A Comprehensive Trust-based Information Security Model for Online Social Networks

Nadav Voloch

LIBERTY
Academic Books

Nadav Voloch

A Comprehensive Trust-based Information Security Model for Online Social Networks

LIBERTY
Academic Books

10.09.2022
by Liberty Academic Publishers
New York, USA

For citation purposes, cite as indicated below:
Nadav V., A Comprehensive Trust-based Information Security Model for Online Social Networks ; Liberty Academic Publishers : New York, USA, 2022.

Liberty Academic Publishers works to counter discrimination on grounds of gender, race, disability, age and sexuality.

Cover design by Andrew Singh

ISBN: 978-1-955094-23-8

ISBN 978-195509423-8

9 781955 094238

Ben-Gurion University of the Negev

Faculty of Natural Sciences

Department of Computer Science

A Comprehensive Trust-based Information Security Model for Online Social Networks

Nadav Voloch

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Under the supervision of **Prof. Ehud Gudes** and **Prof. Danny Hendler**.

August 2022

Advisor: Prof. Ehud Gudes. **Advisor Signature:** ________

Advisor: Prof. Danny Hendler. **Advisor Signature:** דני הנדלר

Acknowledgments

First and foremost, I want to thank my thesis advisor Ehud Gudes, for giving me the opportunity to perform this research, guiding and helping me in every aspect. I believe that today I am a better researcher thanks to these years of guidance with him. Second, I wish to express my deepest gratitude to Nurit Gal-Oz, whom I've known professionally for many years, and who has been a most important part of this research work almost from the beginning. Her work with me on our research papers has made me improve every aspect of my research abilities and helped us publish our work in excellent publications. Third, I would like to pay my special respects to Danny Hendler, my academic advisor and Department Head, who helped us in all of the formal procedures of the thesis and PhD stages from day one. I am in deep debt to my amazing wife, Dganit and our two amazing daughters, Nitzan and Niva, for being so supporting of my endeavors during these years; I wouldn't have been able to do it without you. I would also like to thank my office mates, Geoffrey and Amit, for giving me feedback on my work, and being great company as well. I would also like to thank the Ben Gurion University Cyber Center, and the Computer Science Department for funding this research. Finally, I would like to thank the people of the Computer Science Department: Mazal, Roni, Sigal and everyone else who have helped me tremendously during these years.

Table of contents

Abstract

Online Social Networks (OSNs) have become a central means of communication and interaction between people around the world. The essence of privacy has been challenged throughout the past two decades as technological advances have enabled benefits and social visibility to active members that share content in online communities. While OSN users share personal content with friends and colleagues, they are not always fully aware of the potential unintentional exposure of their information to various people, including adversaries, social bots, fake users, spammers, or data-harvesters. Preventing this information leakage is a key objective of many security models developed for OSNs, including access control, relationship-based models, trust-based models and information-flow control. In this research, we assert that a combined approach is required to overcome the shortcomings of each model. In this thesis we present a new model to protect users' privacy.

The first part, the basic trust-based model, is composed of three main phases addressing three of its major aspects: trust, role-based access control and information flow. This model considers a user's sub-network and classifies the user's direct connections into Trust-based roles. It relies on public information, such as the total number of friends, age of user account, and friendship duration, to characterize the quality of the network connections. It also evaluates trust between a user and members of the user's network to estimate whether these members are acquaintances or adversaries, based on the paths of the information flow between them. Finally, it provides more precise and viable information-sharing decisions and enables better privacy control in the social network. We have evaluated this trust-based model with extensive experiments using both synthetic and real users' networks to demonstrate its ability to provide a naïve user with a good means of privacy protection. We have validated separately every phase of the model. The results show a strong correlation between the decisions made by the algorithm and the users' decisions.

The second part of the thesis has four main subsections: the first one is an analysis of the robustness of the model by creating attacks on it and proving their futility.

We simulated attack scenarios carried out by a community of malicious users that attempt to fake the OSN features of the model. We then analyzed an attack by an alleged trustworthy clique of adversaries and showed the futility of such an attack due to the strength of the model's parameters and combination of trust, access control and flow control.

The second subsection treats context-based content and builds a context-based trust model in personal networks. We validated this model by analyzing trust using sentiment analysis of posts in a real network. This part of the model created a much more accurate picture of OSN users and their data and helped to reveal the sources of sensitive data exposures and to prevent them from happening.

The third subsection is an extension of the second one and comprises a fake-news-propagation-prevention solution using Machine Learning (ML) algorithm based on trust and content. We used reinforcement learning to detect problematic users by analyzing data items that are fake or misleading.

The fourth part focuses on the General Data-Protection Regulation (GDPR) enforcement. We created a solution for social networks that deals with various data types and their type of control. We used different aspects of trust and consent and also used the context-based model for enforcement of the GDPR, and we presented corresponding evaluations for it.

This work appears in two journal papers (Voloch, Gal-Oz, & Gudes, 2021; Voloch, Gudes & Gal-Oz, 2022) and eight conference papers (Gudes & Voloch 2018; Voloch & Gudes, 2019; Voloch, Levy, Elmakies, & Gudes, 2019A & 2019B; Voloch, Gudes & Gal-Oz, 2021A & 2021B & 2021C; Voloch, Gudes, Gal-Oz, Mitrany, Shani, & Shoel, 2022).

1. Introduction

Privacy and security problems in Online Social Networks (OSNs) have been a key challenge for researchers in the past decade. As more users are becoming active participants in social networks, the threat to their security and privacy is growing due to the potential risk of data leakage to adversaries. The major consequences for users, as defined by (Kayes & Iamnitchi, 2017), are the inappropriate sharing of personal information, i.e., leakage and exploitation of personal details using active mining (information linkage). OSNs are therefore required to provide a means that allows users to share data while controlling the dissemination of their personal information. In this research we combine aspects from both access control and flow control to achieve a comprehensive and efficient model for preserving privacy in social networks. Discretionary access-control policies provided to users by OSNs make it possible for users to limit access to their information using constraints. However, ordinary users do not have the proper knowledge to make informed privacy decisions and cannot anticipate the spread of their information and the possible consequences of releasing personal information, as described in (Li, Li, Yan, & Deng, 2015). According to (Misra & Such, 2016), there is very little, or no, actual user awareness of the spreading of personal data throughout the network, and the extent to which the data is spread is seldomly evaluated correctly. Some solutions to this problem involve handling the OSN information-sharing instances as an access control system, in which there is selective restriction of access to the network's resources. Access-control models are implemented to prevent unauthorised access to sensitive information and to mitigate security and privacy risks. A survey presented in (Sayaf & Clarke, 2012) lists OSN access control models, elaborating the functionalities of the different types of models. (Hirschprung, Toch, Schwartz-Chassidim, Mendel, & Maimon, 2017) suggest an architecture for privacy in OSNs that includes access control by reducing the built-in privacy settings controlled by user preferences. The OSN flow-control problem was investigated in several recent papers. An OSN community is described as a connected graph, where each user is a node, and an edge connecting two nodes indicates a relationship between two users. (Ranjbar & Maheswaran, 2014) define a user's community, denoted myCommunity, as the largest sub-graph of users who are likely to receive and hold the user's information without leaking.

The flow-control aspect of the model operates when there is no coherent role definition within the OSN. While the roles of direct friends are well defined, as they are familiar with the source user to a certain extent, the users that are not directly connected lack a formal role and form a potential privacy hazard. In this part of the model the edges to cut are selected according to parameters of credibility. OSN users know that the information they share is exposed to their friends, but they do not necessarily know that an act of their friends (share, like, etc.) on this data, exposes it to different, and maybe unknown, users' networks.

We use the information-flow part of the model to create a trustworthy network of users, to whom the data being spread from a source user is monitored. This is done by applying one of two alternatives: the first is using a known graph algorithm for finding a Minimum Spanning Tree (MST), such as Kruskal's algorithm (Kruskal, 1956); the second alternative identifies possible adversaries by the flow-control model (Gudes & Voloch 2018), using Dinic's algorithm for finding all the paths from a source user node to a target node (an unknown user). We first identify each user as either a potential adversary or an acquaintance with respect to the source user, by the evaluation of different attributes, such as total number of friends, friendship duration, age of user account, etc. A trust value is then computed for each of the nodes and edges, and, based on a pre-defined trust threshold, edges are cut to block information flow to adversaries.

Several attacks on private information in social networks have been described in (Heatherly, Kantarcioglu & Thuraisingham, 2012). A common type of attack in OSNs aims at a specific user or network, and attempts to access or act on its information, e.g., spread false data or spam for different purposes. Trust-based systems must deal with attacks in which malicious users initially behave properly to gain a positive reputation, but then start to misbehave and inflict damage on the community. In the part of 'analyzing attacks on the model', we show the robustness of our model and focus on the latter type of attack, where a user, or its network, is the target of an attack initiated by malicious users. The main scenarios we simulate include a community of spammers, whose profiles conform with the OSN attributes that constitute the trust aspect of the model. We use a graph algorithm (minimal vertex cover) to select an optimized set of candidate nodes to compromise, and show that even in this case, such an attack is futile.

After building the basic model and analyzing its robustness, one of our primary goals was to refine the trust between OSN users to be context aware. Our aim is to identify interactions between users and to characterize the quality of these interactions in a way that may imply trust between them in a certain context.

The motivation for the context evaluation part of this research is to extend the basic trust model and to make an important distinction between different types of data instances that differ in their subject's sensitivity. For example, a political post might be more sensitive for its publisher than a post discussing food.

OSN user's friends are not homogeneous by nature, and accommodate different perspectives and views; therefore, we can expect different users to perceive the same subject with different levels of sensitivity. Moreover, users may wish to share a piece of sensitive information on one subject with people they trust more concerning that subject but avoid sharing with these same people sensitive information on other subjects.

The General Data-Protection Regulation (GDPR) (Regulation (EU) 2016/679) is a regulation for data protection and privacy for citizens of the EU that affects most of the commercial companies, government institutions, and other sectors that maintain personal information on their customers or audiences. The enforcement of the GDPR represents a great challenge for OSNs, which are required to make significant changes to achieve compliance with these regulations. (Kotsios, Magnani, Vega, Rossi, & Shklovski. 2019) provided practical guidelines to GDPR-compliant social network data, covering aspects such as data collection, consent, anonymization, and data analysis. The insufficiency of the approach with respect to privacy policies addressing GDPR of tech giants like Facebook and Google has been criticized by the media through the years. Our mechanism to enforce GDPR uses techniques such as digital rights management (DRM) system (Peinado, Abburi, & Bell, 2006) and watermarking that will be elaborated on in the related work section.

In the experimental work in this part of the research, we used three different datasets, all taken from real Facebook networks. For the first part, concerning context evaluation, we used a network for which we devised several data categories. We assessed the trust level of every user, and in each category, we analysed their posts trust-wise.

For the second part, concerning sentiment analysis, we used two datasets, for which we collected specific trust scores for the posts, and their sentiment analysis. Our purpose was to find the effect of sentiment in a post to the user's trust in a certain context. For the third part of the GDPR implementation we used three different datasets and two ego networks for numerical estimations of the DRM implementation of the model.

In the fake news part of this research, we use the context-based model as a basis to address the problem of fake news in OSN. Fake news is a term without a fixed definition. It usually refers to fabricated or misleading information that lacks accuracy and credibility.

Today, social media presents a major problem regarding the spreading of fake news - users spread information that seems right to them and is not necessarily procured from a reliable source did not undergo proper fact checking. In extreme cases, fake news can get a lot of exposure and cause harm. For example, texts regarding fabricated data about vaccines can cause a significant percentage of a population to not get vaccinated, which can consequently cause death. Social network users are exposed to a lot of information on a daily basis. This information is brought to them by their network - posts by people with whom they are connected, or posts shared by their connections.

The detection and propagation-prevention of fake news is done by extending this model with context-awareness and user profiling trust-wise, and then use Machine Learning (ML) to find users that have a high probability of being fake-news propagators. The extension of the basic trust model is done to make an important distinction between data instances that differ in their subject. After creating a unique trust value for all the ego users' friends for each data category, we can evaluate their data in the sense of facticity used in the model. This part of the research focuses on preventing low-trusted users from spreading sensitive information. Our model uses the technique of blocking these users, and thus stopping the propagation of false information.

In summary, our main contributions are:

- Developing a comprehensive trust-based model for security and privacy in OSNs by using Trust, access control and flow control.

- Analysing attack scenarios on the model and establishing it strength and robustness.
- Building a context-based extension for the model, that refines its accuracy, and making it a viable solution for real users, with an emphasis on their different types of OSN content.
- Applying the extended model for GDPR enforcement, that will help improve the suitability of OSN to some of the requirements in the regulation.
- Preventing Fake News propagation in OSN, using our extended model, with Reinforcement Learning, with an emphasis of detecting problematic users that could be Fake News propagators.

We hope our model can provide a better solution for these issues that will create a stronger infrastructure for social networks.

2. Background and related work

This section is divided into several subsections, corresponding to the different parts of this research.

2.1 Privacy and security in OSN

Access-control models, and specifically ones describing OSN privacy, have been studied extensively over the past decade. The main access-control model used in OSN is Role-Based Access Control (RBAC), which has many versions, as presented in (Sandhu, Coyne, Feinstein, & Youman, 1996), and limits access by creating user-role assignments. The user must have a role that has permission to access that resource.

The most prominent advantage of this method is that permissions are not assigned directly to users but to roles, making it much easier to manage the access control of a single user, since it only must be assigned the right role.

An addition to the trust factor of this model is proposed in (Lavi & Gudes, 2016); it is based on the network users' interaction history, which could be problematic in assessing the trust of relatively unknown new connections. In this research we circumvent this problem by adding independent user attributes to the trust estimation. An example of using RBAC in Facebook is provided by (Patil & Shyamasundar, 2017), where the use of roles and the possible breaches that can occur due to flexible privacy settings of the network are described.

Another important model we rely on is Relationship-Based Access Control (ReBAC), presented in (Cheng, Park, & Sandhu, 2012), which is based on user-relationships in OSN. The model is topology-based and establishes relationships between users in the social network with a sequence of binary conditions. (Fong, 2011) presents a model that implements the contextual nature of relationships, in which a policy language for ReBAC is devised based on modal logic, for composing trust-based access-control policies with an applicative paradigm. In (Crampton & Sellwood, 2014), the formal ReBAC model was developed into a two-stage method for evaluating policies. These policies were defined by semantics for path conditions, similar to regular expressions, which were used to develop a policy evaluation method.

A model based on relationship strength between friends – RSBAC (Relationship Strength-Based Access Control) is presented in (Kumar & Rathore, 2016). The model calculates the level of closeness between users according to their social activities and their profile similarities. The authors argue that OSN users that have profile similarity (in terms of attributes) and communicate frequently, necessarily have a high degree of closeness, and therefore should get broader permissions to each other's data instances. (Squicciarini, Paci & Sundareswaran, 2014) suggest an automated mechanism for determining access rules by the user's privacy preferences. An interesting approach of this paper is the suggestion of categorizing data instances by their types for this mechanism. (Bahri, Carminati & Ferrari, 2018) survey two main approaches of dealing with OSN privacy – decentralized and centralized architectures, where each has its unique advantages and challenges.

The problem of spammer detection (Cohen., Gordon & Hendler, 2018) is closely connected with with the information leakage problem since the misuse of private data is the common ground for both, and the prevention of privacy breaches is in their mutual interest. Basing this detection on user attributes is handled in (Benevenuto, Magno, Rodrigues & Almeida, 2010), specifically for Twitter, where several important user attributes (such as the age of the user account, the fraction of tweets with spam words and other factors) were checked on real data from Twitter, and with these values a spammer profile could be characterized. In (Zheng, Zeng, Chen, Yu & Rong, 2015) this detection is applied on Facebook datasets, where it is shown that spammers usually have noticeable differences in the values of certain attributes, such as the number of friends, tags and mentions. (Viswanath, Post, Gummadi & Mislove, 2011) describe a trustworthy network of users that can be identified by the clusters around trusted nodes. This implementation creates a relatively credible environment of users, in which data can be safely transferred since this network is comprised of generally trustworthy users. (Fogues, Murukannaiah, Such & Singh, 2017) give a privacy solution for the unwanted sharing problem by giving an agent-based approach, that has an incremental learning method, reducing the user's involvement in multi-user scenarios by asking the user for input only if necessary. Anonymizing an OSN for enhancing privacy is a related problem. Changing a graph for anonymization purposes is discussed in papers such as (Das, Eğecioğlu & El Abbadi, 2010) and

(Tassa & Cohen, 2013), where the anonymization of the OSN is done by sequential clustering.

(Lin, Steiert, Morris, Squicciarini & Fan, 2019) investigate the risks of image exposure in the OSN, especially in images that have shared ownership. The paper gives an interesting approach for calculating the probability of exposure to problematic users by reviewing the sharing history of the image.

2.2 Trust based models for OSN

Employing trust in OSN is widely used in different models, and even in relatively early research, such as (Ali, Villegas & Maheswaran, 2007), the idea of involving trust in access control for OSN user data is handled by creating trust criteria for different subjects (users) and objects (data instances). A trust-based access control model for OSN is presented in (Wang & Sun, 2010), in which a policy refers to an access right that a subject can have on an object, based on relationships, trust, purpose and obligations in the network.

Using information-flow control for preserving privacy in OSN was investigated in several early papers on OSN. (Lucas & Borisov, 2008) present a privacy architecture that reduces potential information-leakage threats whilst preserving good accessibility to the user's important data. Another early paper that handles these issues is (Gross & Acquisti, 2005), that presents a heuristic method for network security based on user identification and shows a novel method of basing the credibility of a certain user on its relationship with other users.

In (Patil & Shyamasundar, 2017), the user-relationship approach for information-flow security in OSN is further developed. The OSN is portrayed in a modular manner, in which a deeper resolution of the graph is given; the vertices represent different data instances whilst the edges are connections between them, such as friendships between users, sharing of posts or pictures, etc.

The model is dynamic and fits a real-life applicative form as it shows the different graph instances in several timestamps, monitoring the changes over time. This model specifically uses the Facebook jargon of OSN activities such as 'wall posts', 'sharing', 'tagging', etc. This choice is well justified for Facebook as a multi-functional OSN, serving as a professional and social OSN, as well as serving other purposes and including numerous additional features (the Facebook model is also used in our research).

(Misra, Such & Balogun, 2016) present a model named IMPROVE - Identifying Minimal Profile Vectors for similarity-based OSN privacy control. It elaborates on the importance of user and connection attributes for setting a credibility level of an OSN data instance by giving ranking to these attributes. This ranking is based on information gained from each attribute, assessing its importance from the closeness approximation between users and evaluating the information-sharing willingness.

A major problem that exists especially in OSN is information flow to unwanted entities, violating the privacy of individuals. Even with a proper access-control model, it is desirable to prevent such flow, and this is a subject of recent research. In (Levy, Gudes, & Gal-Oz, 2016) a privacy-control model is established by defining the other users in three other closeness categories (close friend, acquaintance, and adversary); then, an algorithm is presented that moves edges from the OSN graph, having the amount of information flowing to the adversaries minimized, while the information flow to friends and acquaintances remains intact. The access given to information instances is decided by a Min-Cut algorithm, dividing the community graph for the purpose of preventing data leakage to unwanted entities such as spammers or other potential adversaries. In this model, the access granted to the user's information is based on user-to-user relationships, and we have based our access-control method on this approach.

(Squicciarini, Karumanchi, Lin, & Desisto, 2014) identify hidden circles in social networks by focusing on common interests. An interesting aspect of this research is the importance given to the security aspects of new users and new data instances (uploaded by all users). In this paper an approach is proposed that helps users to distribute their social network contacts into relevant groups automatically and helps users to set up their privacy policies automatically for their uploaded content. (Laleh, Carminati & Ferrari, 2016) give an interesting risk assessment scheme for OSN privacy that is based on user-anomalous behaviors based on their OSN activities, such as comments, likes, etc., and identifying possible attacks and adversaries by the analysis of these behaviors.

(Taheri-Boshrooyeh, Küpçü & Özkasap, 2015) present a wide range of security and privacy issues in OSN, with a distinction between two types of architectures: centralized and distributed. They analyze the issues of data integrity, data privacy and secure social searches. (Boshrooyeh, Küpçü, & Özkasap, 2018) introduce a novel system named PPAD as the first privacy-preserving group-based advertising system.

It protects users' privacy by using as a third-party an external non-colluding privacy-service provider. The system performs user and advertisement matching without requiring them to be online. Another group-based advertising mechanism is presented in (Boshrooyeh, Küpçü & Özkasap, 2020), which runs on several servers, each provided by an independent provider. The great advantage of using such a system is that user privacy is protected against an active malicious adversary, even if it controls all providers but one, all the advertisers, and a large fraction of the users. The system uses a zero-knowledge-proof mechanism and different encryption schemes to make it as secure as possible. However, neither (Boshrooyeh, Küpçü, & Özkasap, 2018) nor (Boshrooyeh, Küpçü & Özkasap, 2020) use a trust model to refine the privacy policies, as we present in this research.

The model we present in this research is inspired by the approaches and research discussed in this section. Our goal is to create a trustworthy solution for OSN users wishing to protect their data privacy and avoid unwanted information leakage. The novelty of the access control part of our model is that the relationships and their strengths do not determine access control directly, but are used along with other characteristics to compute the trust that an OSN user has towards another user. This trust may be used differently for different roles, depending on the model policy.

In the following parts of this research, we use OSN attributes that were investigated in the above-mentioned approaches to construct a comprehensive integrated access- and flow-control model that creates a trusted user network. This trusted user network is then used to facilitate a reasonable automatic access decision on behalf of the user with respect to others that do not have a clear social status in the user's network.

2.3 Robustness of Trust-based models

There are two major types of attack scenario in social networks. A common spammer attack is an attack that does not aim at a specific user or network, but rather sends spam or harvests data from anywhere it can. The other type of attack aims at a specific user or network and attempts to access or act on its information, e.g., spread false data or spam for different purposes. Trust-based systems must deal with attacks in which malicious users initially behave properly to gain a positive reputation, but then start to misbehave and inflict damage on the community.

In this research we show the robustness of our model and focus on the latter type of attack, where a user (or its network) is the target of an attack initiated by malicious users.

Collusion attacks, in which a group of malicious users act together with strong trust relations between them to manipulate the system and gain a high reputation, and then cause damage in the social networks, are described in (Li, Shen & Sapra, 2012), (Sun, Zhu & Fang, 2010) and (Viswanath et. al, 2014). Our simulated attacks on the model differ from collusion attacks on high-reputation systems, such as the one described in (Li, Shen & Sapra, 2012), as we attack the privacy of the ego user, based on the trust criteria established in our above-mentioned model.

The problem of such attacks on trust criteria is presented in (Sirur & Muller, 2019), where a reputation lag attack is described as a formal model capturing the core properties of the attack, in which the reputation of a user fails to reflect its behavior due to a delay, and a malicious user exploits this delay for personal gain.

Attacks on social networks are presented in relatively early papers such as (Lee, Caverlee & Webb, 2010), where a conceptual framework of a 'social honeypot' is described for uncovering social spammers who target online communities. The idea of creating social honeypots is also developed in (Paradise et. al, 2017), where the honeypot profiles were assimilated into an organizational social network. The honeypot then received suspicious friend requests and mail messages that revealed basic indications of a potential forthcoming attack.

An interesting form of attack that is related to our presented attack scenarios is the 'friend-in-the-middle' attack (Huber, Mulazzani, Weippl, Kitzler & Goluch, 2011), in which a legitimate friend in the social network is used as a gateway for spammers who harvest social data. This data can then be exploited for large-scale attacks, such as context-aware spam and social-phishing. The network used in this attack scenario is specifically Facebook. Our vertex-cover algorithm can be seen as a generalization of the friend-in-the-middle attack.

2.4 Context-based models

The context-based model part of the research presented in this thesis extends our basic model and relies on the context of data in the OSN, dividing it into different topics with different characteristics. In (Wang, Lei & Guanfeng, 2015), a contextual social network model that uses personal characteristics as the independent social context, and mutual relations, is presented. It proposes social-context-aware trust inference in OSN that is used for recommendations on service providers. In (Du, Yu, Mei, Wang, Wang & Guo, 2014) there is a very interesting use for context and content analysis in OSN, in an attempt to predict event attendance in event-based OSN (such as Facebook). (Sara, Tassa & Bonchi, 2016) handles the problem of preserving users' individual privacy when publishing relatively rich information in OSN. This is done by anonymization and context-related trust, by referring to connections between users in different topics.

The use of NLP in information security was made even in relatively early papers such as (Atallah, McDonough, Raskin & Nirenburg, 2000) and (Tsoumas & Gritzalis, 2006), and it is, of course, widely used in social media (Louis, 2016). Sentiment analysis is also used for these types of research, such as the one presented in (Hutto & Gilbert, 2014). In OSNs such as Facebook or Twitter, there is no mechanism for providing explicit feedback per interaction.

We consider comments to a post as an alternative to feedback and analyse them using sentiment analysis in order to determine users' attitudes to the post's content. Sentiment analysis is about categorizing a document (e.g., a post, an article, a comment) as expressing an overall positive or negative sentiment. Most recent detection systems are based on deep neural networks (Zhang, Wang, & Liu, 2018), and deal with various related problems, such as aspect-level sentiment classification (Xue & Li, 2018). Multiple labelled datasets (Blitzer, Dredze, & Pereira, 2007) and tools (Socher, Perelygin, Wu, Chuang, Manning, Ng, and Potts, 2013) are available for sentiment analysis. Many models of trust management ignore the fact that trust is subject to context, and, for simplicity, consider all trust-related information to be part of a general-purpose trust. Context-aware models calculate various aspects of interactions along with the corresponding evaluation values for a single trustee (Granatyr, Botelho, Lessing, Scalabrin, Barthès, & Enembreck, 2015).

Models like (Mokhtari, Nooria. Ladani, & Nematbakhsh, 2011) not only consider multiple contexts, but also describe adapting values between contexts, which may lead to a more accurate evaluation of trustees.

2.5 Fake news propagation

Fake news involves two major concerns, which have become subjects for research.

a. Identification and detection of the news as fake. This part is a difficult one and has seen only partial success. The detection on social media is defined and presented in (Shu, Sliva, Wang, Tang & Liu, 2017), while in (Tandoc, Lim, & Ling, 2018) an important typology of fake news is given in categorizing six different categories of fake news: native advertising, news satire, propaganda, manipulation, news parody, and fabrication. The most severe form of fake news is fabrication, since it has a very high level of the author’s immediate intention to deceive, and a very low level of veracity. (Kumar & Geethakumari, 2014) present an approach that uses psychological estimation of OSN users to detect misinformation spreading in the network, and (Levi, Hosseini, Diab, & Broniatowski, 2019) use semantic context to detect and analyze different categories of fake news.

b. Prevention of fake news propagation. (Vosoughi, Roy, & Aral, 2018) deal with the propagation of fake news and show that the spreading of fake news is done in a fast and thorough manner, since its nature is that of an extensive content, which has the potential of extremeness.

In (Li, Li, Yan & Deng, 2015), different types of data-spreading scenarios are described. Most of these vulnerabilities occur from discretionary privacy policies of the OSN users. These privacy policies create misleading knowledge of the number and type of users exposed to this shared data. Most of the solutions suggested demand changes in these specific policies. As mentioned above, (Misra & Such, 2016) presented the fact that there is very little or no actual user-awareness of the spreading of personal data throughout the network and the extent to which the data is spread is seldomly evaluated correctly. Detecting fake news by different types of learning is the topic of many very recent research papers such as (Choudhary & Arora, 2021) that proposes a linguistic model to find the properties of content that will generate language-driven features, and (Monti, Frasca, Eynard, Mannion & Bronstein, 2019) that uses geometric deep learning to detect fake news in OSN.

(Helmstetter & Paulheim, 2018) use supervised learning for this detection, and the work was done specifically on Twitter datasets, which are naturally very accessible due to twitter's publicly open infrastructure. (Pierri & Ceri, 2019) survey fake news in a comprehensive manner, in terms of detection, characterization and mitigation of false news that propagates on social media, using a data-driven approach.

2.6 GDPR for social networks

GDPR has become an important topic for both industry and government throughout Europe, yet academic papers on enforcing GDPR in social networks are very few. Handling the effects of GDPR on social networks is crucially important, since the regulations have been enforceable since May 2018. It is hard to find comprehensive solutions for GDPR, since there are many aspects: ethical, legal, technological and more, and a lot of resources have been invested in seeking suitable infrastructures that can satisfy these demands. Compliance with the GDPR is a complex pursuit that requires different types of solutions (e.g., security, international transfers, accountability, etc.). This is especially hard in social networks, where endless amounts of data are spread constantly.

