A

CADIVa: Cooperative and Adaptive Decentralized Identity Validation Model for Social Networks

Amira Soliman, Royal Institute of Technology (KTH)
Leila Bahri, Insubria University
Sarunas Girdzijauskas, Royal Institute of Technology (KTH)
Barbara Carminati, Insubria University
Elena Ferrari, Insubria University

Online Social Networks (OSNs) have successfully changed the way people interact. Online interactions among people span geographical boundaries and interweave with different human-life activities. However, current OSNs identification schemes lack guarantees on quantifying the trustworthiness of online identities of users joining them. Therefore, driven from the need to empower users with an identity validation scheme, we introduce a novel model, Cooperative and Adaptive Decentralized Identity Validation *CADIVa*, that allows OSN users to assign trust levels to whomever they interact with. *CADIVa* exploits association rule mining approach to extract the identity correlations among profile attributes in every individual community in a social network. *CADIVa* is a fully decentralized and adaptive model that exploits fully decentralized learning and cooperative approaches not only to preserve users privacy, but also to increase the system reliability and to make it resilient to mono-failure. *CADIVa* follows the ensemble learning paradigm to preserve users privacy and employs gossip protocols to achieve efficient and low-overhead communication. We provide two different implementation scenarios of *CADIVa*. Results confirm *CADIVa*'s ability to provide fine-grained community-aware identity validation with average improvement up to 36% and 50% compared to the semi-centralized or global approaches, respectively.

Additional Key Words and Phrases: Identity Validation, Online Social Networks, Distributed Systems, Privacy Preservation, Decentralized Online Social Networks

1. INTRODUCTION

Online Social Networks (OSNs) have changed the way people communicate and have provided new forms of communication and social interactions. Online interactions span geographical boundaries and interweave with the different daily life activities. The realm of OSNs design shows variety in purpose for different types of interactions among people. Some sites keep a very professional approach (like LinkedIn[1]), while most sites mix professionalism with personalization (like Facebook[2] and Google+[3]). However, all of these sites employ a lightweight process for obtaining membership identities (i.e., confirming a valid email address) to facilitate their smooth joining and fast adoption. Moreover, when users create their profiles on these OSNs, they are given

[1]www.linkedin.com
[2]www.facebook.com
[3]https://plus.google.com/

This work is under the umbrella of the iSocial EU Marie Curie ITN project (FP7-PEOPLE-2012-ITN).
Author's addresses: A. Soliman and S. Girdzijauskas {aaeh, sarunasgg@kth.se}, Software and Computer Systems, School of Information and Communication Technology, KTH, Stockholm, Sweden; L. Bahri, B. Carminati and E. Ferrari {leila.bahri, barbara.carminati, elena.ferrarig@uninsubria.it}, DISTA, Insubria University, Italy.

DOI: http://dx.doi.org/10.1145/0000000.0000000

the complete freedom to fill up the records of their profiles without validating them. Consequently, such convenience increases the vulnerability of such networks to undergo security threats such as spam, malware, and phishing attacks [Huber et al., 2011; Luo et al., 2009; Jagatic et al., 2007]. One of the recently trending threats to OSNs is the spread of fake accounts that are seeking to get social [Robinson, 2015; Stringhini, 2014].

Fake accounts are nothing new to the online world in general and to OSNs in particular. Despite all the efforts to aid the detection of fake accounts, they still make a considerable proportion of the active online population of today's major OSNs. For instance, as of December 2015, Facebook has been reported to have 1.49 billion accounts out of which at least 83 million are known to be fake.[4] What is more dangerous than the existence of these fake accounts is their exploitation to build social trust; hence making honest targets more willing to trust dangerous content or putting the privacy of their information at risk [Robinson, 2015; Stringhini, 2014]. This social trustworthiness is mainly achieved by means of creating personal connections with honest users. Indeed, most of the techniques available for fake accounts detection rely on the premise that fake accounts exhibit tendencies of densely connected groups that are weakly connected to the rest of the OSN, or outlying behavior that is skewed compared to common trends [Yu et al., 2006a; Yu et al., 2008]. As such, once a fake account succeeds at befriending honest users, its chances of getting detected would be considerably diminished. Moreover, the established connections may allow the fake account to inherit some of the trust accorded to the befriended honest account; thus give to the fake account more credibility resulting in higher chances of fooling other honest users [Stringhini, 2014]. This suggests that there may be a need for a mechanism that facilitates the validation of profiles in an OSN to allow honest users to take better informed decisions before accepting a new connection in the network.

Several approaches have been proposed to address the problem of identity validation of users in OSNs. Particularly, online identity validation targets the estimation of trustworthiness of an OSN profile in terms of linking this profile to a true social human identity. However, all of the existing approaches tend to compromise users' privacy in their trial to achieve some security goals. For example, some of them identify users by utilizing their sensitive information such as geo-locations they usually visit and time-stamps of the information they share [Goga et al., 2013]. In [Chairunnanda et al., 2011], authors use typing patterns to identify users, whereas chatting patterns are exploited in [Roffo et al., 2013]. Additionally, other validation approaches have suggested to rely on human feedback. For example, in [Sirivianos et al., 2012] the authors suggest to evaluate an identity on a given network based on feedback of her connections on another one. Generally, all of these techniques are derived from the incentive to validate online identities, yet they fail to limit the boundary of information to be used to fulfill their objective without violating users privacy or revealing their sensitive information to other entities who are not privileged to access it.

More importantly, further privacy concerns also emerge as a result of the centralized architecture of todays popular OSNs. In particular, this centralized architecture has critical consequences such as the necessity for a high degree of trust in the OSN provider, censorship of users behavior and the utilization of users' data for business-related purposes [Debatin et al., 2009; Dwyer, 2011]. Therefore, in the last decade, researchers and the open source community have proposed various decentralized OSNs (DOSNs) (e.g., [Koll et al., 2014; Nilizadeh et al., 2012; Kapanipathi et al., 2011]) that remove dependency on a centralized provider. The main objectives behind decentralization are to preserve users privacy in both shared content and communication, and

[4]https://zephoria.com/top-15-valuable-facebook-statistics/

also to provide complete freedom from any form of censorship or profiling. DOSNs operate as distributed information management platforms on top of networks of trusted servers or P2P infrastructures [Datta et al., 2010]. Thus, DOSNs provide a privacy preserving alternative to current OSNs, where users have full control of their data.

Although the DOSNs paradigm presents promising ways for preserving users privacy, it creates even more challenges when it comes to validating users identities. Indeed, in the absence of a central management entity, designing mechanisms to control online identities in a DOSN brings up several challenges. First, all the information that could be exploited to validate a user's identity is solely owned and managed by him/her and is not available to any other entity. Second, with the privacy preservation set as a first requirement, an appropriate solution should not exploit any information outside of its owner's boundaries. In addition to that, an adequate solution should not subvert the decentralized architecture of DOSNs. That is, collaboration between peers should ideally be exploited in a fully decentralized manner without introducing subcentral entities or super peers that might constitute single points of failure or privacy breach entities.

1.1. Motivation

Starting from the requirement of preserving users' privacy in suggesting an appropriate solution for identity validation of peers in a DOSN, it sounds crucial to limit the exploited information to publicly available one only and to not move data across its ownership boundaries. A possible approach is to utilize the provided profile information of a user. Particularly, the evaluation can be done based on the integrity of the provided profile information, and veracity of reflecting actual real identity of the profile owner. This idea has been suggested in [Bahri et al., 2014] where the authors suggest using community feedback to assign trustworthiness levels to users on a social network based on the profile information they exhibit. The authors show that there exists a dependency among different profile attributes such that their corresponding values are expected to exhibit some correlation within any truthful profile. They do this by gathering human feedback from a trusted set of users on a centralized profiles training dataset. Once these correlations are identified, again they engage users' feedback, to estimate the identity trustworthiness level of a target profile. In particular, the computed trustworthiness level of any target profile indicates the homogeneity between values in the user's profile and the identified correlated attributes. Although, the proposed approach succeeds in limiting the required information to identify identity trustworthiness by relying only on profile information, using users' feedback might be violating users privacy. Moreover, this solution relies on the existence of a central repository of profiles from which the correlations between attributes could be extracted.

Basing the learning of correlations between profile attributes on all the users profiles as one unified entity (i.e., the global correlations that are generated using all the profiles) would capture generalizations across all of the users in the network and might result in discriminatory validation patterns to minorities. For example, if we consider a network of one million users, extracting statistically significant correlations from the whole population provides broad commonalities shared across the whole population such as interests in specific sports. This might not apply to all users at a micro scale, and would result in incorrectly validating their profiles. At this point, it comes logical to exploit network relationships as well as they reflect groupings of people that might be representative of common identity trends. In fact, it has been found that social networks exhibit a clustering phenomena by which users topologically cluster into communities [Krivitsky et al., 2009; Ferrara, 2012]. Furthermore, users inside every community typically have high similarity to each other sharing common identity and

background trends [Ferrara, 2012]. Thus, it sounds more realistic to validate identities within single communities instead of considering all the users base as one flat entity and resulting in identity validation patterns that could be very vague and too general to apply to users at micro levels. For example, if the majority of users work in information technology jobs and share the same interests in electronics and gadgets, global learning will enforce a correlation rule between job and interests. On the other hand, smaller communities of people who work on different careers (e.g., school teachers) will be penalized for mismatching this rule.

Furthermore, it is intuitively observable that people have multiple community memberships. For example, a person usually has connections to multiple groups or communities, such as family members, colleagues, friends, and co-workers. Therefore, it is more reasonable to compute multiple trustworthiness levels per user according to the different communities that he or she belongs to. Therefore, the objective of identity validation systems is to go beyond the existing solutions of binary classification (i.e., classifying a new profile as legitimate or fake) and provide a community-aware identity validation. Community-aware identity validation systems will have the ability to identify the existing communities in the social network and extract the correlated attributes inside each community. Consequently, identity validation of new users can be performed by quantifying the overlapping among the profile attribute values of the new users with respect to the existing correlated attributes of the communities which these users want to join. Moreover, new users can have multiple trustworthy levels to the communities they join. For example, a new user who is a computer science student in some university has a high trustworthiness level for the community of computer science students as they study the same subjects. However, the same student may have a lower trustworthiness level for the community of music bands as he/she is not interested in music.

In addition to these limitations, relying on a central learning repository to unveil identity trends does not align with our target scenario of DOSNs. Thus, to design a solution tailored to DOSNs, and to overcome the limitations related to the centralized and supervised approach exploited in [Bahri et al., 2014] to extract the correlations among profile attributes from a profile schema, we previously proposed in [Soliman et al., 2015], a decentralized identity validation (DIVa) model that adopts a quasi-decentralized approach. Instead of supervised learning that requires human feedback, DIVa successfully conceptualizes users online identities by extracting the correlations among profile attributes from the user population. Additionally, DIVa provides community-based validation by mining the correlations form individual communities not from the user population as a whole. DIVa achieves this in a three phase process that starts by each node learning the collection of its local correlated attribute sets (LCAS) by exploiting association rule mining over the profiles of its direct friends only. In the second phase, a community detection mechanism is deployed to define the communities existing in the network. Thereafter, every node, knowing the communities to which it belong, communicates its learned collection of LCAS to the super nodes of its membership communities. These super nodes, referred to as *diva nodes*, are unique in each community and are responsible of receiving all LCAS collections from all the nodes in their community and aggregating them to generate the community level correlated attribute sets (i.e., CAS). As such, DIVa provides stronger fine-grained validation rules (i.e., a set of CASes per community) that reflect the existing patterns inside every existing community instead of the global trends that any new profile can maintain. Thereafter, every user can use these correlations to evaluate the truthfulness of new profiles he or she desires to become friend with. In particular, the evaluation of identity trustworthiness depends on the coherence of its claimed identity against the discovered correlations of the targeted community.

