International Journal of Transportation Science and Technology 8 (2019) 68–81

Contents lists available at ScienceDirect

International Journal of Transportation Science and Technology

journal homepage: www.elsevier.com/locate/ijtst

Reputation assessment mechanism for carpooling applications based on clustering user travel preferences

Athanasios Salamanis *, Dionysios D. Kehagias, Dimitrios Tsoukalas, Dimitrios Tzovaras

Information Technologies Institute, Centre for Research & Technology Hellas, PO Box 60361, GR57001 Thermi, Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 20 January 2018
Received in revised form 27 August 2018
Accepted 31 August 2018
Available online 12 October 2018

Keywords:
Carpooling
Reputation systems
Clustering
k-means

ABSTRACT

One way to ensure sustainable and environmental friendly mobility is the use of less vehicles for carrying more passengers, and carpooling is a means to achieve this goal. One major concern in carpooling services is related to trust, as carpooling users need to either share their vehicles, if they act as drivers, or travel with strangers if they act as passengers. One way to tackle trust concerns is the utilization of user reputation assessment mechanisms, whose objective is to provide ranking of users with respect to their behavior, based on feedback provided by other users. This paper presents a newly introduced reputation assessment mechanism for carpooling applications, which, in addition to feedback provided by other users, takes into account user travel preferences. Preliminary experimental results have shown that the proposed mechanism is robust against attacks by malicious users, on their attempt to jeopardize the trustworthiness of the system, as it preserves the real reputation scores of the users, when the penetration rate of malicious users increases.

1. Introduction

In the resent years, a rapid growth has been noted in the development of carpooling services and associated supporting applications. The main purpose of those services is to create an online community, whose members can act as either drivers offering rides or, passengers, seeking for rides from one place to another. The users maintain a profile in the community that includes personal information, as well as their travel preferences. Carpooling service is responsible for matching potential drivers with passengers based on their profiles. A number of relevant smartphone applications for carpooling end-users, have been developed for supporting the process of matchmaking for carpooling users. The carpooling service model implies significant trust concerns, especially derived from the fact that users must share their vehicle with strangers. Fulfillment of security requirements is closely related to user satisfaction regarding the overall offered service, both for drivers and passengers. For instance, a driver may drive aggressively, or a passenger may exhibit impolite behavior during the trip.

In order to tackle the case of inappropriate user matchmaking reputation assessment systems have been introduced recently in order, not only to early identify a problematic user, but also help other reliable users while they build their trust capital. Most of the proposed reputation systems are quite simplistic and use summation or averaging schemes in order to compute reputation scores, whereas the more sophisticated of them are based on probability theory. In this paper, we

* Corresponding author.
E-mail address: asal@iti.gr (A. Salamanis).

https://doi.org/10.1016/j.ijtst.2018.08.002
2046-0430/

present a novel centralized reputation assessment mechanism for estimating the reputation of users registered in an online carpooling service. Our mechanism takes as input the ratings for a user submitted by other users at the end of each ride, and computes the reputation score. Ratings may refer to various aspects of the journey, such as the total time for reaching the destination, the driving behavior, vehicle condition, etc. Each rating parameter can take an integer value between 1 and 5 (measured as number of stars). The reputation assessment mechanism initially transforms each incoming rating for a specific user into a scalar value using an averaging scheme. Then, it aggregates again the separate scalars using a weighted averaging scheme. The produced value, after translated to positive or negative feedback, is inserted into the final component, which computes the reputation score based on the beta probability distribution.

Our main contribution is the introduction of a reputation assessment mechanism that manages to fine-tune the weights of the existing weighted averaging based approaches, taking into account the travel preferences of the users, which have never been addressed before for this purpose. To this end, we created clusters of users with similar travel preferences by the means of a *modified k-means* algorithm. In this context, when the reputation score of a user needs to be updated, the ratings of the users that belong to the same cluster, or clusters with close centers, are valuated more than the ones that belong to a distant cluster. Preliminary evaluation results has shown that the proposed reputation assessment mechanism exhibits robust behavior against attacks by malicious users, i.e. ones who wish to inject false information into the system, in purpose. To our knowledge, our reputation assessment mechanism is the only one that takes into account the travel preferences of the users during the reputation estimation process.

The rest of the paper is organized as follows. Section 2 presents state-of-the-art on reputation assessment systems. Section 3 presents the RideMyRoute carpooling application in which the proposed reputation assessment mechanism was incorporated. Section 4 describes the overall reputation assessment mechanism introduced in this paper, and Section 6 describes preliminary evaluation results. Finally, Section 4 concludes the paper, outlining its main contributions and indicating directions for future work.

2. Related work

The presence of a user reputation assessment mechanism in carpooling applications is of paramount importance for enabling acceptable quality of service. Existing assessment mechanisms (or reputation systems) are present in various online communities and application domains, developed initially for e-commerce purposes and later used for site reviews and file sharing networks, among others. This section reviews a list of the most popular reputation assessment systems implemented either by commercial or research community.

Reputation assessment systems were firstly developed for e-commerce applications, such as eBay (2017) and Amazon (2017). For instance, Amazon allows its users to provide reviews for the products they purchased. After a review is completed, other users can rate it as "helpful" or "not helpful". More "helpful" reviews appear higher in the list of reviews. Another application area, in which reputation mechanisms are used is Question & Answering (Q&A) online communities. Such a well-known Q&A community and corresponding reputation assessment system is StackOverflow (2017), which is a dedicated community for software developers. Users may ask questions or provide answers regarding various technical topics. Similar examples include Advogato (2017), an community of open source developers, and Turkopticon (2017), a third-party reputation assessment system for crowdsourcing workers who use the Amazon Mechanical Turk (AMT) platform.

In addition to the purely commercial online reputation assessment systems, several platforms were introduced as academic research projects, such as the one proposed by Coetzee et al. (2014), and EigenTrust Kamvar et al. (2003), developed specifically for peer-to-peer (P2P) file-sharing networks. EigenTrust calculates and assigns a global trust value to each network peer based on its history of uploads. In this context, Aberer and Despotovic (2001) presented a fully decentralized system for computing the reputation of an agent, based on former interactions with other agents. Similarly, Xiong and Liu (2004) introduced PeerTrust, a decentralized reputation assessment system for P2P networks. Moreover, Tian and Yang (2011) introduced a system for measuring the trustworthiness of peers, based on both reputation and risk, which is widely applied to unstructured P2P networks.

An interesting iterative ranking method with high performance in both accuracy and robustness was proposed by Liao et al. (2014) who introduced a reputation redistribution process to enhance the influence of highly respectful users. This method defined two penalty factors in order to increase the resilience of the algorithm against malicious behavior. Moreover, Alswailim et al. (2016) presented a reputation system to evaluate participants in participatory sensing applications. The objective of this work was to introduce a reputation system for evaluating individual users or groups of users in order to reduce the probability of a participatory sensing application being corrupted due to inaccurate data sensed. Another similar work by Li et al. (2015) revealed that user reputation is topic-biased, and based on this observation they proposed six topic-biased algorithms for estimating user reputation. It was shown after experiments that those algorithms are more effective compared to traditional reputation assessment algorithms.

