Combining Trust and Skills Evaluation to Form E-Learning Classes in Online Social Networks

Pasquale De Meo[d], Fabrizio Messina[b,*], Domenico Rosaci[a], Giuseppe M. L. Sarné[c]

[a] *DIIES, Università "Mediterranea" of Reggio Calabria, Italy*
[b] *DMI, University of Catania, Italy*
[c] *DICEAM, Università "Mediterranea" of Reggio Calabria, Italy*
[d] *University of Messina, Italy*

*DIIES, Università "Mediterranea" of Reggio Calabria, Via Graziella, Loc. Feo di Vito, 89122, Reggio Calabria, Italy, e-mail: domenico.rosaci@unirc.it

Abstract

E-Learning class formation will take benefit if common learners' needs are taken into account. For instance, the availability of trust relationships among users can represent an additional motivation for classmates to engage activities. Common experience also suggests that there are many similarities within dynamics of formation for thematic social network groups and e-Learning classrooms. Since Online Social Networks provide data concerning users interactions (e.g. trust relationships), we propose a model aimed at managing the formation and the evolution of e-Learning classes based on information available on Online Social Networks — skills, interactions, and trust relationships — which are properly combined in a unique measure. The aim is to suggest both to a user the best classes to join with and to the classes themselves the best students to accept. The proposed approach has been tested by simulating an e-Learning scenario within a large social network. Experiments show that this proposal is able to support students and class managers in order to satisfy their expectations in a scalable manner.

Keywords: e-Learning, Class formation, Online Social Networks, Skill, Trust

1. Introduction

E-Learning is a cost-effective solution to lifelong learning and on-the-job work force training [7, 18], providing a learner-centered environment and giving to e-learners (hereafter, only learners) the opportunity of exploiting time and location flexibility, low costs, unlimited access to knowledge and archival capability for information sharing [30].

In this context, learners progresses depend on many factors, as their own attitudes and initial skills (i.e. background) and, last but not least, the cognition of the "human" virtual environment where interactions take place. There is also a relation between "learner-teacher", or "learner-learner" interactions and student learning and satisfaction issues [34, 35]. In fact, the attitude of the learner to start interactions with her/his peers also depends on the *mutual trust* level [29].

1.1. Motivations

Based on the previous considerations, the design of a proper process of class formation – in order to improve the "quality" of learner-learner/teacher *interactions* such that actual knowledge improvements can be maximized [11, 24] – should be supported by *i*) an analysis of learners' skills and attitudes, and by *ii*) the knowledge of the mutual trust between learners. Such an analysis can benefit from information available on Online Social Networks (OSNs) and, therefore, e-Learning activities can take further advantages from the potential synergies with OSNs. In addition, OSN platforms, like Facebook [15] or Twitter [40], support activities of thematic *groups*[1], which embrace a wide

[1]Each day more than 100,000 groups are created only on Facebook [28]

range of topics [22] and evolve by following the dynamic of human interactions [3]. Due to their large relevance, social groups have been studied by researches with particular emphasis to their formation [20], evolution [3, 27] and failure [28].

In this context, intelligent software agents can support the process of e-Learning class formation by providing suggestions for students (resp., classes) about the best classes (resp., students) to join with (resp., to accept into a class) [6, 41, 19]. Class formation is important for the generic student because he can join with a class potentially capable to satisfy his/her expectations at the best and the increment of utility for his/her future mates in accepting him/her in their class.

Moreover, some studies confirmed that the users' attitude to mutually interact and share information depends also from the mutual trust [12]: the larger the reciprocal trust among users, the larger their interest to start interactions [8, 39]. Besides, several studies show that, in forming OSN groups, existing trust relationships can provide a relevant contribution, in addition to a similarity criterion [12, 36]. The use of trust relationships when forming groups is motivated also by the common practical problems of computing similarities between users [2, 31, 38]. Indeed, processing the overall amounts of data about OSN users and groups is not easy, due to the size and limitations in crawling these data for OSN and/or group administrators policies. Therefore, such algorithms cannot really examine the entire space of involved OSN groups to suggest suitable solutions for learners' needs.

At the best of our knowledge, any of the existing approaches to form e-Learning classes considers the recent results described above that should

lead to form groups based on both students' skills and trust relationships. From the analysis reported in Section 5, the past proposals are mainly based on considering the skills [10] as the sole criterion for groups formation, and only a few of work is addressed on the use of trust information. In any case, any of these past approaches explicitly study roles and mutual interactions between skills and trust component in forming effective groups.

1.2. Contribution

Based on the considerations above we propose a model aimed at managing formation and evolution of e-Learning classes by using skills and trust information already available on OSNs. The main novelty of our proposal is of combining skills and trust data in a unique measure, named *convenience*, used to (*i*) suggest the best class to join with or to leave to a user and (*ii*) to suggest the best students to accept or remove to the class itself. The information on the skills of the generic student on a set of topics of interest, as specified above, is a fundamental aspect to consider for giving teaching-homogeneity to the class [37]. Indeed, by analyzing the nature and the number of such activities carried out in the past by the involved users, it is possible to give a significant contribution in forming classes. Based on such information, "supply" and "offer" of interactions can be balanced in class formation, such that unfitting requests and users profile resulting incompatible might be avoided.

Trust relationships is the second component and it is combined (i.e. weighted) with the "results" derived by these interactions (i.e. the learners' satisfaction level) and the relevance they assign to them, also in order to limit malicious behaviors. Although existing trust relationships can be

found within OSNs, in order to consider specific concerns related to the mutual interactions within learning classes, the proposed solution includes a trust model exploiting reliability and reputation criteria. The model includes some countermeasures aimed at filtering erroneous or malicious opinions.

Students and classes are driven by personal software agents, respectively named learner and class agents, delegated to create, manage and update the *profiles* of their respective owners (i.e. a learner or a class) on the basis of information found in the OSNs. The *convenience* measure is exploited by a distributed procedure, named Class Formation (CF), that allows learner/ class software agents to cooperate in order to form classes.

A number of experimental trials was performed to test the proposed approach. A first set of results highlights that i) by adopting the discussed trust model, learners can identify malicious peers in several different scenarios. A second set of experiments proved the ability of the CF algorithm ii) to improve the average value of the convenience within classes during their formation processes. Finally, the same results also prove that the increment of average convenience within classes leads to an increment of satisfaction of the learners. The rest of the paper is structured as follows: Section 2 introduces the context of the work as well as the definition of Behavioral, Trust and Convenience measures.

Section 3 illustrates the supporting multi-agent architecture and the distributed CF algorithm. Section 4 presents our experimental tests. In Section 5 we compare our work with related literature and, finally, in Section 6 we draw our conclusions and illustrate some possible future works.

2. Basic scenario and exploited measures

2.1. The OSN Scenario

The e-Learning scenario is built on the generic OSN community, whose members are able to form classes and perform e-Learning activities with their peers and their teachers. Let *i)* $\mathcal{N}$ be the space of the OSN members, with $||\mathcal{N}|| = N$; let *ii)* $\mathcal{C}$ be the set of classes, with $||\mathcal{C}|| = C$; let *iii)* $c \in \mathcal{C}$ be any e-learning class, which consists of a number of learners, and at least a teacher, that is also its administrator; *iv)*learners are assisted by software agents [17], denoted as a_i for any user $u_i \in \mathcal{N}$. Each software agent a_i is able to analyze the behavior of its owner to obtain a detailed view on his/her background and attitudes and assists the learner in joining with or leaving classes; *v)* class administrators are assisted by software agents, denoted as A_i. Each software agent A_i assists its associated class manager in getting decision about the acceptance of a new member for its own class.

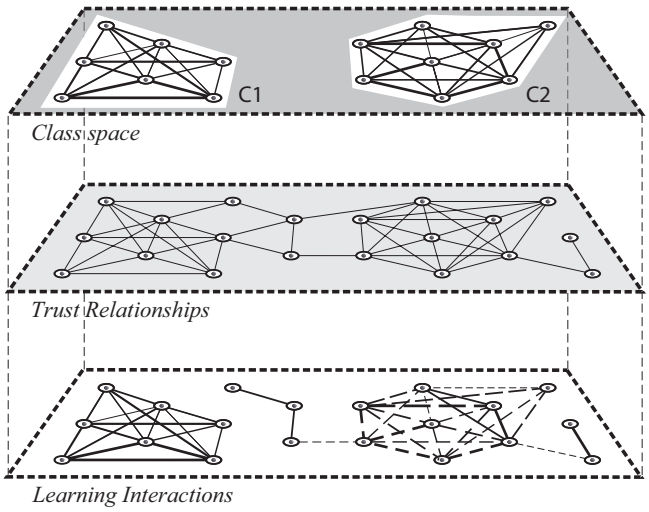

Figure 1: A simplified schema of class formation process in the OSN e-Learning scenario.