1.2. Contribution

The provided model in [Soliman et al., 2015] demonstrated good results in meeting the goal of designing an identity validation model for DOSNs that uses minimal information (i.e., profile information only); however, this model does not provide a fully decentralized solution. In fact, [Soliman et al., 2015] assumes the availability of some super nodes (i.e., diva nodes) that are exploited as central hubs within each community and are used to aggregate the final community-based profile attribute correlations. These super nodes might constitute single points of failure or performance bottlenecks in the system as the process depends on their availability and on their ability to perform the tasks entrusted to them. Moreover, the assumption of super nodes does not fully align with the fully decentralized spirit of DOSNs. In addition to that, the work in [Soliman et al., 2015] bases on static assumptions across all communities for the threshold values adopted to learn significant correlations within each community. That is, all communities adopt the same threshold value for the learning of a valid correlation between profile attributes of their members, ignoring the specific characteristics of every community such as size, homogeneity, etc.

To address these limitations, in this paper we suggest a cooperative and adaptive decentralized identity validation model (CADIVa), that is fully decentralized and adaptive. CADIVa exploits gossip learning to provide fully decentralized and cooperative learning, not only to preserve users privacy, but also to increase the system reliability and to make it resilient to mono-failure. Furthermore, CADIVa tunes the statistical significant threshold for selecting profile attribute correlations according to the number of nodes belonging to each community. Adaptive thresholds increase the freedom of each community to have the value that reflects the level of homogeneity among its constituent members.

CADIVa operates based on a gossip-based algorithm to cut off the role of the super nodes in aggregating their communities CASes, and to engage all of the nodes in a community instead. As this might result in a communication overhead, we demonstrate the trade-off between convergence and network overhead and propose two different implementations of CADIVa. In the first one, the community detection phase is executed first and performed separately from the aggregation phase, whereas in the second implementation we combine both the community detection and LCASes aggregation to minimize the overall communication overhead. The results show that both versions of CADIVa achieve improvement up to 36% and 50% than DIVa and global approach, respectively.

Furthermore, the main motivation behind CADIVa is to quantify the trustworthiness of new users joining the social network and recommend neglecting the connections from untrustworthiness users. However, the users have the complete freedom either to follow CADIVa's recommendations or to neglect them. Therefore, we developed CADIVa as adaptive and self-correcting model that continuously updates communities validation rules while new nodes being added to the communities. The first part of CADIVa's adaptability lies on computing different threshold values according to the statistical strength of attribute pairs frequency inside every community independently. Secondly, CADIVa monitors the topological changes in the communities after adding the new nodes/edges. Afterwards, CADIVa re-performs the CAS learning in the regions where communities topologically change. We perform a set of experiments following hierarchical community detection to show how CASes change with the increase of community size. The results emphasize the ability of CADIVa to extract the community-level CASes that reflect the topological structure of the underlying communities and the properties of the user population belonging to each community.

The main contributions of this paper can be summarized as follows:

— A cooperative, massively parallel and reliable identity validation model that preserves users privacy and operates without super nodes support, hence it suitably fits DOSNs.
— Community-aware identity validation model that reveals mostly frequent fine-grained identity patterns inside every community more accurately than existing semi-centralized or global approaches.
— Adaptive identity validation model that is capable of tuning the model parameters to reflect the existing homophily level inside every community.
— Incrementally learning model that monitors the evolving changes in the underlying social graph and updates communities validation rules.

The rest of the paper is organized as follows. Section 2 provides background on the proposed identity validation scheme. In Section 3, we describe the CADIVa model and detail its two suggested implementations. Section 4 provides security and complexity analysis, whereas Section 5 presents and discusses experiments results. In Section 6 we survey the related work and then conclude the paper in Section 7.

2. BACKGROUND: COMBINING PROFILE AND NETWORK DATA FOR IDENTITY VALIDATION

The goal of our proposed model is to provide users in a DOSN with an identity validation service that would help them in assessing the trustworthiness of their new online contacts. Our target requirements are to achieve this goal without subverting the privacy preservation guaranteed by the DOSN design and to offer a solution that is fully decentralized without relying on super nodes that might constitute single point of failure. To answer our goal, we suggest a model that exploits detected correlations between profile attributes in a profile schema to provide communities in a DOSN with sets of correlated attribute sets (CAS) that reflect the identity trends of their members. These CASes can be used by community members to assess the trustworthiness of new contacts desiring to connect with them. Basically, our model bases on two assumptions. First, social networks exhibit a clustering feature by which users topologically cluster into communities with connections within a community denser than across communities. Second, people within communities share common identity trends and patterns that could be extracted and that could be used to validate the identity of new members desiring to connect with them.

To extract those correlations in a profile schema, we exploit the principles of Association Rule Mining (ARM) [Agrawal et al., 1993]. ARM is a data mining model that has been extensively used in market-basket analysis, to extract rules on how a subset of items influences the presence of another subset [Agrawal et al., 1993; Agrawal et al., 1994; Kotsiantis and Kanellopoulos, 2006; Hipp et al., 2000]. Similarly in our scenario, we are interested in finding the set of correlated attributes and quantifying the dependency relations among them. Hence, for identity validation an association rule can be, "a user who is employed at company X also lives in city Y". To infer such rules, the proposed model extracts the frequent profile attributes values inside each community and identifies their equivalent profile attributes as Correlated Attribute Set (CAS). As such, the evaluation of identity trustworthiness of a profile can be performed based on the coherence of its claimed attribute values against the discovered CAS and their values.

To be aligned with the DOSN design and with our target privacy preservation requirements, we design our suggested model based on a node-centric approach and structure its operation in three main phases, as illustrated in Figure 2. First, all the nodes in the network collaboratively execute a decentralized community detec-

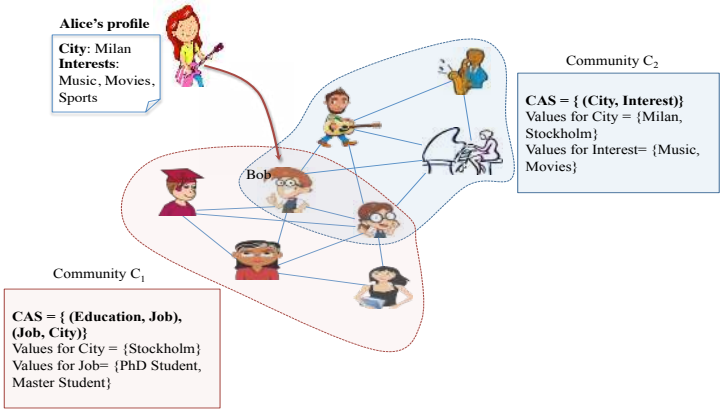

Fig. 1. Illustrative scenario for Example 2.1. Alice sends a friendship request to Bob who evaluates Alice's profile w.r.t the validation rules of the communities he belongs to.

tion algorithm[5] to detect topological communities existing in the social network. Second, every node independently executes ARM learning using its local data (i.e., the profile information of its direct friends only) to extract its local correlated attribute set (LCAS) that exist among its direct friends. Thus, profile information is processed locally and any possibility of mismanagement or accidental disclosure of profiles information is diminished because users' data are not moved outside their trusted zone. Finally, nodes participate in a voting mechanism to formulate the community consensus from LCASes and reach a common community-level CASes.

Once the nodes agree on their communities CASes, nodes can evaluate the integrity of profile information of a new user desiring to connect with them. To illustrate the validation process, we provide the scenario in Example 2.1.

Example 2.1. Let us assume we have two communities C_1 and C_2 in the OSN where we found that {Education, Job}, and {Job, Current City} form the C_1's CAS, while { Current City, Interests} forms C_2's CAS. The existence of the two attribute pairs in CAS of C_1 is agreed on by and communicated to all its nodes, and the same applies for C_2. Consider Bob to be a member of the two communities and assume Alice is a new user who wants to connect with Bob (see Figure 1). Bob knows that in his first community (i.e, C_1), trustworthy nodes should demonstrate homogeneity between Education and Job and between Job and Current City. Therefore, Bob can estimate the trustworthiness of Alice's profile by checking the values she provides to these attribute couples. The estimation of Alice's profile indicates that Alice has a low trust level to be a member of C_1, however, she is more trustworthy to be in C_2. So, Bob can accept Alice's friendship and consider her to be a member of C_2 but not C_1. To assist users to have clearer judgment, the model also provides the top-n values associated with each CAS in their community.

[5]We exploit the community detection algorithm suggested in [Rahimian et al., 2014] as it provides a fully decentralized solution.

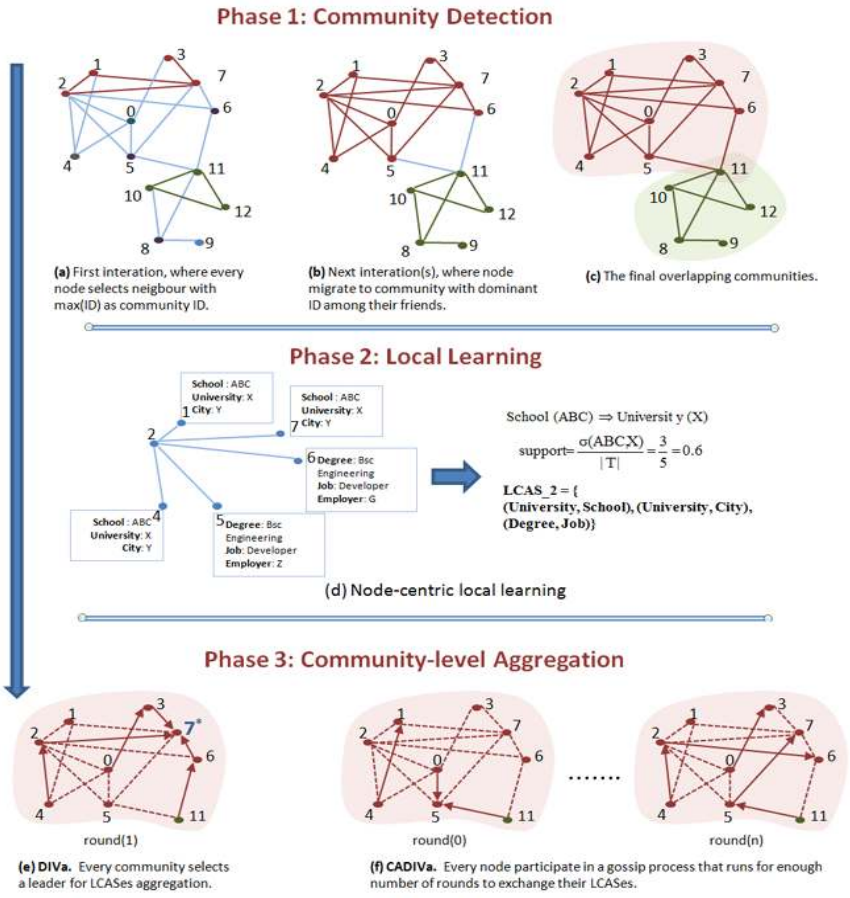

Fig. 2. The three phases of our identity validation scheme. First, communities boundaries are identified by detecting densely connected regions, then every node performs local learning, and the final step is aggregate rules per community using gossip-learning.