(Cohn-Gordon, Damaskinos, Neto, Cordova, Reitz, Strahs, & Papagiannis, 2020) define various types of deleting information from OSN, comparing users for whom deletion is a tool for removing what they have shared and controlling their data with the OSN administrators. The latter are obliged to their users, thus risking the remaining of some data in other sources. The authors suggest a safe and comprehensive deletion framework for OSN that will help preventing cases of unwanted data remaining, compromising the users' privacy.

(Kotsios, Magnani, Vega, Rossi & Shklovski, 2019) examine the principles outlined in the GDPR in the context of social-network data and analyze the consequences of their implementation in OSN. Some of the most important issues they highlight and recommendations they provide include the following. First, they recommend that the OSN administration puts an emphasis on the initial data processing consent by the data owners (OSN users) vs. the public nature of the social network. Second, primary and secondary data collection and transparency issues. Third, the depth and spread of OSN data (for which the flow part of our model provides a good solution).

Fourth, some analysis techniques lead to a serious breach of data ownership and privacy, and user profiling should adhere to Art. 22 in the GDPR, meaning that the data subject (user) will have the right to not be exposed to any decision made based on this profiling. Last is the issue of data storage limitation, which may breach the important GDPR principal of the 'right-to-be-forgotten'. The efficacy of OSN compliance with the GDPR requirements that are relevant to the data handled by the OSN is presented in (Patil & Shyamasundar, 2018), where the challenges of implementing the regulations on OSN are outlined, and the association of their causes with the nature of the communication are presented in general, together with an indication of the problematic aspects of data spreading in the network. This research specifically addresses the right-to-be-forgotten issue in Facebook. (Goldsteen, Garion, Nadler, Razinkov, Moatti & Ta-Shma, 2017) present a consent-management solution that can be used for the implementation of GDPR on different platforms. A solution for different users that may or may not access certain data can be collaborative access control, as suggested specifically for social networks in (Amsterdamer & Drien, 2019).

Using DRM in social networks is presented in several papers, such as (Rodríguez, Rodríguez, Carreras, & Delgado, 2009) and (Marques & Serrão, 2013), that describe the privacy approach needed in OSN that can be manifested by the DRM, and (Liu, Liu & Shao, 2014) that presents the aspect of the misuse of digital rights in OSN and use access control to multimedia in the social network. Implementing GDPR-compliance solutions is presented in (du Toit, 2018), although there is no one easy fix for all the obligations in the GDPR. The use of watermarking to identify data in social networks was discussed by (Iftikahr, Kamran, Munir & Kahn, 2017). The scheme described there, called 'reversible watermarking', enables both the identification of the original owner watermark, and the recovery of the data in the case where some parts of it were deleted or changed. DRM can be used to strengthen the control of sharing and spreading of the data in the OSN. (du Toit, 2018) describes a DRM system called DEPRIM, which may be used to protect private data and implement GDPR. A DRM scheme that enables controlled sharing was suggested by (Davidson, Tassa & Gudes, 2016).

3. The trust-based model

3.1. Preliminary motivation and basic concepts

We represent a social network as an undirected graph, where nodes are the OSN users, and edges represent relations between them such as friendship relations. An ego node (or ego user) is an individual focal node, representing a user whose information flow we aim to control. An ego node along with its adjacent nodes are denoted an ego network.

Any OSN should provide means to control the dissemination of information items, and therefore any ego user should be able to explicitly restrict the flow of their information to selected target nodes based on their evaluated trust and their role within the ego user's network.

Users know that their information instances (e.g., pictures, posts, personal details, etc.) are revealed to their direct OSN friends. However, an information leakage may occur when one of these friends acts upon the user's information instance, e.g., comments on a post, likes or shares a picture, or any other form of OSN action. This action allows any friend of the actor who is not a direct friend of the ego node to access this information instance. Figure 1 describes an ego user's information leakage to friends of friends, triggered by an action taken by the ego user's direct friend on the ego user's data. The key objective of our model is to provide a general OSN solution to prevent data leakage to undesired users.

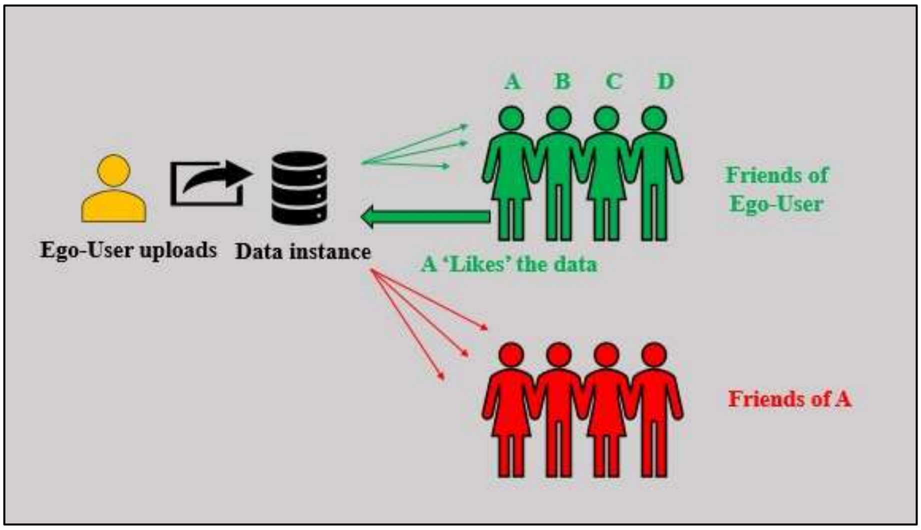


Fig. 1 - Data leakage of an OSN data instance due to a friend's activity

The model we present is composed of the following three main phases addressing three of its major aspects: trust, role-based access control and information flow.

Phase I: Trust

In the first phase, we assign trust values on the edges connecting direct friends to the ego node in their different roles, e.g., family, colleagues, etc. These trust values are calculated based on seven different parameters, as will be explained in section 3.2.

Phase II: Role-based access control

In the second phase we remove direct friends that do not have the minimal trust values required to grant a specific permission to their roles. A cascade removal is carried out in their ego networks as well. After this removal, the remaining user nodes and their edges are also assigned trust values. All removals in this phase are virtual, as will be explained in section 3.4.4.

Phase III: Information flow

In the last phase, we remove from the graph edges and nodes that are not directly connected to the ego user, by using different graph algorithms, to construct a privacy-preserving trusted network.

The result of these three phases is a new trusted network for the ego user with minimal information leakage. In the following sections we thoroughly explain and demonstrate the three phases of the model. The explanations are illustrated via the toy example in Fig. 2, representing a social network of a user denoted ego node. In the rest of this section, we introduce some basic definitions and concepts we use in our role- and trust-based access control (RTBAC) model. The evaluation is described in chapter 8.

Definition 1: An RTBAC instance is a tuple <*Ego*, *u_id*, *R*, *P(R)*, *UTV*, *MTV*> where:

Ego is the user identified as the ego node,
u_id identifies the user who is a candidate for accessing the ego node's information,
R is the role assigned to *u_id*, the same as in RBAC,
P(R) is a permission granted to a role R,
UTV denotes the trust value of the *Ego* user in *u_id*,
MTV is the minimal trust value for a permission *P* in role *R*, *P(R).*

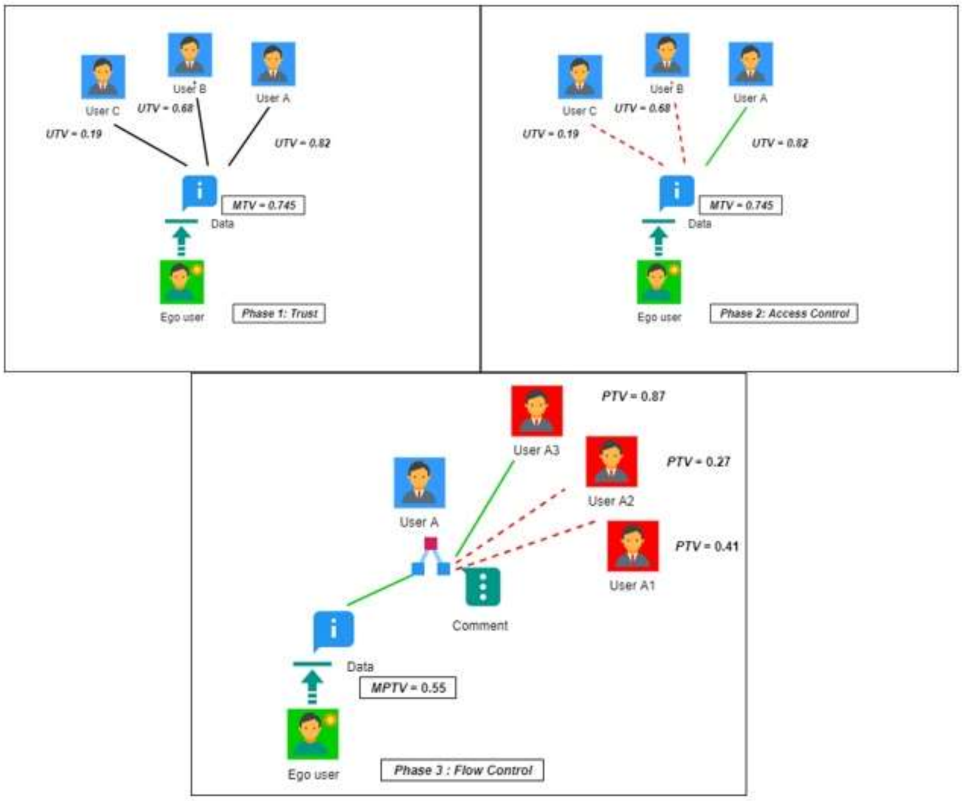


Fig. 2. The model's phases for creating a trustworthy network

3.2. First phase – assigning trust values

3.2.1. Trust parameters

Trust values in our model are calculated based on some of the measurable parameters suggested by (Misra & Such, 2016). We consider the following two major categories of these parameters.

Connection strength (*c*): connection strength between two user nodes is determined by characteristics that indicate their level of closeness such as friendship duration (*FD*), mutual friends (*MF*), etc. The full characteristics list and their notations are shown in Table 1.

User credibility (*u*): user credibility is determined by attributes that derive the user's OSN reputation and trustworthiness level, such as total number of friends (*TF*) and age of user account (*AUA*), calculated from the time the user joined the OSN, etc. The full list and notations are shown in Table 1.

The resemblance attributes (*RA*) that are taken into consideration are: gender, age (range), education, workplace, and relationship status (married, single, etc.).

Table 1 – Characteristic trust variables for the model.

Variable	**Attribute**	**User / Connection**
TF	Total number of Friends	User
AUA	Age of User Account (OSN seniority)	User
FFR	Followers/Followees Ratio	User
MF	Mutual Friends	Connection
FD	Friendship Duration	Connection
OIR	Outflow/Inflow Ratio	Connection
RA	Resemblance Attributes	Connection

We use the *u* prefix to denote user characteristics and the *c* prefix to denote connection characteristics (e.g., u_{AUA} stands for the value for the Age of User Account attribute, and c_{MF} stands for the value for the Mutual Friends attribute). A trust value ranges between 0 and 1 to reflect the probability of sharing information with a certain user: 0 represents total restriction, and 1 represents definite sharing willingness. The parameters that we use to calculate the values of the attributes were studied in an experiment that is discussed in the evaluation section of this paper.

The threshold values are denoted here as $T^{property}$ (e.g., for the *TF* attributes the threshold value is T^{TF}), and their experimental values are presented in the results section. We next provide a detailed explanation for the user credibility attributes and for the connection attributes.

User credibility attributes

User credibility attributes are generally defined for any individual node in the graph.

Total number of Friends: u_{TF} value is based on the Total Friends (*TF*) attribute (Dunbar, 2016), with respect to the average number of friends of fake profiles, social bots, etc. A profile of T^{TF} friends and above is a genuine user profile with a high probability.

$$u_{TF} = \begin{cases} \frac{TF}{T^{TF}} & (TF < T^{TF}), \\ 1 & (TF \geq T^{TF}). \end{cases} \qquad (1)$$

Age of User Account: u_{AUA} value is calculated in months based on the estimation of the Age of User Account (*AUA*) attribute (Zheng, Zeng, Chen, Yu & Rong, 2015), assuming that an active spammer profile will not remain active in the long term, due to OSN security updating policies.

A profile with a seniority of T^{AUA} months and above is a genuine user profile with a high probability.

$$u_{AUA} = \begin{cases} \frac{AUA}{T^{AUA}} & (AUA < T^{AUA}), \\ 1 & (AUA \geq T^{AUA}). \end{cases} \qquad (2)$$

Followers/Followees Ratio: u_{FFR} value is derived directly from the ratio between the number of people the user follows (Followees) and the number of people following the user (Followers).

This attribute reflects the fact that spammers and fake profiles follow more users and are usually less followed themselves.

$$uFFR = \begin{cases} FFR & (FFR < 1), \\ 1 & (FFR \geq 1). \end{cases} \qquad (3)$$

Connection strength attributes

Connection attributes are generally defined for any pair of connected nodes in the graph. In equations 4, 5 and 6 we use the pair *Ego* node and *uid*.

Resemblance Attributes: For the c_{RA} (Resemblance Attributes) value we take into consideration the following 10 user attributes: gender, age (range), current educational institute, past educational institute, current workplace, past workplace, current town, hometown, current country and home-country.

The c_{RA} value is calculated as the ratio between the number of non-null resembling attributes between the ego user and the other user, and the total number of non-null attributes of the ego user.

Let TA_{ego} be the total number of non-null attributes of the ego user and let $TRA_{ego,\ other}$ be the number of non-null resembling attributes of the ego user and the other user.

Then c_{RA} is defined as follows:

$$c_{RA} \qquad = \qquad \frac{TRAego,other}{TAego} \quad . \qquad\qquad (4)$$

This value cannot be larger than 1, since the number of common attributes is always less than or equal to the number of ego attributes. The IMPROVE model (Misra, Such & Balogun, 2016) shows that user-features' similarity can be a good estimation for sharing probability, since friends often have common ground.

While the Pearson correlation coefficient is often used for calculating this similarity (Benesty, Chen, Huang, & Cohen, 2009), it provides a symmetric value for both ends of the connection. Using the resemblance attributes ratio (Misra, Such & Balogun, 2016), we provide an asymmetric measure, where the user resemblance is evaluated with respect to the ego one, and not vice versa. In this work we preferred the resemblance attributes ratio over of the SimRank (Jeh & Widom, 2002), since the latter requires much more information of user profiles, which is not always available.

Mutual Friends: c_{MF} value is based on the *MF* attributes. Fake profiles, social bots, or even adversaries have a small number of mutual friends, if any. A profile of T^{MF}mutual friends and above is a close friend with a high probability.

$$c_{MF} = \begin{cases} \frac{MF}{T^{MF}} & (MF < T^{MF}), \\ 1 & (MF \geq T^{MF}). \end{cases} \quad (5)$$

Friendship Duration: c_{FD} value is calculated in months based on the Friendship Duration (*FD*) attribute. The ego user is less likely to share with a relatively new friend, see also (Wiese, Kelley, Cranor, Dabbish, Hong & Zimmerman, 2011).

A friendship of T^{FD}months and above is most likely to be a genuine connection.

$$c_{FD} = \begin{cases} \frac{FD}{T^{FD}} & (FD < T^{FD}), \\ 1 & (FD \geq T^{FD}). \end{cases} \quad (6)$$

Outflow/Inflow Ratio: We define outflow (inflow) as the number of interactions between the source (target) node and the target (source) node. The attribute of c_{OIR} describes the ratio of Outflow/Inflow (Ranjbar & Maheswaran, 2014). It is an important attribute, since unwanted users or even spammers and fake profiles create towards the user much more inflow actions (advertisement, data-harvesting, etc.) than outflow actions created by the user itself (sometimes this information is not available; therefore, in most of our experimental evaluation we did not use this feature).

$$c_{OIR} = \begin{cases} Outflow/Inflow & (Outflow < Inflow), \\ 1 & (Outflow \geq Inflow). \end{cases} \quad (7)$$

To support the access permission decisions, we compute the values of user credibility and connection strength as described above.

User credibility (*u*) is calculated as a weighted average of all user credibility attributes, and connection strength (*c*) is calculated as a weighted average of all connection attributes. The weights are set according to the significance of every attribute, as inferred from an empirical survey we conducted (see the results section).

$$u = \langle WiUi \rangle = \frac{\sum_{i=1}^{|u|} WiUi}{\langle W \rangle |u|} \quad , \quad (8)$$

$$c = \langle WiCi \rangle = \frac{\sum_{i=1}^{|c|} WiCi}{\langle W \rangle |c|} \quad . \quad (9)$$

User Trust Value (*UTV*) is calculated as the weighted average of user credibility and connection strength. The weight is set according to the relative number of attributes in each category (in this model there are 7 attributes: 4 connection attributes and 3 user attributes):

$$UTV = \frac{c\cdot|c|+u\cdot|u|}{|c+u|} \quad . \qquad (10)$$

Based on this trust measure we will define trust-based permissions, as detailed in the next subsection.

In Table 2 we can see an example, portrayed in Fig. 2, where there is a difference between two users (C and B) that have the same role, but not the same *UTV*, thus not getting the same permission.

Table 2 – Difference in *UTV* between same-role users.

User	*uTF*	*uAUA*	*uFFR*	*cRA*	*cFD*	*cOIR*	*cMF*	***u***	***c***	***UTV***	***MTV***
C	0.44	0.33	0.89	0.4	0.67	0.13	0.22	**0.55**	**0.36**	**0.44**	**0.745**
B	0.78	0.59	0.91	0.8	0.86	0.96	1	**0.76**	**0.91**	**0.84**	**0.745**

The *MTV* set for this specific role and permission (e.g., the family role and tagging permission) is 0.745. User C achieves a *UTV* value of 0.44 and does not get the permission, whilst User B achieves a *UTV* of 0.84, thus gets the permission, as seen in the figure.

3.2.2. Dealing with a strict threshold for the parameters

As mentioned in the previous section, the numerical aspect of the threshold values could be different if we take into consideration different attribute weights, different aspects of the network, or other experimental results. Therefore, we handle the uncertainty of these threshold values with a Certainty Factor (*CF*), as presented in (Ravi, 2016). We derive the access-granting decision subject to the threshold values, with levels of certainty that may require user approval in borderline cases.

The ranges of these values are set by the standard error of the mean (SEM - $\sigma_{T^{property}}$), that is derived as the standard deviation (σ) of the threshold values as follows:

$$\sigma_{T^{property}} = \frac{\sigma}{\sqrt{n}} = \frac{\sqrt{\frac{1}{n}\sum_{n}^{1}(T_i^{property} - T^{property})^2}}{\sqrt{n}} \, . \; (11)$$

σ is set for each property differently, with $T^{property}$ being the average value of all thresholds from the experimental evaluation ($T_i^{property}$). These property values are shown in the experimental evaluation.

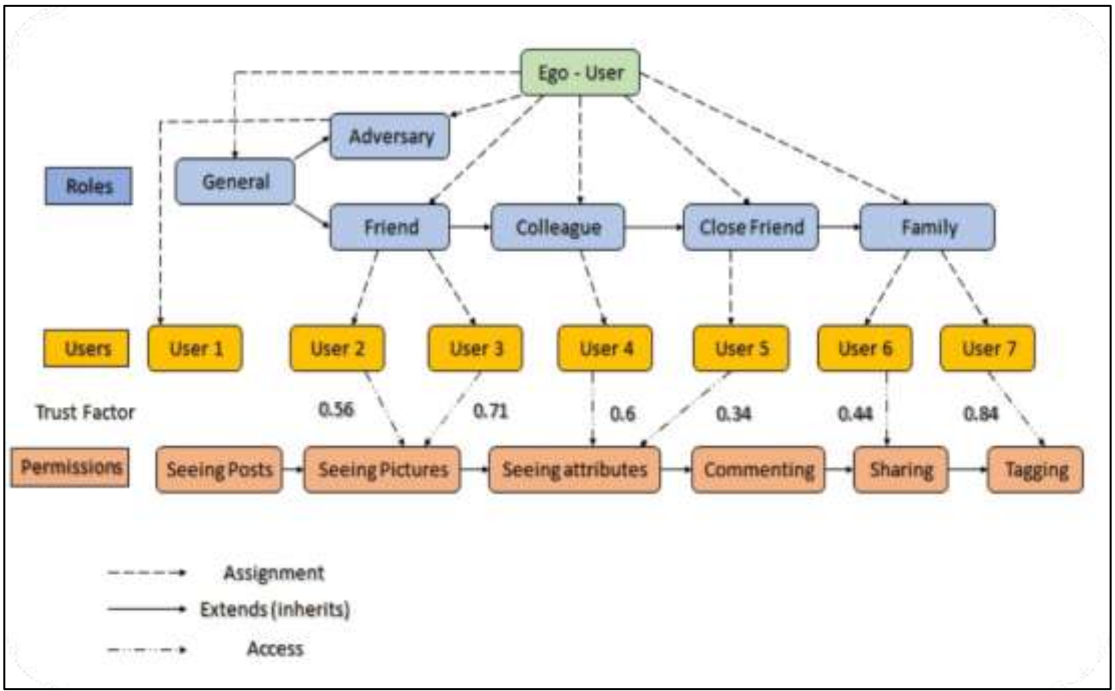
Now, we can set the permission decision (*P*) according to the needed certainty level *(CF)* of the specific privacy preferences. When the uncertainty is in the grey area (in the range of the standard error of the mean (SEM - $\sigma_{T^{property}}$)), we ask for user approval; a certainty level above this will allow access, and below this will deny it.

$$P(T^{propert}) = \begin{cases} Allow\,, & (CF\,(T^{property}) \geq T^{property} + \frac{\sigma_{T^{property}}}{2}), \\ Deny, & \left(CF\,(T^{property}) \leq T^{property} - \frac{\sigma_{T^{property}}}{2}\right),. \\ User\ approval, & \left(T^{property} - \frac{\sigma_{T^{property}}}{2} < CF\,(T^{property}) < T^{property} + \frac{\sigma_{T^{property}}}{2}\right) \end{cases} \qquad (12)$$

3.3. Second phase – access control for direct friends

The model we propose extends the basic RBAC model to allow flexible and fine grained access control. In addition to roles, each user is assigned a level of trust, and permissions to different data instances are granted only if the trust level is above a certain threshold. The inclusion of trust in the access control model has three major purposes: it provides an additional stage of screening on top of the RBAC roles, it enables better control of data distribution, and it allows dynamic granting of permission over time, reflecting the dynamic nature of OSN and the possible changes of users' trust in each other.

In this phase we are concerned only with direct friends of the ego node and their access capabilities. The second part of Fig. 2 illustrates this decision in this second stage where User A and its network are disconnected since A is not in the family role



(meaning *P(R)* =0), and User C, who has the family role, but is disconnected along with its network since it did not achieve the required *MTV* of 0.745, as explained above. Algorithm 1 depicts the access-decision phase and Fig. 3 demonstrates the algorithm in a Facebook-like network. The ego user in the figure is the user sharing the information, while the other 7 users are assigned different roles with respect to the ego user. Assume that in this example, we define an *MTV* of 0.745 for a family member role to access the permission of 'tagging'. Both users 6 and 7 have a family role, but only User 7 achieves a trust value bigger than 0.745 and gets the tagging permission that is denied from User 6, but does have the sharing permission.

```
Algorithm 1. GrantPermission
Input: Minimal Trust value: MTV, User U, Role R, Permission P
Output: granted or denied
    if $P \in R$
        if $U.UTV \geq P.MTV$
            return granted
        else
            return denied
    else
        return denied
```

Fig.3 Access decision of the model

3.3.1. Partial access for data anonymization

Our model's algorithm enables complete access of information to highly trusted users, or blocks it completely to undesirable ones. In this section we suggest an extension of the model such that the information access is generalized or anonymized based on the user's trust level and distance from the ego user.

We demonstrate this idea using image anonymization, but it can also be applied to text, profile attributes and other information instances, similarly. The main idea of the model's extension of partial access is that a certain instance of data is not fully seen or unseen but can be partially scaled in its appearance. This option gives wider information access, with the benefit of secure data anonymity. In image anonymization, this feature helps to reduce data leakage from facial recognition algorithms, which are vastly used in OSN and other web applications.

In Fig.4 we can see the manifestation of such a partial access, where the ego user's profile picture is anonymized in the access-granting step. For the given scenario we assume the value of 0.7 as the *MTV*, and the permission handled is the 'visible pictures' that User 2 and User 3 obtain.

User 1 does not see the image at all (left part of Fig.4) since it does not have a fitting role (he is 'general' and the relevant role is 'acquaintance'). User 2 has a fitting role, but has a *UTV* of 0.56; hence, he gets a more blurred image (middle part of Fig.4) than User 3 obtains. User 3 has the fitting role and the needed trust value (*UTV*=0.71); thus, he gets the full image (right part of Fig.4). It is important to state here that this extension of the model is only relevant to those permissions that logically enable partial access. Permissions of 'sharing' or 'tagging' are binary by nature, and hence cannot comply with a partial access model.

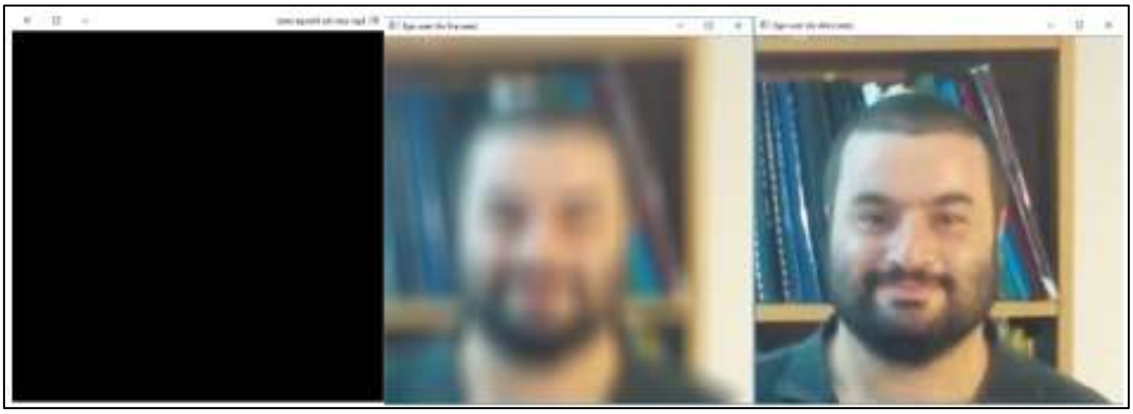


Fig.4 Visibility of profile picture as seen by the users of Fig.2, with a threshold *MTV* of 0.7
Left: User 1, Middle: User 2, Right: User 3

3.4. Third phase – flow control for the more distant network

3.4.1. The trust assignments of the remaining network

After the removal of the unauthorized ego user's direct friends and their networks, the remaining graph consists of all authorized friends and their networks. In this phase we take only four attributes into consideration, due to the limited evaluation ability of users (nodes) that are not directly connected to the ego nodes. The attributes we consider are: *TF*, *AUA*, *MF* and *FD*.

In a real OSN, some of the other attributes are anonymized (like resemblance attributes) or unmeasurable (like Followers/Followees Ratio).

Let *G = (V, E)* be an undirected graph that describes the OSN, where V represents the set of users and E represents the set of social connections between them. We define the user credibility and connection strength attributes as follows.

Definition 2: User credibility attribute

For user $v_i \in V$, the credibility attribute v_i.u is defined as the average of the two user's attributes: v_i.u $= \langle uTFi, uAUAi \rangle$.

Definition 3: Connection strength attribute

For a social connection $e_i \in E$, the connection strength attribute e_i.c is defined as the average of the two connection attributes: $ei.\mathrm{c} = \langle cMFi, cFDi \rangle$.

In Fig.5 we can see an OSN graph in its preliminary form with its raw attribute values (for *TF*, *AUA*, *MF* and *FD*), before calculations. In this graph there are 4 networks of the ego node's fiends (Alice, Bob, Charlie and David), from which we demonstrate the graph algorithms in the following subsection. These networks will be used in the evaluation section as well. For the access control model Alice and Bob have the *Family* role while the rest of ego's friends have the *Colleague* role. Charlie and David will be omitted from this network, due to not having a proper role (*P(R*) = 0), since the permission is just for the *Family* role. Bob, who has the family role will be excluded, due to not having a sufficient level of trust value (*UTV*< *MTV*). Alice's network will therefore be the only one remaining, and in the following subsection her network will go through the third phase of the trust model.