Relevant reviews and comparative studies were also conducted in order to investigating the value of trust and reputation in different application domains, e.g. Khalid et al. (2013) and Wahab et al. (2015). Especially, trust management comprises a very important component in the development of intelligent transport systems (Ma et al., 2011). In particular, the development of vehicular ad hoc networks (VANETs) introduced a new need for ensuring trust regarding the messages that are exchanged between the various network nodes. Reputation systems can sufficiently contribute in this area. Take as an example VARS (Dötzer et al., 2005), a completely distributed reputation assessment system for evaluating the trustworthiness of the information that is exchanged in an ad hoc VANET. Similarly, Chen et al. (2010) introduced a trust-based message propagation and evaluation framework for VANETs, in which the network users share information regarding the road conditions for road safety applications. The exchanged information was then reviewed by other users who rated the level of trustworthiness of the content that other users produce. The idea of users cooperation for enduring the validity of shared data is not new in VANETs, as a previously introduced reputation management scheme (Patwardhan et al., 2006) associated trust with cooperativeness of peers for ensuring accuracy of the data shared in the same user network.

In another approach (Minhas et al., 2010), role-, experience- and majority-based trust is used to model trustworthiness between communicating peers in VANETs. Similarly, Wang et al. (2016) designed and proposed a reputation management scheme for VANETs that was more robust against tactical (e.g. self-promoting) attacks and preserved the privacy of the users through pseudonymization. In this context, a cryptographic primitive for public updating of reputation score in VANETs was introduced (Chen et al., 2017), based on the Boneh-Boyen-Shacham short group signature scheme. Additional relevant issues in VANETs, such as information cascading and oversampling were successfully tackled (Huang et al., 2014).

Research work on reputation assessment systems for carpooling and ridesharing applications is still limited. Among those, it is worth noting those efforts, which are based on the concept of six degrees of separation from the social networks analysis field, and on the votes provided by the users at the end of each ride, such as Caballero-gil et al. (2015) and Martn-Fernndez et al. (2015) that exhibit robust behavior against attacks from malicious users (e.g. Sybil attacks). Another interesting work, quite different from the previous ones, is the use of fuzzy logic for assigning priority to users with positive feedbacks in the route-matching process (Collotta et al., 2012). Finally, another interesting approach is based on a series of interviews with users, but also non-users of carpooling and ridesharing applications in order to identify the various sources of trust, mistrust and risks associated with those types of application (Créno and Cahour, 2015).

As opposed to the previously reviewed reputation assessment mechanisms, the one presented in this paper takes into account the travel preferences of the users in the reputation estimation process. In particular, its novel characteristic lies on the construction of users profiles using a modified unsupervised machine learning technique that operates on multivariate mixed-type data. Our work comprises a unique example of the application of machine learning techniques for market segmentation as a core component of a carpooling reputation assessment mechanism.

3. The RideMyRoute application

RideMyRoute is a carpooling application developed as part of the research project SocialCar funded by European Commissions Horizon2020 program. The main objective of SocialCar is to integrate public with private mobility services including carpooling, car sharing and other on-demand mobility services. RideMyRoute incorporates our reputation assessment mechanism to assist users on their selection of driver or co-passenger. The matchmaking process takes into account the degree of reputation of the users, as calculated by our mechanism, so that a user with higher reputation is more likely to be selected. Through the RideMyRoute application, our reputation assessment mechanism acquires all the necessary data for estimating the reputations of each user.

Travel preferences of the users are provided as the first input to our reputation assessment mechanism. Table 1 presents the various travel preferences and shows the preferred travel modes (e.g., bus, car, etc.), gender of the driver, maximum affordable cost for a ride, etc. Most of these travel preferences take values from a predefined set, whereas three of them, i.e. the *Maximum Transfers*, the *Maximum Cost* and the *Maximum Walk Distance*, take arbitrary real values specified by the users. When a user registers to the RideMyRoute application, she provides values for these variables in order to describe her profile as a traveler. All these variables (14) are used for clustering the users. As the clustering process is a core

Table 1
User travel preferences defined in the RideMyRoute application.

Parameter	Values
Maximum Transfers	User specified
Maximum Cost (€)	User specified
Maximum Walk Distance	User specified
GPS Tracking	Yes, No
Travel Modes	Bus, Carpooling, Feet, Metro, Rail, Tram
Optimize Travel Solutions By	Price, Comfort, Speed, Safety, Distance
Carpooler Gender	Male, Female
Carpooler Age Range	$[18, 30)$, $[30, 40)$, $[40, 50)$, $[50, 60)$
Impaired	Visual, Hearing, Elderly, Wheelchair
Smoking	Yes, No
Food	Yes, No
Music	Yes, No
Pets	Yes, No
Luggage	Yes, No

component of the proposed reputation assessment mechanism, the registration process of a new user in the *RideMyRoute* application cannot proceed unless all values for those variables are set.

The second input that our reputation assessment mechanism receives is the ratings of other users. At the end of each ride, the passengers rate the driver (and vice versa) based on a set of *rating features*, which include:

1. Comfort Level
2. Driving Behavior
3. Satisfaction Level
4. Compliance with Travel Rules

For all of the above features, the user selects a rating from 1 to 5 stars.

Compliance with Travel Rules was included as a rating feature in order to evaluate the case of a driver not abiding to the calculated route, which is also recommended by the application, e.g. the driver makes a detour thus increasing the total travel time. This case may be rated low by a passenger who wants to arrive to the destination at a specific time, as initially agreed.

4. Reputation assessment mechanism

In this section the design and implementation details of the proposed reputation assessment mechanism are presented. In particular, the reputation assessment mechanism consists of two main components, namely the travel preferences clustering component, which is responsible for creating groups of users based on their travel preferences, and the reputation estimation component, which uses the generated groups along with reviews from other users in order to estimate and update the reputation value. These components are described in detail in what follows.

4.1. Travel preferences clustering

The proposed reputation assessment mechanism assumes that the travel preferences of the users include useful information for the estimation of their reputation and hence they should be taken into account in the reputation assessment process. For instance, let us assume that a driver who likes listening to music while driving, is traveling with a passenger who does not like it. At the end of the ride, it is likely that the passenger will give a low score to the driver, as been disturbed by his music preference, ignoring other positive aspects of his, such as kind behavior, etc. Obviously, the two users in the aforementioned example belong to different user clusters with respect to their preference over music during the ride, hence the rating from one to the other is most likely biased.