2.2. Measures

Two different measures are introduced: *i) behavioral measure*, related to the interactions carried out by a learner, (Section 2.2.1), and *ii) trust measure*, i.e. level of trust that an OSN member places on another OSN member (Section 2.2.2).

A simplified schema of the approach to compose classes is depicted in Figure 1. The bottom layer represents a portion of the OSN space where edges identify the set of learning interactions measured among the same users: the different kind of interactions (i.e., the set of skills) have been represented by means of different type of lines due to the different set of considered skills, and their thick is proportional to the difference among their behavioral measure. The middle layer describes trust relationships among students of the same OSN. Class formation is driven by the computation of the convenience measure which combines trust and behavioral measures as described in Section 2.2.3. The result of this final process, i.e. the formation of two e-Learning classes, C_1 and C_2, is depicted in the top layer of Figure 1.

2.2.1. Behavioral Measures

The concern of behavioral measure is based on the concept of "skill", i.e. the level of knowledge of a particular learner at a certain time. We model the concept that *each learner is interested in improving his/her own knowledge also by interacting with other users*. Therefore, learners are interested in joining with specific classes also because the other participants have suitable skills. On the other hand, when classes are deficient in skills of interest to their own members, class managers have the interest to include those users holding some specific skills which can potentially help other users to perform

better in their e-Learning activities.

Classes. Let's define a class c as a tuple $\langle S, W, V_c, o \rangle$ where: (i) $S = \{s_1, s_2, \ldots, s_m\}$ is the set of skills required by a class manager for participating to c; (ii) $W = \{w_1, w_2, \ldots, w_m\}$ is the set of weights used by the class agent to evaluate the skills of their students; (iii) V_c is the minimum overall skill grade required to join with the class; (iv) o is the reference topic or subject or goal of the course itself (e.g., "Course for Cambridge FCE").

More in detail, the overall skill grade V is defined as the estimated grade (or level) of an OSN user, computed over a specific skill set S. More formally, for an OSN user u_k and a skill set S, V is computed as $V(k, S) = \sum_{i=1}^{m} w_i \cdot g(k, s_i)$, where $g(k, s_i) \in [0, 1] \in \mathbb{R}$ is the knowledge grade of the user u_k coming from the evaluation of the skill $s_i \in S$ [2], while $w_i \in [0, 1] \in \mathbb{R}$ is specifically set by the class manager to weight the knowledge grade g_i and such that $\sum_{i=1}^{m} w_i = 1$. For example, a class manager may focus on the language requirements for joining a class; a set of specific skills related to the language may be $S = \{s_1, s_2, s_3\}$ with: $s_1 = \{$"Speaking fluently"$\}$; $s_2 = \{$"Reading technical documents"$\}$; $s_3 = \{$"Extracting relevant technical information"$\}$. Contextually, the set of weights adopted by the class manager is $w_1 = 0.2$, $w_2 = 0.4$ and $w_3 = 0.4$.

Historical attitude. Let H be the "historical" attitude of the OSN user u_k to require and/or offer interactions related to its own skills, $H(k) = \alpha \cdot H(k) +$

[2] Although the grade for a skill (e.g., Ability to fluently speak English) generally will take non numerical values from a finite set (e.g., {A, B, C, D}), they can be always converted into a set of numerical normalized values.

$(1-\alpha)\cdot\overline{H}(k)$, with $\alpha\in[0,1]$. Therefore, the new value of the real number $H\in[0,1]$ is given by combining the previous value and the new contribution $\overline{H}$ due to the requests and supply for interactions occurred at the last step. $\overline{H}$ is calculated as:

$$\overline{H}(k)=\begin{cases}1-\dfrac{|h(k)_{req}-h(k)_{off}|}{h(k)_{req}+h(k)_{off}} & \text{if } h(k)_{req}+h(k)_{off}\neq 0\\ 0.5 & \text{otherwise}\end{cases} \quad (1)$$

with

$$h(k)_{req}=\frac{1}{N_{req}}\sum_{i=1}^{N_{req}} g(k,S_{req,i}) \qquad h(k)_{off}=\frac{1}{N_{off}}\sum_{i=1}^{N_{off}} g(k,S_{off,i}) \quad (2)$$

where, with respect to the OSN user u_k, $h(k)_{off}$ is the evaluation of the offered interactions for a skills subset $S_i\subset S$, and $h(k)_{(req)}$ is the similar evaluation for the requested interactions. Moreover, the smaller the overall skills in S_{req}, the greater the resulting factor. Indeed, it will result that, when $h(k)_{req}\approx h(k)_{off}$, then $\overline{H}\approx 1$, i.e. the attitude of the user u_k is to ask and provide interactions to the same extent. Vice versa, if $h(k)_{req}\ll h(k)_{off}$ ($h(k)_{req}\gg h(k)_{off}$) then $\overline{H}\approx 0$, it means that the attitude of the user u_k is to offer (require) interactions.

Class Behavior. The class behavior for the class c_j, denoted as $B(j)\in[0,1]$ is defined as a sort of "footprint" over the class itself, in order to characterize the tendency of the whole class c_j to offer or require interactions. It is defined

as follows [3]:

$$B(j) = \frac{1}{||c_j||} \sum_{k=1}^{||c_j||} H(k) \tag{3}$$

2.2.2. Trust

The second measure is based on the concept of *trust*, which can be assumed as "*The quantified belief by a truster with respect to the competence, honesty, security and dependability of a trustee within a specified context*" in accord to [21]. In the model described in this work, trust is computed by combining the reliability and the reputation factors. The former is the measure of perceived trust derived by the direct knowledge between truster and trustee due to their past *interactions*. The reputation represents an indirect knowledge, i.e. the perceived trust due to the past interactions occurred among the trustee with counterparts different from the current truster [1].

An interaction between the two generic OSN learners u_p and u_r consists of a process where the learner u_p starts an interaction (i.e. one or more learning tasks) with the learner u_r (e.g. checking homework, practice on a specific topic, asking some explanations). Consequently, the software agents a_p and a_r, respectively associated with u_p and u_r, can observe the interactions carried out by their owners to *i*) register the interaction features (type, topic, duration) and *ii*) collect feedbacks about other OSN users to compute their respective reputations. Such feedbacks will reflect the quality of the interactions among the two learners, e.g. whether each counterpart has complied

[3] Anyway, depending on the distribution of the numbers $H(k)$, it can be defined as its average or median value, as the latter is not sensitive to the outliers

with the fixed meetings, or deadlines to provide feedbacks on homework, and so on (remember that users' skills are evaluated by the *behavioral measures*).

Let $\eta_{p,r}$ and $\rho_{p,r}$ be respectively the measures of reliability and reputation that the generic OSN user u_p (i.e., agent a_p) computes for the OSN user u_r (i.e., agent a_r). The trust measure $\tau_{p,r}$ is obtained by combining the reliability ($\eta_{p,r}$) and the reputation ($\rho_{p,r}$) by means of a real coefficient $\beta_{p,r} \in [0,1] \in \mathbb{R}$:

$$\tau_{p,r} = \begin{cases} 0.5 & \text{if } I_{p,r} = 0 \\ \beta_{p,r} \cdot \eta_{p,r} + (1 - \beta_{p,r}) \cdot \rho_{p,r} & \text{if } I_{p,r} > 0 \end{cases} \tag{4}$$

where $I_{p,r}$ is the number of previous interactions between agents a_p and a_r. For new coming learners the initial trust/reputation is set to 0.5 to mitigate penalization [32] but to sufficiently contrast whitewashing strategies [43]. To compute $\beta_{p,r}$, we consider that, first of all, its value should increase according to the number of interactions occurred between the learners and its peers, because the direct knowledge that u_p (i.e. a_p) has of u_r (i.e. a_r) will consolidate over time. Moreover, it should increase as the recommenders reliability in providing suggestions decreases. Then we also consider that each OSN user (i.e. agent) has only a partial view of his/her (i.e. its) community and the trust measures computed in his/her (i.e. its) own might differ from those computed by including the opinions of the whole community. Finally, the reputation or one or more peers may differ due to malicious behavior. For the above considerations, the coefficient $\beta_{p,r}$ is computed as:

$$\beta_{p,r} = Max(\beta_1, \beta_2) \quad \text{with} \quad \beta_1 = min\left(\frac{I_{p,r}}{I_{max}}, 1\right) \quad \text{and} \quad \beta_2 = 1 - \Omega_{p,r}^{(t)} \tag{5}$$

where $\Omega_{p,r}^{(t)}$ is the average *confidence* computed at time t for the current set of recommenders which provided at least a suggestion to a_p for agent a_r, where $R_{p,r}$ is the set of agents which provided an opinion about a_r (i.e. u_r):

$$\Omega_{p,r}^{(t)} = \frac{1}{||R_{p,r}||} \sum_{i=1}^{R_{p,r}} |\sigma_{p,r}^{(t-1)} - \tau_{q,r}| \tag{6}$$