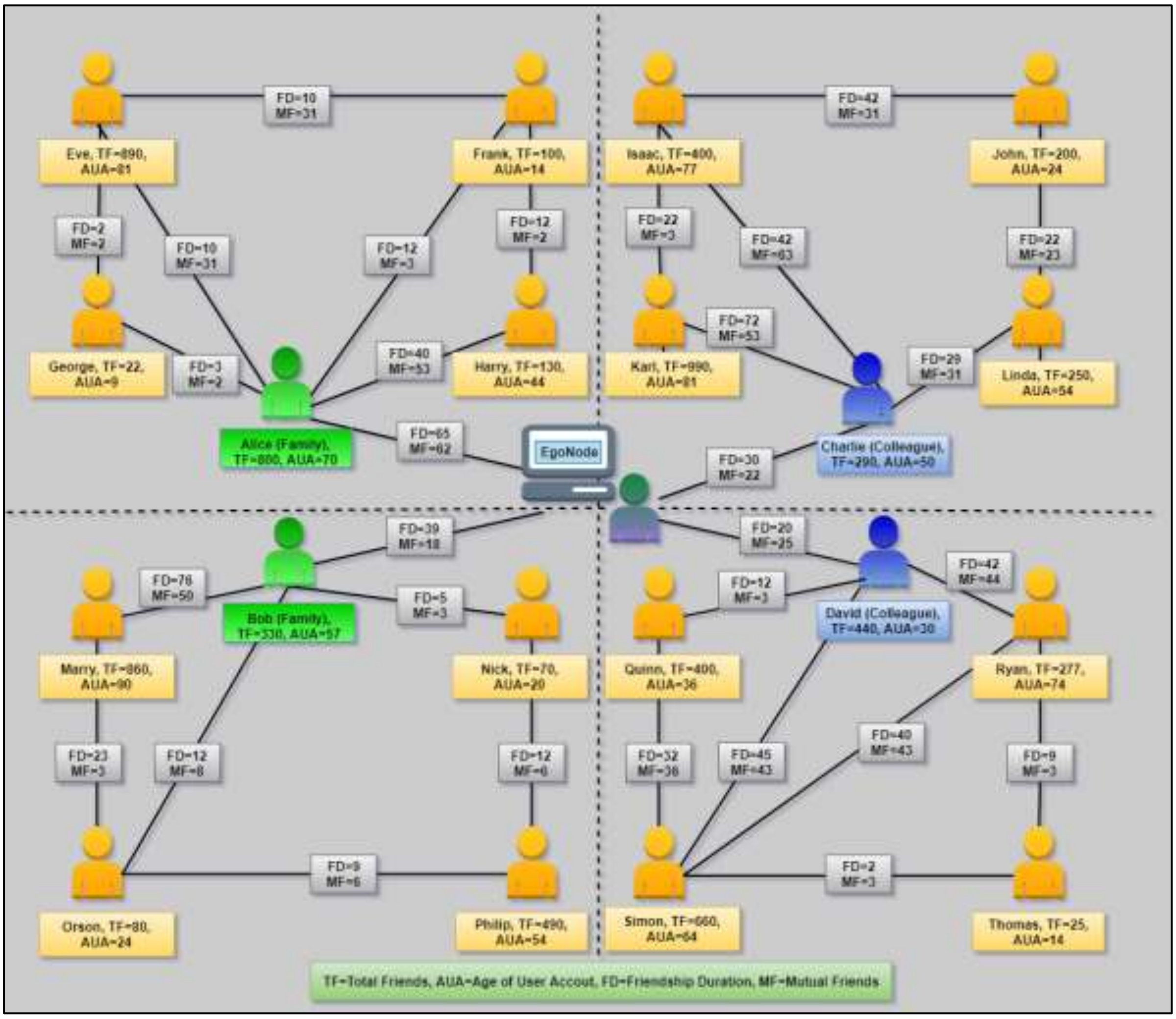


Fig. 5 – OSN graph, in its preliminary form with its raw attribute values

For the third phase of the model, we need to assess the vulnerability of the remaining friend networks, and to deal with it by creating secure information flow. In this phase the network of a direct friend is exposed to the ego user's data only if that direct friend acts on the data, e.g., by sharing a post, liking a picture, etc. While in the access control phase, users are explicitly denied access to the ego user's data, in this phase we attempt to prevent a possible leakage caused by a friend's activity. For this final phase we have applied two information-flow methods. The first method emphasizes the friends' networks; it creates a secure network by using the MST algorithm.

The second method puts more emphasis on finding potential risks; it identifies possible adversaries by calculating the trust on all the paths from a source node to a target one. A comparison between the methods is provided in the experimental evaluation.

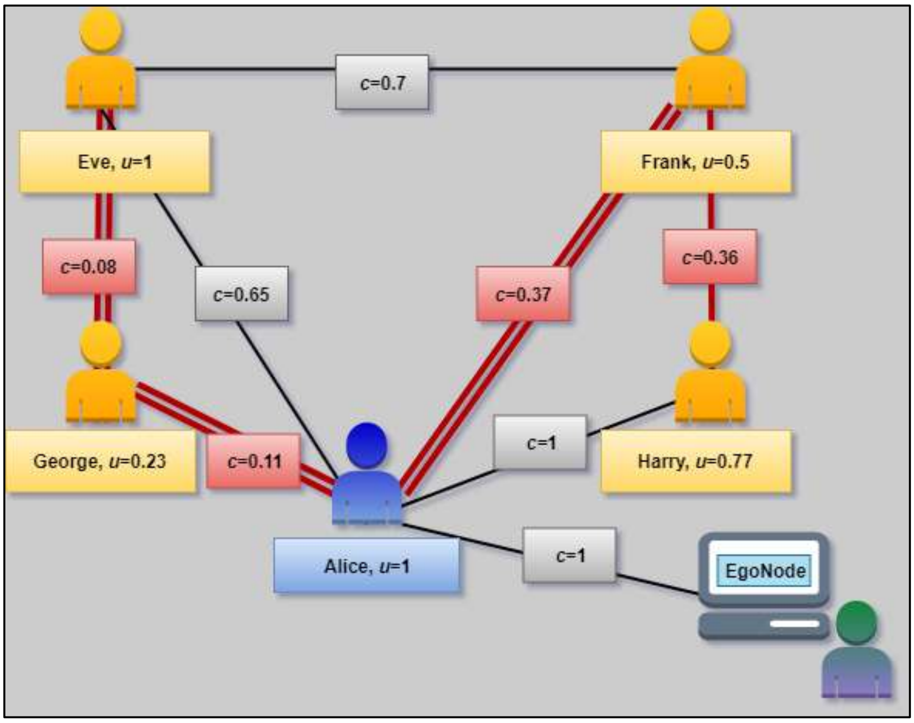


Fig. 6 – Creating the MST in the ego user's sub-graph by the evaluation of trust attributes

Table 3 – Calculation of trust attributes for all users on Alice's network.

User/Connection	Attribute calculation
Eve	$u = \langle uTF, uAUA \rangle = \langle 1{,}1 \rangle = 1$
George	$u = \langle uTF, uAUA \rangle = \langle \frac{22}{245}, \frac{9}{24} \rangle = 0.23$
Frank	$u = \langle uTF, uAUA \rangle = \langle \frac{100}{245}, \frac{14}{24} \rangle = 0.5$
Harry	$u = \langle uTF, uAUA \rangle = \langle \frac{130}{245}, 1 \rangle = 0.77$
Alice	$u = \langle uTF, uAUA \rangle = \langle 1{,}1 \rangle = 1$
Eve – George	$c = \langle cMF, cFD \rangle = \langle \frac{2}{37}, \frac{2}{18} \rangle = 0.08$
Eve – Frank	$c = \langle cMF, cFD \rangle = \langle \frac{31}{37}, \frac{10}{18} \rangle = 0.7$
Eve – Alice	$c = \langle cMF, cFD \rangle = \langle \frac{11}{37}, 1 \rangle = 0.65$
George – Alice	$c = \langle cMF, cFD \rangle = \langle \frac{2}{37}, \frac{3}{18} \rangle = 0.11$
Frank – Alice	$c = \langle cMF, cFD \rangle = \langle \frac{3}{37}, \frac{12}{18} \rangle = 0.37$
Frank – Harry	$c = \langle cMF, cFD \rangle = \langle \frac{2}{37}, \frac{12}{18} \rangle = 0.36$
Harry – Alice	$c = \langle cMF, cFD \rangle = \langle 1, 1 \rangle = 1$
Ego – Alice	$c = \langle cMF, cFD \rangle = \langle 1, 1 \rangle = 1$
Eve	$u = \langle uTF, uAUA \rangle = \langle 1{,}1 \rangle = 1$

We demonstrate the calculations of trust of all paths from the source node to the target node on the upper left part of the ego user's network in Fig. 5, which is Alice's network. The values of the trust attribute calculations are shown in Table 3 and are the basis for the manifestation of the MST algorithm (see Fig. 6, where the double red line is the MST edges). In general, we use a simple average for the attribute calculations, although a weighted one can also be applied to comply with any OSN policy or preferences.

3.4.2. Creating a trustworthy network by using a MST algorithm

Our goal is to construct a strongly trusted subnetwork for the ego user by removing the weakest links of trust and leaving the network connected through the strongest edges. We can iterate this process as needed until we reach the desired level of trust. Finding the MST of a graph is a well-known problem that has been dealt with in different aspects and applications, and by many efficient algorithms, as presented comprehensively in (Gabow, Galil, Spencer & Tarjan, 1986).

One of the best-known algorithms is Kruskal's algorithm (Kruskal, 1956), which can be summarized as follows:

Given an undirected connected weighted graph G = (V, E), let S=E be the set of all edges removed from the graph, sorted by their weights.

While S is non-empty, and G is not yet spanning:

Remove an edge e ∈ S with the minimum weight from S and add it to E.

If the removed edge e does not create a circle, put it in G.

The result of the algorithm is an MST of the graph. The MST of the OSN graph is the weakest (thus, insecure) connected subgraph of the OSN graph. Next, Algorithm 2 constructs the trustworthy network out of the original graph by disconnecting the weak edges and vertices from the OSN graph. Removing the edges of the MST implies removing the weakest links of trust from the network, leaving the graph connected through the strongest edges.

Algorithm 2 constructs the trustworthy network out of the original graph. For each ego user's friend, it creates the friend's network graph (as illustrated in Fig. 5 with 4 friends) and a matching MST using the Kruskal algorithm (as demonstrated in Fig. 6 for Alice's network).

The MST is the weak (untrusted) part of the network, and therefore its edges are considered for removal one by one. An edge is removed if it does not disconnect any vertex or if it does disconnect a vertex, but the trust value of the user represented by the vertex is lower than the predefined *MTV* threshold. This step is depicted for Alice's network in Fig. 7, where the user George is first disconnected and then removed since $u(v_{George}) < MTV$. After applying Algorithm 2 for disconnecting the weak edges and vertices from the OSN graph, we get an information-flow trust network, where the information can be freely shared. We can assess the overall network's trust value by averaging the trust value of all the edges and nodes. For a higher level of trust, we can apply the algorithm iteratively until a desired threshold is reached. An advantage of this algorithm is that highly connected vertices that do not necessarily have a high *MF* attribute will remain in this trust network. This phase, as well as the other phases, is scalable to networks with friends of third degree (a friend of a friend of a friend). The scale of the problem, when extended on networks beyond that degree, makes these calculations not feasible. An example of the algorithm's implementation for the upper left part of the ego user's graph of Fig. 5 (Alice's network) is provided in Figs. 6 and 7: Fig. 6 presents the MST before the removal and Fig. 7 presents the MST after the removal, where the user George has been disconnected and then removed since $u(V_{George}) = 0.23$ while $MTV= 0.5$. If we run Algorithm 2 again, all the remaining edges in Fig.7 are candidates for removal, but only Frank will be disconnected due to his low trust value. As we can see in this figure, the link between Alice and Frank has been removed, yet Frank belongs to the trusted network due to his connection to Eve. This means that he will have access to the data if Eve acts on it, due to the strong trust values of both Alice-Eve and Eve-Frank. This is a relatively common situation: Frank is a close friend of Eve, and not very acquainted with Alice (e.g., Eve introduced Frank to Alice a few days ago), so the direct connection is not yet strong, but Alice trusts Eve and Eve trusts Frank and together, using the transitive attribute of trust, they construct a strong path.

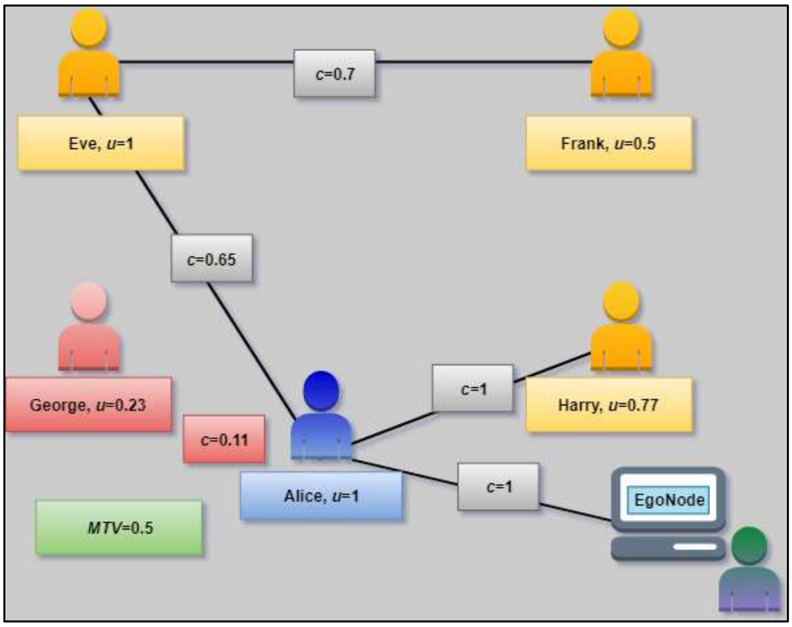


Fig. 7 – Removing the unsecure node that is disconnected by the algorithm, due to its having a trust value lower than the minimal threshold

This means that Frank belongs to the network only since we found a path from Alice to Frank through Eve that expresses a chain of trust, and thus justifies the presence of Frank in the trusted network.

Algorithm 2. ConstructTrustworthyNetwork

Input: $G = (V, E)$ OSN Graph, Ego-user, *MTV* - Minimal Trust Value
Output: Trustworthy Network Graph

```
For i=1..sizeof(Ego-user.friendsList)
   $G_i$ = G.subNetwork( i)
   $G\_MST_i = G_i$ .kruskalMST()
   For j=1..sizeof($G\_MST_{i.}$.EdgeList)
      if not isDisconnectingEdge(Gi, $e_j$)
         $G_{i.}$.removeEdge($e_j$)
      else
        $v_j = e_{j.}$endPointVertex
        if ($v_i$.u < MTV)
          $G_{i.}$.removeEdge($e_j$)
 Return G
```

3.4.3. Identifying possible adversaries in a user's network

The second alternative for information-flow control is to identify possible adversaries in the user's network, as we explain next.

Let $G = (V, E)$ be an undirected graph that describes the OSN, where V represents the set of users and E represents the set of social connections between them. $v_{src} \in V$ is the ego source node, that holds the information to be shared, and $v_{trgt} \in V$ is the target node, which may or may not get the information from v_{src}.

Definition 4: A *Path* from a source to a target node, denoted $PATH^{src \rightarrow trgt}$, is a set $\{v_{src}, E^{src \rightarrow 1}, v_{1,} E^{1 \rightarrow 2}, \ldots.. v_k, E^{k \rightarrow trgt}, v_{trgt}\}$, where the number of intertwined user-nodes is k.

For example, in Fig. 8 we can see that there are 3 possible paths from ego to User E: *Ego → C → E*, *Ego → D → E*, and *Ego → B → E*. In these 3 paths $k=1$, since there is a single node between the source (*Ego*) and the target (User E).

There are several efficient algorithms for finding all possible paths between the source and the target nodes, such as the Ford-Fulkerson algorithm (Ford & Fulkerson, 1956), the Edmonds–Karp algorithm (Edmonds & Karp, 1972) for computing the maximum flow in a flow network $O\ (V E^2)$, and the Dinic algorithm (Dinic, 1970), which performs even better with time complexity of $O\ (V^2 E)$. To determine whether an information instance will be passed from an ego node to a target node, we seek all possible paths, which is the same problem covered by the Dinic algorithm. We calculate the trust value of a path $PATH^{src \rightarrow trgt}$ defined by $\{ v_{src}, E^{src \rightarrow 1}, v_{1,}\ E^{1 \rightarrow 2}, \ldots.., E^{k \rightarrow trgt}, v_{trgt} \}$ by multiplying the trust values of each node and each edge on the path, where the trust of node v_{i} is determined by user-credibility attributes ui and the trust of edge i $E^{i\text{-}1 \rightarrow i}$ is determined by connection attributes ci:

$$PTV(PATH^{src \rightarrow trgt}) = \prod_{i=1}^{k} ci \cdot ui. \qquad (13)$$

The trust of the source node v_{src} (indexed in the formula as v_0) is omitted since the ego user does not need to be checked for credibility. Table 4 presents sample values assigned to the nodes and edges of the graph shown in Fig. 8, with User E as an adversary target node and user F as an acquaintance target node.

Table 4 – User and Connection attribute variable values for the graph in Fig.7.

User/Connection	***u***	***c***
B	**0.78**	-
C	**0.68**	-
D	**0.97**	-
E	**0.56**	-
F	**0.95**	-
Ego→B	**-**	**0.58**
Ego→C	**-**	**0.58**
Ego→D	**-**	**0.83**
B→E	**-**	**0.29**
D→E	**-**	**0.74**
C→E	**-**	**0.23**
D→F	**-**	**0.91**
C→F	**-**	**0.68**

The values of *u* and *c* are calculated as described in the previous subsection (for *MF*, *TF*, *AUA* and *FD*). Table 5 shows the *PTV (*Path Trust Value*)* calculation (equation 13) of all the *PATH*s in the graph, using the values of Table 4. In order to decide whether the target node is an acquaintance or an adversary, we first set a numerical threshold value for *PATH*, thus including the decision of information sharing by defining the *PATH* as being safe or not safe. This threshold defines the Minimum Path Trust Value (*MPTV)*. The value in the example of Fig. 8 and Tables 4 and 5 is 0.5, where *PTV* ≥ 0.5 means that the *PATH* is safe, and that the target node is necessarily an acquaintance, not an adversary. The acquaintance identification algorithm is described in Algorithm 3.

Table 5 - *PTV* values for all the *PATH*s of the graph.

PATH	*PTV*
Ego→B→E	0.58*0.78*0.29*0.56 = **0.07**
Ego →C→E	0.58*0.68*0.23*0.56 = **0.05**
Ego →D→E	0.83*0.97*0.74*0.56 = **0.33**
Ego →D→F	0.83*0.97*0.91*0.95 = **0.7**
Ego →C→F	0.58*0.68*0.68*0.95 = **0.25**

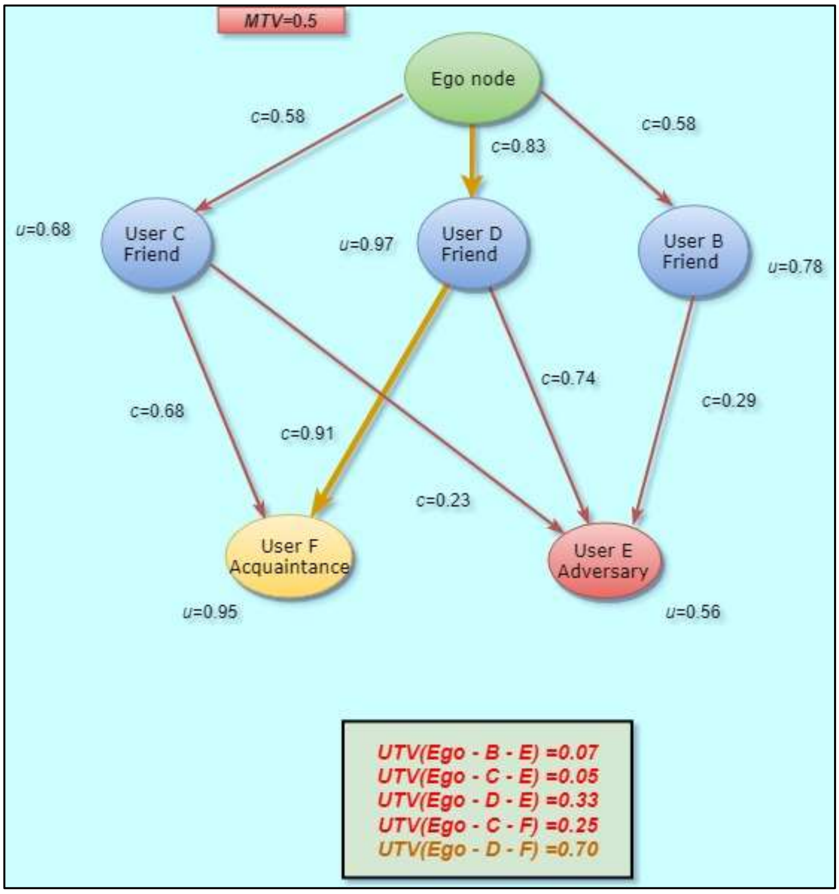


Fig. 8: Graph instance of six OSN users, with vertex and edge values, where the outflow checked is from Ego to Users E and F

Algorithm 3. ***isAcquaintance***

Input: Graph *G*, Vertex v_{src}, Vertex v_{srgt}, *MPTV -trust value threshold*

Output: true if the target user is identified as an acquaintance

```
$AllPaths^{src \rightarrow trgt}$ ← $GetAllPathsByDinic\ (G, v_{src}, v_{srgt})$
For each path in ($AllPaths^{src \rightarrow trgt}$)
    if $PTV(path) \geq MPTV$
        return true
return false
```

Algorithm 3 starts by finding all possible paths between the source and target nodes by applying the Dinic algorithm for a maximum flow network. Next, it calculates the *PTV* for each path, and if the *PTV* of at least one path is higher than the threshold, it returns *true* indicating that the node is an acquaintance.

If no path has a sufficient trust value, it returns *false*, indicating that the target node is an adversary. Once an adversary is identified, a blocking algorithm should be applied, e.g., (Levy, Gudes, & Gal-Oz, 2016). Since *PTV* is calculated using a predefined set of user attributes and connection attributes, this calculation is of constant complexity, and the algorithm complexity is $O(V^2 E)$.

Figure 8 demonstrates the detection of an adversary (User E) and an acquaintance (User F). None of the *PATH*s from *Ego* to *E* have enough trust (PTV) value to satisfy the *MPTV*; therefore, *E* is considered an adversary. The trust value of *PATH A – D - F* satisfies the *MPTV*; therefore, *F* is an acquaintance. When a sufficient value of *PT*V is found on one of the paths from the source to the target node, the target node is declared as an acquaintance with whom we are willing to share information. In chapter 8 we evaluate and compare the two algorithms.

3.4.4. Implementation issues in a real OSN

The implementation of the algorithms we have described in this section does not involve permanent disconnection of users in the network; the network graph edges are cut just in order to obtain the resultant list of users that can access the data instance. This trusted network can be saved for each user within the administrative control of the OSN. To address the dynamic nature of an active OSN (e.g., friends are added or removed, *AUA* increases) the algorithm can be executed periodically, as defined by the user or by the OSN administrator. During the runtime, every sharing request is checked against the list of trusted network users in order to decide which node can be exposed to this data. Although social networks may contain millions of nodes and tens of millions of edges, the ego node's trustworthy network is relatively small, having all users at a distance of two or three hops from the ego node. The limited number of nodes and edges involves makes the information-flow algorithms feasible in terms of runtime. The algorithm can be executed in any number of iterations to suit the desired threshold value. Another important aspect of real OSNs is the possible privacy violation that occurs when a user is denied access to a data instance accessed by his connections. This violation is prevented in our model since any action that an authorized user performs on a data item (e.g., sharing or transferring) is visible only to authorized friends (the ones that were screened by the model).

The values of *MTV* and *MPTV* may differ from one network to another. New networks (ego users that have just joined the OSN) tend to have low trust values, so the *MPTV* and *MTV*, may be adjusted accordingly. Strongly based networks tend to have higher trust values, and, accordingly, a higher *MPTV* and *MTV* may be set so that the screening could be more rigid in terms of trust. We suggest defining the *MTV* default value as the average of all of the *UTV*s of friends in the ego network, and, accordingly, the *MPTV* default value as the average of all of the *PTV*s in the ego network.

However, the actual decision is a matter of policy. It can be set as a default value by an OSN administrator and overridden by an ego user under her/his own privacy policy.

4. Robustness of the model – analyzing attacks

4.1. Attack definitions and scenarios

An attempt to examine the vulnerability of our trust-based comprehensive model led us to question the strength of the trust attributes that are used to determine the levels of trustworthiness in the ego user's network. To gain a high user trust level (*UTV*) an attacker must fake all the values of the relevant attributes required to build this trust level. In this section we consider possible attacks on these attributes and analyze the feasibility of such attacks and show that creating fake users with high trust value is very difficult.

To create a fake user that appears genuine, an attacker should make sure that the user is connected to other users. An attack on the model is the creation of a set of fake users such that each fake user has its own ego network. The success of an attack depends on the network of the fake users, so usually it would be a collaborating network of fake users. To formalize this attack, we provide the following definition:

Definition 5: An attack is a tuple of the form $<G, T^{TF}, T^{AUA}, G^{spm}, t_{spm}>$

where:

T^{TF} is the *Total Friends* threshold value of the ego user network.

T^{AUA} is the *Age of User Account* threshold value of the ego user network.

G – is the graph of the ego user that is under attack.

G^{spm} (V^{spm}, E^{spm}) – is the spammer graph that is created in the attack.

t_{spm} – is the elapsed time before the attack can take place.

The result of the attack is denoted:

$G^{\psi} = G \cup G^{spm}$ – the spammed graph after the attack.

For an attack to take place, the model's major trust attributes must be faked: *MF, TF, AUA and FD*. We divide these attributes into two groups: attributes representing connecting quantities (*MF* and *TF*), and attributes representing durations *(AUA* and *FD*).

Connecting quantities imply that a user is well connected, and a user that has enough mutual friends with others demonstrates human circles of relations within an OSN (family, work, neighborhood, etc.). Duration attributes represent the steadiness of the profile, as genuine users usually create their profile once. To fake a user attribute such as *MF* or *TF,* an adversary must connect the fake user profile to other profiles in the network, genuine or not. The minimal number of fake users to be created must exceed the threshold of every attribute. To impersonate a real user network, an attack must consist of a network of trustworthy users who need to adhere to all the model's properties. We consider the extreme scenario of spammers that are only friends with each other, making the *MF* property as similar as possible to the *TF* property. This attack simulates a closed spammer network G^{spm} (V^{spm}, E^{spm}) that is a clique; therefore, every node (user) is connected to another node in the graph.

In this type of attack the *MF* attribute is correctly faked, since all the users are connected to each other. As all the nodes are connected in the spammer clique the size of the spammer graph must be at least:

$$|V^{spm}| \geq T^{TF} \quad . \qquad (14)$$

For the duration attributes, *AUA* and *FD*, we also consider the extreme scenario of spammers that are only friends with each other, making the *FD* property as similar as possible to the *AUA* property. These properties must also hold for all the users in the spammer's network. This is specifically hard, due to OSN policies that require a reasonable duration for a user account to be considered a genuine one (Zheng, Zeng, Chen, Yu & Rong, 2015). This means that before the attack can take place the elapsed time should be:

$$t_{spm} \geq T^{AUA} \quad . \qquad (15)$$

This attack process is shown in Fig. 9, where the two properties are created. The creation of a spammer network (specifically in OSN) for malicious purposes is described in (Shrivastava, Majumder & Rastogi, 2008), where the following attack is described: a malicious user that creates a set of false identities and uses them to communicate with a large, random set of innocent users (Random Link Attack -RLA). The research shows and proves that this is, in fact, an NP-complete problem.

Practically, it means that this kind of attack, carried out naively without heuristics, is very hard to perform.