In order to remove the impact of the aforementioned bias from the user rating process, we generate groups of users based on their travel preferences. These preferences may describe several aspects of the traveling experience and may have been defined differently for each carpooling application. Our reputation assessment mechanism uses the 14 travel preferences described in Table 1 as attributes for performing clustering of the users.

From a theoretical point of view, the travel preferences of a user can be represented as multidimensional vectors of mixed-type data. For instance, *Maximum Cost* is a continuous variable, as opposed to *Carpooler Gender* which is nominal, or *Food*, which is binary. Hence, in our work we address the problem of clustering multidimensional, mixed-type vectors.

The most essential part for this clustering process is to define a way to measure the distance between the mixed-type vectors. To this end, we used the *Gower's Squared Euclidean Distance* (Gower, 1971), which, for any two mixed-type vectors i,j is defined by the following equation:

$$d_{ij}^2 = \frac{\sum_{k=1}^{N} w_{ijk}\left(x_{ik} - x_{jk}\right)^2}{\sum_{k=1}^{N} w_{ijk}} \tag{1}$$

where N is the number of different variables in vectors and w_{ijk} the weight that corresponds to the variable k. The weight w_{ijk} is either 0 or 1 depending on whether the comparison is valid for the variable k or not. As shown, the distance is normalized by the number of variables. The Gower's distance metric, defines the contribution of each variable to the overall distance based on its type (continuous, nominal, ordinal and binary). We consider valid the comparisons for all variables defined in Table 1, hence all weights w_{ijk} are set equal to 1. The final value derived from the Gower's distance ranges between 0 and 1.

The next step for clustering mixed-type vectors, is to define an averaging scheme, i.e., given m mixed-type vectors, to find the one which is the average of m. The resulting vector will have values for each variable, computed differently depending on whether the variable is continuous or not. Specifically, for $x_1, x_2, \ldots, x_m$ vectors with n variables, the variable k of the average vector x_{av} is given by the following equation:

$$x_{av,k} = \begin{cases} \frac{1}{m}\sum_{i=1}^{m} x_{i,k}, & k \text{ continuous} \\ \text{majority voting}, & k \text{ not continuous} \end{cases} \tag{2}$$

By *majority voting* we refer to the value assigned to the non-continuous variable k of the average vector x_{av}, which equals to the value that occurs most often between the x_i vectors.

After defining the distance metric for the mixed-type vectors, we apply the k-means partitioning algorithm. In our work, k-means utilizes the Gowers Squared Euclidean Distance as distance metric and the aforementioned averaging scheme, in order to compute the centers of the clusters. This variation of the k-means algorithm forms the proposed *modified k-means* algorithm. In order to estimate the optimal number of clusters, we used the so called Elbow method. As shown in Fig. 1, three optimal numbers of clusters were identified, i.e., $K = 3, K = 7$ and $K = 10$. We use all three values of K in our analysis in order to identify the one that leads to the optimal clustering process. For the selection of the initial centers in all cases, K uniformly distributed random points were selected.

4.2. Reputation estimation process

In this section we describe the process of how we compute the reputation of a user based on the generated clusters. The reputation estimation process takes place at the end of each ride, when the users rate each other. More specifically, the passengers rate the driver and the driver rates the passengers (the passengers cannot rate each other). These ratings are fed to the proposed centralized reputation estimation mechanism and the clustering process for each user is initiated.

A user can rate another user based on N_f different rating variables, which describe different aspects of the user's satisfaction from the ride. As already mentioned, the rating provided by a user forms a vector that consists of four values ($N_f = 4$), one for each rating feature that RideMyRoute application supports. The value of each feature is an integer number between 1 and 5. A rating vector, submitted to the system, has the following form:

$$R = \left.\begin{matrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{matrix}\right\} \tag{3}$$

where f_i is the value of feature i. From each rating vector, a single average value R_{av} is computed using the following formula:

$$R_{av} = \frac{1}{N_f}\sum_{i=1}^{N_f} f_i \tag{4}$$

The total rating for a target user, equals to the weighted average of the average ratings of the other co-passengers, and is given by the following equation:

$$R_{tot} = \frac{1}{N_r}\sum_{i=1}^{N_r} w_i \cdot R_{i,av} \tag{5}$$

where w_i is the weight that corresponds to the user u_i, and N_r is the total number of users. The weight w_i of a random user u_i is defined based on the clusters of travel preferences. In particular:

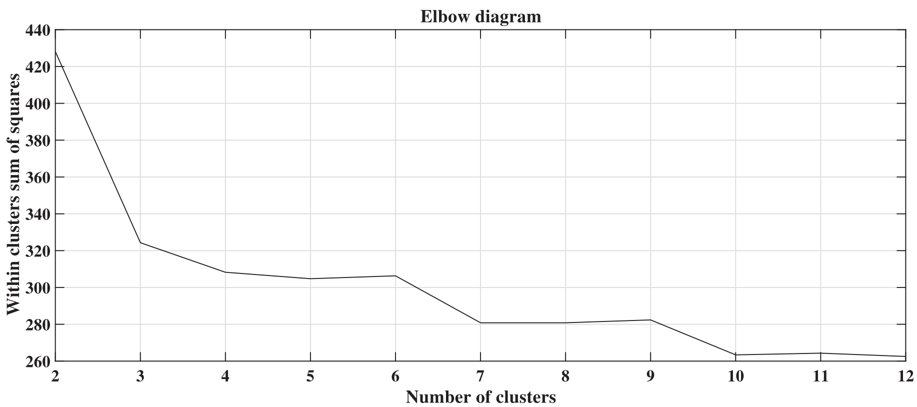

Fig. 1. Elbow diagram for determining the optimal number of clusters.

- Each user belongs to a cluster (i.e. user u_i belongs to cluster c_i).
- The distance d_i between the two centers, i.e., the one of user's u_i cluster and target user's cluster is calculated.
- The weight w_i is defined by the following equation:

$$w_i = 1 - \frac{d_i}{d_{\max}} \tag{6}$$

where $d_{\max}$ is the maximum distance of the center of cluster to which the target user belongs from the centers of the clusters of the reviewing users. The distances between the cluster centers are calculated at the completion of the clustering process, based on the Gower's Squared Euclidean distance metric. As a result of this, a symmetric square matrix D_{KXK} is constructed that contains the distances between all pairs of cluster centers, so that its element d_{ij} is equal to the distance between clusters c_i and c_j. For each user u_i, the distance d_i of cluster c_i from the cluster of the target user can take values in the interval $[0, d_{max}]$, hence weight w_i takes values in the interval $[0, 1]$. If a reviewing user belongs to the most distant cluster from the evaluated user ($d_i = d_{max} \Rightarrow w_i = 0$), her rating is not taken account in the reputation estimation process. We call this choice *fairness policy* for the evaluated user.