In what follows, we describe in more details each of the three phases of the proposed model.

2.1. Community Detection Phase

In social networks, it is intuitively observable that people have multiple community memberships. For example, a person usually has connections that span multiple groups like family, friends and classmates, co-workers etc. Furthermore, the number of communities a user can belong to is unlimited as a person can simultaneously associate with as many groups as he/she wishes. Thus, it is more reasonable to cluster users into overlapping communities rather than disjoint ones. Therefore, in our community detection phase we allow nodes to join multiple communities. In particular, the community detection algorithm that we exploit (i.e., the proposal in [Rahimian et al., 2014]) run in conductive rounds, such that in every round nodes maintain an ordered list of the communities according to the number of the direct friends belonging to these communities. Basically, as illustrated on Figure 2, a node starts a community by itself and initializes its community identifier with the maximum ID among itself and its

direct friends. Afterwards, nodes exchange their sates with their direct friends, such that every node updates the community membership list with the number of of the community that the majority of its friends are located in. Then, every node allocates itself to the community containing the highest number of its neighbours (i.e, the dominant community as the case of node 8), if such a community exists; otherwise, the node assigns itself to the community with the maximum ID.

Once every node is aware of the communities it belongs to, the second phase of the model is run locally by each node.

2.2. Local Learning Phase

Our proposed scheme extends the ensemble learning paradigm in distributed machine learning and works on fully distributed datasets without collecting the data into one central location. As shown in Figure 2(d), every node uses its local data repository that contains the collection of its direct friends profiles, and generates a set of distributed models by exploiting principles of Association Rule Mining (ARM). The formal statement of ARM was firstly stated by Agrawal [Agrawal et al., 1993] to extract the association rules of the causal dependencies of buying different items. In our model, the items are the profile attributes and the association rules are the correlated attribute sets. For example, a node can learn from examining the profiles of its direct friends that users who study at university x also live in city Y. If a node sees such an observation is frequent enough in the profiles of its direct friends, the node deduces that attributes University and City are correlated. We provide the details of this step under Section 3.2.

Once all the nodes have learned their local correlated attribute sets, they engage in a collaborative process to form consensus on the community level correlated attribute sets.

2.3. Forming Community Consensus Phase

The last phase is to agree on the communities CASes. Communities CASes are computed in an incremental fashion by aggregating the individual LCASes at the level of all the nodes in every given community. This achieves the same predictive and analytic power, as applying the learning on a centralized repository, in a distributed fashion without moving individual data outside its ownership boundaries and thus, without violating users privacy. In our previous work [Soliman et al., 2015], the node with the maximum ID (also referred to as diva node) inside every community is responsible for receiving the LCASes form all other nodes in the community and performing weighted voting mechanism in order to reach the final CAS for the community. For example, as illustrated in Figure 2(e), node 7 is the community's diva node and in round (0) of the aggregation process this node receives LCASes generated from its direct friends. In the following round(s), the node continues to receive LCASes from other nodes that are not directly connected to it (i.e., nodes 0, 4, and 11) . Particularly, the arrows in Figure 2(e) show the paths that nodes 0, 4, and 11 use to send their LCASes to their diva node. Afterwords, diva node generates the community CAS by performing the voting mechanism, then propagates the CAS to all of the nodes belonging to the community.

Obviously, depending on diva nodes for LCASes aggregation reduces the reliability of the system and makes it vulnerable to a single point failure. Therefore, in this paper, we propose two different implementations in CADIVa to perform LCASes aggregation in a fully decentralized manner without relying on any super nodes. More precisely, in the first implementation, we allow all of the nodes in every individual community to cooperate in an aggregation process using gossip-based algorithm. In particular, every node keeps a cache repository to store some nodes inside its community that are not direct neighbors to it, and stores hop-to-hop route to reach them. Afterwards, during a

gossip exchange process that is executed in successive rounds, nodes select a random node from their caches, and exchange a subset of their random peers stored in their caches. After a sufficient number of rounds, nodes inside every community will end up having a uniformly random sample of its community members. Then, the nodes can exchange their LCASes with the random sample nodes from their communities and locally they can merge these LCASes to reach the final community CAS. Thus, every node can select the top-k attributes to be the community CAS. As illustrated in Figure 2(f), in the first round node 0 performs LCAS exchange with node 5, whereas in the last round node 11 is not a direct friend of node 7, thus the exchange message is sent via the intermediate node 5. We detail this implementation in Section 3.3.1.

Regarding the second implementation, the local learning is performed first, then the community detection and LCASes aggregation are combined together into one phase to minimize the communication overhead. In particular, nodes start by extracting their LCASes and while they are exchanging their community IDs, they add a random sample of their direct neighbors. Upon receiving such messages, nodes store into their caches the received random nodes information if they belong to the same communities. Afterwards, after the community detection algorithm converges, nodes forward their LCASes to the random nodes stored in their caches. We present this second implementation under Section 3.3.2.

3. CADIVA: UNLEASHING THE COOPERATIVE WORK

In this section, we present the core of CADIVa. First, we detail the local learning algorithm, followed by the overlapping community detection and gossip-based algorithm for LCASes aggregation. Afterwards, we present a second implementation of CADIVa that iteratively applies a combined process of community detection with LCASes aggregation. Before we present the algorithms, we proceed with a few definitions.

3.1. Notations and Definitions

We consider the social network as an undirected graph G = (V, E), where V is the set of nodes and E is the set of edges. $e_{ij} \in E$ denotes a relationship between nodes v_i and $v_j \in V$. We denote with S = $\{A_1, A_2, .., A_m\}$, the profile schema adopted in the social network. Given a node $v_i \in V$, p_i denotes the set of its profile values: $p_i = \{p_i.a_1, p_i.a_2, ..p_i.a_m\}$, where $p_i.a_k$ is the value provided by v_i for $A_k \in S$.

We denote by *Local Profile Collection (LPC)*, the set of profiles of a node's friends. That is, given $v_i \in V$, and $DF_i = \{v_j \in V | e_{ij} \in E\}$ representing the set of v_i's direct friends, $LPC_i = \{p_k | v_k \in DF_i\}$ denotes the collection of their profiles and is referred to as v_i's local profile collection.

Given LPC_i, the *Local Frequent Attributes* LFA_i is the set of attributes for which the values are highly repetitive in LPC_i. Formally we define:

DEFINITION 3.1. ***Local Frequent Attributes***. *Let $v_i \in V$ and LPC_i be its local profile collection. Let $A_k \in S$ be an attribute from the profile schema and let $P_k^{\vartheta} \subseteq LPC_i$ be the set of profiles in LPC_i having the same value for attribute A_k. That is, P_k^{ϑ} = $\{p_m \in P_k^{\vartheta} \ | p_m.a_k = \vartheta$, where ϑ is a given value$\}$. Let $LFA_i \subseteq S$ be the set of attributes such that, LFA_i = $\{A_k \in LFA_i | \ \frac{|P_k^{\vartheta}|}{|LPC_i|} \geq \epsilon\}$, where ϵ is the average frequency of the repetitive attribute values in LPC_i.*

For a given pair of attributes from LFA_i, its support is defined as:

DEFINITION 3.2. ***Support of an attributes pair***. *Let $v_i \in V$ be a node in the OSN. Let LPC_i be its local profile collection and let LFA_i be its local frequent attributes set. Let BA = (A_j, A_h) be a pair of attributes from LFA_i. The support of BA defines the*

percentage of co-occurrence of the same paired values for the two attributes A_j and A_h to the total number of values in LPC_i:

$$Support(BA) = \frac{\textit{values-co-occurrence } (A_j,\ A_h)}{\textit{all-values } (A_j,\ A_h,\ LPC_i)} \tag{1}$$

Where,

values-co-occurrence $(A_j,\ A_h)$=$|\{(p_e, p_m) \in LPC_i | p_e.a_j = p_m.a_j \wedge p_e.a_h = p_m.a_h\}|$.

and,

all-values $(A_j,\ A_h,\ LPC_i)$=$|\{\vartheta\ | \exists p \in LPC_i$ *s.t.,* $p.a_j = \vartheta \vee p.a_h = \vartheta\}|$

Based on Definition 3.2, we define a Local Correlated Attribute Set a follows:

DEFINITION 3.3. ***Local Correlated Attribute Set - LCAS****. Let $v_i \in V$. Let $LFA_i \subseteq S$ be its local frequent attributes set. Let BA = (A_j, A_h) be a pair of attributes from LFA_i. BA is a local correlated attribute set, denoted as LCAS, if: Support(BA) $\geq$ β, where β is the average support value of attribute pairs over LFA_i.*

3.2. Local Learning

As aforementioned, we implement a node-centric ARM algorithm to extract the associations which reflect the causal structures among different profile attributes. However, investigating the causality among all possible attribute pairs could be computationally expensive. Therefore, nodes start by finding the candidate attributes for the ARM according to Definition 3.1. Specifically, nodes consider the set of profiles of their direct friends (i.e., LPC) and select those attributes with high frequency of repetitiveness in their values. For example, a job value *teacher* that is repeated in more than 25% of the profiles in LPC makes the attribute *Job* a frequent one; whereas, one satisfying less than this threshold is not a frequent attribute. Moreover, the threshold value is determined locally by every node, such that it is equal to the average frequency of the repetitive attribute values in LPC. So as, having different threshold values increases the degree of freedom and flexibility to reflect the repetitive patterns at every node independently in the social graph.

Example 3.1. Assume Alice as another OSN user. To learn his LCAS, Alice collects the available profile attributes from all her direct friends to construct LPC_{Alice}. Assume he finds that *Education* and *Interests* are in LFA_{Alice}; that is, their values are highly frequent in LPC_{Alice}. Alice computes the number of profile pairs in LPC_{Luka} for which these two attributes' pair have similar values. Assume in more than 40% of the profiles in LPC_{Alice}, this pair is co-occurring. Also, he calculates the average support value and finds it to be 27%, thus the LCAS threshold β = 0.27. Thus, the pair *(Education, Interests)* is an LCAS for Alice.

Given the LFA_i, a node v_i investigates all the possible attribute pairs and computes the support of each attributes pair using formula mentioned in Equation 1. Once the support is calculated for all pairs from its LFA_i, node v_i selects the ones for which the support is high enough according to the Definition 3.3 to represent its local correlations (i.e., LCAS). Similarly, we define the minimum value of the support of an attribute pair to be selected in $LCAS$ to be greater than the average support of all extracted attribute pairs. Algorithm 1 shows the pseudocode of the steps executed by all nodes to extract their LCASes. Nodes start by initializing their LCASes with all possible attribute pairs from LFA with support value equals to 0. Afterwords, nodes iterate over their LPCs to estimate the support of attribute pairs according to the existing repeated values. Accordingly, nodes get the sets of overlapping words in different

profiles for an attribute pair (A_1, A_2) values using *tokenize()* method that retrieves all the words associated with an attribute. Then, the support of an attribute pair is increased by the normalized support value computed by dividing the size of the smallest overlapping between word lists over the total number of words in the two attributes. After computing the support for the existing attribute pairs, nodes calculate the average support value and assign it to the threshold for selecting their LCASes. Therefore, any attribute pair with support lower than the threshold is removed from the LCAS.