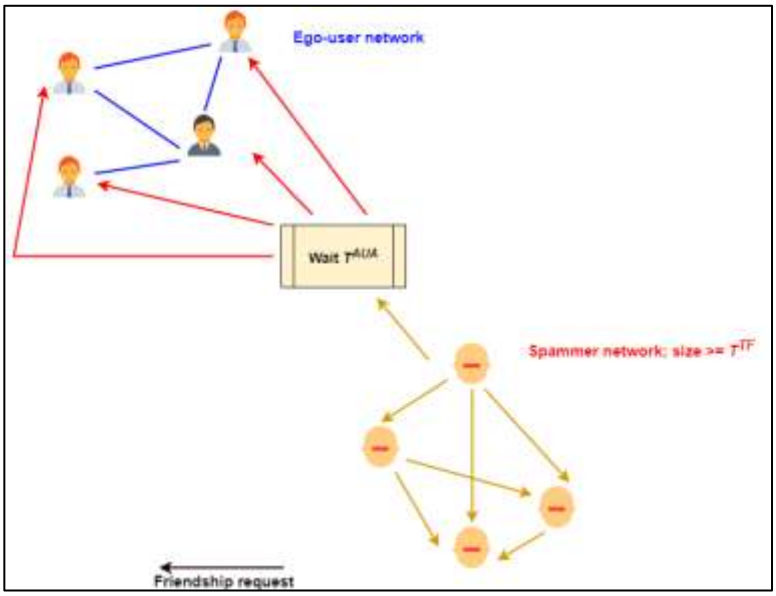


Fig. 9: The attack of a spammer network on the Ego network

We extend this form of basic attack one step further as we take into consideration the attributes of these nodes, making the attack even more difficult to implement.

To perform an efficient attack, we need to assume that some of the requests of the spammer network will be denied or blocked by the OSN administration; thus, the attack has to involve as many friends as possible from the ego network. The robustness of our model is derived from its resilience to these attacks in term of the actual OSN size - the bigger the network is, the harder it is to fake the attributes of the model.

We now define four types of attack, based on their strength and complexity.

A regular attack: This is a blackbox attack that does not include preliminary knowledge on the ego user network. In this attack the spammer network tries to connect to k direct friends in the ego network, where $1 \leq k \leq \frac{T^{TF}}{2}$ (the reason k is usually greater than 1 is that a single spammer network connected to a single direct friend may be detected quite easily by the network administrator).

In this attack, the number of friend requests to be made is the number of edges from the spammed network, that is $|E^{spm}|$.

since it is a clique, $|E^{spm}| = \frac{|V^{spm}|\,(|V^{spm}|-1)}{2}$, and the size of the connected network is:

$$|E^{\psi}| = \frac{|V^{spm}|\,(|V^{spm}|-1)}{2} + k\cdot\ T^{TF} = \frac{T^{TF}\,(T^{TF}-1)}{2} + k\cdot\ T^{TF}.\ (16)$$

Since usually *MF* is much smaller than *TF*, even if many of the friend requests are denied, *MF* will be fulfilled. We therefore use the extreme case where *MF = TF*.

A strong attack: This attack is also a Blackbox attack, in which the spammer network will try to connect to all the direct friends of the ego network. In this case, the number of friend requests to be made, which are the number of edges from the spammed network, is:

$$|E^{\psi}| = \frac{|V^{spm}|\,(|V^{spm}|-1)}{2} + (T^{TF})^2 = \frac{T^{TF}\,(T^{TF}-1)}{2} + (T^{TF})^2\ .\ (17)$$

A very strong attack: This is a knowledge-based Whitebox attack, which includes the pre-requisite of being familiar with the ego network structure.

In this attack the spammer network will try to connect to all the friends in the network within a distance *d* from the ego user. In this case, the number of friend requests that should be made, which are the number of edges from the spammed network, is:

$$|E^{\psi}| = \frac{|V^{spm}|\,(|V^{spm}|-1)}{2} + (T^{TF})^d = \frac{T^{TF}\,(T^{TF}-1)}{2} + (T^{TF})^d\ .\ (18)$$

An optimized very strong attack: an attack that uses an optimization algorithm to conduct an efficient attack. In the upcoming subsection, we describe a minimization heuristic that a smart spammer would perform, but as we show this problem is still NP-complete. The complexity of the problem of creating a fake friends' network becomes harder as the attack strength grows, and therefore it is not viable in terms of OSN sizes of user networks.

4.2. Optimizing the attack: minimizing the connections of fake users by reduction from minimum vertex cover

An attempt of a spammer's network to reach out to the entire ego network could create an anomalous amount of action in the OSN, which may raise the suspicion of the OSN administration or community.

Certain techniques for minimizing this amount of activity may involve graph algorithms to allow the attacker an efficient connection to several nodes in the ego users graph instead of connecting to the entire ego network. In graph theory, a vertex cover of a graph is a set of vertices such that each edge in the graph is incident to at least one vertex of the set.

Formally, a vertex cover *V'* of an undirected graph *G= (V, E)* is a subset of *V* such that $uv \in E \wedge (u \in V \vee v \in V)$. It is a set of vertices *V'* where every edge has at least one endpoint in the vertex cover *V'*. Such a set is said to cover the edges of *G*.

The problem of finding a minimum vertex cover in a graph is an optimization problem (Dinur & Safra, 2005). We formulate the problem assuming that every vertex has an associated cost *c(v)*≥ 0 and define:

minimize $\sum_{v \in V} c(v)x_v$ (minimize the total cost)

subject to $x_v + x_u \geq 1$ for all $\{u, v\} \in E$ (cover every edge of the graph)

$x_v \in \{0, 1\}$ for all $v \in V$ (every vertex is either in the vertex cover or not)

To correlate this problem to an attack on our model, we assume that a potential attacker would create fake attributes only on the vertices (users) in *V'*, which are in the minimum vertex cover; thus, the attacker is able to control all of the connections for a minimal number of users, and hence the creation of the size of the fake property requires less actions than the amount described in the previous attacks section. To reduce the problem to the spammer attack, we define the cost $c^{\psi}(v) \geq 0$ as the number of actions required for the creation of G^{ψ}, and formulate it as follows:

minimize $\sum_{v \in V} c^{\psi}(v)x_v$ (minimize the total number of actions)

subject to $x_v + x_{v^{spm}} \geq 1$ for all $\{v^{spm}, v\} \in E^{spm}$ (cover every edge of the connected spammed subgraph that connects a spammer node with a friend node)

$x_v \in \{0, 1\}$ for all $v \in V$ (every vertex is either in the vertex cover or not)

$G^{\psi} \leftarrow G^{spm} \cup V'$ (the connection of the spammer network is to the vertex cover)

An example of such a minimal vertex cover is seen in Fig. 10. *V'= {A, B}* is a vertex cover, since all of the edges are connected to either *A* or *B*. The futility of such an attack is explained as follows: first, the problem of finding the minimal vertex cover is NP-complete (Dinur & Safra, 2005), and second, the networks of the allotted users in *V′* remain very large and must be created with fake attributes, as presented in the previous subsection.

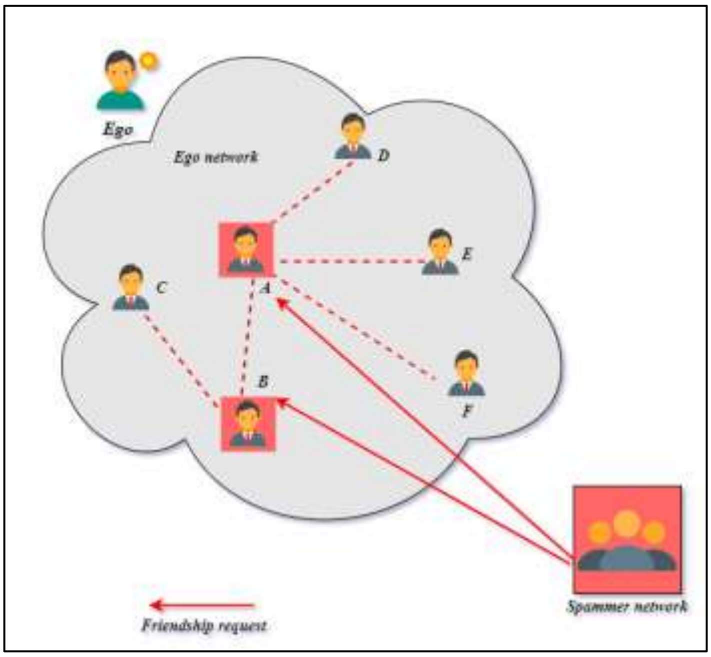


Fig. 10: Minimal vertex cover for an attack on OSN attributes

Finally, after the creation of the spammer network, the attack is delayed by t_{spm} . This delay in time could be very significant in terms of the OSN structure: as time goes by, properties change, users are added and removed, and the network can be different from its preliminary status. The changes of the network create difficulty for an attack that was pre-ordained to the original network and might not be relevant after the delay of T^{AUA}. This limitation makes the attack even harder to implement in a real OSN, since it needs to adhere to the dynamic changes of the network over time.

The full attack is described in Algorithm 4:

Algorithm 4. ***SpammerCommunityMinimalAttackOnOSN***

Input: Total Friends threshold T^{TF}, Age of User Account threshold T^{AUA} , Graph G, Spammer Vertex v_{spm_0};

Output: Spammed Graph G^{ψ}

For i =1 to T^{TF}

```
 $v_{spm_i} \in V^{spm}$ // creating $T^{TF}$ fake users
Graph $G^{spm}$ ← {$V^{spm}$, $E^{spm}$}; // creating a spammer network
Wait ($T^{AUA}$) // the threshold time must pass to authenticate the AUA attribute
$V'$ ←minimalVertexCover (G)
For each $v$ in $V'$ and e in $E'$; $0 \leq i \leq |V'|$
$e_i$ ← { $v_{spm_i}$, $v_i$ } // spammer connects to minimalVertexCover of Ego network
$G^{\psi} \leftarrow G \cup G^{spm}$
return $G^{\psi}$
```

In chapter 8 we show that the above attack is futile.

5. Context-based model

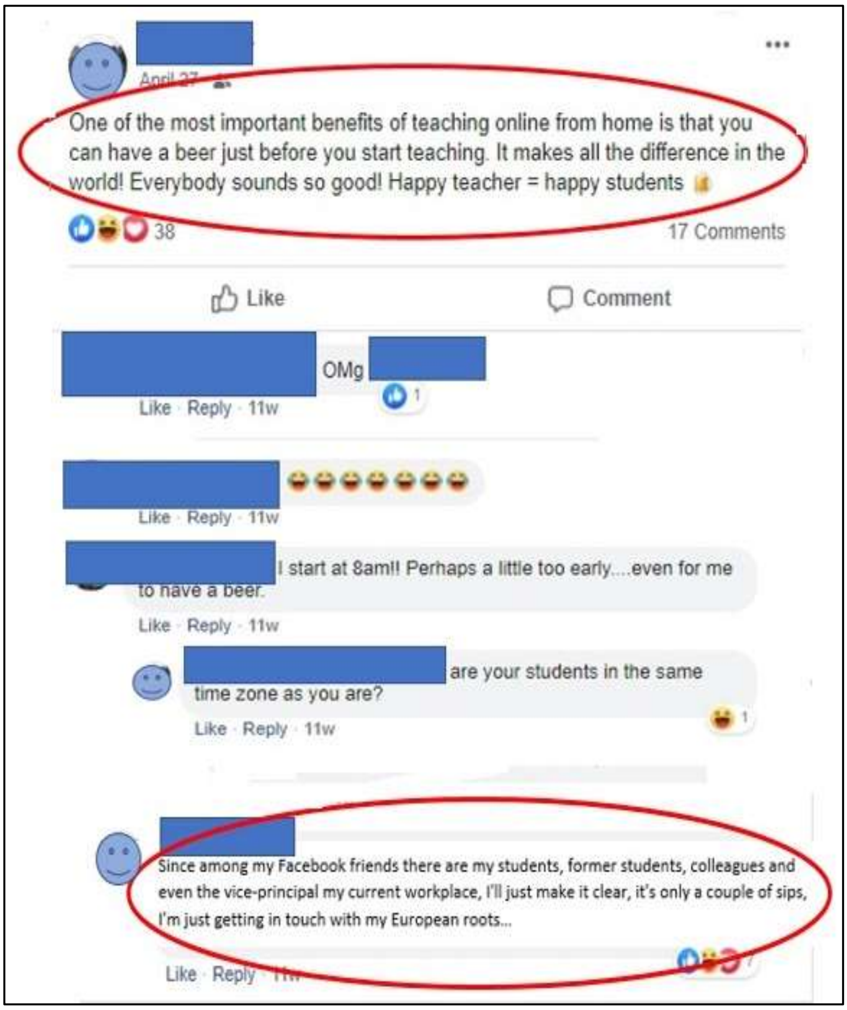
5.1. Basic definitions and motivation

The basic model treats all the users equally, and as demonstrated in the example, users are not a homogenous group and may hold different preferences in various topics and data categories. A certain friend of the ego user can be very trustworthy, but with radical and very different political opinions, a fact that might encourage the ego user not to share political posts with him.

An example of such a problematic case is seen in Fig. 11, taken from a real OSN. In this example the user writes a humoristic innocent post as if he was talking to his friends. While writing, he does not think that one of the readers of this post could be the vice-principal of his workplace at the conservatorium. In this case, the user understands the problematic aspect of his post only after he gets the reactions and comments about it. He then, accordingly, adds an apologetic comment about it. This example demonstrates that sensitivity and privacy of social network data is context dependent.

The main idea of this research is to extend the basic trust model defined above and make an important separation for different types of data instances. Some are inherently more sensitive than others, and thus there is a need to be treated discreetly. A user’s perspective on a data instance is very subjective, and some ego users might see the same content in different lights. This may happen because of different political views, different preferences, different types of personality and more.

In (Misra & Such, 2016) we can see that users tend to trust the OSN that it will both preserve and protect their data. There is a gap between this trust and what happens to their data in terms of misuse and spreading. This gap can be closed by increasing the transparency of what happens with the users' data. A part of our solution solves this issue by controlling the spread of data, thus creating an awareness of what happens with it. There is a distinction between knowledge that users have on their OSN features that are understood and accepted by them (such as their posts, likes, shares and number of friends), and the knowledge that they do not have, which is where this data is used in the OSN besides in their own network, who has access to it, and if and how it is analyzed for different purposes, such as commercial, political or statistical.



[redacted]

April 27 ·

One of the most important benefits of teaching online from home is that you can have a beer just before you start teaching. It makes all the difference in the world! Everybody sounds so good! Happy teacher = happy students

38 · 17 Comments

Like · Comment

[redacted] OMg [redacted]

Like · Reply · 11w · 1

[redacted] 😂😂😂😂😂😂😂

Like · Reply · 11w

[redacted] I start at 8am!! Perhaps a little too early....even for me to have a beer.

Like · Reply · 11w

[redacted] are your students in the same time zone as you are?

Like · Reply · 11w · 1

[redacted]

Since among my Facebook friends there are my students, former students, colleagues and even the vice-principal my current workplace, I'll just make it clear, it's only a couple of sips, I'm just getting in touch with my European roots...

Like · Reply · 11w · 7

Fig. 11: Inappropriate data sharing due to unawareness in OSN

When a certain user creates a post, a comment, or any other type of data instance in the OSN, this action can be used to learn the user's tendencies in different topics and categories, and can be used to create a contextual, dynamic trust level per user per topic or data category. For context evaluation, we categorize different users in the ego network by their trust per context. An ego user gives his/her friends different trust values for every category κ, meaning that they have a Subjective Trust Value, denoted here as STV_{κ}. These κ categories can be varied by nature, and may include different subjects, such as, e.g., politics, sales, sports, social friendship interactions, etc. Ideally, the ego user sets the value of STV_{κ} for each user in his/her network and this process is carried out at the beginning of the friendship (e.g., approval of a friendship request). However, it would be unrealistic to expect the ego user to cooperate in such a tedious process. Moreover, a user's trust value changes dynamically over time to reflect the user behavior in the OSN as expressed by the actions the user does.

If we could evaluate every action trust-wise, we could approximate the STV_{κ} by the following model. Let $STV_i A_{\kappa}$ denote the level of trust from the ego user to a friend in category κ, as observed by action i. For every action there is a weight (w_i) that represents the effect of the action (some actions can affect the user's subjective trust value more than others). These weights can change according to the importance given to them by the user. In this research they are equal by default. $STVA_{\kappa}$ is computed as a weighted average of the trust values given to each of the actions, normalized by the number of actions, as detailed in equation 19:

$$STVA_{\kappa} = \langle w_i \, STV_i A_{\kappa} \rangle = \frac{\sum_{i=1}^{|STVA_{\kappa}|} w_i \, STV_i A_{\kappa}}{\langle w \rangle |STVA_{\kappa}|} . \qquad (19)$$

At this point we can calculate the total trust level of the user in a certain κ category. We denote it as UTV_{κ}, and it consists of the basic model's $UTV_K$and $STVA_{\kappa}$, and, considering the weight (w) of every factor, as follows:

$$UTV_{\kappa} = \frac{w_{UTV} UTV + w_{STVA_{\kappa}} STVA_{\kappa}}{\langle w \rangle} . \qquad (20)$$

The UTV_{κ} is the trust value computed by our model for approximating the actual STV_{κ}. We can see an example for such a set of UTV_{κ}s and access granting for certain data instances in Fig. 12 (*SAF* will be explained in the next sub-section), which is based on real users (their names were altered in the figure for privacy) from our experimental evaluation. In the figure we can see the different factors according to the context categories and their values, based on calculations using equations 21 and 22 (which will be presented in the upcoming subsection). As we can see in the figure, three out of four users hold the necessary trust value (UTV_{κ}) for κ=Online Shopping, and thus have access to it, while only one user (Arik) holds the necessary UTV_{κ} for κ=Elections and has access to it. The purpose of the estimation of UTV_{κ} for each user in the network is to give as accurate as possible an estimation of the ego user's trust of each friend in each category k, STV_{κ}, which represents the ground truth. In the experimental section we will demonstrate the ability of the computed UTV_{κ} to predict the ground truth STV_{κ}.

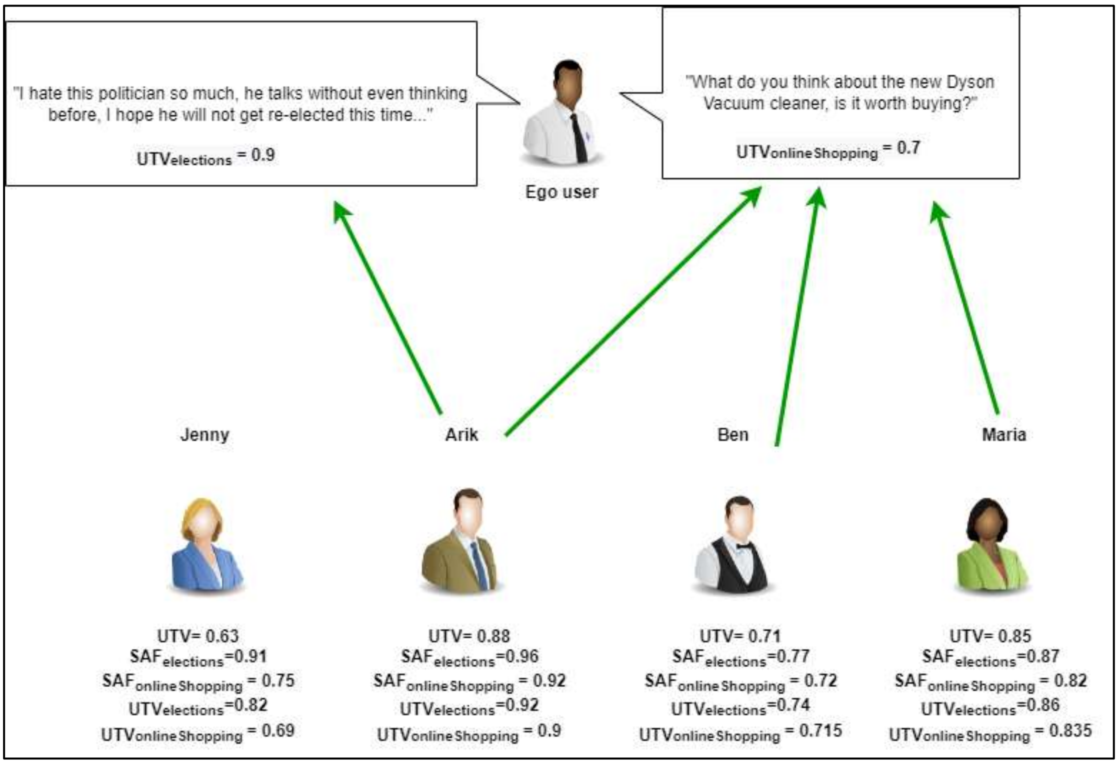

"I hate this politician so much, he talks without even thinking before, I hope he will not get re-elected this time..."
UTVelections = 0.9
Ego user
"What do you think about the new Dyson Vacuum cleaner, is it worth buying?"
UTVonline Shopping = 0.7
Jenny
Arik
Ben
Maria
UTV= 0.63
SAFelections=0.91
SAFonline Shopping = 0.75
UTVelections=0.82
UTVonline Shopping = 0.69
UTV= 0.88
SAFelections=0.96
SAFonline Shopping = 0.92
UTVelections=0.92
UTVonline Shopping = 0.9
UTV= 0.71
SAFelections=0.77
SAFonline Shopping = 0.72
UTVelections=0.74
UTVonline Shopping = 0.715
UTV= 0.85
SAFelections=0.87
SAFonline Shopping = 0.82
UTVelections=0.86
UTVonline Shopping = 0.835

Fig. 12: Access decisions to data instances in different categories and trust values

5.2. The effect of sentiment analysis on contextual trust

As was discussed previously, computing the value of each action is not simple. In this section we show that NLP analysis of the text within actions can be used to adjust the value of context-based trust. In previous research (Collomb, Costea, Joyeux, Hasan & Brunie, 2014) it was shown that 'sentiment' influences trust. This is specifically relevant to social networks and their content (Alahmadi & Zeng, 2015). The effect of sentiment on trust is relevant in our social context only in the cases of mutual positive sentiment, that is, the ego user's friend published a post that is recognized as positive by the sentiment analysis, and then the ego user responds positively to it (with a positive comment, a like, etc.). This mutual positive sentiment is a strong confirmation of trust between the two. This effect is not relevant in cases of negative sentiment, due to the possible disambiguation of the mutual action. For example, a friend can post: "I hate Bob, he is a liar, and I will never vote for him!", and the ego could then reply: "You are right! I strongly think so too and think it's good that you say that!" which creates a positive sentiment and strengthen the Trust between the two, but he can also respond: "Agreed, Bob is a liar and I hate him too!" which creates a negative sentiment, but then also strengthen the trust between the two.

(Therefore, in our experimental evaluation we consider positive sentiments only and leave negative sentiments for future work). The sentiment analysis in our model is used to estimate the trust of each action. A question arises about the influence of the system on the freedom of information, when gaining contextual trust may tilt the balance against (potential) isolation from information that one does not agree with. To address this problem, we can give different weights to the sentiment analysis factor and the basic *UTV*.

If there is a preference for also seeing posts and data that do not necessarily adhere to the 'echo chamber' (the resemblance of opinions in the close network to the ego user's opinions) we created, we can reduce this influence by changing the desired weights, hence giving more balance to different opinions in our network.

The sentiment analysis factor for an action *i* in category κ is denoted here as SAF_iA_κ. This is summed up as the total estimation of actions, and here we define the sentiment refined value as $STVA_\kappa^S$:

$$STVA_\kappa^S=\langle w_iSAF_iA_\kappa\rangle = \frac{\sum_{i=1}^{|STVA_\kappa^S|}(w_iSAF_iA_\kappa)}{\langle w\rangle|SAF_iA_\kappa|} \quad . \qquad (21)$$

This value will, of course, also create a refined UTV_κ denoted here as UTV_κ^S:

$$UTV_\kappa^S = \frac{w_{UTV}UTV+w_{STVA_\kappa^S}STVA_\kappa^S}{\langle w\rangle} \; . \qquad (22)$$

These refined parameters will give us a good basis for the comparison and analysis of the contribution of sentiment analysis to the contextual trust in the OSN content and connections for the ego user.

Using context-based trust we can now apply all the algorithms presented in our previous work, including access control and flow control, but applied to a specific context. In the experimental evaluation section, we compare the two computed estimates: $UTVk$ and UTV_{κ}^{S} to the ground truth of $STVk$.

6. Fake-news-propagation prevention

An important case of the extension of the context awareness models is the detection of fake news and the prevention of its propagation. In this model, users gain or lose trust values in different data categories, and consequently may not be exposed to some data instances. As a result, these users will not be able to spread fake news. This is specifically important in categories, such as politics, which is one of the main categories for fake news, especially during election times. The model analyzes the network trust wise in a deep and comprehensive manner. The users and their social content are monitored and users that are not trustworthy are suspected as potential of spreaders of false data, and even as the possible source of this data.

Figure 13 describes an ego user's information spreading to friends of friends (Users A1, A2 and A3), triggered by an action (a comment in this example) taken by the ego node's direct friend (User A) on the ego node's data. In this part we use the sentiment analysis factor for action α in category κ, denoted $SAF_{\alpha}A_{\kappa}$, as an important indicator of fake news. This is because fake news usually contains polarized emotions (very positive or very negative), as described thoroughly in (Cui, Wang, & Lee, 2019).

The second indicator we use is the user's trust value UTV_{κ}^{j}, which was described in the previous section. The actions that were used to compute the $SAF_{i}A_{\kappa}$ for the UTV_{κ}^{j} are different from $SAF_{\alpha}A_{\kappa}$- which is a new action that we now examine to determine whether it should be considered as fake news or not.

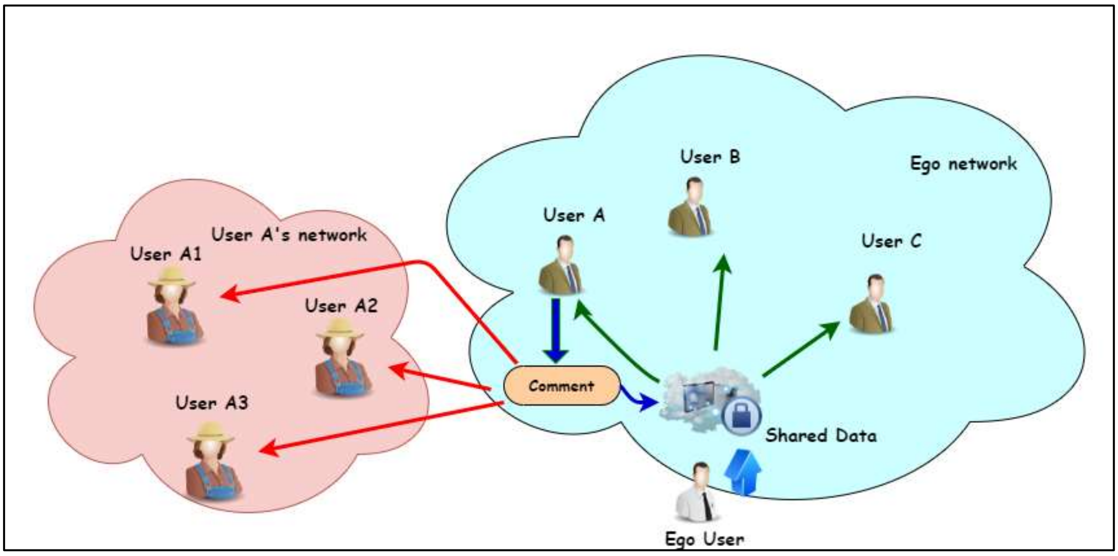


Fig. 13: Data spread, not necessarily intended, in OSN

Q-learning is a reinforcement learning algorithm that aims to learn the value of an action in a particular state, first introduced by (Watkins & Dayan, 1992). Reinforcement learning involves an agent, a set of states – *S*, and a set *A* of actions per state. By performing an action *a*, the agent transitions from state to state. Executing an action in a specific state provides the agent with a reward. The goal of the agent is to maximize its total reward. So let us formally define these parameters on our fake-news-prevention model, based on trust and context:

- System – The ego network.
- Agent – ego user.
- Action – a user's action in the OSN- e.g., post, share, etc.
- Reward – one point given for accurate prediction of fake news; a smaller reward may be given for prediction of non-fake news.
- State - the user's condition after an action – including reward.
- Initial state of the system – all rewards are zero; no actions taken yet; threshold values set for $SAF_{\alpha}A_{\kappa}$ and UTV_{κ}^{j}.
- Training of the model – comparing the actions and user data to a ground truth of fact checking.