The weights are defined in such a way so that the opinion of users who have similar travel preferences with the evaluated user count the most, in the reputation estimation process. If a reviewing user u_i belongs to the same cluster as the evaluated one, their corresponding distance d_i is 0 and the weight w_i is 1. Hence, the rating of this user has the maximum impact in the calculation of the total rating. On the other hand, if a user u_i belongs to the most distant cluster from the evaluated user, their corresponding distance d_i is d_{max} and the weight w_i is 0, i.e. this user has zero influence on the calculation of the total rating.

It should be pointed out that, based on the aforementioned assumption, a positive feedback from a user who belongs to a distant cluster does not create a negative impact on the reputation of the target user, but a low positive one.

After the total rating is calculated, it is transformed into positive or negative *feedback (f)* based on the following inequalities:

$$\begin{aligned} & f \geqslant 0, R_{tot} \geqslant 2.5 \\ & f < 0, R_{tot} < 2.5 \end{aligned} \tag{7}$$

This feedback is used in order to update the reputation value of the evaluated user. For this purpose, we used the *Beta Reputation System (BRS)* (Jøsang and Ismail, 2002), which is based on the beta distribution, one of the most widely used distributions on user reputation systems, because of its ability to take into account both current and historical user provided information. The beta distribution is indexed by two parameters α and β. The beta probability density function is given by the following equation:

$$beta(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \cdot \Gamma(\beta)} p^{\alpha-1} (1-p)^{\beta-1}, 0 \leqslant p \leqslant 1, \alpha, \beta > 0 \tag{8}$$

where Γ denotes the *gamma function*. The expectation value of the beta distribution is:

$$E[beta(p|\alpha, \beta)] = \frac{\alpha}{\alpha + \beta} \tag{9}$$

When nothing is known, we have the *a priori* beta distribution with $\alpha = 1$ *and* $\beta = 1$. Then, after r positive and s negative feedback, we take the *a posteriori* beta distribution with $\alpha = r + 1$ and $\beta = s + 1$. We assume that feedback for a target user u follow a beta distribution and hence the reputation $R(u)$ is calculated as the expectation value of the beta distribution, modeled as follows:

$$R(u) = \frac{r + 1}{r + s + 2} \tag{10}$$

where r is the number of the past positive feedbacks and s is the number of the past negative feedbacks for user u. The r and s parameters are updated using feedback f as follows:

$$\begin{aligned} & r = r + 1, f \geqslant 0 \\ & s = s + 1, f < 0 \end{aligned} \tag{11}$$

If no past feedbacks exist for a target user (i.e. $r = 0$ and $s = 0$), the value of $R(u)$ is 0.5.

4.3. Calculation

This section demonstrates the reputation estimation process, based on a numerical example. Let u be the driver and u_1, u_2, u_3 the passengers on a ride set by the RideMyRoute carpooling application. We assume that the clusters, to which users u, u_1, u_2 and u_3 belong, are c, c_1, c_2 and c_3, respectively. We also assume that passenger u_3 belongs to the same cluster with the driver u (i.e. $c_3 = c$). Also, the distances of the center of cluster c from the centers of clusters c_1, c_2 and c_3 are given by:

$$\begin{aligned} d_1 &= d(c_1, c) = 2 \\ d_2 &= d(c_2, c) = 4 \\ d_3 &= d(c_3, c) = 0 \end{aligned} \tag{12}$$

The maximum distance of cluster c from all the other clusters is selected as $d_{max} = 4$. Finally, we assume that the driver has already 6 positive ($r = 6$) and 4 negative ($s = 4$) ratings, hence the current reputation value of the driver is approximately 0.583. At the end of the ride, the passengers submit their ratings for the driver to our reputation assessment mechanism. The rating vectors, the average ratings and the weights of the users u_1, u_2 and u_3 are calculated from Eqs. (4) and (6), and are shown in Table 2. It should be noted that the weight of user u_2 is $w_2 = 0$, because user u_2 belongs to the most distant cluster from the driver, i.e., $d_2 = d_{max} = 4$. The total rating, given by Eq. (5), is:

$$R_{tot} = \frac{1}{2}\sum_{i=1}^{3} w_i \cdot R_{i,av} = \frac{1}{2}(w_1 \cdot R_{1,av} + w_2 \cdot R_{2,av} + w_3 \cdot R_{3,av}) = \frac{1}{2}(0.5 \cdot 3 + 0 \cdot 4 + 1 \cdot 2) = 1.75 \tag{13}$$

In Eq. (13) the denominator is 2, instead of 3, because $w_2 = 0$. This is due to the implementation of the fairness policy, as described in Section 4.2. Based on Eq. (7), the feedback is negative, and therefore the parameter s takes the value $s = 5$. Finally, the new updated reputation value of the driver, based on Eq. (10), is approximately 0.538.

We can see that even though the driver had two average rating values greater than 2.5, namely rating 3 from user u_1 and rating 4 from user u_2, the overall reputation value is finally lower than both of them. The reason for this is that the average rating from user u_3, who belongs to the same cluster as the driver was less than 2.5. As already mentioned, in our reputation assessment mechanism the positive or negative ratings of the users that belong to clusters close to the cluster of the evaluated user, are the ones that mostly affect the overall reputation value.

5. Evaluation

In this section, the evaluation process and the corresponding preliminary results of the proposed reputation assessment mechanism are presented. In particular, experiments on travel preference data from real users (from the RideMyRoute application) were performed in order to assess the convergence and the clustering results of the modified k-means algorithm. Moreover, we conducted a simulation process with artificially generated user ratings in order to showcase robustness of the proposed reputation mechanism, against attacks from malicious users.

5.1. Clustering evaluation

A key component of the proposed reputation assessment mechanism consists of the modified k-means algorithm. Given the fact that this algorithm does not use the standard Euclidean distance metric in order to compute distance between clusters, it is necessary to evaluate the convergence of the algorithm. The typical k-means algorithm is guaranteed to converge to a local minimum, because it uses the typical Euclidean distance metric. In order to check the convergence of the modified k-means algorithm, an empirical analysis was performed.

In particular, travel preferences data from 172 anonymized real users of the RideMyRoute application were included in our evaluation process. The algorithm run for the three optimal number of clusters identified by the Elbow method. In each iteration, the within-clusters-sum-of-squares (*wcss*) was computed and its values were plotted against the iteration id. As shown in Figs. 2–4, in all cases, *wcss* decreased with the iteration number, finally reaching its local minimum value at the last iteration. Based on these results, it is empirically evident that the algorithm converges to a minimum vale of *wcss* in all cases.

Fig. 2 shows that, after four iterations and three iterations, on the first and second plot, respectively, the *wcss* remains constant. This occurs as after the given number of iterations, *wcss* decreases less than in the first iterations. The same applies to the plots shown in Figs. 3 and 4.