Looking at Equation 6, the confidence factor minimizes the impact of untrustworthy opinions by giving more relevance to those mentors that a_p evaluates as the most similar to it. I_{max} is a threshold representing the number of interactions after which the "knowledge" of an OSN user about another OSN user is assumed maximum. It is incremented at each step as:

$$I_{p,r} = min(I_{max}, I_{p,r} + 1) \tag{7}$$

The ratio of Formula 7 is that the number of interaction between a_p and a_r is increased of one unit until the threshold I_{max} is reached, since I_{max} is considered as a sort of saturation value for $I_{p,r}$, in order to give a practical limit to the increment of $I_{p,r}$. In other words, the ratio adopted in computing $\beta_{p,r}$ is to provide a different relevance to the reputation with respect to the reliability based on the experience acquired by u_p (i.e., a_p) about u_r (i.e., a_r). As a result, the contribute of the reputation in computing trust decreases as much as the number of the interactions occurred between the two involved learners constantly increases.

Computation of Reliability. Reliability measure, denoted as $\eta_{p,r} \in [0, 1]$, is computed by u_p (i.e., a_p) about u_r (i.e., a_r) as $\eta_{p,r} = \vartheta_{p,r} \cdot \sigma_{p,r} + (1 - \vartheta_{p,r}) \cdot \eta_{p,r}$, where $\vartheta_{p,r}$ weights in a complementary way the contributes of i) the feedback

parameter $\sigma_{p,r} \in [0,1]$ computed for the last interaction between u_p and u_r at time-step t and ii) the value of $\eta_{p,r}$ computed at time-step $(t-1)$. The parameter $\vartheta_{p,r}$ should take into account the *relevance* assigned to the interaction between u_p and u_r, let be $\Psi_{p,r}$. In principle, malicious behaviors focused to gain good reputation on low value interactions ($\Psi \ll 0.5$) with high reliability ($\sigma \gg 0.5$) may occur. In this case, malicious users may start interaction of high relevance ($\Psi \sim 1$) due to good reputation providing poor performance ($\sigma \sim 0$). Therefore, the closer the ratio Ψ/σ to 1, the higher the value of ϑ; the farther the value Ψ/σ from 1, the lower the value of ϑ[4]. A possible choice for ϑ is given by the adoption of the Gaussian centered in 1, as follows:

$$\vartheta = e^{-(\Psi/\sigma - 1)^2/v^2} \tag{8}$$

By Equation 8, ϑ will perform as a "filter" for those values of σ which, for the analogous values of Ψ, may be a malicious behavior. Parameter v^2 is useful to tune the filter. Small values of v will select only values of σ for which σ/ϑ is close to 1, while large values of v will ensure that almost the whole history of feedbacks σ is considered in computing the reliability η defined above.

Computation of Reputation. The reputation measure $\rho_{p,r}$ is computed by u_p (i.e., a_p) with respect to u_r (i.e., a_r) as a value ranging in $[0,1] \in \mathbb{R}$:

[4]For convenience, let's suppose that $\sigma > 0$

$$\rho_{p,r} = \frac{1}{||R_{p,r}||} \sum_{q=1}^{||R_{p,r}||} \tau_{q,r} \tag{9}$$

Through the usual meaning of these indexes, 0 means that u_r is totally unreliable and, conversely, 1 means that u_r is totally reliable.

2.2.3. Convenience Measure

Behavioral and trust measures are combined to measure the *convenience*, for a user, to join with the class c_j. Due to the asymmetric nature of the trust measure, convenience is also an asymmetric measure. Therefore, let $\gamma_{u,c}$ and $\eta_{c,u}$ be the convenience of the user u to join with the class c and that of the class c to accept the affiliation request of a user u defined as follows:

$$\gamma_{k,j} = \frac{(1 - |H(k) - B(j)|)}{||c_j||} \sum_{a_i \in c_j} \tau_{k,i} \quad \phi_{j,k} = \frac{(1 - |H(k) - B(j)|)}{||c_j||} \sum_{a_i \in c_j} \tau_{i,k} \tag{10}$$

where $||c||$ is the number of users (i.e. agents) affiliated to c. Convenience increases with the difference between the behaviors of a_k and c_j. The asymmetric part is due to the different trust measures $\tau_{k,i}$ and $\tau_{i,k}$, with $a_i \in c$.

2.2.4. Discussion

Reliability and reputation defined above are rooted into the relationships among OSN users. Indeed, even e-Learning bring students to have interactions in virtual environments, users have the attitude to create closed relationships, which help them to benefit from the resulting interactions. In defining trust and its components, a number of additional factors have been introduced to improve their computations. In particular, the definition of

$\beta_{p,r}$ in Eq. 5 takes account of the suggestion "quality", by the definition of Eq. 6, and how much the direct knowledge, in terms of number of interactions already taken with the counterparts, by the definition of expression for $I_{p,r}$ in Eq. 7. Finally, the definition of Eq. 8 is an attempt to limit "malicious" behaviors of participants, as detailed above, basing on the ratio Ψ/σ.

Furthermore, based on the fact that $\eta_{p,r} \neq \eta_{r,p}$ and, therefore, the trust computed by the agent a_r about the agent a_p is different of that computed by a_r vs a_p. For such a reason, Equations in 10 assume different values for the agents a_p and a_r. Consequently, the procedure described in Section 3 is distributed among the agents assisting learners and those related to class administrators. As it will be discussed in the experimental Section, the aim of the distributed procedure is to let the system to reach a balance in terms of convenience among all the considered actors of the proposed scenario.

3. The distributed procedure for Class Formation (CF)

In the proposed approach, we suppose that OSN users (i.e. learners) are supported by intelligent software agents [42] capable to perform all the activities aimed at organizing the e-Learning classes basing on the measures presented in Section 2. Such a multi-agent architecture is sketched in Figure 2: on each of the three classes in Figure 2, black circles represents humans (i.e. learners and class managers), white circles, built around the black ones, software agents (i.e. learner and class agents) and black circles, built around the white ones, the class agents built around the class managers.

In particular, each agent will execute a set of tasks briefly summarized below. Each *Learner Agent* has to update behavioral measures, as well as

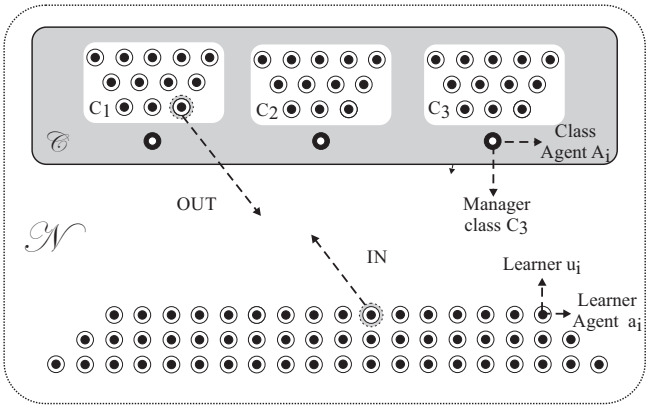

Figure 2: The multi-agent architecture for the e-Learning system.

reliability and convenience whenever one or more interactions occurred with the user' peers. The generic learner agent is also in charge of sending, on a periodical basis, behavioral and trust measures to the class agents representing its own classes once they have been recomputed. Finally, the generic learner agent will assist its own user to decide about joining with or leaving classes by executing the CF algorithm detailed in Section 3.1. For this aim, it will receive behavioral and trust measures from its own class agents.