ALGORITHM 1: LCAS Learning at node v_i

Data: Require LFA_i and LPC_i
Result: Ensure list of correlated attribute pairs with their local support: $LCAS_i$
forall the $(A_1, A_2) \in LFA_i$ **do**
  insert($LCAS_i$,(A_1, A_2), 0)
  $W_1 =$ tokenize(LPC_i, A_1)
  $W_2 =$ tokenize(LPC_i, A_2)
  forall the $p_k, p_j \in LPC_i$ **do**
    $X =$ tokenize($p_k.a_1$) $\cap$ tokenize($p_j.a_1$)
    $Y =$ tokenize($p_k.a_2$) $\cap$ tokenize($p_j.a_2$)
    if $X \neq \emptyset$ *and* $Y \neq \emptyset$ **then**
      $s = \frac{min(|X|,|Y|)}{|W_1|+|W_2|}$
      $LCAS_i[get\_index(A_1, A_2)].s$ += s
    end
  end
end
$\beta =$ average_support(LPC_i)
forall the $(A_1, A_2) \in LCAS_i$ **do**
  if $LCAS_i[getIndex(A_1, A_2)].c < \beta$ **then**
    remove($LCAS_i$, (A_1, A_2))
  end
end

3.3. Forming Community CAS

Our model extends the ensemble learning paradigm such that nodes generate their LCASes by accessing only their local data, so as user's privacy is maintained. Therefore, the next step is to build up the final CAS by aggregating the locally generated LCASes. We perform community-aware aggregation of the locally generated LCASes to reflect the underlying topological structure of the social network. In particular, in our model topological communities are identified and that all the nodes belonging to a community exchange their LCASes. Commonly, finding communities is well-know as community detection and is defined as:

DEFINITION 3.4. ***Community Detection***. *A community detection Φ, also known as graph clustering, is a mapping*

$$\Phi : G \rightarrow G'_1 \times ... \times G'_c \tag{2}$$

that partitions G into c non-empty, node-disjoint subgraphs $G'_1 \times ... \times G'_c$ representing a set of communities or clusters. A widely used quality measure for community detection is the modularity Q of the clustering Φ(G) [Newman, 2006], which is a mapping

$$Q : \Phi(G) \rightarrow \mathbb{R} \tag{3}$$

that assigns a quality value q ∈ [-0.5,1] to the clustering Φ(G), as defined by

$$q := \sum_i (e_{ii} - b_i^2) \tag{4}$$

Where $b_i = \sum_j e_{ij}$*, and* e_{ij} *is the fraction of edges in community i for which the target node of the edge lies in community j. The higher the quality value q is, the better the detected community is. One possible definition for* Φ *is to maximize Q over all clustering Φ(G) [Newman, 2006], which was shown to be an NP-hard problem [Brandes et al., 2008].*

Majority of research in community detection focuses on partitioning social networks into disjoint communities. However, in social networks, every person typically belongs to more than one community, such as the community of family members, that of friends and classmates, that of co-workers, etc. Therefore, for high quality results it is initiative that we perform overlapping community detection. Thus, in our model nodes are allowed to have multiple community memberships. As aforementioned, we provide two different implementations of CADIVa. In the first one, the community detection phase is executed first and performed separately from the aggregation phase, whereas in the second implementation we combine both the community detection and LCASes aggregation to minimize the overall communication overhead.

3.3.1. Separate Phases: Community Detection followed by LCAS Aggregation.

The first implementation of CADIVa follows the modular design where each phase is an independent module that contains everything necessary to execute the desired functionality. So as, the community detection module is separate from the community aggregation one.

Overlapping Community Detection. For DOSNs, a compliant solution for community detection should follow the decentralization requirement by which every node can only be aware of and contact its direct neighbors. Therefore, CADIVa employs recently developed decentralized diffusion-based community detection strategy [Rahimian et al., 2014]. In particular, every node starts by joining the node with the maximum identifier among its direct friends to form a community. Afterwards, in successive iterations every node chooses to quit its current community and join one of its neighbour's if this brings some modularity gains. For example, as illustrated in Figure 2(a), in the first iteration nodes 1, 2 and 3 join the community of node 7 that has the largest identifier among them. Then, nodes inform their direct neighbors with their current status by sending a message that contains the community they belong to. Later on, nodes 0, 4 and 5 reevaluate their states and join the community of node 7 as it becomes the dominant identifier among their direct friends (see Figure 2(b)).

Moreover, every node calculates the modularity gain locally by finding the dominant community identifier among its direct friends. If a node does not find a dominant identifier among its neighbours, it changes to the highest ID between its own and the ones of the communities of its neighbours. This step is iteratively repeated until no node wants to change its community identifier as it already represents the dominant one of all its neighbors. To allow nodes to join multiple communities, every node keeps a membership lists to order the top dominant communities identifiers in the surrounding neighbors. When the community detection algorithm converges, every node in the network becomes aware of the communities to which it belongs.

Gossip for aggregating LCASes. In our algorithm, we apply gossip-based peer sampling where peers periodically exchange small random subsets of the identifiers of their direct friends and paths to reach them. Thus, after sufficient number rounds, nodes

ALGORITHM 2: Gossip Exchange Scheme at node v_i

Data: Require community identifier C_i, number entries n for exchange
Result: Ensure a random subset of community members: CRM_i and community CAS
$CAS \leftarrow LCAS$
Procedure `GossipSampling()`
  Loop
    wait($\triangle$)
    $RM \leftarrow$ select_random_member(C_i)
    $RCE =$ select_random_cache_entries(C_i, n)
    send(RM, RCE)
  EndLoop
Procedure `OnReceivedCRM(`*message m*`)`
  forall the $e \in m.RCE$ **do**
    if *new_entry(e)* **then**
      add_new_entries (CRM_i, e)
    end
  end
Procedure *ExchangeLCASes*
  forall the $v_j \in CRM_i$ **do**
    sendLCAS ($v_j, LCAS_i$)
  end
Procedure `OnReceivedLCAS(`*message m*`)`
  forall the $lcas \in m.LCAS$ **do**
    if *new_entry(lcas)* **then**
      add(CAS, $(lcas.A_1, lcas.A_2)$, $\frac{lcas.s}{2}$)
    else
      $new\_s = average(CAS[get\_index(lcas.A_1, lcas.A_2)].s, lcas.s)$
      $CAS[get\_index(lcas.A_1, lcas.A_2)].s = new\_s$
    end
  end

are going to have a local random sample of the nodes belonging to their communities and the routing paths towards them. The advantage of gossip-based sampling in our setting is that samples are available locally and without delay. Furthermore, the messages related to the peer sampling algorithm can piggyback the locally generated LCASes, thereby avoiding any overheads in terms of communication overhead. More formally, each node maintains a fixed-sized cache of c entries (with typical value 20 or 50 entries). A cache entry contains identifier and routing path of another node in the community. Each node v_i repeatedly initiates a neighbor exchange operation, by executing Algorithm 2. As shown, the algorithms consists of 4 procedures. The first procedure *GossipSampling* is the one responsible for constructing a random sample of the communities members for each node. Initially, every node maintains a local repository named CRM to refer to community random members, and stores the identifiers of those random nodes and paths to reach them. Periodically, every node randomly selects a partner from its CRM for the gossip exchange and selects a random subset entries from its CRM to be send in the gossip message. On receiving a reply form the contacted node during the gossip exchange, the receiving node updates its CRM by adding the entries of the new nodes that are not included in CRM as described in procedure *OnReceivedCRM*. Thereafter, by executing the gossip exchange for sufficient number of rounds, nodes start to execute the procedure *ExchangeLCASes* by sending their LCASes to all the nodes in their CRM. Then, on receiving LCASes from other nodes, every node starts to update its repository that represents the community CAS

by averaging the received LCASes from other nodes as described in procedure *OnReceivedLCAS*.

3.3.2. One Phase: Combining Community Detection and LCAS Aggregation.

It is important to be noticed that there is high similarity between the communication process that is executed during the community detection phase and random peer sampling during the aggregation phase. Therefore, in our second implementation of CADIVa we combined the two phased into one part to eliminate the communication overhead of the entire phase of LCASes aggregation. So, the nodes executes only two phases starting with LCAS learning and then start the phase of community detection and LCAS aggregation. Basically, in the second phase the messages to be exchanged for community detection additionally contain a random sample of nodes direct neighbors. Thus, nodes execute the same steps to identify the communities they belong to, in the mean while they are going to update their CRMs to add random member that are belonging to the same communities. Consequentially, there is no need to execute the procedure *GossipSampling* in Algorithm 2 and nodes proceed with executing the other procedures after the community detection algorithm converges.

4. SECURITY, PRIVACY, AND COMPLEXITY ANALYSES

In this section, we study the security and the privacy properties of CADIVa and we provide its complexity analysis.

4.1. CADiVa Security Properties

We consider a malicious adversary model whereby an attacker would try to subvert the correct functioning of the system's processes. Given that the outcome of our system is collections of CASes that reflect identity trends of communities and that would be used to validate the identities of new members to the OSN, the most prominent interest of an attacker would be to corrupt the CASes learning to reflect identity patterns of malicious nodes and not of honest ones. This can be achieved in one of two ways. First, a malicious attack could target invalidating a valid CAS in a community. Second, it can work on introducing another fake CAS that would match the malicious behavior. In both cases, the goal of the attacker would be to change the CASes in a target community to confirm the identity trustworthiness of the malicious nodes or to invalidate the honest nodes. As discussed in [Soliman et al., 2015], this could only be achieved by infecting a target community, introducing into it a number of malicious nodes that is high enough to reflect corrupted CASes.

In [Soliman et al., 2015], we have provided a quantitative analysis of the effort required, in terms of number of required malicious nodes, to maliciously introduce a fake CAS to a community or to corrupt a valid one in it. As detailed in [Soliman et al., 2015], a community CASes could be corrupted if a malicious attack succeeds at introducing enough fake nodes (i.e., sybil nodes [Yu et al., 2006b]) into it; that is, befriending enough honest nodes in the community to become member of it. As CADIVa adopts the same strategy as DIVa in learning the CASes in a community, the same security properties presented in [Soliman et al., 2015] apply to CADIVa as well. In fact, CADIVa adopts the same technique of learning a CAS based on the co-occurrence frequency of similar attribute values within a community as in [Soliman et al., 2015]. The difference in CADIVa is w.r.t the process by which the aggregation of values is performed. This does not affect the security properties of the system as an attacker would need, in both DIVa and CADIVa, to introduce the same number of malicious nodes to introduce corrupt CASes to a community. We recall these properties as follows.

4.1.1. Introducing a Fake CAS. Introducing a new CAS, CAS_{new}, to a community requires that the community holds enough nodes within its boundaries that exhibit profile values confirming CAS_{new}. For a node x to be considered within the boundaries of a community C, it has to befriend enough other members of C. The effort required to befriend the needed number of an honest community's members, to become one of them, cannot be defined quantitatively. However, it is expected to be high given that CADIVa is deployed and offered as a service for users to evaluate their new friendship requests before accepting or denying them. Moreover, the structure of communities is not known to users; hence it may not be straightforward to predict which nodes needs to be befriended to become member of a community when this information is not available.

Regardless of what it requires for a stranger to befriend honest nodes in a community, we deterministically define z, the number of fake nodes required to become member of a target community C, to be able to introduce a fake CAS_{new} that reflects the identity trends of these z fake nodes.