Finally, the validity of the generated clusters was evaluated. As there were no known class labels for the clusters of users, the results of the clustering process could not be evaluated with external evaluation criteria. Hence, an internal evaluation criterion was used, namely the Davies-Bouldin index. This index calculates and compares the average distance of the points to the centers of their clusters with the average distance between the centers of the clusters. A low value of the Davies-Bouldin index indicates low intra-cluster distances, i.e., high intra-cluster similarity, but also high inter-cluster distances, i.e., low inter-cluster similarity. The value of the index was calculated for all three optimal number of clusters, identified

Table 2
User rating vectors, average ratings and weights.

u	f_1	f_2	f_3	f_4	R_{av}	w
1	2	2	3	5	3	0.5
2	4	4	4	4	4	0
3	1	2	2	3	2	1

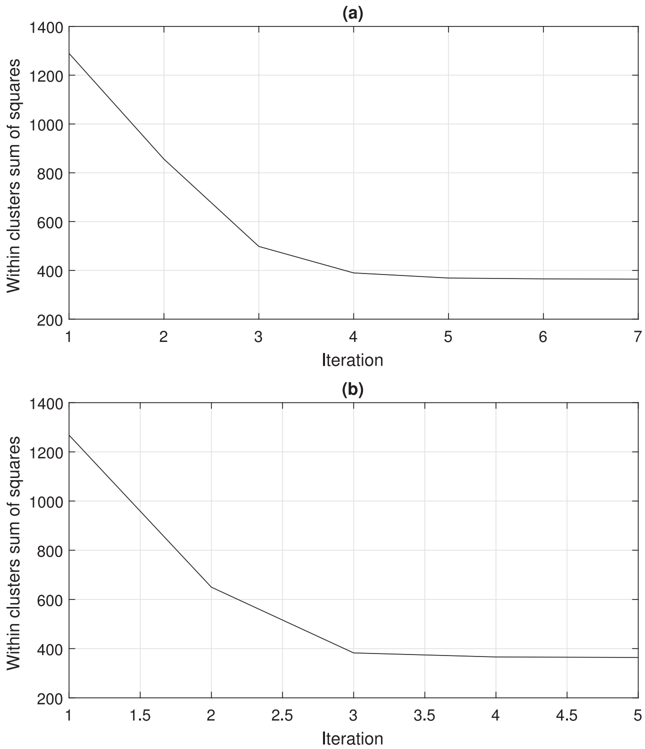

Fig. 2. Convergence results for the modified k-means algorithm with $k = 3$ clusters and (a) three random points or (b) the first three points as initial clusters.

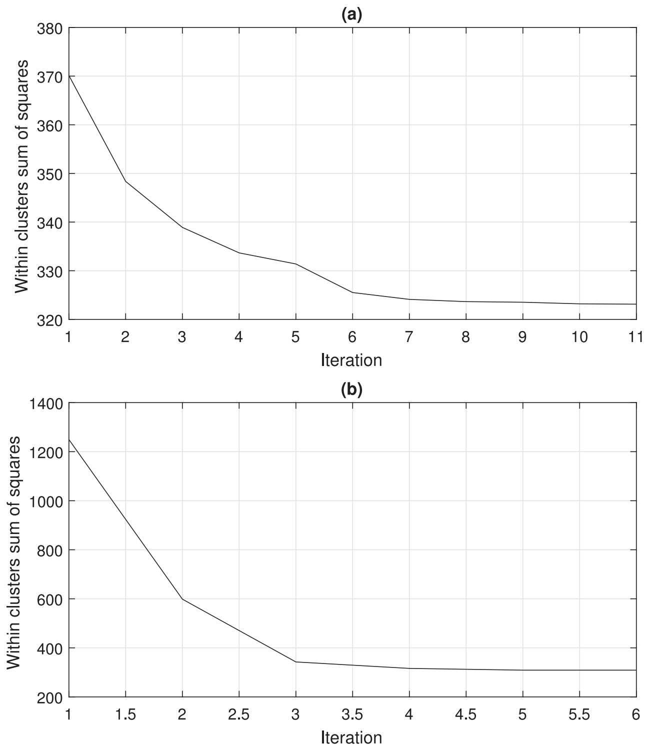

Fig. 3. Convergence results for the modified k-means algorithm with $k = 7$ clusters and (a) three random points or (b) the first three points as initial clusters.

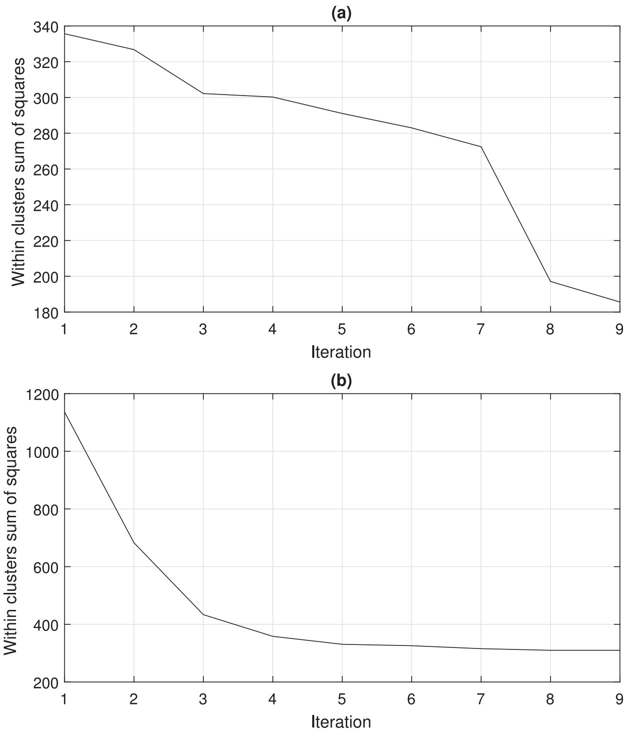

Fig. 4. Convergence results for the modified k-means algorithm with $k = 10$ clusters and (a) three random points or (b) the first three points as initial clusters.

by the Elbow method and the results are shown in Table 3. The lowest value of the Davies-Bouldin index occurs for three clusters and increases, as the number of clusters becomes higher.

5.2. Robustness evaluation against attacks from malicious users

One way to evaluate the performance of a reputation estimation mechanism is to check its robustness against attacks from malicious users. A mechanism that can be affected very quickly and easily by the behavior of malicious users cannot be considered robust. In order to evaluate the robustness of the proposed reputation assessment mechanism against attacks from malicious users, a simulation framework was designed and implemented using artificially generated user ratings. We imposed using artificial data due to the lack of sufficient number of users in the RideMyRoute application at the time of this writing.