The generic *class agent* waits for messages coming from learner agents in order to update behavioral, trust and convenience measure of the entire class. Furthermore, it has to send updated behavioral measure to allow learner agents to compute their convenience in updating their own convenience measure. Finally, as specified in Section 3.1, each class agent must be capable to assist its own class administrator to decide about the requests coming from learner agents to join with or leave classes.

3.1. The distributed CF procedure.

The CF procedure is executed by every learners, in order to join with a set of classes that will maximize the resulting convenience index. We suppose that agents can query a distributed database named CR (Class Repository) storing the list of the classes of the e-Learning system. Details are explained below.

The CF procedure performed by the learner agents is formalized into the pseudocode listed in Figure 3. Let T be the time between two consecutive executions (steps) of the CF procedure, X_n the set of the classes which the agent a_n is affiliated to, and N_{MAX} the maximum number of classes which an e-Learning agent can analyze at time t, with $N_{MAX} \geq |X_n|$. Moreover, let us suppose that a_n stores into a cache the class profile of each class contacted in the past and the timestamp d of the execution of the CF procedure. Finally, let ξ_n be a timestamp threshold and $\chi_n \in [0, 1]$ be a threshold set by the agent a_n. The ratio behind the procedure executed by the learner agent is represented by the attempt, of the learner agent, to improve the convenience in joining with a class. For this aim, the values of convenience are recalculated if they are older than the threshold ξ_n (lines 1-4). Then, candidate classes are sorted in a decreasing order with respect to their Convenience value (line 5). The two loops in lines 7-15 represent the core of the procedure, on which N_{Max} classes are selected. If some classes in the set L_{good} are not in the set X_n, agent a_n can potentially improve convenience of the user u_n, if they accept the user itself to join with. The only constraint of the algorithm is the maximum number of classes the user want/can join with.

The CF procedure performed by the class agent is formalized into the

Input: $X_n, N_{MAX}, \xi_n, \chi_n$; $\quad Y = \{c \in C\}$ a random class set : $|Y| \leq N_{MAX}$
$X_n \bigcap Y = \{0\},\ Z = (X_n \bigcup Y)$

1: **for** $c \in Z : d_c > \xi_n$ **do**
2: Send a message to A_c to retrieve the profile P_c.
3: Compute $\gamma_{u_n,c}$
4: **end for**
5: Let be $L_{good} = \{c_i \in Z : i \leq j \rightarrow \gamma_{u_n,c_i} \geq \chi_n\}$, with $|L_{good}| = N_{MAX}$
6: $j \ \rightarrow \ 0$
7: **for** $c \in L_{good} \wedge c \notin X_n$ **do**
8: send a join request to A_c
9: **if** A_c accepts the request **then**
10: $j \ \rightarrow \ j + 1$
11: **end if**
12: **end for**
13: **for** $c \in \{X_n - L_{good}\} \wedge j > 0$ **do**
14: Sends a leave message to c
15: $j \ \rightarrow \ j - 1$
16: **end for**

Figure 3: CF algorithm: Learner Agent

pseudocode in Figure 4. Let K_c be the set of the agents affiliated to the class c, where $||K_c|| \leq \ K_{MAX}$, being K_{MAX} the maximum number of learners allowed to be within the class c. Suppose that the class agent A_c stores into its cache the profile P of each user u managed by his/her learner agent $a \in K_c$ and the timestamp d_u of its acquisition. Moreover, let ω_c be a timestamp and $\pi_c \in [0, 1] \in \mathbb{R}$ be a threshold set by the agent A_c. The procedure performed by the class agent A_c is triggered whenever a join request by a learner agent a_r (along with its profile P_r in the interest of its user u_r) is received by A_c. First

Input: $K_c, K_{MAX}, W, \omega_c, \pi_n$, a_r, $Z = K_c \bigcup \{a_r\}$;

1: **if** $(V(r, S_c) < V_c \vee |K_c| \geq K_{MAX})$ **then** Send a reject message to a_r
2: **else**
3: **for** $a \in K_c$ **do**
4: **if** $d_u \geq \omega_c$ **then** ask to a its updated profile
5: **end if**
6: **end for**
7: **for** $a \in Z$ **do**
8: compute $\phi_{c,a}$
9: **end for**
10: Let be $K_{good} = \{a \in Z : \phi_{c,a} \geq \pi_c\}$
11: **for** $a \in K_c - K_{good}$ **do**
12: send a *leave* message to a.
13: **end for**
14: **if** $a_r \in K_{good}$ **then**
15: the request of a_r is accepted
16: **end if**
17: **end if**

Figure 4: CF algorithm: Class Agent

of all, parameter K_{MAX} represents the maximum number of students that can join with a given class[5]. In fact, if the class has reached this maximum, no more students can be accepted to join with the class. By lines 3-6 the class agent asks the updated profile of the components of the class itself, therefore the convenience $\gamma_{c,a}$ is computed for all these agents (lines 7-9) and a new, sorted set $K_{good} \subset \{K_c \bigcup a_r\}$ is built (line 10). Then, the class agent

[5]We assume that it is the same for all the classes and topics for convenience

Sc.	Unreliable users		Malicious users	
	Ratio (ru)	Behaviors	Ratio (rmal)	Behaviors
A	0.1	Feedbacks generated by the PdF in row no.2 of Table 2.	–	–
B	0.1	Feedbacks generated by the PdF in row no.2 of Table 2.	0.2–1.0	Malicious learners give false recommendation.
C	$0.2 - 0.8$	also malicious	rmal=ru	Unreliable users are reliable in interactions having low relevance and are unreliable on those with high relevance. See Eq. 8 and Table 3

Table 1: Trust model, simulated users behavior. *ru*=ratio unreliable; *rmal*=ratio malicious

will send a *leave* message to all the learner agents a showing a convenience $\gamma_{c,a}$ (lines 11-13). Finally, if $a_r \in K_{good}$ (line 16), its request is accepted.

4. Experiments

To evaluate the described approach, we obtained two different sets of results. A first set of simulations was devoted to test the effectiveness of our trust model in identifying untrustworthy users or those that assume anomalous behaviors (e.g. cheating to gain positive reputation). Results are described in SubSection 4.1. By the second set of simulations we studied the convergence of the CF algorithm described in Section 3, and the benefits in improving the quality of the learners' interactions. Results are described in SubSection 4.2.

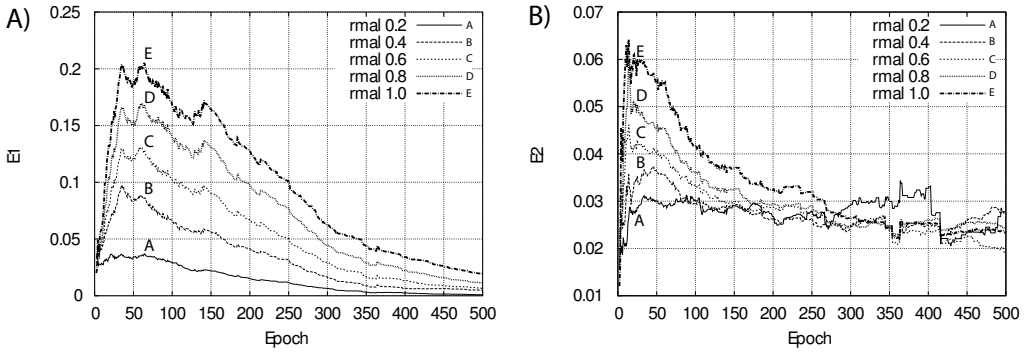

Figure 5: Scenario B - Time evolution for different ratio of malicious users $rmal$=0.2 – 1.0 of **A**) E1 and **B**) E2.