Our basic premise in this part of the model is that actions that have very high, or very low (polar) values of $SAF_{\alpha}A_{\kappa}$, taken by users who have low UTV_{κ}^{j} values, have the potential of being fake news. For this purpose, at the initial state of the system, we set threshold values for these parameters that can dynamically change in the process of learning. Although SAF_iA_κ is used as a part of the calculation of UTV_κ^j, it serves a different purpose here - not as trust estimator, but as a detector of polarized sentiment of data; thus, it must be considered separately. The reward is set to be given for accurate prediction of Fake News by the Fact Checker, and for predicting the accurate thresholds set for the other parameters. At the end of the learning process, we aim to detect the users that have the most prominent potential of being fake news propagators. These values can be adapted to another important parameter mentioned above, *TF*. The higher this number is, the higher is the potential harm of this user. Thus, we can apply stricter thresholds for users that have a large network. The reward in reinforcement learning is actually given to the agent, which is the ego user, but we relate it to a certain user, so we can detect our potential fake news propagators. The algorithm for detecting and preventing fake news propagation with trust and reinforcement learning is as follows:

Algorithm 5. FakeNewsPropogatorsDetection

```
Input: System Ego_Network, Fact Checker FC, false_counter=0;
Output: $T_{MAX}^{SAF}$, $T_{MAX}^{UTV}$, $reward^{MAX}$, max_index
    1. for every user j in Ego_Network // initializing users' parameters
        o calculate $UTV_\kappa^j$ ; calculate $SAF_\alpha A_\kappa$; $reward^j$ ← 0
        o if FC= False
            ▪ false_counter ← false_counter +1
  //Setting the system initial state (0) with threshold values $T^{UTV}$, $T^{SAF}$
    2. $T_0^{UTV}$ ← AVG of $UTV_\kappa^j$ in Ego_Network
    3. $T_0^{SAF}$ ← AVG of $SAF_\alpha A_\kappa$ in Ego_Network
//The training of the model with s States
    4. $reward^{MAX}$ ← 0; max_index ← 0; $T_{MAX}^{SAF}$ ← $T_0^{SAF}$; $T_{MAX}^{UTV}$ ← $T_0^{UTV}$
    5. for every state $0 \le s \le |States|$
        o $reward^s$ ← 0; UTV_counter ← 0; SAF_counter ← 0
        o for every user j in Ego_Network
            ▪ if $UTV_\kappa^j \le T_s^{UTV}$ then UTV_counter ← +1
            ▪ if $SAF_\alpha A_\kappa \ge T_s^{SAF}$ then SAF_counter ← +1
            ▪ if $UTV_\kappa^j \le T_s^{UTV}$ and $SAF_\alpha A_\kappa \ge T_s^{SAF}$ and FC= False
$reward^j$ ← 1; $reward^s$ ← $reward^s$ + $reward^j$
```

```
        o  if $reward^{s} \geq reward^{MAX}$
              ▪  $reward^{MAX} \leftarrow reward^{s}$; $max\_index \leftarrow s$
              ▪  $T_{MAX}^{SAF} \leftarrow T_{s}^{SAF}$; $T_{MAX}^{UTV} \leftarrow T_{s}^{UTV}$
        o  if $s <$|States| // not the last state
              ▪  update $T_{s+1}^{UTV}$ and $T_{s+1}^{SAF}$ using UTV_counter, SAF_counter,
                 $reward^{s}$ and false_counter * as follows:
              ▪  $T_{s+1}^{UTV} \leftarrow T_{s}^{UTV} \cdot \frac{UTV\_counter}{UTV\_counter-1} \cdot \frac{false\_counter}{false\_counter-1} \cdot \frac{reward^{s}}{reward^{s}-1}$
              ▪  $T_{S+1}^{SAF} \leftarrow T_{s}^{SAF} \cdot \frac{SAF\_counter}{SAF\_counter-1} \cdot \frac{false\_counter}{false\_counter-1} \cdot \frac{reward^{s}}{reward^{s}-1}$
6. return $T_{MAX}^{SAF}$, $T_{MAX}^{UTV}$, $reward^{MAX}$, max_index
```

* The update is done to the thresholds of the next state, and determined by the parameters mentioned, for the purposes of refining these thresholds and giving optimal results in terms of reward, and a high probability of prediction of fake news propagators.

For example, if *SAF_counter* is very high relative to the *reward^s*, it means that there were many actions with a $SAF_{\alpha}A_{\kappa}$ that were higher than the threshold T_{s}^{SAF}, we will update T_{s+1}^{SAF} to be higher, and therefore more strict, to get more accurate results. In the update of T_{s+1}^{UTV} and T_{s+1}^{SAF} at every iteration we can see above that these thresholds are being multiplied by the relative change to the previous iteration of *UTV_counter, SAF_counter, reward^s* and *false_counter*. It is important to state here that these updates are done only after a certain number of iterations, since fractions of small numbers give us changes that are above 1 (e.g., multiply a threshold of 0.3 by 2 and then again by 2, we will already pass 1).

The purpose of the algorithm is to get the most refined thresholds for $SAF_{\alpha}A_{\kappa}$ and UTV_{κ}^{j}, that will be able to predict most accurately users that may be fake news propagators, in comparison with the ground truth of the fact checker. In stage 1 we calculate the trust and sentiment parameters described in the previous section. In stages 2 and 3 we set the system initial state (0) with threshold values that are simple averages of the parameters of the ego network. Stages 4-6 are the training of the model, which includes comparison to the fact checker results, and, accordingly, first updating the parameters and then the thresholds, and finally returning the results of the optimized thresholds. We can see one iteration of the algorithm in Fig. 14, where the initial state of the system has two users in state 0; the system initial state is $T_{MAX}^{UTV} = 0.3$ and $T_{MAX}^{SAF} = 0.6$.

If UTV_{κ}^{j} is less than 0.3 and $SAF_{\alpha}A_{\kappa}$ (simplified in the figure as *SAF*) is higher than 0.6 or lower than -0.6 (relatively very good or very bad), and the action that the user made was proven to be fake news (by the external fact checker), then the reward is received.

On the other hand, if a trustworthy user creates an action that is not fake news, and his UTV_{κ}^{j} is more than 0.6 and $SAF_{\alpha}A_{\kappa}$is between -0.6 and 0.6 (moderate sentiment), the reward is also given. This is how the model tries to predict the veracity of the data, and how the learning is done. We can see that User A holds both conditions for fake news, and it is also verified to be fake by the fact checker; thus, a reward is given.

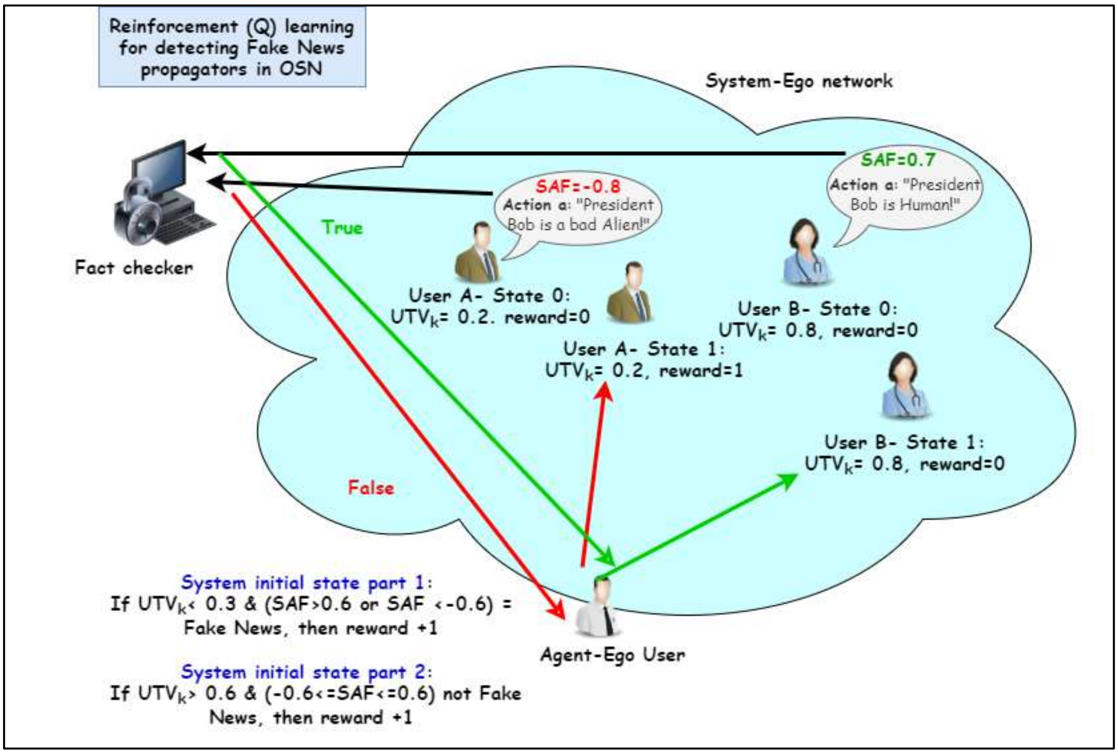

Reinforcement (Q) learning for detecting Fake News propagators in OSN
System-Ego network
SAF=0.7
Action a: "President Bob is Human!"
SAF=-0.8
Action a: "President Bob is a bad Alien!"
Fact checker
True
User A- State 0: UTVk= 0.2. reward=0
User B- State 0: UTVk= 0.8, reward=0
User A- State 1: UTVk= 0.2, reward=1
User B- State 1: UTVk= 0.8, reward=0
False
System initial state part 1:
If UTVk< 0.3 & (SAF>0.6 or SAF <-0.6) = Fake News, then reward +1
Agent-Ego User
System initial state part 2:
If UTVk> 0.6 & (-0.6<=SAF<=0.6) not Fake News, then reward +1

Fig. 14: Learning process of the Ego network for the detection of fake news spreaders

After this iteration the threshold are then updated in State 1, as the result of the algorithm. If we assume that this is the 10th and 11th iteration (because as explained above we do not do the update in the first iterations), then the thresholds will be updated to 0.6 · 1.1 · 1.1 · 1.1 = 0.8 for T_{s+1}^{SAF} and 0.3 · 1.1 · 1.1 · 1.1 = 0.4 for T_{s+1}^{UTV}.

These threshold values, and the changes we can do to them, are important from both the aspect of prediction - we aim to know, with some level of certainty, in which values we can infer fake news propagators' actions, and how strict should be the values with a high *TF* value (highly influencing users).

7. GDPR compliance for social networks by DRM, context and trust

The context-based model presented above enables the limitation of the distribution of context specific data in the network. In this section, we leverage this model to enforce privacy following the GDPR. We divide OSN activities into atomic ones and non-atomic ones. Non-atomic actions are ones that can create linked actions; for example, writing a post can create comments and likes from friends in the network. Writing a comment is also non-atomic since it can create likes and sub-comments. Atomic actions, such as likes, accordingly, cannot create linked actions. Non-atomic actions have a challenging aspect of data ownership. If the ego user writes a post, and then Alice comments on this post, who does the comment belong to? It might seem negligible if the comment is of a simple nature, but comments are a platform that sometimes go well beyond a simple data instance and can be elaborating, especially if the post itself is of a sensitive nature to begin with. Comments themselves are non-atomic actions that can create linked actions, e.g., Bob replies to Alice's comment, or likes it. In terms of OSN, this ownership problem is quite important, especially considering the GDPR, although GDPR does not mention data ownership, but data subjects and control in processing this data. Comments, for example, can be deleted either by the ego node (who writes the post) or by the commenter himself. Let us clarify the ownership concept by another example. Supposed Bob publishes a picture where Alice appears. The data subject is Alice and according to GDPR she has the right to erase it. However, since Bob published the picture, we delegate this responsibility to Bob and call him the owner (See ownership agreement below). The challenges here are to find the proper ownership outline in order to handle the data distribution without privacy or ownership conflicts. We divide the GDPR-related tasks in social networks into two parts: data dispersion in the network and erasing data from the network (right-to-be-forgotten).

7.1. Handling data dispersion in the network

There are three main types of data instances in OSN:

1. Governed data – a data instance that can be shared, commented on, etc., but never fully and separately copied by another user. The original data will remain as the data owner's instance, and no other instances (objects) will be created.
   For example – Alice posts a picture, her friends comment and share it, but none of them copies and uploads it as a separate picture in his/her feed.
2. Governed data with leakage – a data instance that can be shared, commented on, etc., but can be fully and separately copied by another user, which implies that other different instances (objects) are created. For example – Alice posts a picture, and one of her friends' copies and uploads it as a separate picture in his/her feed.
3. Controlled data – a data instance that cannot be shared but can only be viewed or liked as an atomic action. This type of data is very relevant to video or audio files that need to be restricted in their dispersion. It can be shared only with followers or subscribers and cannot be freely shared with other parts of the network. For example – Alice is a singer, who shares a new song only with her followers in their private group and disables the sharing option for this song.

For the problem of data dispersion in the network, which is relevant to both types of governed data, but not to the controlled data, subject to GDPR, we suggest a three-stage process for non-atomic OSN actions:

1. Ownership agreement- In the first stage, we look at the data instance origin as it is generated. This origin can consist of multiple ownerships, e.g., a photo uploaded with several tagged users in it, or a song uploaded by an artist that involves the record company, which is tagged in the post. For this preliminary stage we need the consent of all the original data owners. (Note that this is a policy issue; one may define a policy that even with multiple ownership, permission of one owner is sufficient, but we will not deal here with different possible policies).

2. In the second stage, we monitor the spread of the data, and decide whether a certain user, who is connected to the data owner, is allowed to access the data. In this stage we use the trust-based model described in the previous section to determine which user can gain access to the data instance.

   Users with a low trust value could be denied access in order to prevent them from making unlawful use of the data instance or merely to deny them the knowledge (exposure to the content). These access decisions are transparent to the user (the ego node), who knows the restrictions are based on trust.

   Since from a GDPR perspective, users should be informed regarding the processing of their data and the recipients of such data, an important part of this stage is the contextual validation phase. A data instance can be characterized by its context (e.g., politics, sports, etc.), and the trust measure must be refined by this context. For context evaluation, we categorize different users in the ego network by their trust per context. We calculate this trust for the friends in the ego network - different trust values for every category, as explained in the previous sections.

3. Usage agreement- In the third stage, which regards non-atomic actions and is done after the completion of the second stage when we know the granted access for users, we need to create an access and usage agreement - even before a user creates the sub-data instance itself (comment on the original post, share of the post, etc.), and by doing that he/she are actually spreading this data instance to their network.

The manifestation of this process is portrayed in Fig. 15. We can see that in Stage 1 there is an ownership agreement between Ego A and Ego B, who are the owners of the original data instance. Stage 2 includes the screening of the user named Alice - she must gain the necessary contextual trust to act on the data (in this case, commenting on the post). In stage 3 we can see that the final approval of acting on the data (and thus, spreading it) is dependent on the usage agreement for Alice. We can see that Bob, who is a friend of Alice, wishes to reply on her comment, thus creating another branching in the data tree. The usage agreement is between Bob and Alice; therefore, it is also derived from the usage agreement between Alice and the ego user. If Bob wishes to read the comment and not act upon it, all he needs is the sufficient trust level from Alice in the relevant context.

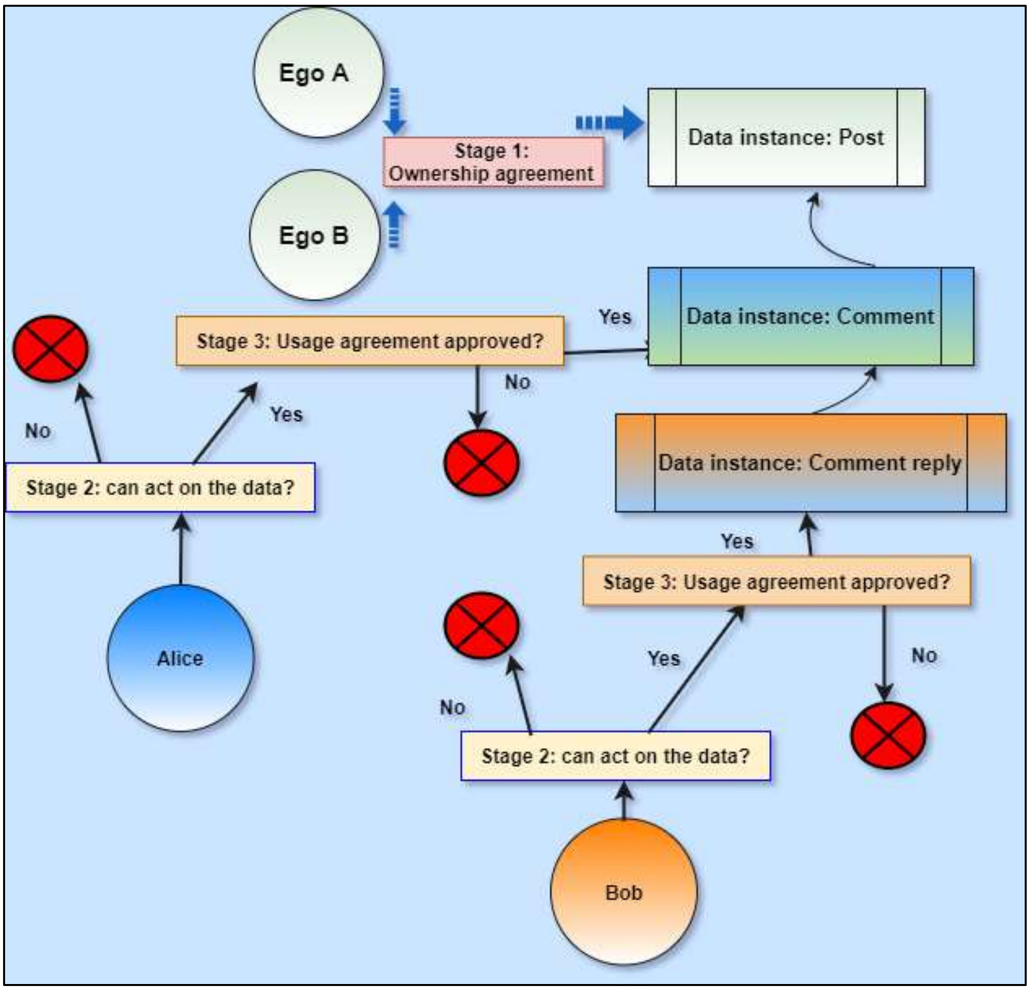

Ego A
Stage 1: Ownership agreement
Data instance: Post
Ego B
Yes
Data instance: Comment
Stage 3: Usage agreement approved?
No
No
Yes
Data instance: Comment reply
Stage 2: can act on the data?
Yes
Stage 3: Usage agreement approved?
Alice
Yes
No
No
Stage 2: can act on the data?
Bob

Fig. 15: The process of monitoring non-atomic OSN actions

This approach is a viable solution to some of the challenging aspects of GDPR in social networks and can create a sustainable solution to the dispersion and ownership problem of data instances. For example, in Art. 25 of GDPR, there is a requirement that the controller shall implement appropriate technical and organisational measures for ensuring that, by default, only personal data which are necessary for each specific purpose of the processing are processed. That requirement can be fulfilled by using our controlled sharing algorithm.

7.2. Erasing data from the network – right-to-be-forgotten

For the problem of erasing data from the network we suggest a two-stage process for non-atomic OSN actions:

1. In the first stage, we look at the data instance origin. As mentioned in the previous subsection, this origin can consist of multiple ownerships. For this preliminary stage we need the consent of all the original data owners that the data instance can be erased. We use the same ownership agreement mentioned in the first stage of the dispersion solution, that involves the implementation of a consent management solution (Goldsteen, et al., 2017). In a case where we do not delete the original post, but a comment in the chain of comments, the second stage starts from this comment only.
2. In the second stage of this process, we erase all the data instances that were spread in the network along the chain. This task is particularly hard due to the OSN sizes of users and data; however, we use the context-based trust screening results we have for

controlling the dispersion of data. The amount of data that needs to be checked is considerably reduced, because we check only the subset of the active users (those who created data or acted upon data) from the subset of the trustworthy (above a certain *MTV_{κ}*) users in a certain category. The efficiency is substantial, as we demonstrate in the experimental part of this paper.

The use of this approach is an efficient solution to the complex aspects of the GDPR right-to-be-forgotten in social networks, due to its efficient search on a considerably smaller subgraph of the network graph.

7.3. GDPR implementation using DRM and watermarking

Implementing GDPR-compliance solutions depends on the type of data instance described in the subsection of *Handling Data dispersion in the network*. In this section we discuss the dispersion and erasure of each type of these data instances.

Governed data – the dispersion of the data is subject to the context-based trust model; however, since no new copies of the data instance are created, all shares are linked recursively to the data owner.

The erasure can be done easily by a 'recursive chain algorithm', since each action points to its originating 'father' node in the chain up to the data owner.

This algorithm begins the search with the data owner and recursively searches all the nodes that shared this data and erases the related data instances.

The process is depicted in Algorithm 6 and carried out as follows.

Algorithm 6- Recursive data erasing (X – a data instance to erase)

```
If Children(X) are null // stop condition:  no sub-data instances to data of X
  Erase data_instance of (X)
 Else
   X← Children(X)// iterating down the data instances tree
   Recursive data erasing (X)
End
```

The algorithm is called with X as the data owner.

Governed data with leakage – the dispersion here, as well, is controlled by the context-based trust model. The erasure requires two actions. The first is to identify the object that was copied or created from the original data instance – this is done by watermarking or by DRM. Once all copies have been identified they can be erased. The main advantage of our model here is that the search for the watermarked objects is limited to the set of users who are permitted access by the contextual trust model, and not the entire network. The safeguards taken here also include notifications for the involved parties, and that the data will not spread outside the trusted circle since the basic thresholds are determined by the OSN attributes and contextual trust. This is in light of Art. 17 in GDPR: the controller, taking account of available technology and the cost of implementation, shall take reasonable steps, including technical measures, to inform controllers that are processing the personal data that the data subject has requested the erasure. Erasing this data for implementing the right-to-be-forgotten, requires the following algorithm, Algorithm 7 –'contextual search algorithm':

Algorithm 7-Contextual search algorithm (X- a data instance***)***

```
   Y ← action(X) // Y is a set of objects, action is 'like', 'share' or 'comment'
  While Y is not empty
       Remove X from Y
        Erase (X)
 End
```

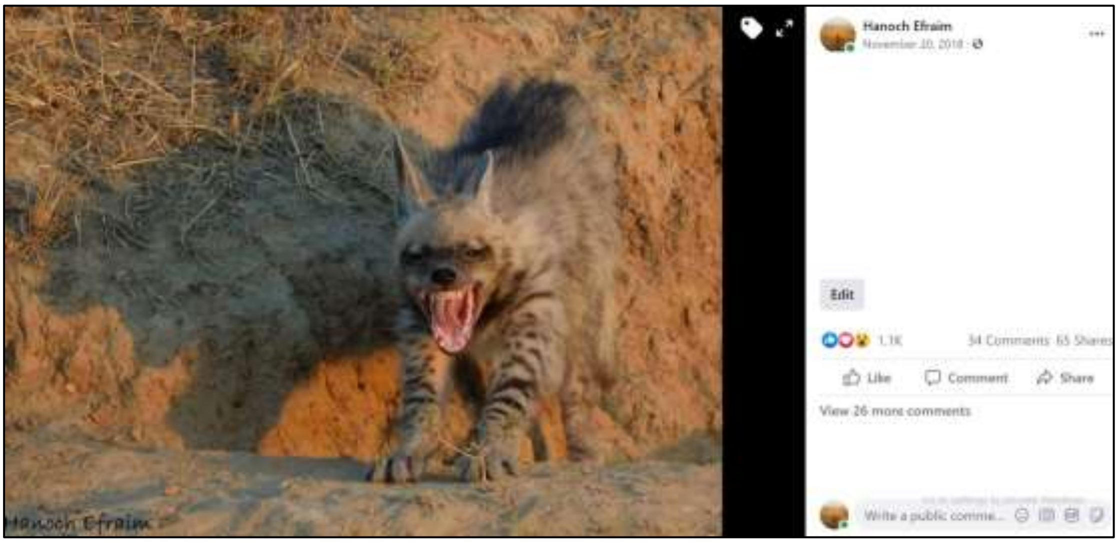


Fig. 16: Watermark use for credit in OSN data instances. **Credit**: Hanoch Efraim

To be able to perform this contextual search we need to identify the data instance and tie it back to the original owner. This can be done by using a DRM method such as watermarking. In Fig. 16 we can see an example of the use of a watermark in an image taken from a real OSN, posted originally in a nature photographers Facebook group. This is done mainly to reclaim credit for original content uploaded to the network.

Controlled data – a DRM scheme similar to (Davidson et. Al, 2016) may be used. Consider the following case where Alice is the ego user: Alice likes to share a data instance with some friends. She encrypts the data instance and saves the key. Alice also sends the encrypted data instance to a trusted third party. If Alice's friend Bob wants to access the object, he asks her permission. Alice requests a public key from Bob and then generates a content license that contains the key encrypted with Bob's public key, but not the encrypted data instance. To access the data instance Bob needs to provide this key to the trusted third party, who will enable Bob to see the object, but not to copy it (like a movie DRM scheme). The erasure is very easy, since the ego user has full knowledge of all the shared data instances, and therefore simply can erase them. The algorithm is depicted in Fig. 17.

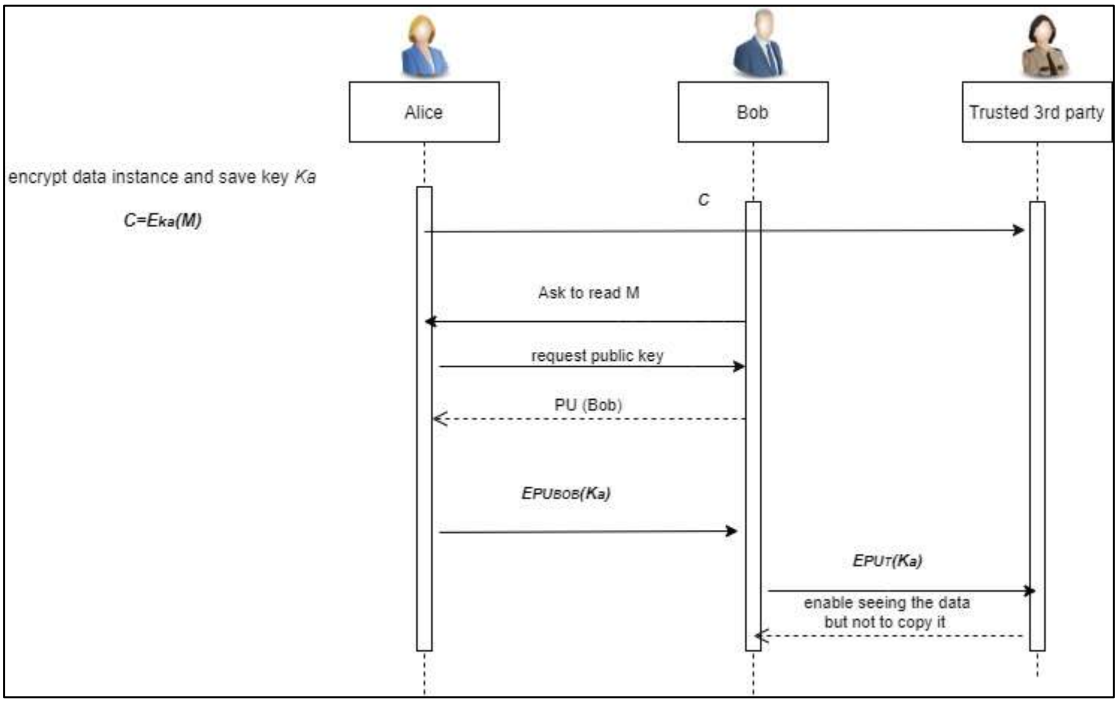


Fig. 17: Controlled sharing algorithm

The algorithm's main limitation is that it requires re-encryption of the object for each share request. In future work we plan to implement both the watermarking and the DRM schemes and derive some performance results on their respective overhead. The sequence diagram in Fig. 17 describes the algorithm specifically as a toy example of Alice, Bob and a trusted 3rd party, because we describe a general scheme of the algorithm that includes all of the sharing requests and encryptions necessary for the scheme.

8. Results – experimental evaluations

8.1. The trust-based model

The experimental evaluation and validation of the model consists of several parts, each concerning a different aspect or phase of the model. In this section we provide a detailed explanation of these parts and demonstrate the results.