The design and implementation of the simulation process was based on a set of logical assumptions. Firstly, a *malicious* user is defined as a user who registers in the system with the sole purpose of lowering the reputation of a specific target user, also known as *slandering attack*. When a malicious user participates in a ride with the target user, always gives the lowest possible value (i.e. 1 star) on all rating features, hence resulting in the lowest average rating value. This forms the selected rating policy for the malicious user. A malicious user can operate either on its own or in coalition with other malicious users, thus forming an *orchestrated attack*. The objective of our simulation process was to evaluate the robustness of the reputation assessment mechanism for an increasing penetration rate of malicious users, i.e., percentage of malicious users in the total set of RideMyRoute application users. For this purpose, the penetration rate was iteratively set in each iteration, from 0%, i.e., no malicious users, up to 90% with step s, given by Eq. (14), as shown below:

Table 3
Davies-Bouldin index values for the three optimal number of clusters.

Clusters	Davies-Bouldin index
3	0.836
7	1.982
10	2.089

$$s = \frac{100}{n_{sc}} \tag{14}$$

where n_{sc} the number of iterations, also known as *simulation cycles*. Finally, we define the *non-malicious* user as a typical user of the application who rates a target user in a uniform way, i.e., each rating variable follows a uniform distribution in the interval $[1,5]$. Hence, at the end of each ride a non-malicious user may submit a valuation from 1 to 5 stars for a random rating feature, with the same probability. This forms the selected rating policy for non-malicious users.

The first step of the simulation process was to create groups of users with similar travel preferences. For this purpose, real anonymized users from the RideMyRoute application were used, namely 172 users with complete travel preferences data. Based on this dataset, three different groupings of users were created for $3,7$ and 10 clusters, respectively, using our modified k-means algorithm. After clustering is executed, the assignment of each user to a cluster, as well as the distances between the clusters, are calculated.

The core simulation process is started in an iterative fashion. Initially, the malicious users are randomly selected from the total number of users, following a uniform distribution. In particular, the total set of users is split into two subsets, one of malicious and one of non-malicious users, respectively. Next, we define an assumed ride, in which the target user is the driver and the reviewing users are the passengers. We selected to have two passengers in total because this is the most typical number of passengers in carpooling applications. The passengers were randomly selected from all users. The possibility of a malicious user to appear is proportional to the penetration rate, which is set at a higher value at each iteration. Finally, all passengers, being either malicious or non-malicious, submit their ratings to the system, initiating the reputation assessment process.

In conclusion, in each iteration the following steps are executed:

1. The malicious users are selected from the total set of users.
2. For each user:
   (a) A hypothetical ride is defined. In this ride, the current user is the driver.
   (b) 2 users are selected randomly from the total set of users as passengers of the hypothetical ride.
   (c) The passengers provide ratings about the driver based on their type, i.e., if they are malicious or non-malicious, and the defined rating policies.
   (d) The ratings are submitted to the reputation assessment mechanism for execution.

At the beginning of each simulation cycle, all users have the same initial reputation value. We consider two different initial values for user reputation, in particular:

1. 0.5 for users with no previous ratings
2. 0.9 for users with only positive previous ratings

For each one of the two initial reputation values and the three different groupings of users, two separate simulations are executed for 100 iterations each. In the first simulation, the percentage of malicious users in the system remained constant to 0 in all iterations, whereas in the second one the percentage of malicious users increased from 0 to 90% by step s at each iteration. The simulation results for a specific target user are shown in Figs. 5 and 6.

As it is shown in Fig. 5, for no malicious users, the reputation of the target user initially increases from 0.5 to approximately 0.9 and then, from the 20*th* iteration onwards, it remains relatively constant. On the other hand, when the number of malicious users increases, the reputation of the target user lowers, but at a steady pace. In particular, the reputation decreases abruptly when the penetration rate becomes larger than 50%, 20% and 30% for $3,7$ and 10 clusters, respectively. This result can be also used for verifying the optimal number of clusters as suggested by the Elbow method. In particular, in our case, selecting the number of clusters with the highest gradient in the Elbow diagram, i.e. $K = 3$, leads to the most robust behavior against malicious attacks.

Again, in absence of malicious users the reputation of the target user initially shows a small drop from 0.9 to approximately 0.8, and after the 20*th* iteration, it remains relatively constant. On the contrary, in presence of malicious users, the reputation of the target user becomes lower, but more sharply this time, compared to the previous case. Also, when the number of clusters is 3 the reputation starts to drop more abruptly, when half of the users become malicious.

Considering the aforementioned evaluation results, a handful application is related to the design of reputation assessment mechanisms for carpooling systems, with a desired degree of tolerance against malicious attacks. Let us, for instance, define as a system requirement, the maximum tolerable deviation of user reputation, as a result of slandering attacks, to be 50%. In this case, if a target user has initial reputation of, let us say 0.8, a deviation of the reputation value up to 0.4 is acceptable as a result of attacks from malicious users. Based on the previously shown evaluation results, we can claim that our reputation assessment mechanism can handle penetration rates of malicious users up to 100%, 90% and 95% for $3,7$ and 10 clusters, respectively, when the initial reputation value of the target user is set to 0.5. Similarly, if the target user has an initial reputation value of 0.9, our mechanism satisfies the aforementioned requirement for maximum penetration rates of 70%, 50% and 65%, for the three numbers of clusters, as shown in Table 4. Based on those results, it becomes apparent that our mechanism can provide a suitable solution for carpooling systems, as it exhibits high tolerance against considerable penetration rates of malicious users, as shown above.