4.1. The Trust model

The first set of experiments consisted in a number of simulations where learners were grouped in classes. A simulation consisted in a certain number of interactions between Learners of the same class. We carried out these experiments by an ad-hoc simulator written in C++[6]. At each simulation

[6]The simulator can be downloaded at the following address: `https://globus.dmi.unict.it/~fmessina/suppIS.zip`

Epochs (Sc. A,B,C)	500	Users (Sc. A,B,C)	1000
No. of Classes (Sc. A,B,C)	50	No. of user recc. (Sc. A,B,C)	200
No. of interactions within classes (Sc. A,B,C)	300	Ratio of high values, reliable interactions (Sc. C)	0.3
Class size (Sc. A,B,C)	20	Imax (Sc. A,B,C)	50
Low value interactions (Sc. C)	$\Psi = 0.2$, $\sigma = 0.8$	High value interactions (Sc. C)	$\Psi = 0.8$, $\sigma = 0.2$

Table 2: Trust model, simulation parameters

Generated feedback (σ) P.d.F. (Norm. dist.)			
No.	Profile	mean	stdev
1	Reliable Learner	$0.5 - 0.8$	$0.1 - 0.3$
2	Unreliable Learner	$0.2 - 0.5$	$0.1 - 0.3$

Table 3: Trust model, simulation parameters: generated feedbacks.

Unreliable learners ratio (ru) = 0.1						
mean Rel/Unrel	E_1	E_2	E_1	E_2	E_1	E_2
	stdev=0.3		stdev=0.2		stdev=0.1	
0.5/0.5	0.48	0.23	0.48	0.16	0.48	0.10
0.6/0.4	0.30	0.20	0.28	0.13	0.16	0.05
0.7/0.3	0.25	0.18	0.15	0.11	0.05	0.02
0.8/0.2	0.16	0.15	0.07	0.012	0.01	0.01

Table 4: Results for Scenario A

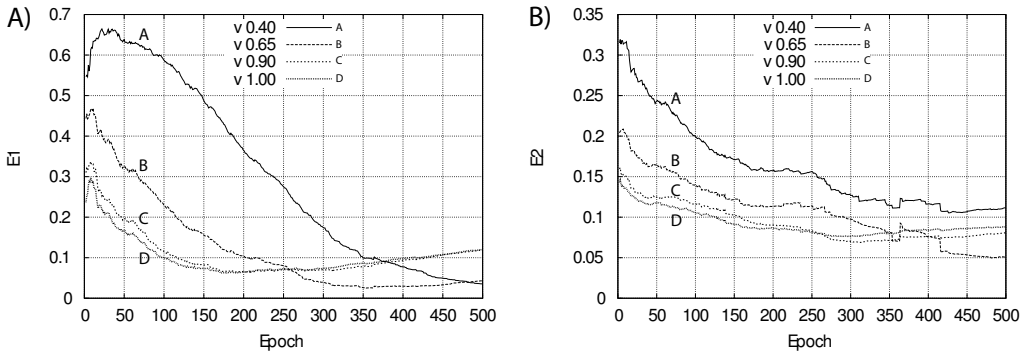

Figure 6: Scenario C - Time evolution for $rmal$=r=0.2 and different values of $v = 0.4 - 1.0$ of **A**) E1 and **B**) E2.

epoch, the following tasks are executed:

1) An e-Learning class in the set of classes is selected at random;

2) for each learner l_1, a set of distinct recommenders is selected. These

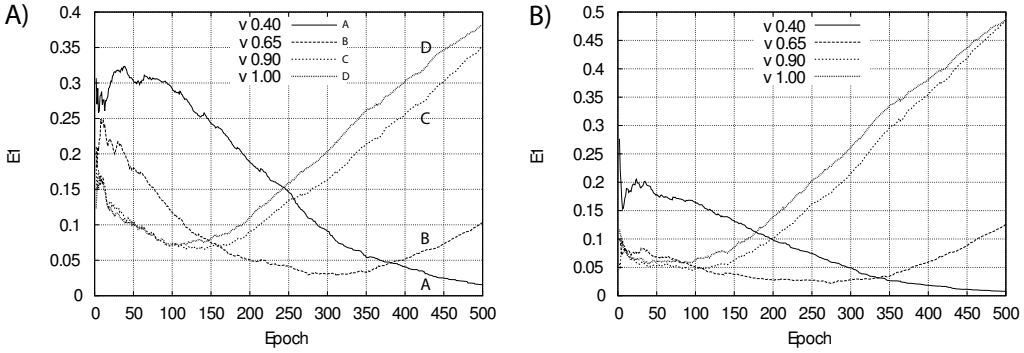

Figure 7: Scenario C - Time evolution of E1 for different values of $v = 0.4 - 1.0$ and of **A**) $rmal$=r=0.6 and **B**) $rmal$=ru=0.8.

recommenders are selected at random among the whole set of learners;

3) an interactions among two learners l_1, l_2 will result in assigning a feedback $\sigma_{l_1,l_2} \in [0, 1]$, as specified in Section 2.2.2.

4) correspondent values of η, ρ and τ (Section 2.2.2) are updated.

Learners behaviors, in terms of reliability and honesty, are simulated by combining the behavioral profiles described below and reported in Table 1.

Scenario A: users will be reliable (generated values of reliability $\sigma >= 0.5$) or unreliable ($\sigma < 0.5$) in their interactions with colleagues.

Scenario B: there will be also malicious learners, which provide false opinion about their peers, i.e. an opinion which does not reflect the actual reliability of the learner.

Scenario C: unreliable learners have a particular malicious behavior which consists in trying to gain reputation by performing *reliable* interactions having *low relevance* (see Section 2.2.2). Moreover, they provide *unreliable* interactions having *high relevance*. Table 2 reports the ratio between the number of high and low relevant (i.e. unreliable and reliable) interactions and the

values of simulated Ψ (relevance) and σ (feedback).

Feedbacks for the different simulation scenarios A and B were sampled by a normal distribution with mean and standard deviation reported in Table 3. Two indexes have been evaluated to evaluate the performance of the trust model described in Section 2.2.2, as follows:

$$E_1 = \frac{fp + fn}{N_{int}} \qquad E_2 = \sum_{i<=fp+fn} \frac{e_i(\sigma, \tau)}{fp + fn} \tag{11}$$

fn – false negative – (resp. fp – false positive), is an integer number representing the number of times a learners received a value of trust τ which does not reflect its real attitude to be reliable (resp., unreliable). Moreover, $e_i(\sigma, \tau)$ is the absolute value of the difference between the value of τ (trust value) and the last value of σ (feedback) computed for the counterparts. This value is computed only for interactions on which a false positive or false negative occurred.

Finally, values of parameters epochs, users, no. of classes, no. of recommenders per user, and no. of interactions within classes were a priori fixed for all the simulations, and reported in Tables 2 and 3. The above five parameters were fixed on the basis of a sensitivity analysis which allowed us to obtain reasonable simulation times. In the following we discuss the results obtained by the simulation of the three scenarios A,B,C.

Scenario A. The first scenario illustrated in Table 1 represents a basic testbed to understand the performance of the trust model with respect to some basic parameters. The ratio of unreliable users has been set as indicated in Table 1, i.e. $ru = 0.1$, and interactions feedbacks generated on the basis of the normal distribution with parameters *mean* and *stdev* indicated in Table 3.

We report the first set of results in Table 4. As expected, both errors E_1 and E_2 are affected by the mean value and the stdev of the generated feedback σ. Indeed, the higher is the standard deviation of the generated feedbacks and the closer is the mean to 0.5[7], the less defined, in terms of reliability, the profile of the users. As a consequence, the worst performance, for E_1, is reached when $mean = 0.5$. In this case the trust system will generate about 50% of false positive/negative. Nevertheless, when the users have a defined behavior, i.e. when a behavioral gap between reliable and unreliable users exists (the mean is far from 0.5 and $stdev < 0.3$), values of the error E_1 is very low, which is a desirable property.

Scenario B. In the second set of simulations, a number of malicious learners was also simulated (their ratio w.r.t. the total number of learners is reported in Table 1), i.e. those users giving false recommendations. To this aim, let denote the malicious recommender as a_q, then we simulated that the recommendation sent about an agent a_r, denoted as $\tau^*_{q,r}$ is set as the value $\tau^*_{q,r} = 1 - \tau_{q,r}$, where $\tau_{q,r}$ is the actual measurement of trust hold by the agent a_q about agent a_r. The ratio of malicious users with respect to the total number of users is denoted as $rmal = 0.2 - 1.0$ (table 1), while the ratio of unreliable users was fixed the same of scenario A (ru=0.1). Furthermore, feedbacks were sampled from a normal distribution by fixing $mean\ rel/unrel = 0.7/0.3$ and $stdev = 0.1$. In this scenario we tested the resilience of the trust model with regards to the different ratio of malicious users *rmal*. Figure 5(a) shows the results for E_1 while Figure 5(b) shows the results for E_2. In particular,

[7]A value of 0.5 will mean, in terms of reliability, "incertitude'

the different plots have been obtained for the different values of parameter *rmal*. From Figure 5(a) we observe that, after a transitory, on which the error reaches a maximum value of about 20% [8], the error decreases to reach values close to zero in all tested cases. In other words, parameter β given in Equation 5 leads the expected results, i.e. it allows agents to assign a low weight to malicious suggestions. Results in Figure 5(b) show a similar trend, with values of E_2 always lower than 10%.