THEOREM 4.1. *[Soliman et al., 2015] Let $\overline{C} \subset G$, $\overline{C} = (\overline{C}.V, \overline{C}.E)$, be a community of size n ($|\overline{C}.V| = n$). Let sup_{lowest} be the lowest support by which a CAS is accepted in $\overline{C}$. For a new CAS, CAS_{new} to appear in $\overline{C}$, it must be inserted a group of fake nodes C_f that successfully join $\overline{C}$ and that show profile information confirming CAS_{new} such that:*

$$z = |C_f| \geq \frac{sup_{lowest}}{(1-sup_{lowest})} * n.$$

PROOF. 1: Consider C_f of size z ($|C_f| = z$) is carrying a correlation $CAS_f = \{A, B\}$, that is unknown to the nodes in $\overline{C}$. Assume all C_f successfully joins $\overline{C}$. Therefore $\overline{C}.V = \overline{C}.V \cup C_f$ and $|\overline{C}.V| = n + z$. That is, the aggregate support of CAS_f in $\overline{C}$ would be: $support(CAS_f) = \frac{values-co-occurrence(A,B)}{n+z}$. Since all nodes in C_f carry the correlation in CAS_f that is unknown to $\overline{C}$ initial n nodes, the support for CAS_f will be: $support(CAS_f) = \frac{z}{n+z}$. According to the proposed method, for CAS_f to be recognized as a CAS in $\overline{C}$ ($support(CAS_f) \geq sup_{lowest}$), this inequality shall hold: $z \geq \frac{sup_{lowest}}{(1-sup_{lowest})} * n$. □

4.1.2. Corrupting a Valid CAS. For an adversary to corrupt a valid CAS in a community, technically by lowering its support to fall below the required threshold, it needs to introduce to the target community a number of new profiles that are not compliant with this CAS. This number has to be big enough to lower the support of the valid CAS below the adopted threshold.

THEOREM 4.2. *[Soliman et al., 2015] Let $\overline{C} \subset G$, $\overline{C} = (\overline{C}.V, \overline{C}.E)$, be a community of size n ($|\overline{C}.V| = n$). Let sup_{lowest} be the lowest support by which a CAS is accepted in $\overline{C}$. For a valid CAS, CAS_{valid} with support S_v, to disappear from $\overline{C}$, it must be inserted in $\overline{C}$ a group of fake nodes, C_f, that does not have profile information confirming CAS_{valid} such that:*

$$z = |C_f| > \frac{S_v * n}{sup_{lowest}} - n.$$

PROOF. 2: Let $CAS_v = \{A, B\}$ be a valid CAS in $\overline{C}$ with aggregate support S_v: $S_v = \frac{m}{n} \geq sup_{lowest}$, where $m = values - co - occurrence(A, B)$. Let C_f of size z ($|C_f| = z$) be not carrying the correlation between attributes A and B. Assume all the nodes in C_f successfully join $\overline{C}$. Therefore $\overline{C}.V = \overline{C}.V \cup C_f$ and $|\overline{C}.V| = n + z$. That is, the aggregate

support of CAS_v in $\overline{C}$ becomes: $S_{v1} = \frac{values-co-occurrence(A,B)}{n+z}$. Since all nodes in C_f do not carry the correlation in CAS_v, $values - co - occurrence(A, B)$ is still equal to m; therefore $S_{v1} = \frac{m}{n+z}$. For CAS_v to no more be a valid CAS, its new support shall be: $S_{v1} < sup_{lowest}$. That is, $\frac{m}{n+z} < sup_{lowest}$. From where $m < sup_{lowest} * (n + z)$. Dividing the inequality by n ($n \in \mathbb{N}^+ and n > 0$), we get: $\frac{m}{n} < \frac{n*sup_{lowest}+z*sup_{lowest}}{n}$. Therefore, dividing the inequality by the positive number sup_{lowest} gives, $\frac{S_v}{sup_{lowest}} < \frac{n+z}{n}$. From that, $z > \frac{S_v*n}{sup_{lowest}} - n$ □

4.1.3. Cloning attacks. In addition to the above detailed attack approaches, that a malicious node can adopt to compromise the CADIVa system, another possible attack is to design fake profiles that would exhibit the correlations and identity trends expected by CADIVa in order to infiltrate within honest communities. This approach may be seen intuitive and inescapable. However, to be able to achieve such an attack, the adversary needs first to be aware of the target OSN graph, of the community structures, and of the CADIVa defined CASes in detected communities. Moreover, the adversary needs to have knowledge on common values for these CASes in the communities target of the potential attack. However, CADIVa is designed for DOSNs where information about the network graph, its structure, and its properties is inherently protected by the nature of the system's design. Therefore, such an attack is mitigated by the underlying design of the system; i.e., the decentralize nature of CADIVa.

There could still be one feasible option for such an attack to succeed. This is related to deploying cloning techniques. Cloning is a known attack in OSNs where an adversary creates a fake account by mimicking the values in a real one [Jin et al., 2011]. The clone account, though fake, would appear as honest as the profile it clones. We admit that CADIVa may be blind to clone profiles; however, we put the accent on the fact that CADIVa employs both public and private profile attributes in the CASes it extracts. It is thus crucial for the clone profile to correctly clone both public and private profile values of the honest profile it copies. As access to private profile values is only possible by befriending the honest node, we consider that clone profiles may not qualify as perfect clones under the validation rules of CADIVa.

4.2. CADIVa Privacy Property

CADIVa guarantees the aggregation process of the LCASes to form consensus on a community's CAS in a fully decentralized manner. This happens by nodes exchanging the LCASes learned at their local level with other nodes in their community. This exchange does not subvert the privacy of the nodes involved in the process as the information communicated between foreign nodes consists of groups of attributes only. As such, an adversary node can only learn that some group of attributes is correlated in a community without being able to learn any specific information about individual nodes. However, we note that one of the strategies of CADIVa is to also form consensus on the top n values related to a given CAS in a community. This might sound to result in revealing private information about nodes; however, this set of top n values for a CAS in a community reflects a statistical representation of the data available in that community as a whole. This is relating to the concept of differential privacy that, in the field of data anonymization, suggests the generation of anonymized data sets based on statistical disturbances to the data [Li et al., 2011]. That is, some statistical perturbation is added to the result of a given query on the data such that, there is a deterministic probability that one data record in the dataset is identified regardless of whether or not it participated in the anonymized dataset [Li et al., 2011]. This expresses a privacy guarantee on every data item as an equal probability to be identified

whether or not it belongs to the anonymized dataset. In our system, every node collects the profile information of all its direct friends. The node aggregates this data to generate its set of LCASes and their corresponding top n values. These LCASes and the corresponding top n values are the result, obtained from the original dataset of all profile information of the node's friends, that is shared by the node with other nodes in the network. This shared result could be viewed as an anonymized data generated from the original dataset containing all the information of the node's direct friends. Following this approach, we model the privacy preservation guarantee of our system.

Assuming the malicious adversary model, a malicious node a would be interested in learning as much information as possible from the system's processes. More precisely, a malicious node a would participate in the system's process to reach consensus on community level CASes for the communities it belongs to, with the intention of collecting data and revealing from it private information about other nodes. We prove that a cannot identify any other node in its community that is not its direct friend, and that it cannot reveal any private information related to them with a deterministic probability of non-disclosure. We formulate this privacy preservation guarantee as follows:

THEOREM 4.3. *Let $C \subset G$, $C = (C.V, C.E)$, be a community in the OSN and let $m \in C.V$ be a member node of it. Let CAS_c be a CAS in C and let d be the number of top values shared for CAS_c in C. Let δ be the support achieved by CAS_c. Assume that node m has private values for CAS_c (i.e., $CAS_c\_Values(m)$ is private information). Let $\mathcal{B}$ be the event that $CAS_c\_Values(m)$ is revealed based on CAS_c and the corresponding d top values. CADIVa guarantees that the probability that $\mathcal{B}$ is true (i.e., $P(\mathcal{B})$) is less than a privacy guarantee threshold expressed as:*

$$P(\mathcal{B}) \leq \frac{\delta}{d}.$$

PROOF. 3: Let C be a community in the OSN and let CAS_c be a CAS in C. Let δ be the support achieved by CAS_c. This means that a percentage of at least δ nodes from C have the same pattern suggested by CAS_c. That is, the probability that a node $m \in C$ exhibits CAS_c is: $P(m \text{ shows } CAS_c) \leq \delta$. Let now d be the number of top values provided for CAS_c in an ordered uniform manner. That is, the top d values are not given in order of importance and the probability to hold any of these d values is equal. Assume the worst case wherein d reflects the number of all available values for CAS_c in the population of C that exhibit CAS_c. This means that the probability that the value of m for CAS_c (i.e., $CAS_c\_Values(m)$) is one of the d provided values is: $\frac{1}{d}$. Assume now that $CAS_c\_Values(m)$ are private. The probability that $CAS_c\_Values(m)$ is revealed based on the d provided values and on CAS_c requires that m exhibits CAS_c and that it holds one of the d values. Therefore, $P(CAS_c_Values(m))$is revealed based on CAS_c and $d) \leq P(m \text{ shows } CAS_c) \times \frac{1}{d} \leq \delta \times \frac{1}{d}$. □

By Theorem 4.3, the probability to reveal private information related to a CAS of a node in a community is relative to the support achieved by the CAS in question. In fact, the higher the support of a CAS in a community is, the more nodes in the community exhibit it. Therefore, the higher the probability to identify a node in the community as exhibiting the CAS in question. However, this would only identify the target node as exhibiting the CAS in question without revealing any information about what the exact values it holds for it are. This information can be revealed from the top n values provided only and can be determined based on the number of these top n values. Therefore, we suggest sharing small numbers of top n values only.

4.3. Complexity Analysis

The model's cost is expected to be low given that every node performs its local computation independently of the other nodes. Besides, the bottleneck that the DIVa nodes could have constituted, as suggested in [Soliman et al., 2015], is overcome by the completely decentralized model exploited by CADIVa. We discuss the complexity of CADIVa in what follows.

First, every node computes its LCAS. The complexity of this is a function of the number of node's friends (i.e., its degree d) and of the number of profile attributes in the profile schema. Indeed, the LCAS learning requires computing for every pair of attributes (a profile schema of m attributes results in $p = \binom{m}{2} = \frac{m^2-m}{2}$ number of pairs), its value-co-occurrence among all the node's direct friends. Therefore, the number of performed checks per attributes pair is, $c = \binom{d}{2} = \frac{d^2-d}{2}$. Accordingly, the LCAS learning's complexity is $\mathcal{O}(c * p)$. By this, the nodes with higher degree would be the bottlenecks in the LCAS learning step; however, this step is node dependent and does not require the simultaneous online availability of all the nodes.

In addition to that, the community detection and gossip exchange of LCASes costs in terms of communication traffic between all the nodes in the OSN. By our adopted work for decentralized community detection, the algorithm's complexity is a $\mathcal{O}(N * D * R)$, where N is the total number of nodes in the OSN graph, D is the average node degree, and R is the total number of rounds needed for the algorithm to converge[6] [Rahimian et al., 2014]. This step requires that all the nodes are online at the time of its execution; however, it is also a process that is performed once and that is incrementally updated only. Moreover, as we demonstrate through experiments on real OSN data, the convergence time of our solution is very realistic and achievable (see Section 5.3).

5. EXPERIMENTS AND RESULTS

In this work our objective is to provide unsupervised and fully decentralized identity validation model using only profile information without violating any privacy constrains. As aforementioned and to the best of our knowledge, Bahri el al. [Bahri et al., 2014] is the solely existing work that addresses online identity validation by exploiting profile information to generate a trustworthiness probabilistic measure for new profiles instead of classifying them as real or fake. However, Bahri el al. [Bahri et al., 2014] neglect the underlying social graph connecting users and process all the profile collection at once. Therefore, in this section we compare CADIVa, with our previously developed semi-centralized model DIVa, and the global approach that processed all profiles at once similar to Bahri el al. [Bahri et al., 2014].