This study is entirely based on data provided voluntarily by the participants and on public data that was anonymized. Informed consent was obtained from all participants included in the study. The surveys done in this part were performed as questionnaires delivered to participants that have an active Facebook account, and it was entirely conducted online, while the relevant parts for their network involved a detailed, anonymized and deep description of parts in their networks.

8.1.1. Parameter validation

In the first part of the evaluation, we conducted a survey with 282 real OSN users to validate the parameters we use in the trust computation phase of the basic model (Section 3). The purpose of this survey was to establish a good understanding of our choice of criteria for the model, and to set threshold values for each parameter accordingly.

The survey initially described the information leakage problem as the one depicted in Fig. 1, referring to the participants as the ego users that share a certain data instance on the network, while one of their friends takes an action on the data instance (shares a post, likes a picture, etc.). Then the survey describes verbally a 'friend of a friend', a user that is not directly connected to the ego user and can see this data instance. The questions presented to the participants were concerned mainly with the parameters of this user for the purpose of assessing the credibility; these were related both to their direct friends and to the friends of their friends. The questions attempt to figure out the values that can define a friend as a trusted user who should be granted access to this data instance. An example for this kind of question is "What is the minimal number of friends a user should have to be considered as trustworthy (for sharing your data)?". The other questions are concerned with the significance of each parameter of the model to the computation of trust. The users were asked about the importance of some attributes in their decisions for granting various permissions to their private data.

The survey included the quantifiable attributes of user credibility and connection strength described in Table 1.

For each attribute, the participants were asked for a minimal value that can indicate trust in another user and the average of their answers was set as a threshold. For example, the participants considered users with a *TF* of at least 245 as trustworthy users, with which they are willing to share information. The results are presented in the trust values calculation section above: T^{TF}, T^{MF}, T^{FD} and T^{AUA}.

Table 6 shows the exact values of the survey results for the trust threshold of the model's parameters. The table also shows the standard error of the mean (SEM - $\sigma_{T^{property}}$) values for these parameters for dealing with the different levels of uncertainty, as discussed in the previous sections. Clearly, the results in this table could turn out differently if we consider other aspects of the network or other measures of privacy for more secure or less secure data instances. Two more aspects were examined in the survey: the importance of every one of the model's trust attributes (from Table 1) and the importance (weight) of every one of the Resembling Attributes (*RA*) on a scale of 1 to 10. The results emphasize the fact that no feature was identified as significantly more important than others. The numerical answers for this survey were averaged, and yielded the threshold values for the four attributes discussed in the previous sections. It is important to state here that the users were not asked to provide trust values; those were derived from the experimental evaluation, as we discuss next. However, any user should be able to override the calculated values to suit their privacy preferences. The results of these two aspects are presented in Fig. 18.

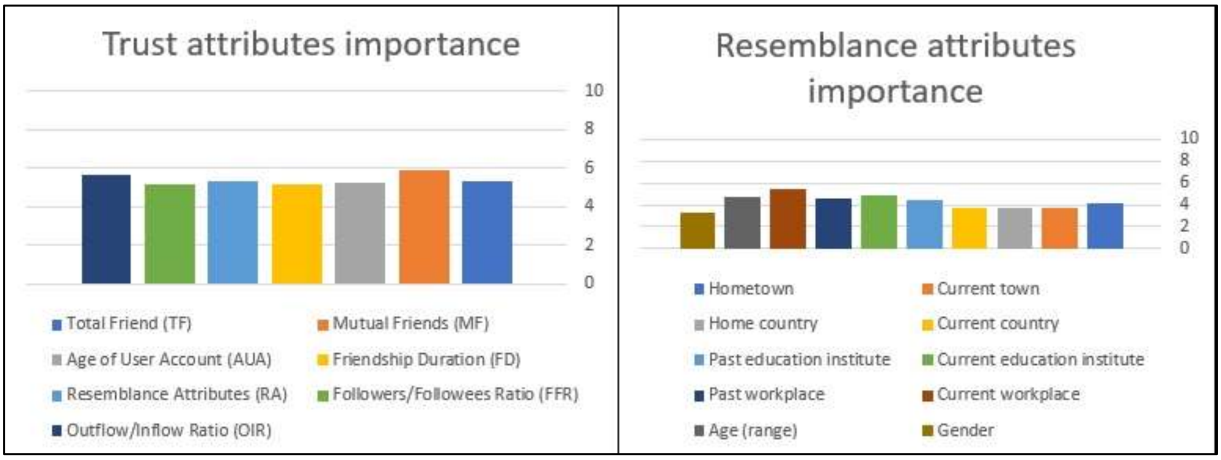


Fig. 18: The importance (weight) evaluation of resemblance and trust attributes

Table 6 – Experimental results for numerical trust values for the model's parameters.

Parameter	**Attribute**	**Experimental value**	**Standard error of the mean (SEM - $\sigma_{T^{property}}$)**
T^{AUA}	Age of User Account (OSN seniority)	23.82	2.33
T^{TF}	Total Friends	244.34	46.24
T^{MF}	Mutual Friends	37	5.6
T^{FD}	Friendship Duration	17.12	1.87

8.1.2. Trust computation validation

In the second experimental evaluation we attempt to validate the trust computation (see *UTV* computation in equation 10 in Section 3) against a real OSN dataset that includes 162 user nodes and their attributes. The real OSN is an actual part of a Facebook ego network that we have examined by finding the attributes of *MF*, *TF*, *AUA* and *FD* of all the ego user's direct friends. From this data we calculated the *UTV* of each friend. The relative importance of the attributes is manifested in the coefficients of equations 8 and 9. The *friend* role was validated based on calculated trust values. The *UTV* of each user node was calculated according to equation 10, and the average *UTV* obtained based on all 162 users was 0.745. If we consider, for example, *MTV*= 0.5, only 3 out of the 162 users are not granted access permission. In Fig. 19 we can see the results of the *UTV* calculations of the 162 user nodes. We also used this experiment to evaluate the threshold values for the trust attributes. Fig. 20 depicts the strong correlation between the trust model attributes comparing the results of the survey and the results obtained by calculation using real OSN data. The Y axis represents the numerical quantities of the attributes. The *TF* attribute has a much higher value in the OSN data since the OSN data presents definitive sharing (actual friends), whilst the survey presents a more general user-preference estimation.

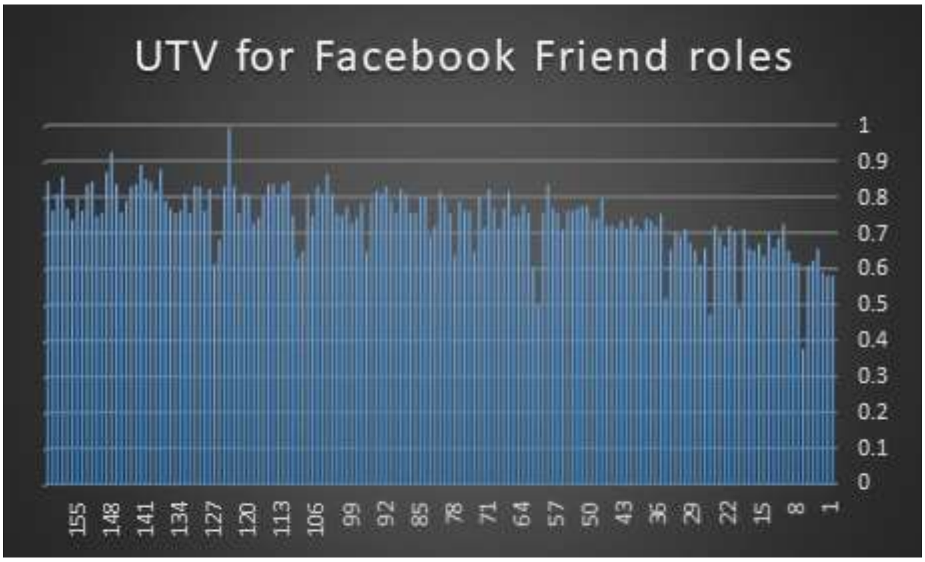


Fig. 19: *UTV* values for 162 user nodes with friend roles

8.1.3. The role in access control

In this part of the experimental evaluation, we attempt to validate the trust computation for users of the same role in the access control part of the model. We conducted an experimental survey including the evaluation of 110 real OSN users provided by 55 participants. Every participant was asked to select 2 users from his/her network that have the same role (e.g., family, colleague), one that could see a certain data instance, and another that could not.

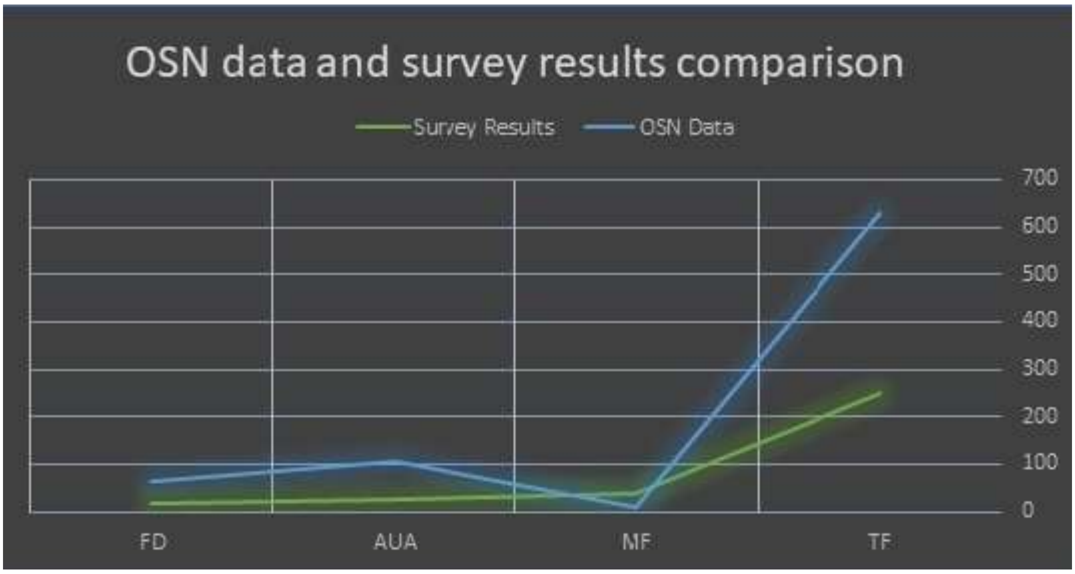


Fig. 20: The compliance of the first two experimental evaluations of the model

The participants were asked first about the friend that was exposed to the data instance: "To what extent, on a scale of 0-10, would you mind if this friend will be exposed to the rest of your OSN content?". A similar question was asked with respect to the friend that was not exposed to the data instance. The purpose of this evaluation was to estimate the correlation between the model's role-based access decisions and the general trust estimation, and, more importantly, the willingness to share information with friends in specific roles. The survey included data from the user's OSN profiles, which was the model's parameters of its raw attribute values (for *MF*, *TF*, *AUA* and *FD*), following the question regarding the willingness to share. The participants were requested to select two users with the same role such that one of them is allowed see a certain data instance, while the other is not allowed.

Afterwards, they were asked to rank for each of these two users the extent to which they would like that user to view their data in general. The results, as shown in Fig. 21, demonstrate a correlation between the Subjective Trust Values (*STV*) as estimated by the participants and the calculated *UTV*. Both were mainly within the range of 0.4 and 1 for the users that were allowed to see the data (*uTrue),* and both are within the range of 0.2 and 1 for users that were not allowed to see the data instance (*uFalse).*

There is, however, a difference in the *STV* and *UTV* between the two groups of users. The average *STV* for *uFalse* was 0.54, and its average *UTV* was 0.74, whilst the average *STV* for *uTrue* was 0.7, and its average *UTV* was 0.81. The differences are large when a user gives a preliminary trust value that is higher that the calculated trust value. This may indicate that this trust may be higher than the desirable one. The results indicate the importance of the trust values for refining the permissions granted to users of the same role. In most cases the two values are quite close.

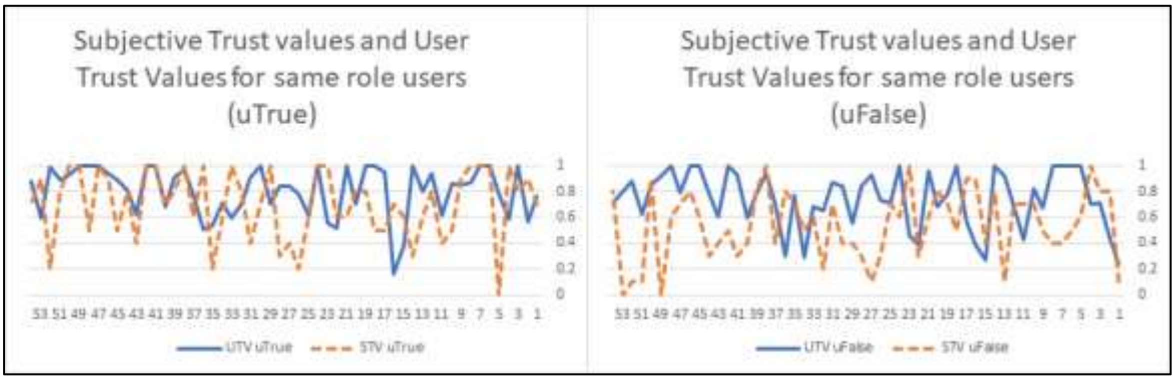


Fig. 21: The juxtaposition of trust values and sharing probability of same role users

8.1.4. Information-flow control

In this experiment we evaluate the flow algorithms from Section 3.4. First, we focus on the MST-based algorithm and then we examine the minimum path trust approach.

8.1.4.1 MST

In the first step to evaluate the information-flow control model, we use the MST algorithm to create of a trustworthy network (section 3.4.1). The purpose of this step is to demonstrate the correlation between the model's cutting decisions and the general trust estimation and sharing willingness of real OSN users.

We conducted an experimental survey including 123 participants. Every participant was requested to provide a *STV* for 16 users that are part of the network graph that is seen in Fig. 5 (two family members that fill the role of Alice and Bob and two colleagues that fill the role of Charlie and David, and their respective friends). Overall, 123 users provided trust values for 1968 users. This was done in the form of questions such as: "In accordance with the current values of attributes seen in Fig. 5, to what extent, in a scale of 0-10 would you mind the user seeing a data instance from your OSN content?". The average *STV* results are presented in Table 7. This allowed us to compare their estimation to the trust values that are calculated by the model.

Table 7 - Sharing willingness experimental results for the graph in Fig. 5, as decided by real OSN users (the bold indicate the minimal SP).

Network of	User	Average SP (Sharing Probability)
Charlie	**John**	**0.413794**
	Isaac	0.482758
	Linda	0.54138
	Karl	0.527586
Alice	Frank	0.427586
	Eve	0.496552
	Harry	0.55862
	George	**0.386206**
David	Ryan	0.534482
	Quinn	0.510344
	Thomas	**0.365518**
	Simon	0.465518
Bob	**Nick**	**0.427586**
	Marry	0.503448
	Philip	0.462068
	Orson	0.434482

We then applied the *ConstructTrustworthyNetwork* algorithm of the flow model on the graph in Fig. 5, to compile a list of nodes which are cut from the ego user network, and assign a trust value to the rest of the nodes. The algorithm results for the same graph are presented in Table 8, showing the MSTs of the friends' networks, as well as the candidates for removal and their respective trust values. We should notice that the users presented in this table are only candidates and will not necessarily be removed if they have high trust values - like John, for example. The results show that the users who were cut from the graph by the algorithm, are the ones that received the lowest *STV* in every one of the 4 networks. The algorithm result for Charlie's network (John) seems a bit high (0.91) but considering the other users in that very strong network, it is the lowest.

8.1.4.2 Path trust

In the second step of the experimental evaluation, we evaluate the *Path-trust* approach of the Information Flow model by using a real OSN. We conducted an experimental survey including 220 real OSN users provided by the same 55 participants described in the previous section. The participants were requested to select one direct friend, and three of its direct friends (friend of a friend), who are acquaintances of the participants, but not direct friends. The purpose of this construct was to convey the "friend of a friend" status, that is manifested in the third phase of our model. Once again, the participants were requested to provide a *STV* for each of the users they selected. For simplicity, the trust values were expressed by the users on a scale of 0 to 10, where 0 stands for unwilling to share and 10 is a definite willingness to share (complete trust). For comparison with the calculated values, we transformed these values to the scale of 0-1.

Table 8 - The model's results for removal of candidates and their trust value, divided by friends' networks.

Friend	MST of the friend's network	Removal candidates and their trust value
Alice	{Alice-George, George-Eve, Alice-Simon, Frank-Harry}	**George:** *u*=0.23
Charlie	{Charlie-Linda, Linda-John, John-Isaac, Isaac-Karl}	**John:** *u*=0.91
Bob	{Bob-Nick, Nick-Philip, Philip-Orson, Orson-Marry}	**Nick:** *u*=0.56
David	{David-Quinn, Quinn-Simon, Simon-Thomas, Thomas-Ryan}	**Thomas:** *u*=0.34

For each participant we considered the average of *STV* provided for the three friends of a friend to be the participant's threshold for granting access to second degree friends. To overcome the possible error caused by the human difficulty to provide exact *STV* values, we defined high and low boundaries of 0.1 for this threshold. A friend of a friend is granted access permission only if the participant has provided a *STV* that falls between the boundaries of the threshold. Otherwise, the friend of a friend is denied access. We calculated the *UTV*, and the *Path-trust* for all the users that were mentioned as friends of a friend by the participants and classified them as granted or denied permission according to each of our two methods, using the average *STV* as the threshold. We compared the classification of each method to the participants' *STV* classification. The results depicted in Fig. 22 in terms of accuracy, precision and recall, and clearly show that the decisions made by the *UTV* method are very close to those made by the participants.

The decisions made by the *Path-trust* approach were less successful. The ground truths for these evaluations were that users that are closer to the ego node and have high trust values, calculated by the parameters, will be evaluated as more trustworthy, and vice versa. Although the permission it granted was mostly to eligible users (precision), it was less sensitive and denied access from other eligible users (recall), and therefore many decisions were not correct (low accuracy).

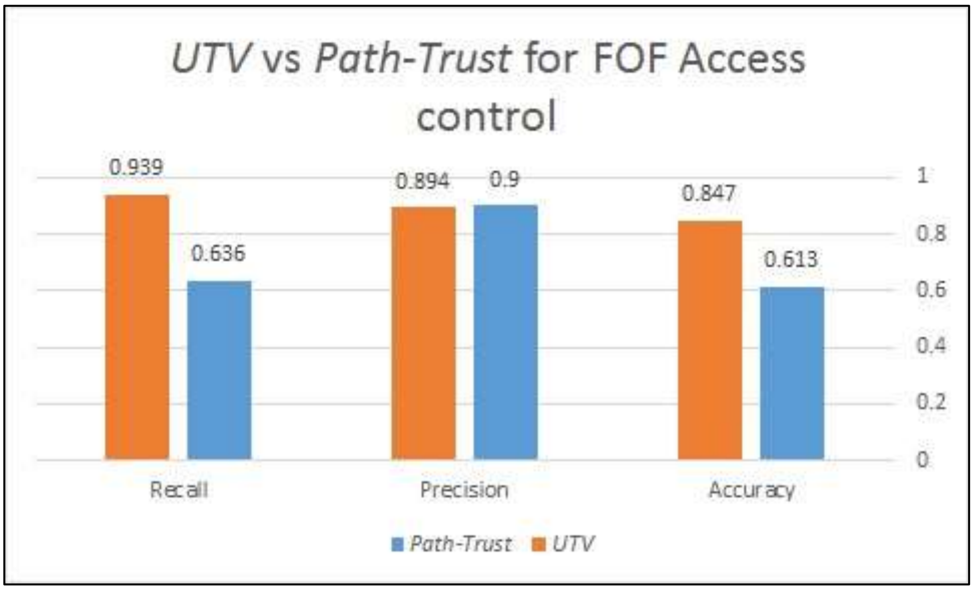


Fig. 22: Performance of *UTV* and *Path-trust* methods as classifiers for FOF

The low accuracy can be explained by the *Path-trust* method's usage of trust in direct members as a discount factor, which lowers the trust in friends of a friend considerably. For example, if a participant provided a trust value of 0.9 to all friends of his friend, it means that he is very willing to share with them. However, according to the *Path-trust* calculation if he trusts the friend 0.92 and the friend's trust in her friend is 0.92, the trust in friend of a friend is lowered to about 0.85. We believe that this method is important to protect the privacy of participants from strangers. In a real social network scenario, the participants are not expected to evaluate all friends of their friends, but just to set a general access threshold to second degree friends - a general group with no names or faces. Only a low threshold will enable access to second degree users.

8.2. Robustness of the model - analyzing attacks

The experimental evaluation estimates the attacker effort in terms of the size of the spammer network that is required for a successful attack to take place. For the OSN attributes threshold we have used the results obtained from the trust-based model's evaluation in the previous subsection. To calculate the sizes of spammer networks we use the experimental results of the thresholds values T^{TF} and T^{AUA} (Table 6). We can see that the basic T^{TF} is 245 for d = 1, and T^{AUA} is 24 months. The size of the spammer network in terms of edges being created is expressed by $|E^{spm}|$, and, as described above, since it is a clique, $|E^{spm}| = \frac{|V^{spm}|\ (|V^{spm}|-1)}{2}$. The resultant graph and values are shown in Fig. 24 (left). The figure shows the number of connections that must be created for a successful attack on the model - for all three types of attacks. For example, for the very strong attack, the size of a spammer network must contain more than 14 million users. In this figure, we also see the size of the spammer network after t_{spm} from the time the attack network was created, when the attack can actually take place. Since there is an annual growth of approximately 10% of users per year in OSN (specifically in Facebook) [17], the number of edges in the spammer network grows. In the left part of Fig.23, the notation of E^{ψ} after T^{AUA} demonstrates this growth after two years. Accordingly, the T^{TF} grows over time, dynamically, forcing the spammer network to add more users, and thus making the attack harder, and even non-realistic in OSN terms. For the optimized attack (set cover attack), the right part of Fig. 23 demonstrates the effort required to attack networks with a connectivity level of 0.5.

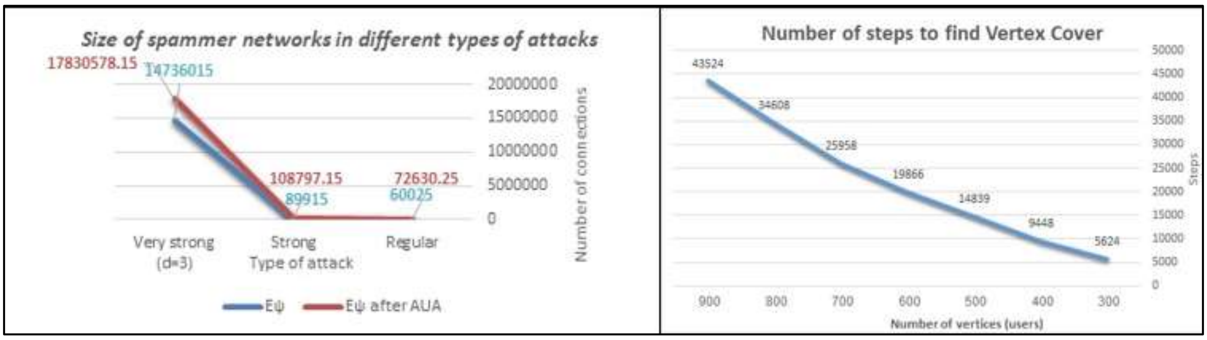


Fig. 23: Left - spammer network sizes of different attacks; Right - optimized attack complexity

We can see that the number of steps required to find the minimal vertex cover is very high relative to the size of the network being attacked. The implementation of the model, with these relevant threshold values for the parameters, is meant to be performed by the OSN administration per each user's network.

8.3. Context-based model

The first experiment tests the accuracy of estimating the trust per context using the trust of the actions (equations 19 and 21). For this experiment the trust of the actions is given by the ego user himself so it can be considered quite accurate. For this part of the experiment, we used a dataset of a real Facebook network of 917 users, which are the direct friends of a single ego user. The ego user first collected all of the users' data relevant to the basic *UTV* - $p_{MF,}\, p_{TF,}\, p_{AUA,}$ and $p_{FD.}$. We then calculated every *UTV* accordingly. There were five different data categories (the κ categories) that were chosen by the ego user, and for each category four actions (posts, shares, etc.) were documented by the user – a total of 20 actions per user. This evaluation was done on all of the users in the experiment. The ego user then gave a specific trust estimation for each friend in every category (STV_κ). He then went over every action and gave his trust estimation for every action in every category ($STV_i A_\kappa$). (This seems like a biased ranking by the ego user itself, later on we replace this ranking by the results of the Sentiment Analysis experiment). A friend's action may reduce or elevate the ego user's trust in a certain category. After these trust estimations, we calculated $STVA_\kappa$ as described in Equation 19. After gathering and calculating all the values, we reached UTV_κ per each category per each user, as described in Equation 20.

The comparison that was done is to the STV_κ value, since our model tries to predict the real trust estimation of the ego user for everyone in his network in different categories. We then divided the users into groups according to their *UTV*:

- *UTV* = 0.7-0.8: relatively low trusted users.
- *UTV* = 0.8-0.9: medium trusted users.
- *UTV* = 0.9-1: high trusted users.

There were no users below *UTV*= 0.7, a fact that suggests that this is a relatively strong network. The purpose of this division was to examine the effect of our model's estimation on different types of users. The results of the average UTV_κs and STV_κs of the different types are presented in Table 9. The division of ranges (0.7-0.8 etc.) was done as a relatively equal scatter of the results to reflect the different levels of basic trust for the users. These results shown in the table are based on the results of the computations done on the raw data according to equations 19 and 20. For example, we have a user (serial id 12) in the data that has a *UTV* of 0.81, and in the topic of elections he has a STV_κ of 0.92 – accordingly, his UTV_κ in elections will be 0.865. Another example is the user with serial id 19, who has a *UTV* of 0.85, and in the topic of online shopping he has a STV_{κ} of 0.93 – accordingly, his UTV_{κ} in online shopping will be 0.89. It is important to notice in the table that there is a difference between *UTV*, which is the basic trust value, and UTV_{κ}, which is the computed context-based trust value, that is affected by STV_{κ} (usually lower values).

In Fig. 24 we can see a graphic representation of the differences between the UTV_{κ}s and STV_{κ}s in the different groups. We can see that the UTV_{κ}s are generally higher than the STV_{κ}s. One of the reasons for this is that the network is relatively strong (high *UTV* values). Accordingly, if we take a closer look we can see that these differences are even stronger in the high trusted users (0.9-1).

In all three groups we can see that the estimated trust values that are based on the actions (*UTVk*) are a bit higher than the values that were originally estimated by the ego user. This can be explained by the fact that people are more judgmental towards close friends and people they highly trust. (Campbell, Sedikides, Reeder, & Elliot, 2000). These close friends also usually appear more on the OSN feed, thus having a larger probability of judgment of their opinions from the ego user.

Table 9 - Experimental results for the first part of the context model.

UTV 0.7-0.8 relatively low trusted users				
$\mathbf{UTV_{\kappa}}1$	$\boldsymbol{UTV_{\kappa}}2$	$\boldsymbol{UTV_{\kappa}}3$	$\boldsymbol{UTV_{\kappa}}4$	$\boldsymbol{UTV_{\kappa}}5$
0.617	0.62	0.59	0.605	0.608
$\boldsymbol{STV_{\kappa}1}$	$\boldsymbol{STV_{\kappa}2}$	$\boldsymbol{STV_{\kappa}3}$	$\boldsymbol{STV_{\kappa}4}$	$\boldsymbol{STV_{\kappa}5}$
0.53	0.465	0.595	0.422	0.443
UTV 0.8-0.9 medium trusted users				
$\boldsymbol{UTV_{\kappa}1}$	$\boldsymbol{UTV_{\kappa}}2$	$\boldsymbol{UTV_{\kappa}}3$	$\boldsymbol{UTV_{\kappa}}4$	$\boldsymbol{UTV_{\kappa}}5$
0.665	0.642	0.658	0.657	0.641
$\boldsymbol{STV_{\kappa}1}$	$\boldsymbol{STV_{\kappa}2}$	$\boldsymbol{STV_{\kappa}3}$	$\boldsymbol{STV_{\kappa}4}$	$\boldsymbol{STV_{\kappa}5}$
0.43	0.494	0.586	0.442	0.487
UTV 0.9-1 high trusted users				
$\boldsymbol{UTV_{\kappa}1}$	$\boldsymbol{UTV_{\kappa}}2$	$\boldsymbol{UTV_{\kappa}}3$	$\boldsymbol{UTV_{\kappa}}4$	$\boldsymbol{UTV_{\kappa}}5$
0.716	0.721	0.722	0.723	0.717
$\boldsymbol{STV_{\kappa}1}$	$\boldsymbol{STV_{\kappa}2}$	$\boldsymbol{STV_{\kappa}3}$	$\boldsymbol{STV_{\kappa}4}$	$\boldsymbol{STV_{\kappa}5}$
0.419	0.505	0.593	0.464	0.454

In the second experiment we replaced the trust given by the ego user to each action by the sentiment analysis of this action; thus, it can be computed automatically without the ego user's involvement. For this part of the experiment, we took two datasets, both containing all the parameters of the first part. Besides the parameters of the first part of context evaluation, in these datasets there were specific trust scores for the posts, and their sentiment analysis. Our purpose was to find the effect of sentiment in a post to the user's trust in a certain context. This was done in order to validate the model's contextual trust parameters described in equations 20 and 21 that were presented in Section 5. There was an initial screening of the results, and only the positive sentiment ones were checked, for the reasons explained in the context section of this paper.