Scenario C. The last trial was aimed to understand whether countermeasure given by Equation 8 is able to recognize malicious patterns described in the third row of Table 1. In this case we obtained results for E_1 and E_2, by driving simulations by means of different values of parameter $v = 0.4, 0.65, 0.9, 1.0$ (see expression of ϑ in Eq. 8), while the ratio of malicious users was fixed as *rmal*=*ru*=0.2. At this regard we remark that, as stated in Section 2.2.2, the smaller the parameter v, the higher the sensitivity over the values Ψ/σ which are far from 1. For this set of experiments the behavior of malicious users is further characterized by the values of Ψ (interaction relevance) and σ (resulting feedback), cf. Table 2.

Results for *rmal*=*ru*=0.2 are shown in Figures 6(A)-(B). Here, as expected, since the percentage of untrusted learners having a malicious pattern is 20% (*rmal*=*ru*=0.2), the whole range of simulations for the different values of v seem to perform very well: the error quickly decreases to values under 10% after about 350 epochs. In addition, figures 7(A)-(B) reveals that, as the ratio of unreliable users is very high, i.e. $ur = 0.6$ and *rmal*=*ru*=0.8,

[8] Although having all the users malicious may seem unrealistic, we tested until this value to perform an exhaustive study of the recommender subsystem of the trust model

Sc.	$\|C\|$	$\|U\|$	K_{Max}	N_{Max}
1	50	200		
2	100	400	20	5
3	200	800		

	τ_r	τ_u	$\{h_{Req}, h_{off}\}$
mean	0.8	0.3	$0.5, 0.5$
stdev	0.2	0.2	$0.2, 0.2$

Table 5: CF algorithm. Simulation parameters

the filter represented by parameter ϑ calculated as in Eq. 8 should be set to be more selective, i.e. related parameter v has to be set with small values. Indeed, only when $v = 0.4$ users are able to recognize untrusted peers, as shown by the values of E_1. This last set of results show that the expression of ϑ in Equation 8 is effective in recognizing malicious patterns, based on the ratio Ψ/σ. Nevertheless, parameter v must be set appropriately, as the higher the ratio of users which show a malicious pattern, the smaller the value of v to set in Equation 8.

4.2. The CF Algorithm.

The second set of experiments was aimed at testing the effectiveness of the distributed algorithm CF reported in Figures 3 and 4. As a measure of the internal convenience measured for a class c_j, we introduced the concept of *Average Convenience* (AC), computed as the average of all the measures $\eta_{j,i}$ computed by the class $c_j \in \mathcal{C}$ for all its students $u_i \in c_j$. Thereafter, to measure the global convenience of all the classes belonging to $\mathcal{N}$ in our simulated scenario, we computed the mean $MAC = \frac{\sum_{c_j \in \mathcal{C}} AC_j}{||\mathcal{C}||}$ and the standard deviation $DAC = \sqrt{\frac{\sum_{c_j \in \mathcal{C}} (AC_j - MAC)^2}{||\mathcal{C}||}}$.

A preliminary test involved three scenarios consisting of 50, 100, and 200 e-Learning classes, as reported in Table 5. To compute the convenience, the

users profiles were build by assuming that 20% of OSN members have an unreliable behavior. Behavioral coefficients h_{req} and h_{off} (useful to compute behavioral coefficients H), and the values of trust (τ), have been sampled from a normal PdF [23] around specific mean and standard deviation (stdev), as reported in Table 5. In particular, τ_r represents the mean of generated trust values for reliable users (e.g. users showing, in average a trusted behavior), while τ_u is the mean for unreliable users. Moreover, for the first set of experiments, the ratio $r = \frac{K_{max} \cdot |C|}{N_{max} \cdot |U|}$ was set to 1 and the starting composition of classes is random. We report, in Table 6, the results obtained by the CF algorithm for the three scenarios reported in Table 5. In particular, we report the initial value of MAC/DAC and the final one, calculated when $T_e = 20$ [9]. The improvement, in terms of MAC, at the end of the experiments, is in the order of 8% for all the tested configuration. Based on the observation above, and since the ratio $r = \frac{K_{max} \cdot |C|}{N_{max} \cdot |U|}$ is the same for the three scenarios, without relevant variations, the subsequent were driven by the variation of r.

For the second set of experiments we assumed a variable value of $r = \frac{K_{max} \cdot |C|}{N_{max} \cdot |U|}$ (Table 7), ranging from 0.1 to 0.9. A value $r < 1$ say us that users,

[9] Indeed, we verified that, after 20 epochs of executions, the MAC reaches a stable value

	Sc 1		Sc 2		Sc 3	
	MAC	DAC	MAC	DAC	MAC	DAC
T_0	0.63	0.12	0.62	0.12	0.62	0.12
T_e	0.67	0.10	0.66	0.10	0.67	0.12

Table 6: Results with $r = 1.0$

	r=0.1		r=0.2		r=0.3		r=0.4		r=0.5	
	MAC	DAC	MAC	DAC	MAC	DAC	MAC	DAC	MAC	DAC
T_0	0.61	0.07	0.59	0.02	0.60	0.03	0.60	0.04	0.60	0.08
T_e	0.61	0.08	0.63	0.08	0.69	0.04	0.70	0.10	0.73	0.06
	r=0.6		r=0.7		r=0.8		r=0.9			
	MAC	DAC	MAC	DAC	MAC	DAC	MAC	DAC		
T_0	0.60	0.07	0.63	0.09	0.63	0.09	0.62	0.11		
T_e	0.70	0.09	0.69	0.08	0.68	0.08	0.67	0.10		

Table 7: MAC and DAC

in overall, can join with more places ($N_{max} \cdot |U|$) than the total allowed ($K_{max} \cdot |C|$). By looking at Figure 8, the best improvement, in terms of MAC, is obtained for $r = 0.4$ (+20%), $r = 0.5$ (+16%) and $r = 0.6$ (+20%). This result can be explained as follows. On one hand, finding a class to improve the personal convenience γ is a bit more difficult for the user when $r < 1$, therefore the CF algorithm will help to improve the MAC for random composition of classes. On the other hand, the algorithm clearly needs a certain degree of freedom to give benefit: indeed, when r is very small, the improvements, in terms of MAC are comparable to those given for values of r closed to 1 (see Figure 8). In overall, these results point out that the CF algorithm gives, on average, a relevant improvement of the convenience for the classes.

To take a step forward in our analysis, some simulations were performed in order to quantify the benefits due to the CF algorithm in terms of "quality of interactions" among learners. Indeed, as stated in the introductory section, mutual trust and behavioral components (i.e. attitude to interact and skills), in principle, influence the quality of the overall learning in a

	r=0.1		r=0.2		r=0.3		r=0.4		r=0.5	
	MQI	DQI	MQI	DQI	MQI	DQI	MQI	DQI	MQI	DQI
T_0	0.60	0.09	0.56	0.18	0.56	0.17	0.57	0.13	0.50	0.18
T_e	0.60	0.10	0.57	0.16	0.72	0.11	0.71	0.14	0.75	0.16
	r=0.6		r=0.7		r=0.8		r=0.9			
	MQI	DQI	MQI	DQI	MQI	DQI	MQI	DQI		
T_0	0.50	0.13	0.54	0.11	0.57	0.13	0.57	0.13		
T_e	0.68	0.16	0.61	0.15	0.61	0.12	0.60	0.16		

Table 8: MQI and DQI

class. Therefore, the execution of the CF algorithm to form classes impacts positively on the learners interactions. To verify this aspect, a number of interactions among couple of users, denoted as n_{int}, were simulated, as in the experiments performed to test the capabilities of the recommender system (SubSection 4.1). In addition, we assumed that the probability of each interaction, say $p_{int}(ij)$, grows with the mutual trust among users, i.e. $p_{int}(ij) = \tau_{ij}$. Besides, to measure the quality of interactions, we defined an index, QI as $QI = \frac{1}{n_{int}} \sum_{i=1}^{n_{int}} (1 - |H_i(u) - H_i(v)|)$, where u and v are the users involved in the ith interaction. Results are shown in Table 8. Based on index QI, the indexes Mean Quality Index (MQI) and standard Deviation of Quality Index (DQI) were computed. The trend of MQI is similar to that of MAC (Table 8). Indeed, around the value $r = 0.5$ the CF algorithm allows class manager to reach the best improvement, in terms of QI. Nevertheless, the overall improvement is larger than that obtained in terms of MAC in the correspondent configuration. In particular, the trend, when MQI is steeper than that of MAC. To offer the best view of this comparison, all the results (Tables 7 and 8) are shown in Figure 8.