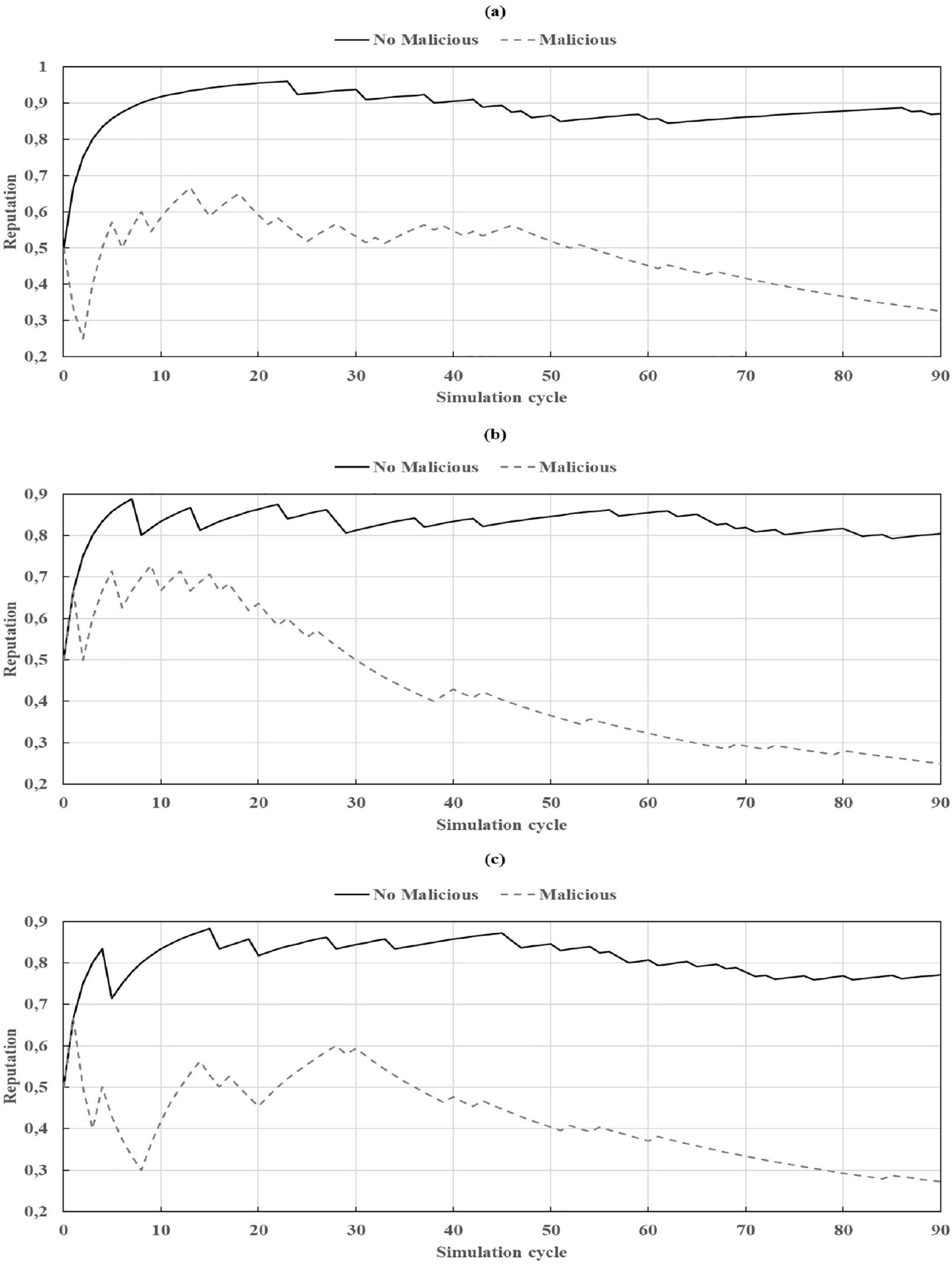

Fig. 5. Simulation results for initial reputation value 0.5 and (a) 3, (b) 7, and (c) 10 clusters.

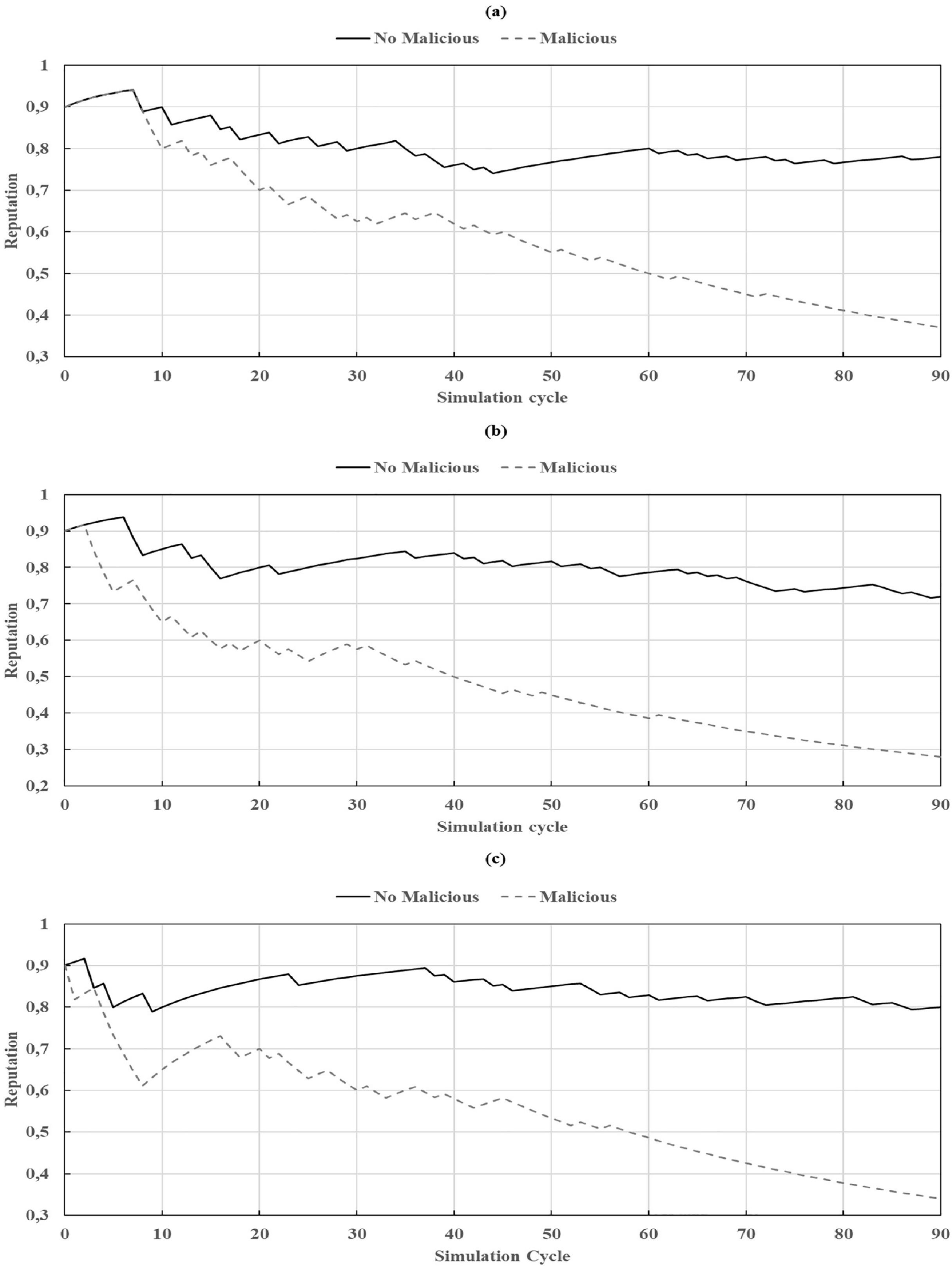

Fig. 6. Simulation results for initial reputation value 0.9 and (a) 3, (b) 7, and (c) 10 clusters.

Table 4
Tolerable penetration rates.