Particularly, we evaluate the other approaches with the different implementations of CADIVa in terms of the ability of providing fine grained community-aware identity validation on real-world datasets. The results for the global approach are obtained by collecting all profiles are at one central repository, then executing the same steps of CAS extraction process by considering the whole profile collection. As aforementioned, DIVa operates in three phases such that nodes start by executing a decentralized community detection algorithm, then generate their LCASes and finally communicate with the diva nodes to decide the final communities CASes. In DIVa, the frequency and support threshold values (Definition 3.1 and Definition 3.3) are equal to 0.2.

Besides, we evaluate the communication overhead of CADIVa. We have implemented two different versions of CADIVa using GraphLab [Low et al., 2012]. The First implementation, CADIVa_S, has two different distributed execution modules, such that the first module executes our adopted community detection algorithm until it converges so

[6] R depends on the topological properties of the underlying graph

Table I. Real OSN datasets used in experiments.

Dataset	Nodes	Edges
Facebook	23,332	28,972
GpJUL	2,417,014	25,016,154
GpAUG	4,349,414	35,544,682
GpSEP	4,388,907	43,060,890

Table II. CADIVa extracted CAS vs. DIVa and global CAS for the Facebook dataset.

CADIVa CAS for Community1		DIVa CAS for Community1		Global CAS	
Attribute Pair	**Support**	**Attribute Pair**	**Support**	**Attribute Pair**	**Support**
1:{education, employer}	0.582	1:{education, employer}	0.582	1:{job, interest}	0.335
2:{education, interest}	0.499	2:{education, interest}	0.499	2:{gender, interest}	0.179
3:{h.country, job}	0.113	3:{h.country, job}	0.113	3:{education, interest}	0.138
4:{f.name, h.country}	0.0798			4:{job, h.country}	0.137
5:{gender, job}	0.0779			5:{gender, h.country}	0.126
6:{f.name, gender}	0.0754			6:{education, job}	0.1
7:{job, employer}	0.0436				

Table III. CAS extraction results for the Google+ datasets.

CADIVa Generated CAS

Google+ July		Google+ August		Google+ September	
Attribute Pair	**Support**	**Attribute Pair**	**Support**	**Attribute Pair**	**Support**
1:{employer, places}	0.148	1:{employer, places}	0.083	1:{employer, places}	0.161
2:{major, employer}	0.122	2:{major, employer}	0.103	2:{major, employer}	0.141
3:{school, employer}	0.273	3:{school, employer}	0.134	3:{major, places}	0.03
4:{school, major}	0.108	4:{school, major}	0.135	4:{school, employer}	0.133
5:{school, places}	0.033	5:{school, places}	0.06	5:{school, major}	0.09

Globally Generated CAS

Google+ July		Google+ August		Google+ September	
Attribute Pair	**Support**	**Attribute Pair**	**Support**	**Attribute Pair**	**Support**
1:{major, employer}	0.153	1:{major, employer}	0.135	1:{major, employer}	0.356
2:{major, places}	0.149	2:{major, places }	0.272	2:{major, places}	0.293
3:{school, major}	0.326	3:{school, major}	0.313	3:{school, major}	0.379
4:{school, places}	0.315	4:{school, places }	0.292	4:{school, places }	0.41

that every node knows the communities it belongs to. Thereafter, the control is moved to the second module that executes the gossip protocol for LCASes aggregation and extracts CASes for every detected community. The second version, CADIVa_C, is implemented as one distributed execution module where every node starts by extracting its LCAS, afterwords it starts engaging in the gossip protocol of detecting the communities and exchanging LCASes. Then after convergence, every node calculates its final communities CASes by averaging the collected LCASes in their caches.

We conducted several experiments to validate the effectiveness of CADIVa using real profile datasets from Facebook and Google+ (shown in Table I). We used the Facebook dataset collected and used in [Akcora et al., 2012], and the Google+ dataset publicly available from [Gong et al., 2011]. The profile schema in the Facebook dataset contains: First Name, Gender, Home County, Education, Job, Current Country, and Interests. Meanwhile, the profile in Google+ datasets has fewer attributes, specifically Occupation, Employment, Education, and Places Lived. The Google+ dataset represents three crawled parts of the OSN collected on July, August, and September in 2011.

5.1. Extracted CASes

Tables II and III list the extracted CASes for the Facebook and Google+ datasets, respectively. Tables show the extracted CASes and their equivalent support values for

different communities using CADIVa, DIVa, and the global approach. It is illustrated that with tuning the support threshold value, CADIVa allows communities to have more attribute pairs in their CASes compared to DIVa. For example, the CAS generated by CADIVa for one of the communities in the Facebook dataset, as show in Table II, contains 7 attribute pairs, whereas CAS extracted by DIVa contains only 3 attribute pairs.

Consequently, by having more attribute pairs inside CAS, CADIVa provides denser identity validation criteria compared to DIVa and the global approach as well. Particularly, using our validation scheme, the trustworthy index of any target profile is calculated by summing up the support values of the attribute pairs compatible with community's CAS in that target profile. Thus, users can make more confident decision regarding regarding the new friendship requests they are going to receive by having more validation rules.

Furthermore, CADIVa validation is not restrictive in terms of prohibiting new users from joining their targeted communities. The results show that these new attribute pairs extracted by CADIVa are not the dominant factors in communities CAS, such that the most important rules are the ones with the highest support values that are commonly extracted by DIVa with the higher threshold value. Therefore, CADIVa is not restricting any new community membership users are seeking to achieve.

5.2. Adaptive Threshold

CADIVa allows communities to have different threshold values derived from the need to tune the threshold according to size of the detected communities and the homophily level expressed in each community. To illustrate further these differences among communities and the need to have adaptive threshold values, Figure 3 shows the support values of different attribute pairs extracted at some communities in the Facebook and Google+ datasets. As a first observation, communities have different patterns not only in terms of extracting different attribute pairs in their CASes, additionally the associated support values have different patterns. The support values represent the statistical significance of attributes pairs, also they represent the homophily existing in each detected community with regard to these profile attributes. Therefore, having one global threshold value disregards reflecting the expressed homophily inside every individual community.

Additionally, a second observation is that the support value is decreasing in non-linear manner, more preciously it decreases sharply after the average value of all extracted pairs from node's LFA. Although, DIVa extracts the stronger pairs with the highest support, however, there are some other pairs that are statistically significant as well. For example, as shown in Figure 3(a), only 2 attribute pairs have support greater than 0.2. On the other hand, if the threshold value is going to be similar to the average support of all extracted pairs, these two communities are going to have 6 attribute pairs in their CASes instead of 2. The same scenario applies with the Google+ datasets as illustrated in Figure 3(b), (c) and (d).

Therefore, in CADIVa we specify a size-dependent lower bound to the threshold value. Thus, the LCAS learning considers a node's LFA (Definition 3.1) with values-frequency greater than the average values-frequency in node's LPC. Similarly, node's LCAS (Definition 3.3) is pruned by considering only the attribute pairs with support greater than the average support value of attribute pairs over node's LFA. Figure 4 depicts the average CAS sizes reported for the detected communities across all the datasets. As illustrated, CAS size varies with respect to community size. For example, in the Google+ datasets the CAS size slightly increases with respect to the increase in community size. On average, CADIVa extracts 4 attribute pairs for every community out of 6 (i.e, the maximum count that equals to $\binom{ac}{2}$, where ac is the number of

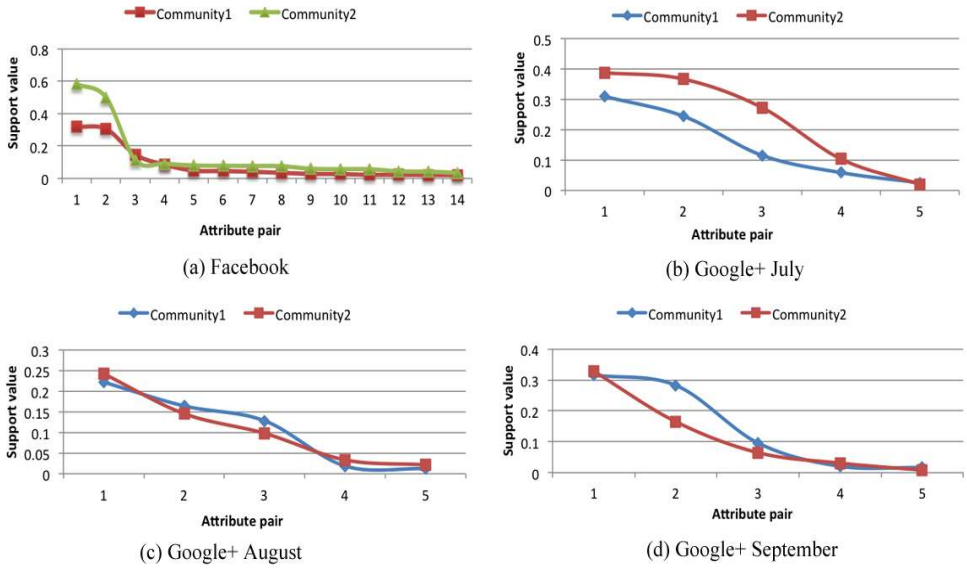

Fig. 3. The support values of different attribute pairs extracted at some communities in the Facebook and Google+ datasets.

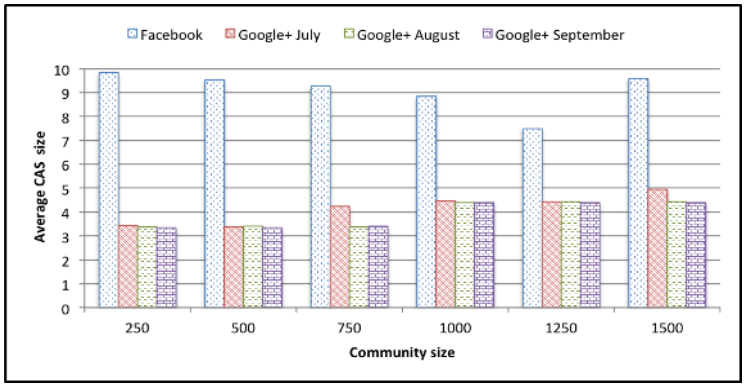

Fig. 4. The average size of communities CASes generated by CADIVa.

attributes in the profile scheme). For the Facebook dataset, communities have on average 9 attribute pairs in their CASes out of 21 possible attribute pairs.

Additionally, Figure 5 shows the average support value that is used for the threshold across all detected communities in the used datasets. Besides, it also depicts a comparison of the average total support of the CASes extracted by different approaches. As illustrated, the results reflect the strength of the validation criteria provided by different approaches. Figure 5 shows that CADIVa provides stronger validation than other validation approaches as the average total support in CADIVa is the highest across all detected communities in the datasets. In general, CADIVa achieves average improvement up to 36% and 50% across all the datasets than DIVa and global approach, respectively.