The ground truth that we compare to is the STV_{κ} - the subjective trust as estimated by the ego user to a certain user in a certain κ category. The topic chosen for the context was Israeli politics – the two datasets of the second part contained only posts from this topic. The sentiment analysis done on the posts of the first dataset was evaluated and analyzed by a prototype that was built specifically for this purpose, using the Python library of Vader-Sentiment, based on (Hutto & Gilbert, 2014).

The results were conclusive, as can be seen in Fig. 25. We found that there is a considerable effect of sentiment on trust, as we preliminarily assumed.

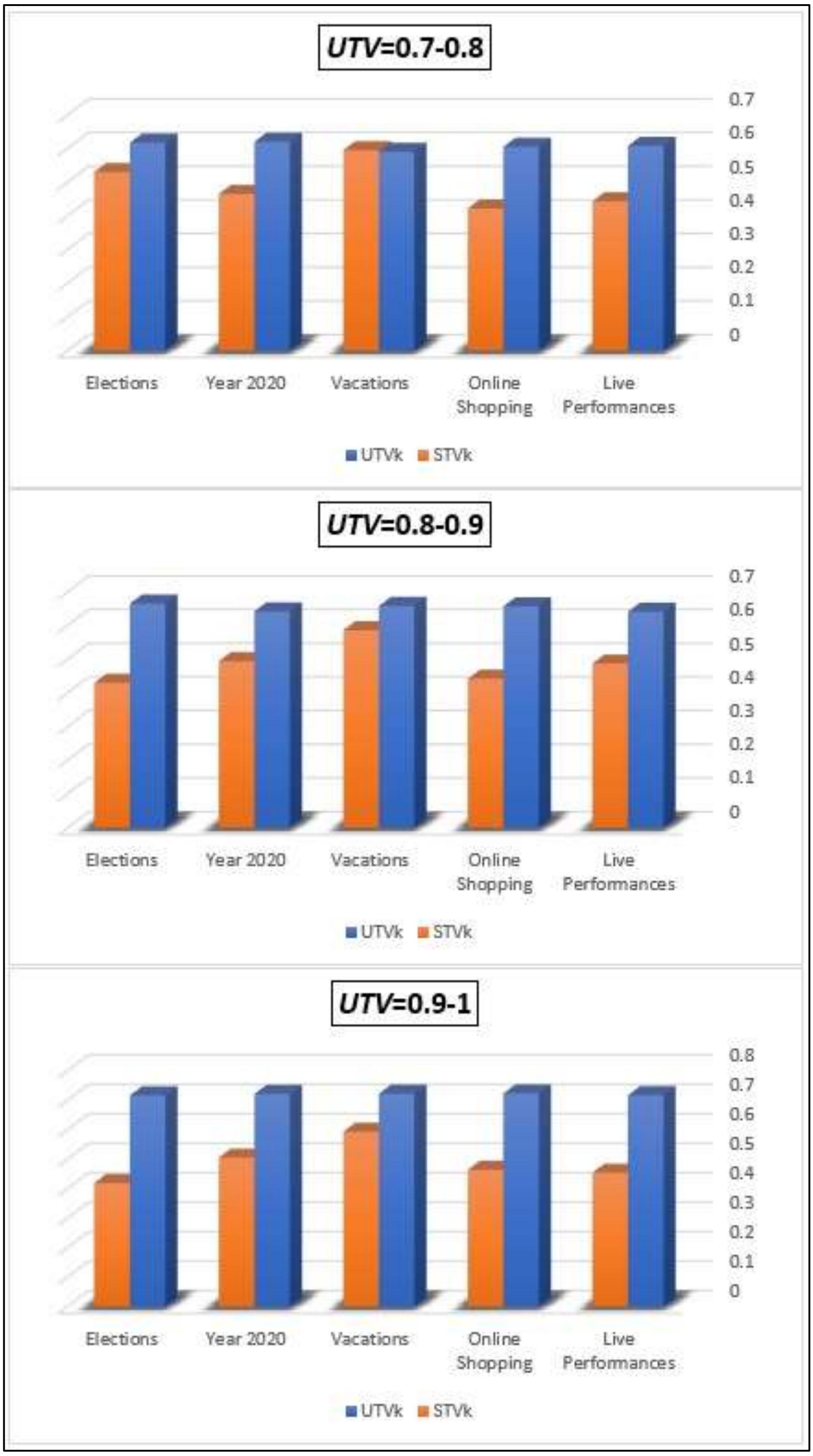


Fig. 24: Comparison of the users' context values in different trust levels

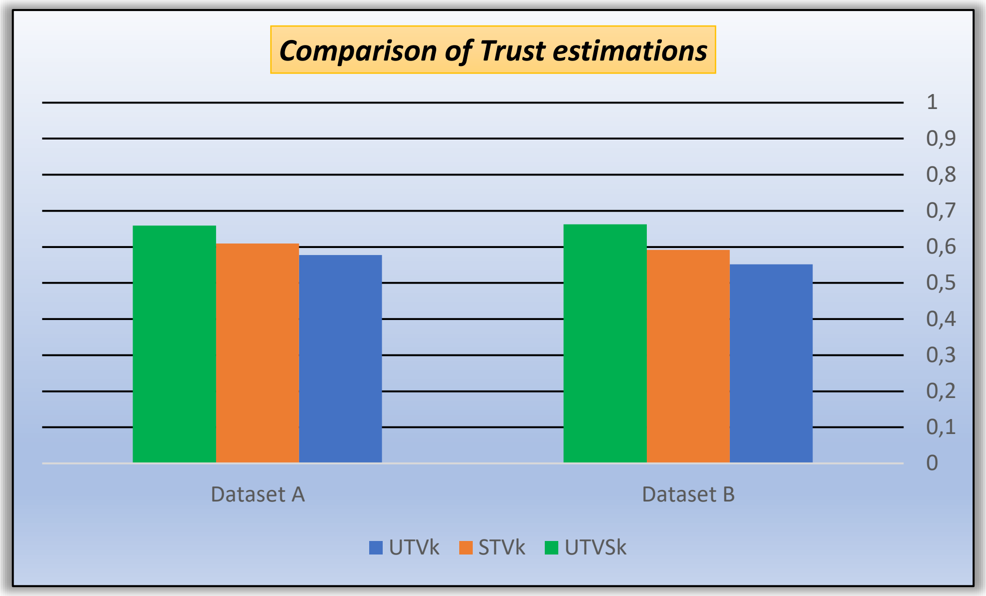


Fig. 25. Comparison of the model's trust values in the experimental evaluation

The results we got are that the trust estimations were close to our ground truth. The UTV_{κ}^{S} (the UTV in category k with the Sentiment Analysis) values were close to the STV_{κ} values, as well as the UTV_{κ}, meaning that the model's addition of the Sentiment Analysis Factor for the Action ($SAF_{i}A_{\kappa}$) is important. In Dataset A STV_{κ} was 0.609, and the UTV_{κ}^{S} was 0.659, while the STV_{κ} was 0.577. In Dataset B STV_{κ} was 0.592, and the UTV_{κ}^{S} was 0.66, while the STV_{κ} was 0.551. These results can, of course, differ in different ego networks and different topics. We can also take into consideration applying different weights to the calculation of the UTV_{κ}s, according to the network and preferences of the ego user, and in topics that are not that controversial or impassioned such as politics, we may get different sentiment results.

The importance of this experiment is that in practice we can avoid asking users to estimate the trust of each action but can instead use sentiment analysis and compute the approximate trust value automatically.

8.4. Fake-news-propagation prevention

For the experimental part of this research, we used a dataset of a real Facebook network of 201 users, which are the direct friends of a single ego user. The ego user first collected all of the users' data relevant to the basic *UTV - pMF, pTF, pAUA*, and *pFD*. We then calculated every *UTV* accordingly.

Next, we needed to find the relevant users and data instances for the κ category, the topic selected was politics, since it was the most discussed topic in the ego network, and usually discussed with strong sentiment. There were 33 relevant users, who had a total of 79 data instances (actions - posts, shares, etc.).

At this point we calculated the sentiment analysis for each of the actions, for the threshold value $SAF_{\alpha}A_{\kappa}$ and the summation of $SAF_{i}A_{\kappa}$ actions per user for the calculation of UTV_{κ}^{j} for user j, as presented in equations 21 and 22. The weights w_i of the factors were 2:1 in favor of the *UTV*, since its *TF* attribute that was discussed above has considerable importance for fake news propagation. This division can, of course, be altered in different circumstances or algorithmic decisions.

As mentioned in the previous section, our purpose is to find the most refined thresholds that will give us the highest probability of detecting potential fake news propagators in comparison with the ground truth of the fact checker. We created software that will help us analyze the data in the sentiment analysis part. The program was written in Python and used the Vader Sentiment library for sentiment analysis of the Facebook posts, that were scraped manually, due to the Facebook crawling restrictions. Out of 79 actions, 9 were discovered to be fake news. All of them had a relatively strong negative sentiment. Most of the users that spread this data were of relatively lower trust values than the others. At this point the training of the learning process begun, and the initial state of the system was set to the default average values: 0.62 for UTV_{κ}^{j} and -0.33 for $SAF_{\alpha}A_{\kappa}$, meaning that for users that have a Trust value lower than 0.62 and their action has $SAF_{\alpha}A_{\kappa}$ lower than -0.33, it is predicted to be of a false nature.

The results for such an initial state were, of course, relatively low: 4 out of 9 fakes were discovered. Out of the total of 34 actions that passed the criteria, these 4 indicate a success ratio of just 12%.

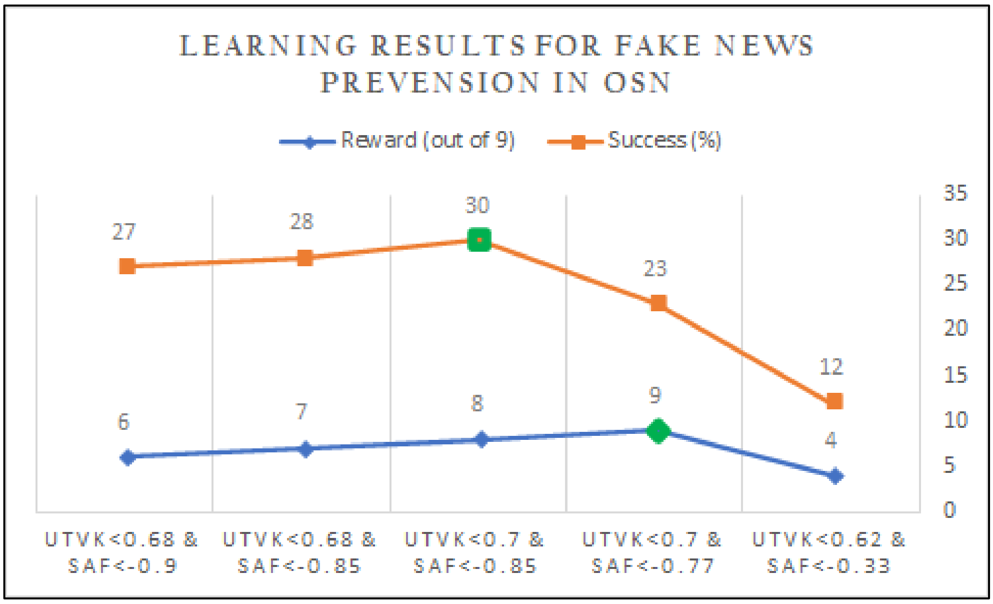


Fig. 26. Learning results for different states of UTV_{κ}^{S} & $SAF_{\alpha}A_{\kappa}$

We can see the results of the different states after the continuation of the training in Fig. 26. There are two different aspects: the reward aspect, exclusively referring to the proven fake news indicators, and the success ratio aspect, also referring to the ones that pass the criteria and are not proven fake. If we wish to optimize the system with respect to the actual proven fake news indicators, most rewards gotten (9 out of 9) are in the state of UTV_{κ}^{j} < 0.7 and $SAF_{\alpha}A_{\kappa}$<-0.77.

If we wish to refer to success ratios, as explained above, the best ratio (30%) is achieved in the state of UTV_{κ}^{j} < 0.7 and $SAF_{\alpha}A_{\kappa}$<-0.85, meaning that users that are within the boundaries of these values have a 30% chance of being fake news propagators.

8.5. Evaluation of the GDPR compliance model

Two experiments were conducted to validate the two approaches related to privacy preserving data dispersion data erasure in OSN. For the data dispersion part, we took three different ego networks and requested each ego user to take 50 actions that involve another person (a friend from the network) from their Facebook activity log. We then asked the ego users to assign every action to a category (like share, comment on a post, etc.), and to state the level of closeness to the friend involved in this action on a scale of 0 to 10 - 0 meaning a stranger (maybe with just a bit of familiarity), and 10 meaning the closest friend (maybe a childhood friend or a close family member).

The relevant data category for these experiments is governed data with leakage, since the ego interacts with its network; hence, the data is shared, and we wish to check if there is a dispersion to undesired users. Finally, we asked the ego user to answer the following question with respect to each action: "If you had to consent before performing this action, that you have joint ownership of the data of this action with the friend involved, thus never being able to erase it by yourself, would you still perform this action?". This question gives us an important estimation of the actual need to better control the data instances in the OSN according to the relevant GDPR parts, as the need for better controlling of data during its process arises. We asked the user to answer on a scale of 0 – 10 where 0 means YES and 10 means NO. We can see the results in Table 10; all the datasets are for non-atomic actions, and we can see their graphic representation in Fig. 27. These experiments were done as a continuation of the two previous experiments that only examined contextual trust, without relating to specific data instances and analyzing them. We can see that there is a correlation between the level of closeness to the friend and the level of consent on performing an action in the OSN. We can see that the lowest average score for closeness is given in Dataset No.2 (4.97) and accordingly, the average level of consent is the lowest (6.52). As for the erasure part, we wanted to find out how efficient it would be to reduce the amount of data needed to be checked, by using only the relevant part of the ego network, meaning we check only the subset of the active users (the ones that created data or acted upon data) from the subset of the trustworthy users (above a certain MTV_{κ}) in a certain category. For this purpose, we took two different ego networks, both in the context topic of politics (in these networks this topic was very relevant and had significant traffic). We checked the ego networks for trustworthy friends in the category of politics ($MTV_{politics} > 0.8$). After this screening we examined this sub-network and looked for only active users who posted or acted on data relating to politics.

Table 10 - Results of consent from option on Facebook activities.

Dataset	*No. of actions*	*Avg. connection closeness level with the friend*	*Avg. consent on performing the action with a consent form*
No.1	36	8.05	8.42
No.2	33	4.97	6.52
No.3	50	8.5	8.22

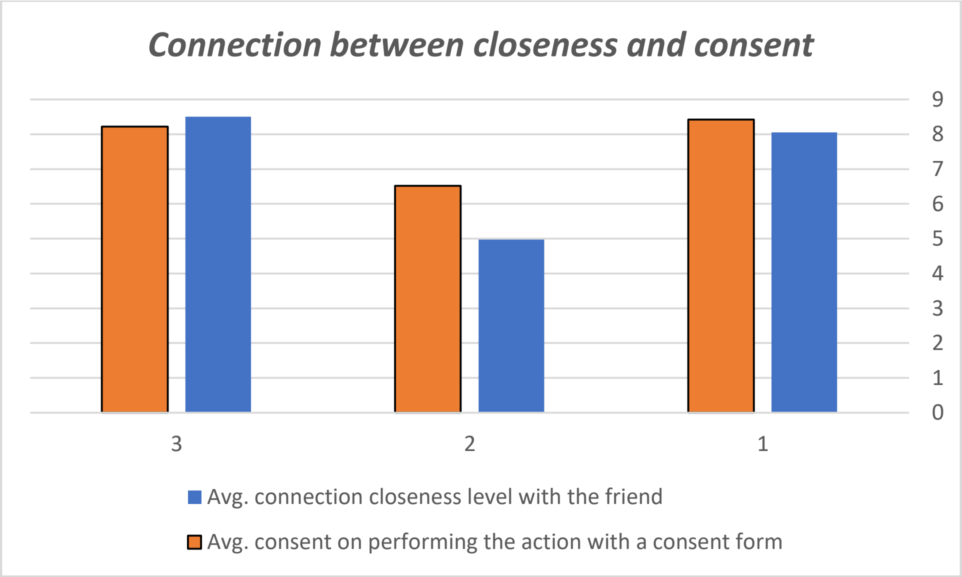


Fig. 27: Results of an anti-vaccination page of the ML fake news analyzer

The results are shown in Fig. 28; we can see the substantial differences in the number of users that are relevant (trustworthy and active) in relation to the entire network. This choice is very efficient: for example, the difference in searching 238 users instead of 933 users, and then erasing data from only 36 users in Network A.

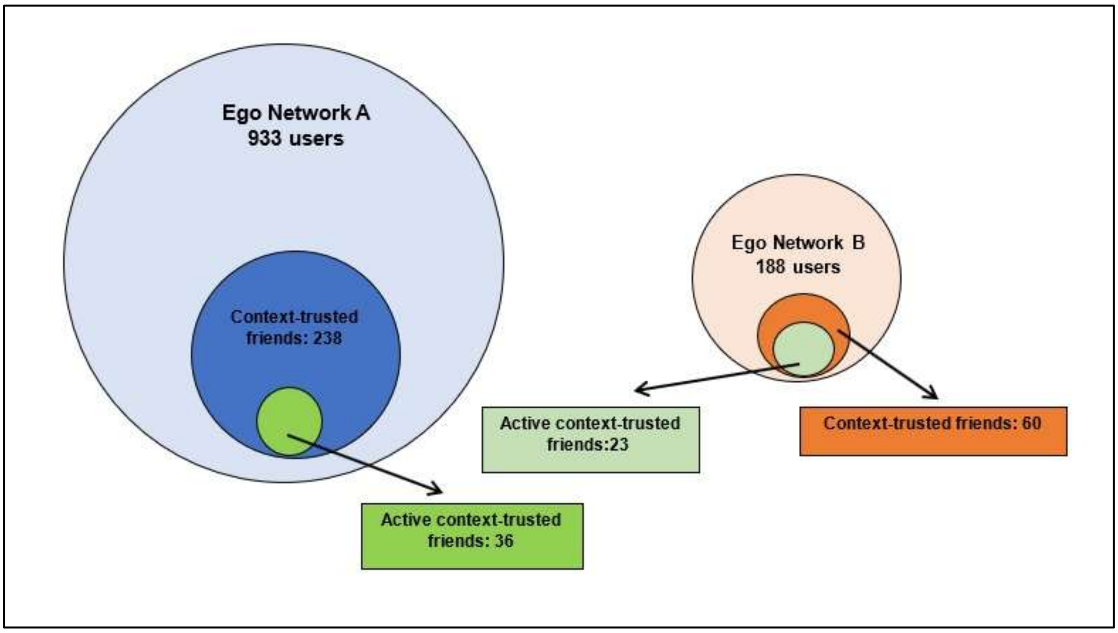

Ego Network A
933 users
Context-trusted friends: 238
Active context-trusted friends: 36
Ego Network B
188 users
Active context-trusted friends:23
Context-trusted friends: 60

Fig. 28: Results of the context phase experiment in sizes of the active sub-networks

8.6 Discussion

In this section we described our experimental evaluation results for each part of our model. We like to note the problem of performing large scale experiments with thousands of users, with OSN that protects the privacy of user actions (the problem is especially difficult with Facebook API and may be less difficult with Twitter and other networks), yet the experiments we succeeded to perform indicate the validity of our models.

9. Summary and future directions

9.1. Summary

In the first part of this research, we presented a combined access and information-flow control model for privacy in OSN. The novelty of our model lies in the combination of user-trust attributes, based on real OSN characteristics, and information-flow control in an RBAC, that usually grants permission solely to roles, thereby improving the privacy features of the network. The problem of data leakage in OSN has many aspects and applications. A major comprehensive solution may not be possible yet, but accurate analysis that contains several main parameters is a good step towards it. The attributes of this model and their values were carefully selected based on previous research and shown to improve information-sharing decisions in OSNs.

The information flow aspect of the model is used for creating a trustworthy network of users and gives a good privacy infrastructure for such a solution. We have used known graph algorithms (Kruskal for finding the MST, Dinic for finding all of the paths from a source node to a target node), along with the combination of several user and connection trust attributes, to find the weak security edges in such an OSN graph, to identify possible adversaries, and to create a stronger, more viable trusted sub-graph, in which users can be relatively safe in terms of information sharing. We have conducted a thorough experimental evaluation and presented the results in comparison to our model's results and showed a satisfactory resemblance in the decision-making application of it. This validation was done for every phase of the model, from the value assignments of the model up to the comparison of the different information flow aspects of it. Our proposed model supports dynamic information flow and access control decisions, which are essential since attribute values may change over time - the user gains or loses friends, the age of a user account grows over time, etc. As in many trust-related access control models, the 'cold start' affects new OSN users, since their trust parameter values, such as *AUA* and *TF*, are very low, even though they could be legitimate users who will be mistaken for fake profiles or spammers. For these specific cases of new users, we can remedy the problem by giving extra weight to the Outflow/Inflow Ratio (*OIR)* attribute, since spammers and bots have a very low value of *OIR* (they mainly outflow data, and rarely inflow), while genuine new user profiles have a high or moderate *OIR* value.

The first extension of the basic trust model handles attack scenarios on the model. The problem of attacks by malicious users in OSN has many aspects and applications. Using several aspects in the comprehensive trust-based model that was presented in this paper is a genuine necessity for OSN privacy.

In this research we have established the strength of the comprehensive model by analyzing the possible attack scenarios of creating a spammer community that may contaminate the model's raw attributes. These attributes are hard to fake since they are built on real OSN user presence and real numerical assets. The comprehensive coverage of access control, flow control and trust provides a solid infrastructure for OSN privacy. We have simulated several attack scenarios based on the preliminary evaluations of the properties from our previous research, and show that the effort required by the attacker makes these attacks infeasible.

The second extension of the basic trust model handles context for the purpose of refining the model, making it more accurate and closer to real–life.

In this context and sentiment part of our model we create a safer environment for the ego user, as users that are less trustworthy in different contexts will be allotted from the data cycle. The experimental part of this paper gives us a comprehensive view of the user's subjective trust value, for the purpose of privacy preservation in the ego network. The less trusted users in different contexts are the ones that the ego user will be most likely to prefer not to show his data to in certain categories. The sentiment analysis on the datasets provided an important validation of the model, since it can be performed automatically on the set of relevant actions. We can see that the course of actions in the OSN can affect the users' trust values. The posts and comments and shares can change the trust values of a certain user, and even a single action can affect this value. We can see that positive actions effect the user's trust.

The third extension of the basic trust model handled is connected to the context part, and it uses the model with ML methods to detect fake news potential propagators and to prevent the dispersion of Fake News in the network.

In this part of the research, we presented a trust-based model that uses context and user evaluation for preventing the propagation of fake news.

These aspects, which include important features of the network data, are very strong in terms of OSN sizes of real user networks. These attributes are hard to fake since they are built on real OSN user presence and real numerical assets. To these aspects we added a reinforcement learning model that helps discover the numerical criteria of users and their actions that can be focused on fake news propagators. After applying this model on the OSN, the users that are marked as potential fake news propagators can be allotted from the ego network flow in a certain category, or, in general, depending on the user preference or on other reasons. We showed the results of our experimental evaluation on a real Facebook ego network, and demonstrated the training of the model and its results, accordingly.

The fourth and last extension of the basic trust model handled is also connected to the context part: we handled the subject of using our model for the compliance of OSN to a part of the GDPR requirements that involve the use of users' private data.

We then showed how the context-based trust model can be used to enforce GDPR. The key objective of the implementation process is to analyze the proper reliable audience for every data instance and to monitor it as it spreads in the social network. We discovered that sentiment has a major effect on contextual trust when it comes to the evaluation of a subjective opinion on a certain post. Users tend to want to see opinions similar to theirs, and when it comes to being more positive about it, the trust rises.

The advantage of the model over other solutions is that non-trustworthy users will be allotted from the data cycle; thus, we reduce the risk of copyright infringement by unauthorized data distributions. The control over the user's data and its monitoring give us the ability to adhere to the GDPR in social networks, hence enabling implementation of the regulation's important feature of the right-to-be-forgotten. If we look at the system's influence on the freedom of information, gaining contextual trust may cause isolation from information that one does not agree with. As mentioned above, to address this problem we can give different weights to the sentiment analysis factor and the basic *UTV*. If there is a preference for also seeing posts and data that do not necessarily adhere to the echo chamber we created, we can reduce this influence by changing the desired weights, hence giving more balance to different opinions in our network.

For example, if the user wished to have access to and to distribute political opinions contrary to those of other users, this requirement can be accommodated for by this different division of parameter weights. The novel issue of handling non-atomic data instances, in accordance with these regulations, gives us a solid solution for shared ownership data assets that are implemented by a consent mechanism.

9.2. Future directions

This research has presented a comprehensive trust-based model for OSNs. All of the model's parts give a viable solution for several unhandled privacy and security issues that social networks are coping with nowadays. Adopting this trust-based approach, in all of its aspect, can provide a better privacy and security infrastructure that will give a better user experience for private users, as well as for organizational ones, and also will help OSN administration to solve some of the more important issues that involve users' data, its processing and its protection. The model helps to create a security infrastructure that can be adapted to users' security needs, thus achieving a much more viable, feasible and accurate privacy model suitable for different kinds of OSN users.

In future work we wish to evaluate the model on additional networks, besides Facebook. Different networks have other privacy issues, and in some cases the data usage is very different. Reddit, Discord, LinkedIn, TikTok, Twitter, are all very different from Facebook. We wish to adapt our model to suit those different networks, in accordance with their unique features. An important direction is fake news prevention in other networks, especially Twitter, because of its open and political characteristics. A lot of fake news traffic happens today in Twitter, but there is no real privacy because of its public nature. For this purpose, we need to adapt our model for every network that acts in a different way. Future work may also improve the ML results and the sentiment analysis results. Furthermore, the thresholds for trust value calculation for non-accounts could be explored. For ML, it is possible to create a larger and more accurate dataset, by collecting more data and reviewing the labels with experts from the field.

Regarding sentiment analysis, it is evident that our method of using translation and analyzing in English has limited results. Currently sentiment analyzers in Hebrew are not abundant, and they are either difficult to use or produce poor results. Work on such analyzers continues to be developed and improves these tools, so hopefully in the future one could perform more accurate sentiment analysis.

In terms of preventing attacks on the model, we can adjust our model for categorizing and analyzing data sensitivity and user psychological profiling. These two aspects have the objective of classifying the proper audience for a data instance. The model can become more context sensitive and user sensitive, making it even more resilient to attacks. In future work we will evaluate the extended models using larger datasets and large sets of users, which we can collect from other networks such as Twitter (Hoaxy Twitter Database), Discord (Kaggle data for Discord) and more.

References

Voloch, N., Gal-Oz, N., & Gudes, E. (2021). A Trust based Privacy Providing Model for Online Social Networks. *Online Social Networks and Media, 24,* 100138.