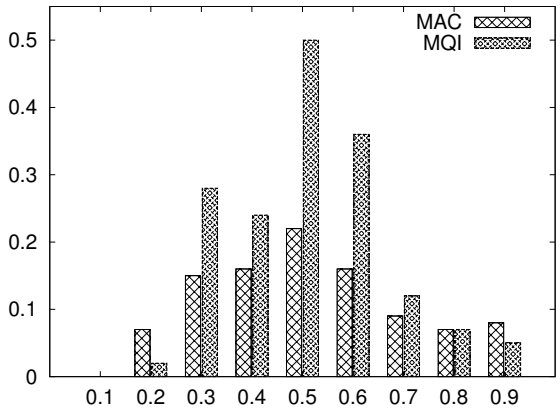

Figure 8: Improvements of MAC and MQI vs $r = \frac{K_{max} \cdot |C|}{N_{max} \cdot |U|}$

4.3. Discussion

In order to test the effectiveness of the trust model along with the distributed CF algorithm we performed two sets of experimental trials. The analysis of the first set of results give us a good confidence that the proposed recommender system, based on the computation of direct and indirect measures (i.e. reliability and reputation), is effective in characterizing the behavior, in terms of trust, of the learners, even when recommenders show a malicious behavior. This model can support learners of OSNs during learning interaction. In particular, the model can start from trust relationships taken from the social network, to be enriched by means of interactions made within e-Learning classes.

In addition, we have verified, by simulations, that the class formation algorithm leads to high and stable values of average convenience. Furthermore, the convenience measure is computed as a combination of trust and behavioral aspects based on the skills of the learners. As both components

represent as incentive, for the users, to start interactions, we also simulated a number of interactions and quantified the benefit ("quality of interactions") due to these interactions. Simulations have shown that the CF algorithm will lead significant benefits in terms of average quality of interactions.

5. Related Work

The learning activity in a structured educational environment may be thought of as a two-step process which involves both the reception and processing of information. In this setting, a learning-style model can be viewed as a classification of students according to where they fit on a number of scales pertaining to the ways they receive and process information [16]. E-Learning systems provide computer-supported collaborative learning (CSCL) technologies to allow the possibility of a group interaction for physically distributed users. These kind of systems extend e-Learning Content Management Systems (LCMS) by adding inter-group communication on learning topics, and learning material can be manipulated by group collaboration. On the other hand, the presence of OSN for personal communication and entertainment leads to an increasing demand for applications that are integrated into the OSN, and people tend to share personal information with connected persons in the OSN. Educational Hypermedia has evolved in the past years from static system to dynamic content presentation and delivery platforms. As shown in [33], CSCL research is focused on finding mechanisms to allow learners to learn in a group of physically distributed people. Popular OSN sites like Facebook or Google+ store personal profiles of each user, and per-

sonal profiles are linked witch each other through a bidirectional relationship, established individually between two users. Therefore, a social network site like Facebook can be seen an undirected graph which edges represent the relationship between users. Building a team in OSNs means to find a subgraph of members in the OSNs user graph which fulfills the following conditions: (*i*) Each user-node has the motivation for collaborative learning on a certain topic; (*ii*) Learning style of a user-node is appropriate for a balanced group; (*iii*) Persons knowledge in the topic is equal among group members. (*iv*) A mutual trust relationship links each member with each other member of the group. members.

In the following we cite the work among the recent literature that, in our view, takes into account the main concerns cited above, i.e. behavioral and trust aspects in group formation processes.

Authors of [13] discuss the issue of forming random groups or classes, as the lack of proportional participation of individuals, or the absence of adequate motivation and low attitude to work in groups may cause a problem within any e-learning group. A more recent work [25] emphasizes that group formation is the first step to design a CSCL on which students can learn and participate to the class activities. They discuss how to select individuals on the basis of their characteristics (e.g. knowledge or learning attitudes) can be usefully combined to create a positive synergy among participants.

A recent survey on algorithms for group/class formation in CSCL is presented in [10], on which 250 works have been analyzed. The authors found that about 50 studies concern the group formation in collaborative e-Learning and, in particular, about 20% of them exploit probabilistic models, while the

remaining studies rely on Genetic Algorithms (GA), swarm intelligence, data mining (e.g. k-means), Bayesian networks and machine learning. Among them, a generic mathematical model for e-Learning group formation is presented in [9], on which it is proposed an optimizer driven by an evolutionary algorithm able to create groups by using the criteria given by the instructors. In their study, the authors focused to find a near optimal solution for the group-formation problem in a reasonable time. As stated by the authors, the power of the approach is represented by the expressiveness of their model.

Strategies for group formation based on individual behaviors are addressed in [26], on which behaviors are analyzed during class discussions, with particular emphasis to the level and kind of participation in small groups. The work is based on the analysis of communication data, i.e. forum posts, either of small of larger forums. The results of the analysis show that the students participation in small groups is correlated with their behavior in the class and, therefore, the authors suggest to instructors to use these information to allocate initial classes into small groups heterogeneously. Nevertheless, the class formation strategy we proposed it is aimed at grouping individuals with similar behaviors, in terms of "positive" and "negative" interactions. Besides, a relevant component in our proposal is given by the trust relationships derived from OSN data that in [26] is neglected.

A recent survey [14] regards recommender systems for Technology Enhanced Learning (TEL), which are capable to recommend a wide variety of items, i.e. any type of learning resources on the Web, foreign language lessons, complete courses or fellow peers. Nevertheless, specific methods must be adopted in order to evaluate TEL recommender systems, as requirements

are very different from other domains like e-Commerce. In our proposal we include a recommender system for learners, focused on the interactions occurring between them with emphasis on those involving e-Learning interactions.

In [5] the authors analyze the state-of-the-art of the "socialization" of e-Learning activities and propose an automated approach to form OSN e-Learning environments by grouping users within the OSN. For this aim the focused on a data model that would extend the OSN itself by introducing e-Learning features. As in our work, they start with the considerations that, in addition to the common criteria exploited to form groups, e.g. knowledge and learning style, social networks offer the opportunity to access the myriad of data related to virtual social relationships. Then they propose suitable metrics in order to weight the "edges" between users. Interestingly, among the several factors taken into account by the users (e.g. learning style, knowledge and group density), the users take into account availability. To form groups, the OSN is explored to find a minimal number of suitable candidates, thereafter the set of candidates is optimized to find the "optimal constellation" for group learning experience. The interesting difference with our work is that we strongly exploit the concept of trust, which is represented, in turn, by combining reliability and reputation within the OSN itself.

Finally, another relevant experience in using an OSN (i.e. Facebook) is [4], where it is analyzed the student use of Facebook at the Cape Town University. Lecturer engagement with students via OSN are also taken into account, and through qualitative interviews they show that while there are real positive benefits in using Facebook in teaching and learning, in particular to build e-Learning micro-communities. On the other hands, certain existing

challenges, i.e. including ICT literacy and uneven access, remain opened.

6. Conclusions

Class formation in e-Learning is a critical task for its impact on the quality of learning activities. In this work we focused on a distributed algorithm supported by a trust model and some behavioral measures to form classes in OSNs. Our proposal uses information coming from the OSN, i.e. trust relationships among users, quality of interactions, as well historical attitude to interact with peers, for improving the metrics related to class composition.