Clusters	Penetration Rate (%) Initial reputation 0.5	Penetration Rate (%) Initial reputation 0.9
3	100	70
7	90	50
10	95	65

6. Conclusions

This paper introduced a reputation assessment mechanism for carpooling applications. After conducting trials with simulated data, we ended up to the conclusion that the reputation of each user depends on both the ratings of other users submitted to the system at the end of each ride, and their travel preferences. Furthermore, we saw that the ratings of those users who had similar travel preferences with the user for whom the provided their evaluation, have a more significant impact on the reputation assessment process, compared to the ratings of users with different preferences. In order to check the validity of this statement, we created various clusters of users based on their travel preferences, by applying a modified k-means algorithm. Custom distance and averaging metrics were defined as part of the clustering mechanism that operated on multivariate, mixed-type vectors. By the time the clusters were constructed, they were used for calculating the reputation of a target user as the weighted average of the average ratings submitted by the evaluating users. In this weighted average scheme, the weights were calculated by the distances of the clusters of the users who submitted their ratings, from the center of the cluster to which the evaluated user belonged. Finally, the weighted average was transformed into a positive or negative feedback, which in turn used by the BRS to estimate the new reputation value. After systematic evaluation, it is safe to conclude that our proposed mechanism provides a robust solution against attacks from malicious users, even for penetration rates up to 100%, i.e. all users are malicious, when a target user has initially a relatively high reputation value, e.g. 0.8 and we require a maximum allowable deviation of user reputation to be less than 40%.

Future work includes extensive evaluation of the proposed mechanism with real users' ratings and different types of malicious attacks. In this context, we plan to provide a thorough mathematical proof of the convergence of the modified k-means algorithm, after examining how the textual description of ratings and the direct communication between the driver and the passengers affect the overall reputation assessment process.

Acknowledgments

This work was supported by the European Unions H2020 Research and Innovation Collaborative Project "SocialCar: Open social transport network for urban approach to carpooling" (Grant Agreement No. 636427).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.ijtst.2018.08.002.

References

Aberer, K., Despotovic, Z., 2001. Managing trust in a peer-2-peer information system. In: Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM'01), pp. 310–317.
Advogato, 2017. Advogato. URL <http://www.advogato.org/> [Online; accessed 27-March-2017].
Alswailim, M. A., Hassanein, H. S., Zulkernine, M., 2016. A reputation system to evaluate participants for participatory sensing. In: 2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings.
Amazon, 2017. Amazon. URL <https://www.amazon.com/> [Online; accessed 27-March-2017].
Caballero-gil, P., Martin-fernandez, F., Caballero-gil, C., 2015. Cooperative Social System based on Trust for Carpooling. IEICE Information and Communication Technology Forum.
Chen, C., Zhang, J., Cohen, R., Ho, P. H., 2010. A trust modeling framework for message propagation and evaluation in VANETs. In: 2010 2nd International Conference on Information Technology Convergence and Services, ITCS 2010.
Chen, L., Li, Q., Martin, K.M., Ng, S.-L., 2017. Private reputation retrieval in public a privacy-aware announcement scheme for VANETs. IET Inform. Secur. 11 (4), 204–210.
Coetzee, D., Fox, A., Hearst, M. A., Hartmann, B., 2014. Should your MOOC forum use a reputation system? In: Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing - CSCW '14. pp. 1176–1187.
Collotta, M., Pau, G., Salerno, V. M., Scata, G., 2012. A novel trust based algorithm for carpooling transportation systems. In: 2012 IEEE International Energy Conference and Exhibition, ENERGYCON 2012, pp. 1077–1082.
Créno, L., Cahour, B., 2015. Perceived risks and trust experience in a service of Carpooling. In: ITS World Congress (October), 2015, pp. 5–9.
Dötzer, F., Fischer, L., Magiera, P., 2005. VARS: a vehicle ad-hoc network reputation system. In: Proceedings - 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM 2005, pp. 454–456.
eBay, 2017. ebay. URL <http://www.ebay.com/> [Online; accessed 27-March-2017].
Gower, J.C., 1971. A general coefficient of similarity and some of its properties. Biometrics 27 (4), 857.
Huang, Z., Ruj, S., Cavenaghi, M.A., Stojmenovic, M., Nayak, A., 2014. A social network approach to trust management in VANETs. Peer-to-Peer Network. Appl. 7 (3), 229–242.

Jsang, A., Ismail, R., 2002. The beta reputation system. In: 15th Bled Electronic Commerce Conference, pp. 2502–2511.
Kamvar, S. D., Schlosser, M. T., Garcia-Molina, H., 2003. The Eigentrust algorithm for reputation management in P2P networks. In: Proceedings of the twelfth international conference on World Wide Web - WWW '03, p. 640.
Khalid, O., Khan, S. U., Madani, S. A., Hayat, K., Khan, M. I., Min-Allah, N., Kolodziej, J., Wang, L., Zeadally, S., Chen, D., 2013. Comparative study of trust and reputation systems for wireless sensor networks.
Li, B., Li, R.H., King, I., Lyu, M.R., Yu, J.X., 2015. A topic-biased user reputation model in rating systems. Knowl. Inform. Syst. 44 (3), 581–607.
Liao, H., Zeng, A., Xiao, R., Ren, Z.-M., Chen, D.-B., Zhang, Y.-C., 2014. Ranking reputation and quality in online rating systems. PLoS One 9 (5).
Ma, S., Wolfson, O., Lin, J., 2011. A survey on trust management for intelligent transportation system. In: Proceedings of the 4th ACM SIGSPATIAL International Workshop on Computational Transportation Science - CTS '11, pp. 18–23.
Martn-Fernndez, F., Caballero-Gil, C., Caballero-Gil, P., 08 2015. A trustworthy distributed social carpool method, pp. 324–335.
Minhas, U. F., Zhang, J., Tran, T., Cohen, R., 01 2010. Towards expanded trust management for agents in vehicular ad-hoc networks 5.
Patwardhan, A., Joshi, A., Finin, T., Yesha, Y., 2006. A data intensive reputation management scheme for vehicular ad hoc networks. In: Mobile and Ubiquitous Systems: Networking Services, 2006 Third Annual International Conference on, pp. 1–8.
StackOverflow, 2017. Stackoverflow. URL <http://stackoverflow.com/> [Online; accessed 27-March-2017].
Tian, C., Yang, B., 2011. R 2Trust, a reputation and risk based trust management framework for large-scale, fully decentralized overlay networks. Future Gener. Comput. Syst. 27 (8), 1135–1141.
Turkopticon, 2017. Turkopticon. URL <https://turkopticon.ucsd.edu/> [Online; accessed 28-March-2017].
Wahab, O.A., Bentahar, J., Otrok, H., Mourad, A., 2015. A survey on trust and reputation models for Web services: single, composite, and communities. Decision Support Syst. 74, 121–134.
Wang, J., Zhang, Y., Wang, Y., Gu, X., 2016. RPRep: a robust and privacy-preserving reputation management scheme for pseudonym-enabled VANETs. Int. J. Distrib. Sensor Networks.
Xiong, L., Liu, L., 2004. PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans. Knowl. Data Eng. 16 (7), 843–857.