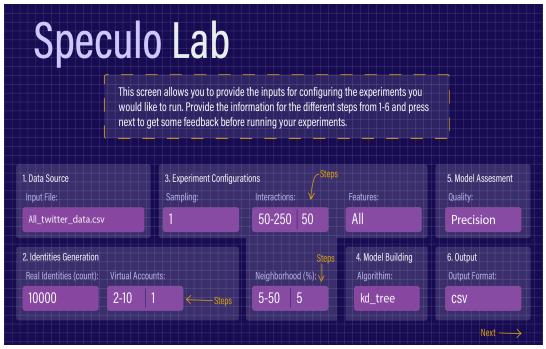

Fig. 2: Interface for setting up the initial parameters

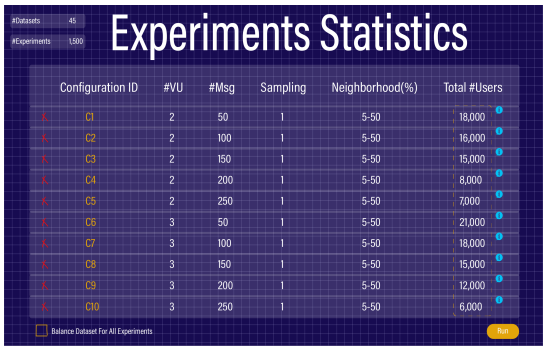

Fig. 3: Interface adjusting the parameters

III. Demonstration Scenarios

SpeculoLab targets researchers interested in exploring questions related to multiple identity detection in the (social) Web. In this perspective, the demonstration will focus on two main scenarios. This platform usage scenario focuses on providing a basic understanding of the different components of the SpeculoLab to allow the user to get familiar with, e.g., the data flow. The scenario will consider a database of Twitter interactions, and a set of experiments configurations as well as their declaration in the system. An important part of this scenario is related to the setup of the parameters (Figure 2) and their impact on the overall evaluations. For example, users can check the outcome of the selected experiment configurations on the data through a dedicated output interface (i.e., Figure 3). They can also fine tune the parameters depending on the summary provided by SpeculoLab. An obvious and useful example of such fine tuning is to decide whether the generated dataset attached to each experiment should be balanced or not.

To understand better the complexity of the task and the capacity of SpeculoLab to simplify it, we make use of two distinct dataset where the first dataset is large enough (in terms of users and associated interactions), while the second is smaller in terms of interactions. The user is expected to appreciate the summary functionality before even running the generation process. Finally, and in order to appreciate the results, the export and reporting process will be executed to send the data to external files for further processing and/or visualization.

The second scenario is for platform expansion which considers the case where a user (researchers) intend to expand the platform with new features, algorithms, or quality measures. To illustrate this, we propose to expand and integrate a new feature in the platform, e.g., the ratio of generated interactions by the user, and its embedding in the full learning and evaluation process. After integrating a new feature, we move to the integration of a new learning algorithm, e.g., Cover Tree. The demonstration will illustrate the procedure to follow to add such algorithm and the needed steps to integrate its output into the following steps, quality assessment and output capture.

Finally, and in order to be complete, this scenario will also demonstrate the way of integrating and using a new quality measure into the platform, e.g., F1 measure [3]. The idea here is to allow a user to use his/her own measures for assessing the quality of algorithms. Again, we will show the integration of the results of new quality into the output capture and storage, which can be used for further visualization, analysis, and comparison.

IV. Conclusion and Future Work

We present in this paper SpeculoLab, an open source platform that implements a strong experiment protocol for multiple identity detection evaluation and comparison. This platform is expected to support researchers and practitioners in objectively assessing their contributions and comparing them with others. The platform supports an end-to-end control of the experimentation process and, more importantly, allows personalizing and extending every part of the process. As a future work, we plan to improve the interaction with the system by making all the capabilities available through a user interface. Also, reporting capabilities will be extended by making these available directly on the interface instead of generating the output into a spreadsheet.

References

[1] Kayode Sakariyah Adewole, Nor Badrul Anuar, Amirrudin Kamsin, Kasturi Dewi Varathan, and Syed Abdul Razak. Malicious accounts: Dark of the social networks. *Journal of Network and Computer Applications*, 79:41 – 67, 2017.

[2] Kahina Gani, Hakim Hacid, and Ryan Skraba. Towards multiple identity detection in social networks. In *WWW'12*, pages 503–504. ACM, 2012.

[3] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *ECIR05*, page 345359. Springer-Verlag, 2005.

[4] Paridhi Jain and Ponnurangam Kumaraguru. Finding nemo: Searching and resolving identities of users across online social networks. *CoRR*, abs/1212.6147, 2012.

[5] Srijan Kumar, Justin Cheng, Jure Leskovec, and V.S. Subrahmanian. An army of me: Sockpuppets in online discussion communities. In *Proceedings of WWW 2017*, pages 857–866.

[6] Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. Surveying stylometry techniques and applications. *ACM Comput. Surv.*, 50(6), November 2017.

[7] Zaher Rabah Yamak. *Multiple identities detection in online social media*. PhD thesis, 02 2018.

[8] Parikshit Ram and Kaushik Sinha. Revisiting kd-tree for nearest neighbor search. In *KDD'19*, page 13781388. Association for Computing Machinery, 2019.

[9] Nikolaos Tziortziotis, Christos Dimitrakakis, and Konstantinos Blekas. Cover tree bayesian reinforcement learning. *CoRR*, abs/1305.1809, 2013.