E-Learning classes are dynamically managed, as the proposed model allows class managers to accept new students, and students itself to join with and leave classes in order to improve the quality of their own learning experiences. This flexibility is reached by combining information about trust and learning interactions in a unique measure named "convenience". The proposed approach has been tested by simulating an artificial scenario including an e-Learning platform within a large OSN. Experiments show that this proposal can support students and class managers to satisfy their expectations. In particular, experimental results highlight that by using our trust model, learners can identify malicious peers in several dierent scenarios. Furthermore, our experiments proved the ability of the CF algorithm to improve the average value of the convenience within classes during their formation processes. This is shown by a simulation on which we measured the potential increment of satisfaction of the learners due to the increment of average convenience within classes. The current main limitation of our results is represented by the fact that they have been obtained on a simulated scenario,

that although having realistic features, cannot be considered as completely representative of real social network environments. An important research question is how our results could change in presence of social networks having higher size than that we have considered. Our ongoing research is dealing with these issues, and for the future we are planning to test our approach with data extracted from real social networks having high size.

References

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *HICSS '00: Proc. of the 33rd Hawaii Int. Conf. on System Sciences*, volume 6. IEEE Computer Society., 2000.

[2] V. Agarwal and K.K. Bharadwaj. A collaborative filtering framework for friends recommendation in social networks based on interaction intensity and adaptive user similarity. *Social Network Analysis and Mining*, 3(3):359–379, 2013.

[3] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group Formation in Large Social Networks: Membership, Growth, and Evolution. In *Proc. of the 12th ACM SIGKDD Int. Conf.*, pages 44–54. ACM Press, 2006.

[4] T. E. Bosch. Using online social networking for teaching and learning: Facebook use at the university of cape town. *Communication: South African J. for Comm. Theory and Research*, 35(2):185–200, 2009.

[5] S. Brauer and Thomas C. Schmidt. Group formation in elearning-

enabled online social networks. In *Interactive Collaborative Learning (ICL), 2012 15th Int. Conf. on*, pages 1–8. IEEE, 2012.

[6] W. Chen, D. Zhang, and E.Y. Chang. Combinational collaborative filtering for personalized community rrecommendation. In *Proc. of the ACM Int. Conf. SIGKDD'08*, pages 115–123. ACM, 2008.

[7] Ruth C Clark and Richard E Mayer. *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. John Wiley & Sons, 2011.

[8] A. Comi, L. Fotia, F. Messina, G. Pappalardo, D. Rosaci, and G. M. L. Sarné. Forming homogeneous classes for e-learning in a social network scenario. In *IDC IX*, pages 131–141. Springer, 2016.

[9] M. Craig, D. Horton, and F. Pitt. Forming reasonably optimal groups:(frog). In *Proc. of the 16th ACM int. conf. on Supporting group work*, pages 141–150. ACM, 2010.

[10] W.M. Cruz and S. Isotani. Group formation algorithms in collaborative learning contexts: A systematic mapping of the literature. In *Collaboration and Technology*, pages 199–214. Springer, 2014.

[11] M.I. Dascalu, C.N. Bodea, M. Lytras, P.O. De Pablos, and A. Burlacu. Improving e-learning communities through optimal composition of multidisciplinary learning groups. *Computers in Human Behavior*, 30:362–371, 2014.

[12] P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarné. Trust and

compactness in social network groups. *Cybernetics, IEEE Transactions on*, 45(2):205–216, Feb 2015.

[13] P. Dillenbourg. Over-scripting cscl: The risks of blending collaborative learning with instructional design. *Three worlds of CSCL. Can we support CSCL?*, pages 61–91, 2002.

[14] M. Erdt, A. Fernandez, and C. Rensing. Evaluating recommender systems for technology enhanced learning: A quantitative survey. *Learning Technologies, IEEE Trans. on*, 8(4):326–344, Oct 2015.

[15] `https://www.facebook.com`, 2016.

[16] Richard M Felder, Linda K Silverman, et al. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988.

[17] S. Franklin and A. Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Intelligent agents III Agent Theories, Architectures, and Languages*, pages 21–35. Springer, 1997.

[18] D. R. Garrison. *E-learning in the 21st century: A framework for research and practice*. Taylor & Francis, 2011.

[19] J. Gorla, N. Lathia, S. Robertson, and J. Wang. Probabilistic group recommendation via information iatching. In *Proc. of the Int. World Wide Web Conf. (WWW '13)*, pages 495–504. ACM Press, 2013.

[20] P. Grabowicz, L. Aiello, V. Eguiluz, and A. Jaimes. Distinguishing topical and social groups based on common identity and bond theory.

In *Proc. of the ACM Int. Conf. WSDM 2013*, pages 627–636. ACM, 2013.

[21] T. Grandison and M. Sloman. Trust management tools for internet applications. In *Trust Management*, pages 91–107. Springer, 2003.

[22] J. Heidemann, M. Klier, and F. Probst. Online social networks: A survey of a global phenomenon. *Computer Networks*, 56(18):3866–3878, 2012.

[23] K. Hopkins, G. Glass, and B. Hopkins. *Basic statistics for the behavioral sciences* . Prentice-Hall, 1987.

[24] R. Hübscher. Assigning students to groups using general and context-specific criteria. *Learning Technologies, IEEE Transactions on*, 3(3):178–189, 2010.

[25] S. Isotani, A. Inaba, M. Ikeda, and R. Mizoguchi. An ontology engineering approach to the realization of theory-driven group formation. *Int. J. of Computer-Supported Collaborative Learning*, 4(4):445–478, 2009.

[26] N. Jahng and M. Bullen. Exploring group forming strategies by examining participation behaviours during whole class discussions. *European J. of Open, Distance and E-Learning*, 2012.

[27] S.R. Kairam, D.J. Wang, and J. Leskovec. The life and death of online groups: Predicting group growth and longevity. In *Proc. of the 5th ACM int5 conf5 on Web search and data mining*, pages 673–682. ACM, 2012.

[28] R.E. Kraut and A.T. Fiore. The role of founders in building online groups. In *Proc. of the 17th ACM conf. on Computer Supported Cooperative Work & Social Computing (CSCW 2014)*, pages 722–732. ACM Press, 2014.

[29] J. Mason and P. Lefrere. Trust, collaboration, e-learning and organisational transformation. *Int.l J. of Training and Development*, 7(4):259–270, 2003.

[30] J. L. Moore, C. Dickson-Deane, and K. Galyen. e-learning, online learning, and distance learning environments: Are they the same? *The Internet and Higher Education*, 14(2):129–135, 2011.

[31] A. A. Rad and M. Benyoucef. Similarity and ties in social networks: a study of the youtube social network. *Journal of Information Systems Applied Research*, pages 14–65, 2014.

[32] S.D. Ramchurn, D. Huynh, and N.R. Jennings. Trust in multi-agent systems. *Knowledge Engeenering Review*, 19(1):1–25, 2004.

[33] Hendrik Roreger and Thomas C Schmidt. Socialize online learning: Why we should integrate learning content management with online social networks. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 685–690. IEEE, 2012.

[34] D. Rossi, C. Beer, H.M. Janse van Rensburg, R.E. Harreveld, P. A. Danaher, and M.J.G. Singh. Learning interactions: A cross-institutional

multi-disciplinary analysis of learner-learner and learner-teacher and learner-content interactions in online learning contexts. 2011.

[35] A. Sher. Assessing the relationship of student-instructor and student-student interaction to student learning and satisfaction in web-based online learning environment. *J. of Interactive Online Learning*, 8(2):102–120, 2009.

[36] T.A.B. Snijders. Network dynamics. *The Handbook of Rational Choice Social Research. Stanford University Press, Palo Alto*, pages 252–279, 2013.

[37] H Songhao, S Kenji, K Takara, and M Takashi. Towards new collaborative e-learning and learning community using portfolio assessment. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, volume 2008, pages 1270–1275, 2008.

[38] E. Spertus, M. Sahami, and O. Buyukkokten. Evaluating similarity measures: a large-scale study in the orkut social network. In *Proc. of the ACM Int. Conf. SIGKDD'05*, pages 678–684. ACM, 2005.

[39] W. Tan, S. Chen, J. Li, L. Li, T. Wang, and X. Hu. A trust evaluation model for e-learning systems. *Systems Research and Behavioral Science*, 31(3):353–365, 2014.

[40] `https://www.twitter.com`, 2016.

[41] V. Vasuki, N. Natarajan, Z. Lu, B. Savas, and I. Dhillon. Scalable affiliation recommendation using auxiliary networks. *ACM Trans. on Intelligent Systems and Technology*, 3(1):3, 2011.

[42] M. Wooldridge and N. Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(02):115–152, 1995.

[43] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence.*, 14(9):881–907, 2000.