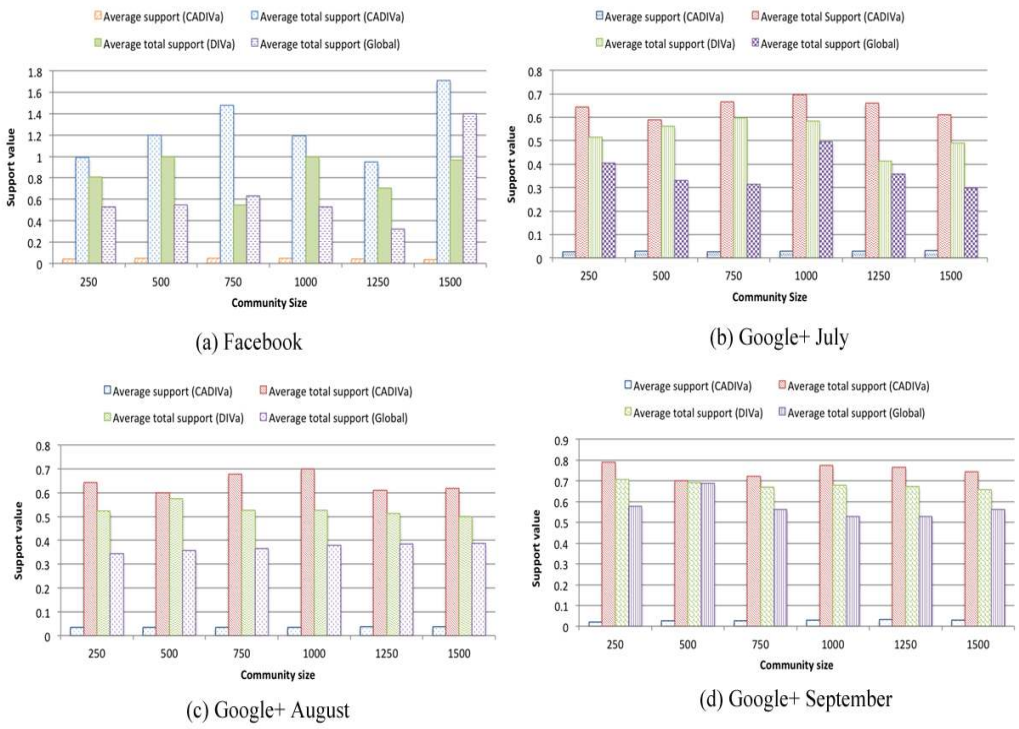

Fig. 5. The average support values computed as a lower bound for the threshold, and comparison of average total support of extracted CASes by CADIVa, DIVa, and the global approach.

5.3. Communication Overhead

We conducted several experiments to evaluate the communication overhead of CADIVa, particularity the overhead of the implemented gossip algorithm for LCAS aggregation (see Algorithm 2). The algorithm is simple: each node knows a small and continuously changing set of other nodes belonging to its communities stored in node's CRM (i.e., local repository for storing random community members information). Then, each node forwards its LCAS to this set of nodes and receives back their LCASes and merges them with its LCAS to construct the final CAS. Therefore, the GossipSampling procedure in Algorithm 2 is executed for sufficient number of cycles to fill nodes RCMes with a random sample of their community members. In each cycle, a node gossips twice with two randomly selected nodes from its CRM: exactly once as an initiator and once as a responder. It, therefore, sends two gossip messages and receives another two for each contacted entry in each cycle. If l is the number of exchanged entries, the gossip message then consists of l cache entries. We used two different values for l such that we executed two set of experiments, in the first one we set $l = 5$, whereas in the second one $l = 10$.

Figure 6 depicts the average converge speed of the gossip exchange algorithm with respect to different exchange size and communities size as well for all the datasets. As shown, the number of rounds increase with both of exchange size and community size. Larger communities require larger number of rounds so that nodes succeed in sampling random members from their communities. Similarly, by comparing Figure 6(a) and Figure 6(b) where the cache size increases from 20 to 50 entries, we can

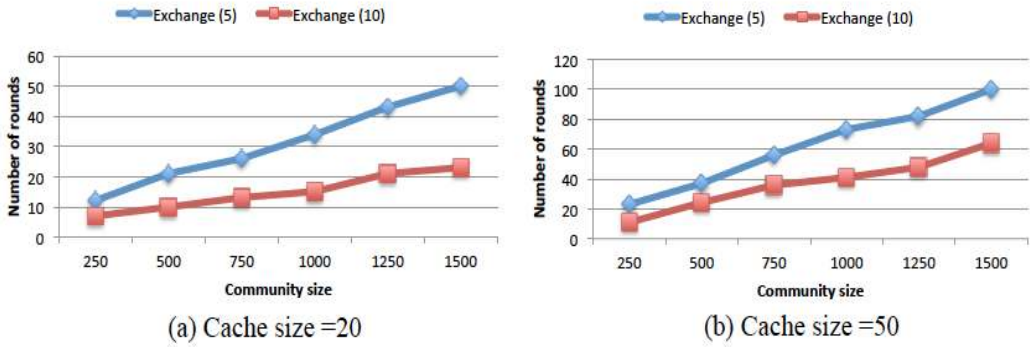

Fig. 6. The convergence speed with respect to community size and number of entries exchanged in a single gossip message.

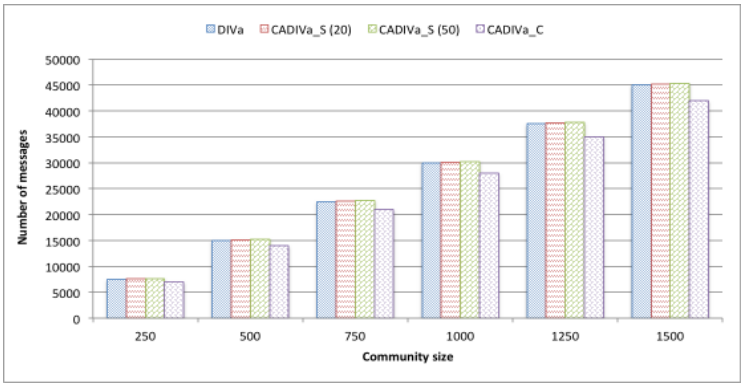

Fig. 7. The communication overhead reported by DIVa and different CADIVa implementations.

see that the number of rounds increases. Consequently, this will directly affect the communication overhead that will increase as well.

Figure 7 shows the total communication overhead. The figure starts by showing the communication overhead of DIVa that is reported during the community detection and LCAS aggregation. Then, the second column shows the overhead of CADIVa_S, the first implementation of CADIVa where the phases of community detection and LCAS aggregation are separate, with CRM size equals to 20 entries. Similarly, the third column is the communication overhead of CADIVa_S but with CRM size equals to 50. Finally, the last column represents the communication overhead of CADIVa_C, the second version of CADIVa where community detection and LCAS aggregation is combined into one single phase. The results show that CADIVa_C has the lowest communication overhead compared to other implementations.

5.4. Incremental Updates on Dynamic Graphs

Each of Google+ datasets contains timeID with values 0, 1, or 2, indicating which snapshot a directed link between two users appeared in. Thus, we execute our experiments incrementally to update the social graph by adding edges among nodes using timeIDs. When a node is added to the graph, this node determines its community memberships based on the dominant communities among existing nodes with which it is going to connect. Furthermore, the new node receives the communities CASes form its direct

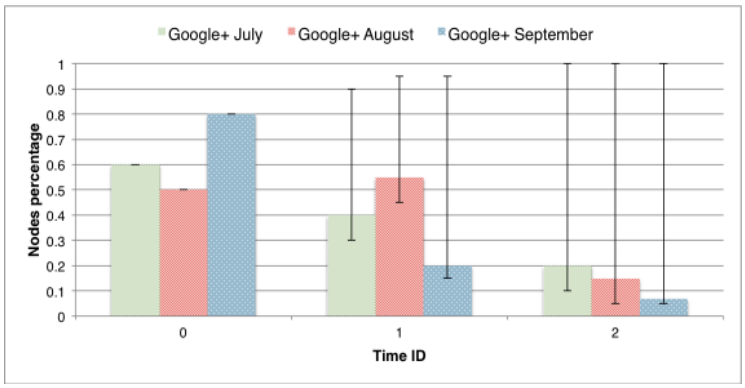

Fig. 8. Percentage of nodes changing their communities at each snapshot.

friends and store the CASes on its cache. Subsequently, this new node starts the *GossipSampling* procedure in order to have random samples from the nodes belonging to the same communities informing them of its existence and exchanging LCASes with them.

5.4.1. Incrementally Updating Communities. In this set of experiments, we study the effect of the newly added nodes and edges in the social structure of previously detected communities, which requires the re-computation of community memberships and CAS aggregation. Specifically, previously existing nodes monitor topological changes that affect their community membership, and re-execute the community detection module followed by CAS aggregation module when required. Figure 8 shows the percentage of nodes re-performing CASes extraction due to topological changes in their communities. In particular, the lower bound of change should be the percentage of new nodes, where only those nodes execute the community detection module and gossip sampling procedure. Meanwhile, the upper bound would mean that the process will start all over from the beginning such that all nodes execute the community detection and LCASes aggregation. The vertical error bars in Figure 8 represent the range of expected change in the graphs after adding the new nodes.

Intuitively, in the first snapshot all nodes execute both modules. As shown in Figure 8, 60%, 50%, and 80% of Google+ July, Google+ August, and Google+ September, respectively were loaded at the beginning. In the second snapshot 30%, 45%, and 15% new nodes were added to Google+ July, Google+ August, and Google+ September graphs, respectively. The results of the three datasets show that, on average, 17% of old nodes got affected by topological changes caused by the new joining nodes and performed community detection followed by LCASes extraction. The average change reported for adding the last snapshot across all three datasets is only 6%. Consequently, in our framework nodes are able to detect the topological changes surrounding them. Moreover, the results show that these changes are localized and require re-computations only for changed regions not the whole graph.

5.4.2. Incrementally Updating CASes. In this set of experiments we analyzed the change occurred in CASes while the communities evolve. We performed hierarchical community detection to show how CASes change with the increase of community size. Figure 9 depicts the average change occurred in CAS size by incrementally loading the nodes belonging to different communities in the Google+ datasets. We started by loading only 20% of nodes and calculating the average CAS size in the detected communities. Then, we incrementally added more nodes to reach 50% of communities members

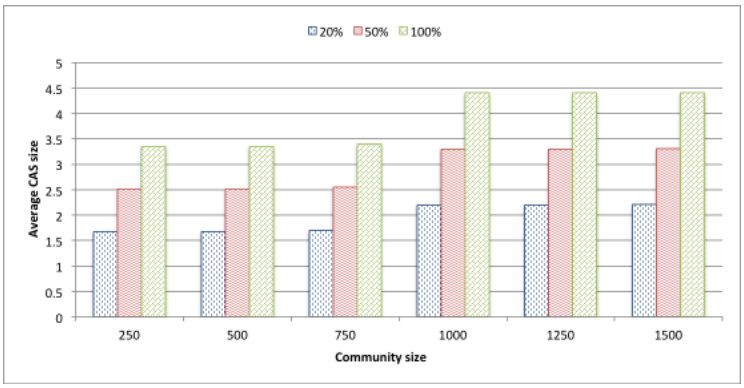

Fig. 9. The average size of communities CASes generated by CADIVa by incrementally adding nodes.

till all the nodes were added to the graph. The results emphasize the ability of our model to extract the community-level CASes that reflect the topological structure of the underlying communities and the properties of the user population belonging to each community.