Voloch, N., Gudes, E., & Gal-Oz, N. (2022) A context-based trust model for OSN and its use for enforcing GDPR. *Online Social Networks and Media.* Elsevier. *Submitted.*

Gudes E., Voloch N. (2018) 'An Information-Flow Control Model for Online Social Networks Based on User-Attribute Credibility and Connection-Strength Factors'. In: Dinur I., Dolev S., Lodha S. (eds) *Cyber Security Cryptography and Machine Learning. CSCML 2018. Lecture Notes in Computer Science, vol 10879.* Springer, Cham

Voloch, N. & Gudes, E. (2019). ' An MST-based information flow model for security in Online Social Networks'. In The IEEE 11th International Conference on Ubiquitous and Future Networks (ICUFN19). Springer, Cham.

Voloch, N., Levy, P., Elmakies, M., & Gudes, E. (2019A, June). 'An Access Control Model for Data Security in Online Social Networks Based on Role and User Credibility'. In The international Symposium on Cyber Security Cryptography and Machine Learning (pp. 156-168). *Lecture Notes in Computer Science, vol 11527.* Springer, Cham.

Voloch, N., Levy, P., Elmakies, M., & Gudes, E. (2019B). 'A Role and Trust Access Control model for preserving privacy and image anonymization in Social Networks'. In 13th IFIP WG 11.11 International Conference on Trust Management (IFIPTM'19). Springer, Cham.

Voloch N., Gudes E., & Gal-Oz N. (2021A) Analyzing the Robustness of a Comprehensive Trust-Based Model for Online Social Networks Against Privacy Attacks. In: Benito R.M., Cherifi C., Cherifi H., Moro E., Rocha L.M., Sales-Pardo M. (eds) Complex Networks & Their Applications IX. COMPLEX NETWORKS 2020 2020. *Studies in Computational Intelligence, vol 943*. Springer, Cham.

Voloch, N., Gudes, E., & Gal-Oz, N. (2021B, July). Implementing GDPR in Social Networks Using Trust and Context. In International Symposium on Cyber Security Cryptography and Machine Learning (pp. 497-503). Springer, Cham.

Voloch, N., Gudes, E., & Gal-Oz, N. (2021C). Preventing Fake News Propagation in Social Networks using a context Trust-based security model. 15th International Conference on Network and System Security (NSS21).

Voloch, N., Gudes, E., Gal-Oz, N., Mitrany, R., Shani, O., & Shoel, M. (2022) Detecting and anlayzing Fake News in Social Networks using Trust, Context and Machine Learning. In International Symposium on Cyber Security Cryptography and Machine Learning. *CSCML 2022.*

Kayes, I., & Iamnitchi, A. (2017). "Privacy and security in online social networks: A survey". *Online Social Networks and Media,* 3, 1-21.

Li, Y., Li, Y., Yan, Q., & Deng, R. H. (2015). "Privacy leakage analysis in online social networks". *Computers & Security, 49,* 239-254.

Misra, G., & Such, J. M. (2016). "How socially aware are social media privacy controls?" *Computer, 49(*3), 96-99.

Sayaf, R., & Clarke, D. (2012) "Access control models for online social networks". *Social Network Engineering for Secure Web Data and Services*, 32-65.

Hirschprung, R., Toch, E., Schwartz-Chassidim, H., Mendel, T., & Maimon, O. (2017). Analyzing and optimizing access control choice architectures in online social networks. *ACM Transactions on Intelligent Systems and Technology (TIST), 8(4*), 57.

Ranjbar, A., & Maheswaran, M. (2014). "Using community structure to control information sharing in online social networks". *Computer Communications, 41,* 11-21.

Kruskal, J. B.. (1956) "On the shortest spanning subtree of a graph and the traveling salesman problem". *Proceedings of the American Mathematical Society. 7:* 48–50.

Heatherly, R., Kantarcioglu, M., & Thuraisingham, B. (2012). "Preventing private information inference attacks on social networks". IEEE Transactions on Knowledge and Data Engineering, 25(8), 1849-1862.

Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)" OJ L 119, 4.5.2016, p. 1‑88

Kotsios, A., Magnani, M., Vega, D., Rossi, L. and Shklovski, I., 2019. An analysis of the consequences of the general data protection regulation on social network research. ACM Transactions on Social Computing, 2(3), pp.1-22.

Peinado, M., Abburi, R., & Bell, J. R. (2006). U.S. Patent No. 7,024,393. Washington, DC: U.S. Patent and Trademark Office.

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). "Role-based access control models". *Computer,* 29(2), 38-47.

Lavi, T. and Gudes, E. (2016). "Trust-based Dynamic RBAC." *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP),* pp. 317-324.

Patil, V. T., & Shyamasundar, R. K. (2017). "Undoing of privacy policies on Facebook". *IFIP Annual Conference on Data and Applications Security and Privacy* (pp. 239-255). Springer, Cham.

Cheng, Y., Park, J., & Sandhu, R. (2012) "Relationship-based access control for online social networks: Beyond user-to-user relationships". *Privacy, Security, Risk* *and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom*) (pp. 646-655). IEEE.

Fong, P. W. (2011). "Relationship-based access control: protection model and policy language." *The first ACM conference on Data and application security* and privacy (pp. 191-202ACM.

Crampton, J., & Sellwood, J. (2014) "Path conditions and principal matching: a new approach to access control." *The 19th ACM symposium on Access control models and technologies* (pp. 187-198). ACM.

Kumar, A., & Rathore, N. C. (2016). "Relationship Strength Based Access Control in Online Social Networks". *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems*: Volume 2 (pp. 197-206). Springer, Cham.

Levy, S., Gudes, E., & Gal-Oz, N. (2016). "Sharing-habits based privacy control in social networks". IFIP Annual Conference on Data and Applications Security and Privacy (pp. 217-232). Springer, Cham.

Squicciarini, A. C., Paci, F., & Sundareswaran, S. (2014). "PriMa: a comprehensive approach to privacy protection in social network sites." annals of telecommunications-annales des télécommunications, 69(1-2), 21-36.

Bahri, L., Carminati, B., & Ferrari, E. (2018). "Decentralized privacy preserving services for online social networks." Online Social Networks and Media, 6, 18-25.

Cohen, Y., Gordon, D., & Hendler, D. (2018). "Early detection of spamming accounts in large-Scale service provider networks." *Knowledge-Based Systems*, *142*, 241-255.

Benevenuto, F., Magno, G., Rodrigues, T., & Almeida, V. (2010). "Detecting spammers on twitter." *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)* (Vol. 6, No. 2010, p. 12).

Zheng, X., Zeng, Z., Chen, Z., Yu, Y., & Rong, C. (2015). "Detecting spammers on social networks." Neurocomputing, 159, 27-34.

Viswanath, B., Post, A., Gummadi, K. P., & Mislove, A. (2011). "An analysis of social network-based sybil defenses". *ACM SIGCOMM Computer Communication Review*, 41(4), 363-374.

Fogues, R. L., Murukannaiah, P. K., Such, J. M., & Singh, M. P. (2017). "SoSharP: Recommending Sharing Policies in Multiuser Privacy Scenarios." IEEE Internet Computing, (6), 28-36.

Das, S., Eğecioğlu, Ö., & El Abbadi, A. (2010). "Anonymizing weighted social network graphs". *2010 IEEE 26th International Conference on Data Engineering* (ICDE 2010) (pp. 904-907). IEEE.

Tassa, T., & Cohen, D. J. (2013). "Anonymization of Centralized and Distributed Social Networks by Sequential Clustering." *IEEE Trans. Knowl. Data Eng.*, 25(2), 311-324.

Lin, D., Steiert, D., Morris, J., Squicciarini, A., & Fan, J. (2019). "REMIND: Risk Estimation Mechanism for Images in Network Distribution." IEEE Transactions on Information Forensics and Security, 15, 539-552.

Ali, B., Villegas, W., & Maheswaran, M..(2007). "A trust based approach for protecting user data in social networks." *Proceedings of the 2007 conference of the center for advanced studies on Collaborative research* (pp. 288-293). IBM Corp.

Wang, R. H., & Sun, L. (2010) "Trust-involved access control in collaborative open social networks". *Network and System Security (NSS),* 2010 4th International Conference on(pp. 239-246). IEEE.

Lucas, M. M., & Borisov, N. (2008). "Flybynight: mitigating the privacy risks of social networking". *Proceedings of the 7th ACM workshop on Privacy in the electronic society* (pp. 1-8). ACM.

Gross, R., & Acquisti, A. (2005). "Information revelation and privacy in online social networks*". Proceedings of the 2005 ACM workshop on Privacy in the electronic society* (pp. 71-80). ACM.

Misra, G., Such, J. M., & Balogun, H. (2016). "IMPROVE-Identifying Minimal PROfile VEctors for similarity-based access control". *Trustcom/BigDataSE/I SPA, 2016 IEEE* (pp. 868-875). IEEE.

Jeh, G., & Widom, J. (2002, July). "SimRank: a measure of structural-context similarity." In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 538-543). ACM.

Squicciarini, A., Karumanchi, S., Lin, D., & Desisto, N. (2014). "Identifying hidden social circles for advanced privacy configuration." Computers & Security, 41, 40-51.

Laleh, N., Carminati, B., & Ferrari, E. (2016). "Risk assessment in social networks based on user anomalous behaviors." IEEE Transactions on Dependable and Secure Computing, 15(2), 295-308.

Taheri-Boshrooyeh, S., Küpçü, A., & Özkasap, Ö. (2015, June). "Security and privacy of distributed online social networks." In *2015 IEEE 35th international conference on distributed computing systems workshops* (pp. 112-119). IEEE.

Boshrooyeh, S. T., Küpçü, A., & Özkasap, Ö. (2018, May). "PPAD: Privacy preserving group-based advertising in online social networks." In *2018 IFIP Networking Conference (IFIP Networking) and Workshops* (pp. 1-9). IEEE.

Boshrooyeh, S. T., Küpçü, A., & Özkasap, Ö. (2020). "Privado: Privacy-Preserving Group-based Advertising using Multiple Independent Social Network Providers." *ACM Transactions on Privacy and Security (TOPS), 23*(3), 1-36.

Li, Z., Shen, H., & Sapra, K. (2012). "Leveraging social networks to combat collusion in reputation systems for peer-to-peer networks". IEEE Transactions on Computers, 62(9), 1745-1759.

Sun, J., Zhu, X., & Fang, Y. (2010, March). "A privacy-preserving scheme for online social networks with efficient revocation." In 2010 Proceedings IEEE INFOCOM (pp. 1-9). IEEE.

Viswanath, B., Bashir, M. A., Crovella, M., Guha, S., Gummadi, K. P., Krishnamurthy, B., & Mislove, A. (2014). "Towards detecting anomalous user behavior in online social networks". In 23rd {USENIX} Security Symposium ({USENIX} Security 14) (pp. 223-238).

Sirur, S., & Muller, T. (2019, July). "The Reputation Lag Attack. "In IFIP International Conference on Trust Management (pp. 39-56). Springer, Cham.

Lee, K., Caverlee, J., & Webb, S. (2010, April). "The social honeypot project: protecting online communities from spammers." In Proceedings of the 19th international conference on World wide web (pp. 1139-1140).

Paradise, A., Shabtai, A., Puzis, R., Elyashar, A., Elovici, Y., Roshandel, M., & Peylo, C. (2017). "Creation and management of social network honeypots for detecting targeted cyber attacks." IEEE Transactions on Computational Social Systems, 4(3), 65-79.

Huber, M., Mulazzani, M., Weippl, E., Kitzler, G., & Goluch, S. (2011). "Friend-in-the-middle attacks: Exploiting social networking sites for spam." IEEE Internet Computing, 15(3), 28-34.

Wang, Y., Lei L., & Guanfeng L. (2015) "Social context-aware trust inference for trust enhancement in social network based recommendations on service providers." World Wide Web 18.1: 159-184.

Du, R., Yu, Z., Mei, T., Wang, Z., Wang, Z., & Guo, B.. (September, 2014), "Predicting activity attendance in event-based social networks: Content, context and social influence." In Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing . pp. 425-434.

Sara, H., Tassa T. & Bonchi, F. (2016) "Individual privacy in social influence networks." Social Network Analysis and Mining 6.1 : 2.

Atallah, M. J., McDonough, C. J., Raskin, V., & Nirenburg, S. (2000, September). "Natural language processing for information assurance and security: an overview and implementations". In NSPW (pp. 51-65).

Tsoumas, B., & Gritzalis, D. (2006, April). "Towards an ontology-based security management." In 20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06) (Vol. 1, pp. 985-992). IEEE.

Louis, A. (2016). Natural language processing for social media.

Hutto, C. J., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Eighth international AAAI conference on weblogs and social media.

Zhang, L., Wang, S. and Liu, B., (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), p.e1253.

Xue, W. and Li, T., (2018, July). Aspect Based Sentiment Analysis with Gated Convolutional Networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2514-2523).

Blitzer, J., Dredze, M. and Pereira, F., (2007, June). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Proceedings of the 45th annual meeting of the association of computational linguistics (pp. 440-447).

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y. and Potts, C., (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language.

Granatyr, J., Botelho, V., Lessing, O.R., Scalabrin, E.E., Barthès, J.P. and Enembreck, F., (2015). Trust and reputation models for multiagent systems. ACM Computing Surveys (CSUR), 48(2), pp.1-42. processing (pp. 1631-1642).

Mokhtari, E., Noorian, Z., Ladani, B.T. and Nematbakhsh, M.A., (2011, May.) A context aware reputation-based model of trust for open multi-agent environments. In Canadian Conference on Artificial Intelligence (pp. 301-312). Springer, Berlin.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). "Fake news detection on social media: A data mining perspective." ACM SIGKDD Explorations Newsletter, 19(1), 22-36.

Tandoc Jr, E. C., Lim, Z. W., & Ling, R. (2018)."Defining 'fake news' A typology of scholarly definitions." Digital journalism, 6(2), 137-153.

Kumar, K. K., & Geethakumari, G. (2014). "Detecting misinformation in online social networks using cognitive psychology." Human-centric Computing and Information Sciences, 4(1), 14.

Levi, O., Hosseini, P., Diab, M., & Broniatowski, D. A. (2019). "Identifying Nuances in Fake News vs. Satire: Using Semantic and Linguistic Cues." arXiv preprint arXiv:1910.01160.

Vosoughi, S., Roy, D., & Aral, S. (2018). "The spread of true and false news online." Science, 359(6380), 1146-1151.

Li, Y., Li, Y., Yan, Q., & Deng, R. H. (2015). "Privacy leakage analysis in online social networks". Computers & Security, 49, 239-254.

Choudhary, A., & Arora, A. (2021). "Linguistic feature based learning model for fake news detection and classification." Expert Systems with Applications, 169, 114171.

Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). "Fake news detection on social media using geometric deep learning." arXiv preprint arXiv:1902.06673.

Helmstetter, S., & Paulheim, H. (2018, August). "Weakly supervised learning for fake news detection on Twitter." In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 274-277). IEEE.

Pierri, F., & Ceri, S. (2019). "False news on social media: a data-driven survey." ACM Sigmod Record, 48(2), 18-27.

Cohn-Gordon, K., Damaskinos, G., Neto, D., Cordova, J., Reitz, B., Strahs, B., ... & Papagiannis, I. (2020). {DELF}: Safeguarding deletion correctness in Online Social Networks. In 29th {USENIX} Security Symposium ({USENIX} Security 20).

Patil, V. T., & Shyamasundar, R. K. (2018, December). Efficacy of GDPR’ s Right-to-be-Forgotten on Facebook. In International Conference on Information Systems Security (pp. 364-385). Springer, Cham.

Goldsteen, A., Garion, S., Nadler, S., Razinkov, N., Moatti, Y., & Ta-Shma, P. (2017, June). Brief announcement: A consent management solution for enterprises. In International Conference on Cyber Security Cryptography and Machine Learning (pp. 189-192). Springer, Cham.

Amsterdamer, Y., & Drien, O. (2019, April). PePPer: Fine-Grained Personal Access Control via Peer Probing. In 2019 IEEE 35th International Conference on Data Engineering (ICDE), (pp. 2012-2015). IEEE.

Rodríguez, E., Rodríguez, V., Carreras, A., & Delgado, J. (2009, June). A Digital Rights Management approach to privacy in online social networks. In Workshop on Privacy and Protection in Web-based Social Networks (within ICAIL’ 09), Barcelona.

Marques, J., & Serrão, C. (2013). Improving content privacy on social networks using open digital rights management solutions. Procedia Technology, 9(0), 405-410.

Liu, E., Liu, Z., & Shao, F. (2014). Digital rights management and access control in multimedia social networks. In Genetic and evolutionary computing (pp. 257-266). Springer, Cham.

du Toit, J. (2018). Protecting Private Data Using Digital Rights Management. Journal of Information Warfare, 17(3), 64-77.

Iftikahr, S., Kamran, M., Munir, E.U. and Kahn, S.U. (2017) A reversible watermarking technique for social network data sets for enabling data trust in cyber, physical and social computing, IEEE systems J., Vol. 11, No. 1, March 2017

Dunbar, R. I., (2016)."Do online social media cut through the constraints that limit the size of offline social networks?" *Royal Society Open Science* 3.1 :150292.

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009) "Pearson correlation coefficient". In Noise reduction in speech processing (pp. 1-4). Springer Berlin Heidelberg.

Wiese, J., Kelley, P. G., Cranor, L. F., Dabbish, L., Hong, J. I., & Zimmerman, J. (2011, September). "Are you close with me? are you nearby?: investigating social groups, closeness, and willingness to share." In Proceedings of the 13th international conference on Ubiquitous computing (pp. 197-206). ACM.

Ravi, M. (2016). "Trust and uncertainty in distributed environments : application to the management of data and data sources quality in M2M (Machine to Machine) systems."

Gabow, H. N.; Galil, Z.; Spencer, T.; Tarjan, R. E..(1986) "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs". Combinatorica. 6 (2): 109.

Ford, L. R.; Fulkerson, D. R. (1956). "Maximal flow through a network" . Canadian Journal of Mathematics. 8: 399–404.

Edmonds, Jack; Karp, Richard M. (1972). "Theoretical improvements in algorithmic efficiency for network flow problems". Journal of the ACM. Association for Computing Machinery. 19 (2): 248–264.

Dinic, Y., (1970). "Algorithm for solution of a problem of maximum flow in a network with power estimation". Doklady Akademii nauk SSSR. 11: 1277–1280.

Shrivastava, N., Majumder, A., & Rastogi, R. (2008, April). "Mining (social) network graphs to detect random link attacks." In 2008 IEEE 24th International Conference on Data Engineering (pp. 486-495). IEEE.

Dinur, I., & Safra, S. (2005). "On the hardness of approximating minimum vertex cover." Annals of mathematics, 439-485.

Collomb, A., Costea, C., Joyeux, D., Hasan, O., & Brunie, L. (2014). "A study and comparison of sentiment analysis methods for reputation evaluation." Rapport de recherche RR-LIRIS-2014-002.

Alahmadi, D. H., & Zeng, X. J. (2015, November). "Twitter-based recommender system to address cold-start: A genetic algorithm-based trust modelling and probabilistic sentiment analysis." In 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 1045-1052). IEEE.

Cui, L., Wang, S., & Lee, D. (2019, August). "Same: sentiment-aware multi-modal embedding for detecting fake news." In Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining (pp. 41-48).

Watkins, C. J., & Dayan, P. (1992). "Q-learning". Machine learning, 8(3-4), 279-292.

Campbell, W. K., Sedikides, C., Reeder, G. D., & Elliot, A. J. (2000). "Among friends? An examination of friendship and the self-serving bias." British Journal of Social Psychology, 39(2), 229-239.

Hoaxy Twitter Database, Indiana Unibversity, https://hoaxy.osome.iu.edu/

Kaggle data for Discord, Kaggle, https://www.kaggle.com/datasets/jef1056/discord-data

תקציר

רשתות חברתיות נעשו בשנים האחרונות אמצעי מרכזי של תקשורת ואינטראקציה בין אנשים בכל רחבי העולם. המהות של פרטיות אותגרה במהלך שני העשורים האחרונים בעקבות התפתחויות טכנולוגיות רחבות היקף, המקנות אפשרויות רבות ונראות חברתית לחברים הפעילים ברשת שמשתפים תוכן רב בקהילות מקוונות. בעוד משתמשים פרטיים משתפים תוכן אישי עם חברים ועמיתים, הם לא תמיד מודעים באופן מלא לחשיפה הבלתי מכוונת של המידע שלהם, שנחשב ליחסית פרטי, או אפילו ציבורי באופן חלקי, לגורמים לא רצויים, כגון יריבים (Adversaries), בוטים, משתמשים מזויפים, ספאמרים או קוצרי- מידע. מניעת דליפת המידע הזה היא מטרתם של מודלי אבטחת מידע רבים, רבים שפותחו לרשתות חברתיות באופן מקיף וכוללים אספקטים שונים. חלק מאספקטים אלו הם אמון (Trust) ואמינות (Credibility) של מידע של משתמשים המופיע בהקשרים שונים ברשת, בקרת גישה (Access Control), בקרת זרימה (Information flow control) ומודלים מבוססי יחסים (Relationship-based).

במחקר זה אנו מציעים מודל משולב הכולל הנדרש בכדי להתגבר על החסרונות של גישות קיימות שונות, מודל זה מגן על פרטיות המשתמשים בו בעת עם שמירה על זרימת מידע אמינה ברשת.

חלקו הראשון של מחקר זה, המודל הבסיסי מבוסס- האמון, מורכב משלושה שלבים עיקריים הפונים אל שלושה היבטים משמעותיים: אמון (Trust) בקרת גישה מבוססת- תפקיד (Role-Based Access Control), ובקרת זרימה (Information flow control). מודל זה לוקח בחשבון את תת הרשת של משתמש מסוים וראשית מסווג את קשריו הישירים של המשתמש לתפקידים. חלק זה נסמך על מידע גלוי כגון סך כל החברים של אותו משתמש, גיל חשבון המשתמש (ותק ברשת), ומשך חברות עם חבר מסוים ברשת ומספר החברים המשותפים עמו, הנועדו לאפיין את איכות הקשר עמו. חלק זה של המודל מבצע הערכת אמון בין המשתמש לחבריו ברשת בכדי להעריך אם חברים אלו הם מכרים או יריבים אפשריים בהתבסס על מסלולי זרימת המידע ביניהם. בסופו של תהליך הערכה זה, המודל מספק תמונה

מדויקת יותר לגבי החלטות שיתוף מידע ומאפשר שליטה טובה יותר על פרטיות ברשת החברתית. חלק זה של המחקר מבוסס על מספר גדול של ניסויים שביצענו לחלקים השונים של המודל, חלקם עם רשתות סינטטיות וחלקם על רשתות משתמשים אמיתיות, זאת בכדי להדגים את יכולת המודל לספק למשתמשי הרשת אמצעי טוב לשמירת פרטיותם והגנה על מידע אישי היכול להיות בעייתי באספקטים מסוימים. התוצאות הראו הקבלה חזקה בין החלטות שהתקבלו על ידי האלגוריתמים של המודל לבין החלטות המשתמשים האמיתיים.

חלקו השני של מחקר זה כולל ארבעה תתי חלקים עיקריים:

בתת החלק הראשון אנו בודקים את חוסנו של המודל ועמידותו בפני התקפות אפשריות, על ידי הדגמה של מספר סוגי התקפות אפשרויות בדרגות חומרה שונות, ואנו מראים את יתירות התקפות אלו מול מבנה המודל שלנו. בחלק זה אנו מבססים את חוזקו של המודל הבסיסי, על ידי בניית התקפות עליו. אנו מבצעים סימולציות התקפה שונות שיכולים להתבצע על ידי קבילת ספאמרים או משתמשים עוינים מסוגים אחרים שמנסים לזייף תכונות של רשת חברתית בהן אנו משתמשים למודל הבסיסי. לאחר מכן אנו מנתחים את התקפות אלו שמתבצעים על ידי רשמת המתחזה לאמינה, ומראים כי התקפות מסוג זה הינן חסרות תוחלת מפאת חוזקו של המודל, שמשתמש בשילוב של אמון, בקרת גישה מבוססת- תפקיד ובקרת זרימה.

בתת החלק השני אנו מעדנים ומדייקים את המודל על ידי שימוש בהקשרים (Context) של תוכן של מידע ברשתות האישיות של המשתמשים, זאת על ידי שימוש בטכניקות של עיבוד שפה טבעית (Natural Language Processing – NLP), לזיהוי מיטבי של מידע שנאסף עבור חברים ברשת המשתמש, העוזר לנו לסיווגם ובניית אמון מדויק בהקשרים שונים. בתת חלק זה אנו מדייקים את המודל על ידי שימוש בקונטקסט של תוכן ואפיון משתמשים שונים לפי קטגוריות שונות של מידע ונושאים, המבוססים על תכונות ופעולות ברשת החברתית. לאחר מכן אנו מבצעים ולידציה של חלק זה של המודל באמצעות Sentiment Analysis בשילוב עם אמון על פוסטים של משתמשים אמיתיים ברשת. חלק זה של המודל נתן לנו תמונה מדויקת הרבה יותר על החלטות שונות של משתמשים ברשת ותאימות להם בהחלטות קבלת גישה שהאלגוריתם המשולב סיפק לנו בצורה המתאימה יותר למשתמשים אלו. כמו כן, חלק זה עזר למציאה טובה יותר של מקורות חשיפה פוטנציאליים של מידע שיכול להיחשב רגיש בהקשרים מסוימים, ולמנוע מחשיפות מסוג זה להתרחש.

בתת החלק השלישי אנו מרחיבים את מודל הקונטקסט לטובת תרחיש השימוש של גילוי ומניעת הפצה של פייק ניוז. לשם כך בנינו אלגוריתם למידת מכונה מבוסס אמון המעבד תוכן של מידע ברשת. בחלק זה אנו מגלים משתמשים בעייתיים אשר יש להם פוטנציאל להפצת פייק ניוז. בחלק זה אנו משתמשים במודל הבסיסי משולב האמון וגם בחלק הקונטקסט, עבור עיבוד התוכן המופץ על ידי משתמשים ברשת- פוסטים, שיתופים וכולי. חלק זה נותן לנו תמונה מדויקת יותר של התוכן המופץ ברשת ועוזר לגלות מידע על מקורות הפצת הפייק ניוז בנושאים שונים ובכך מאפשר מניעה של הפצה זו.

בתת החלק הרביעי והאחרון אנו מתמקדים בפתרון מותאם לתקנות האיחוד האירופי החדשות – GDPR

לרשתות חברתיות, המתמקד בהיבטים שונים של אמון והסכמת משתמשים לטובת פרטיותם ברשת. תקנות אלו המחייבות ארגונים שונים המשתמשים במידע של משתמשים דורשות היבטים רבים של פרטיות שאינם ממומשים כיום. המודל הבסיסי שלנו, בתוספת חלק הקונטקסט, מאפשר התאמה של חלקים שונים בתקנות לרשתות חברתיות ומאפשר שימוש מותאם ברשת החברתית, המגן על פרטיותם של המשתמשים ברשת. ביצענו גם ולידציה לחלקים רלוונטיים אלו על ידי מספר ניסויים שהראו את היתרונות שניתן לקבל על ידי שימוש במודל שלנו ברשת החברתית.

מחקר זה, על כל חלקיו, מופיע בשני מאמרי כתב עת (Voloch, Gal-Oz, & Gudes, 2021; Voloch, Gudes & Gal-Oz, 2022) ובשמונה מאמרי כנסים (Gudes & Voloch 2018; Voloch & Gudes, 2019; Voloch, Levy, Elmakies, & Gudes, 2019A & 2019B; Voloch, Gudes & Gal-Oz, 2021A & 2021B & 2021C; Voloch, Gudes, Gal-Oz, Mitrany, Shani, & Shoel, 2022).

Research-Student's Affidavit when Submitting the Doctoral Thesis for Judgment

I, Nadav Voloch, whose signature appears below, hereby declare that

(Please mark the appropriate statements):

X_ I have written this Thesis by myself, except for the help and guidance offered by my Thesis Advisors.

X_The scientific materials included in this Thesis are products of my own research, culled from the period during which I was a research student.

___ This Thesis incorporates research materials produced in cooperation with others, excluding the technical help, excluding result analysis, commonly applied during such experimental work. Therefore, I attach an additional affidavit stating the contributions made by myself and the other participants in this research, which has been approved by them and submitted with their approval.

_ This thesis in in Manuscript Format, includes one or more papers in which I am an "equal contributor". I therefore attach an additional affidavit signed by other equal contributor(s) stating their contribution to the paper and their approval that that paper could not be included in another Manuscript Format Thesis.

Date: 27.7.2022 Student's name: Nadav Voloch Signature:

LIBERTY
Academic Books

A Comprehensive Trust-based Information Security Model for Online Social Networks

ISBN 978-195509423-8

9 781955 094238