6. RELATED WORK

Personal identity, its formation process, and its components have been the subject of scientific discussion and research work across multiple scientific disciplines such as sociology [Stets and Burke, 2003], psychology [Spears et al., 1997], criminology [Lynch et al., 2000], etc. With the growth of the Internet as a world wide virtual platform that connects data, devices, people, etc, new dimensions for humans' interactions have seen the light of day. Online human to human interactions developed from basic open chat rooms connecting virtual personas to nowadays popular and widespread OSNs with more sophisticated communication and data exchange forms. Within these emerging online socializing realms, identity has had its place as a pole of attraction for researchers from different disciplines. From a computer science perspective, resolving identities in the sense of differentiating between real and fake ones has been the main research concern related to identity. As a result, we find many pieces of work studying and formalizing online identities patterns with the objective of classifying them as good or bad. This gave birth to classifications for bad identities such as sybil (a fake identity operated, along with many other sybils, by one same physical entity)[Yu et al., 2006b; Yu et al., 2008], clone (an identity created by a malicious entity based on information collected about another honest entity)[Jin et al., 2011], compromised (an honest identity but taken control of by a malicious entity)[He et al., 2014] , etc. Therefore, we find works such as SybilyGuard [Yu et al., 2006b] and SybilLimit [Yu et al., 2008] that study OSN topological properties to detect sybil identities. We find [Jin et al., 2011], a framework for the detection of clone identities based on attribute and friends' network similarities, or [He et al., 2014] where the authors address identity theft across multiple social networks. These works, with others on the same line, share the common goal of detecting malicious nodes classified under formalized identity attack trends. However, identity concerns on OSNs go beyond binary classification. For example, some 'good' identities are created with the aim of fooling a category of users, such as child abuse over social networks [Hope, 2013][Chorley, 2012].

Studying identity related attacks is unquestionably an important thread of work, but there is also a parallel need for empowering users themselves to evaluate the trustworthiness and the validity of the online identities they interact with. The liter-

ature provides us with works such as [Sirivianos et al., 2012] where it is suggested to evaluate an identity on a given network based on feedback of her connections on another one. [Cai et al., 2011] suggests people to people recommendations for friendships' acceptance by relying on collaborative filtering techniques. In [Chairunnanda et al., 2011], users are suggested to be identified from their typing patterns; whereas chatting patterns are exploited for users' identification in [Roffo et al., 2013]. More recently, [Goga et al., 2013] suggests identifying users across networks based on geo-location and time-stamp information attached to their posts and on their writing styles. All these pieces of work still do not provide users with a framework to evaluate, by themselves, their perceived trustworthiness of their new online contacts. At this level comes [Bahri et al., 2014] to suggest using community feedback to assign trustworthiness levels to identities on a social network. More precisely, identities in [Bahri et al., 2014] are validated based on community validations of homogeneity between values of some defined correlated profile attributes. However, [Bahri et al., 2014] relies on a central repository of all the profiles of the OSN, on the existence of a group of trusted users for the learning of the correlated profile attributes, and on the responsiveness of the OSN community to evaluate available target identities.

In contrast to the centralized and supervised approach exploited in [Bahri et al., 2014] to extract the correlations among profile attributes from a profile schema, we previously proposed DIVa [Soliman et al., 2015] that adopts decentralized and privacy preserving approach. Instead of supervised learning that requires human feedback, DIVa successfully conceptualizes users online identities by extracting the correlations among profile attributes from the user population. Additionally, DIVa provides community-based validation by mining the correlations form the individual communities not from the user population as a whole. DIVa regulates the validation based on communities; however, it relies on a central role within each community of a diva node that is responsible of aggregating the observed identity patterns. For this, in this paper, we present CADIVa that operates without the reliance on any central roles and that is based on more reliable, scalable and commonly observed assumptions. CADIVa is fully automated, fully decentralized, and proves efficiency and effectiveness with real OSN data. To the best of our knowledge, this work is a first in addressing identity validation based on fully unsupervised and fully decentralized learning from profile information only.

7. CONCLUSION

In this paper, we have introduced CADIVa that is unsupervised, reliable and fully decentralized identity validation model for DOSNs in contrast to existing centralized approaches. CADIVa conceptualizes user online identities by mining the correlations among user profile attributes not from user population as a whole, but from individual communities, where the correlations are more pronounced. Furthermore, CADIVa empowers users with identity validation scheme that they themselves can use to evaluate the trustworthiness and the validity of the online identities they interact with. In our experiments we show that reliance on revealing the highly expressed patterns inside communities resulted in extracting community-aware validation rules with average improvements up to 36% and 50% than semi-centralized and global approaches, respectively. Furthermore, our model maintains users' privacy during the learning phase as users profiles information are processed only by their direct friends. The experiments show the effectiveness and scalability and reliability of our proposed model.

As a natural continuation of the work, we plan to enrich the process of extracting profile attribute correlations with text-based analysis to map words to broader topics. Therefore, CAS learning is enhanced such that exact word matching will be replaced by ontology and topic models.

REFERENCES

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216. ACM.

Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499.

Akcora, C. G., Carminati, B., and Ferrari, E. (2012). Privacy in social networks: How risky is your social graph? In *ICDE'12*, pages 9–19. IEEE.

Bahri, L., Carminati, B., and Ferrari, E. (2014). Community-based identity validation on online social networks. In *Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on*, pages 21–30. IEEE.

Brandes, U., Delcling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2008). On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20(2):172–188.

Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y. S., Compton, P., and Mahidadia, A. (2011). Collaborative filtering for people to people recommendation in social networks. In *AI 2010: Advances in Artificial Intelligence*, pages 476–485. Springer.

Chairunnanda, P., Pham, N., and Hengartner, U. (2011). Privacy: Gone with the typing! identifying web users by their typing patterns. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 974–980. IEEE.

Chorley, M. (2012). How facebook and social networking sites are used by child abuse gangs to groom victims for 'sex parties'. Mail Online.

Datta, A., Buchegger, S., Vu, L.-H., Strufe, T., and Rzadca, K. (2010). Decentralized online social networks. In *Handbook of Social Network Technologies and Applications*, pages 349–378. Springer.

Debatin, B., Lovejoy, J. P., Horn, A.-K., and Hughes, B. N. (2009). Facebook and online privacy: Attitudes, behaviors, and unintended consequences. *Journal of Computer-Mediated Communication*, 15(1):83–108.

Dwyer, C. (2011). Privacy in the age of google and facebook. *Technology and Society Magazine, IEEE*, 30(3):58–63.

Ferrara, E. (2012). Community structure discovery in facebook. *International Journal of Social Network Mining*.

Goga, O., Lei, H., Parthasarathi, S. H. K., Friedland, G., Sommer, R., and Teixeira, R. (2013). Exploiting innocuous activity for correlating users across sites. In *Proceedings of the 22nd international conference on World Wide Web*, pages 447–458. International World Wide Web Conferences Steering Committee.

Gong, N. Z., Talwalkar, A., Mackey, L., Huang, L., Shin, E. C. R., Stefanov, E., Song, D., et al. (2011). Jointly predicting links and inferring attributes using a social-attribute network (san). *arXiv preprint arXiv:1112.3265*.

He, B.-Z., Chen, C.-M., Su, Y.-P., and Sun, H.-M. (2014). A defence scheme against identity theft attack based on multiple social networks. *Expert Systems with Applications*, 41(5):2345–2352.

Hipp, J., Güntzer, U., and Nakhaeizadeh, G. (2000). Algorithms for association rule mininga general survey and comparison. *ACM sigkdd explorations newsletter*, 2(1):58–64.

Hope, C. (2013). Facebook is a 'major location for online child sexual grooming', head of child protection agency says. The Telegraph.

Huber, M., Mulazzani, M., Weippl, E., Kitzler, G., and Goluch, S. (2011). Friend-in-the-middle attacks: Exploiting social networking sites for spam. *Internet Computing*.

Jagatic, T. N., Johnson, N. A., Jakobsson, M., and Menczer, F. (2007). Social phishing. *Communications of the ACM*.

Jin, L., Takabi, H., and Joshi, J. B. (2011). Towards active detection of identity clone attacks on online social networks. In *Proceedings of the first ACM conference on Data and application security and privacy*, pages 27–38. ACM.

Kapanipathi, P., Anaya, J., Sheth, A., Slatkin, B., and Passant, A. (2011). Privacy-aware and scalable content dissemination in distributed social networks. *The Semantic Web–ISWC 2011*, pages 157–172.

Koll, D., Li, J., and Fu, X. (2014). Soup: an online social network by the people, for the people. In *Proceedings of the 15th International Middleware Conference*, pages 193–204. ACM.

Kotsiantis, S. and Kanellopoulos, D. (2006). Association rules mining: A recent overview. *GESTS International Transactions on Computer Science and Engineering*, 32(1):71–82.

Krivitsky, P. N., Handcock, M. S., Raftery, A. E., and Hoff, P. D. (2009). Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social networks*.

Li, N., Qardaji, W. H., and Su, D. (2011). Provably private data anonymization: Or, k-anonymity meets differential privacy. *CoRR, abs/1101.2604*, 49:55.

Low, Y., Bickson, D., Gonzalez, J., Guestrin, C., Kyrola, A., and Hellerstein, J. M. (2012). Distributed graphlab: a framework for machine learning and data mining in the cloud. *Proceedings of the VLDB Endowment*.

Luo, W., Liu, J., Liu, J., and Fan, C. (2009). An analysis of security in social networks. In *Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on*, pages 648–651. IEEE.

Lynch, M. J., Michalowski, R. J., and Groves, W. B. (2000). *The new primer in radical criminology: Critical perspectives on crime, power, and identity*. Criminal Justice Press Monsey, NY.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.

Nilizadeh, S., Jahid, S., Mittal, P., Borisov, N., and Kapadia, A. (2012). Cachet: a decentralized architecture for privacy preserving social networking with caching. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 337–348. ACM.

Rahimian, F., Girdzijauskas, S., and Haridi, S. (2014). Parallel community detection for cross-document coreference. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 2, pages 46–53. IEEE.

Robinson, R. M. (2015). Social engineering attackers deploy fake social media profiles. Security Intelligence.

Roffo, G., Segalin, C., Vinciarelli, A., Murino, V., and Cristani, M. (2013). Reading between the turns: Statistical modeling for identity recognition and verification in chats. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 99–104. IEEE.

Sirivianos, M., Kim, K., Gan, J. W., and Yang, X. (2012). Assessing the veracity of identity assertions via osns. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1–10. IEEE.

Soliman, A., Bahri, L., Carminati, B., Ferrari, E., and Girdzijauskas, S. (2015). Diva: Decentralized identity validation for social networks. In *Advances in Social Network Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 383–391. IEEE/ACM.

Spears, R. E., Oakes, P. J., Ellemers, N. E., and Haslam, S. (1997). *The social psychology of stereotyping and group life.* Blackwell Publishing.

Stets, J. E. and Burke, P. J. (2003). A sociological approach to self and identity. *Handbook of self and identity*, pages 128–152.

Stringhini, G. (2014). *Stepping Up the Cybersecurity Game: Protecting Online Services from Malicious Activity*. PhD thesis, UNIVERSITY OF CALIFORNIA Santa Barbara.

Yu, H., Gibbons, P. B., Kaminsky, M., and Xiao, F. (2008). Sybillimit: A near-optimal social network defense against sybil attacks. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 3–17. IEEE.

Yu, H., Kaminsky, M., Gibbons, P. B., and Flaxman, A. (2006a). Sybilguard: defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review*, 36(4):267–278.

Yu, H., Kaminsky, M., Gibbons, P. B., and Flaxman, A. (2006b). Sybilguard: defending against sybil attacks via social networks. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 267–278. ACM.