Pers Ubiquit Comput (2012) 16:485–506
DOI 10.1007/s00779-011-0420-2

ORIGINAL ARTICLE

TruBeRepec: a trust-behavior-based reputation and recommender system for mobile applications

Zheng Yan · Peng Zhang · Robert H. Deng

Received: 8 November 2010 / Accepted: 20 April 2011 / Published online: 23 June 2011

Abstract Mobile applications are software packages that can be installed and executed in a mobile device. Which mobile application is trustworthy for a user to purchase, download, install, execute or recommend becomes a crucial issue that impacts its final success. This paper proposes TruBeRepec, a trust-behavior-based reputation and recommender system for mobile applications. We explore a model of trust behavior for mobile applications based on the result of a large-scale user survey. We further develop a number of algorithms that are used to evaluate individual user's trust in a mobile application through trust behavior observation, generate the application's reputation by aggregating individual trust and provide application recommendations based on the correlation of trust behaviors. We show the practical significance of TruBeRepec through simulations and analysis with regard to effectiveness, robustness, and usability, as well as privacy.

Z. Yan (✉)
Department of Communications and Networking,
School of Electrical Engineering, Aalto University,
Otakaari 5, Espoo 02150, Finland
e-mail: zheng.yan@aalto.fi; zhengyan.pz@gmail.com

Z. Yan
School of Telecommunications Engineering,
XiDian University, Xi'an 710071, China

P. Zhang
Research Institute of Mobile Internet,
Xi'an University of Posts and Telecommunications,
Weiguo Road Chang'an District, Xi'an 710121, China
e-mail: pzhang@xupt.edu.cn; pengzhangzhang@gmail.com

R. H. Deng
School of Information Systems, Singapore Management
University, 80 Stamford Road, Singapore 178902, Singapore
e-mail: robertdeng@smu.edu.sg

Keywords Reputation systems · Recommendation · Trust · Trust behavior · Mobile applications

1 Introduction

Mobile device has evolved into an open platform to execute various applications. Mobile applications are software packages that can be installed and executed in mobile devices, for example, a mobile email client to access emails in a mobile phone. Generally, mobile applications developed by various vendors can be downloaded for installation. Which mobile application is more trustworthy for a user to consume becomes a crucial issue that impacts its final success.

Trust is a multidimensional, multidisciplinary, and multifaceted concept. We can find various definitions in the literature. Common to these definitions are the notions of confidence, belief, and expectation on the reliability, integrity, ability, etc., or characters of an entity [1]. The trustworthiness of mobile applications relates to their dependability, security, and usability [2], as well as popularity [3]. Many reputation systems of applications evaluate application trust based on the number of download although it is not so accurate. Herein, we define a user's trust in a mobile application as his/her belief on the application that could fulfill a task as expectation. Reputation is public trust derived from direct and indirect knowledge or experiences. In our study, it is defined as the public belief on a mobile application that could fulfill a task according to many people's expectations. Obviously, trust plays an important role in application consumption and usage because it helps users overcome perceptions of uncertainty and risk and engages in "trust-related behaviors", in short trust behaviors. The trust behavior is a user's actions to depend on an application or

believe the application could perform as expectation, e.g., provide personal information to the application to engage in a purchase transaction, or use the application regularly to fulfill a routine task, or continue consuming the application even facing some errors [4]. However, a user's trust in a mobile application is, being highly subjective. It is built-up over time and changes with the use of the application due to the influence of many factors. As trust is an internal "state" of the user, it is hard to measure it directly.

Marsh reasoned that it might prove more suitable to model trust behavior rather than trust itself, removing the need to adhere to specific definitions [5]. Meanwhile, modeling trust behavior overcomes the challenges to measure a subjective concept by evaluating it through objective trust behavior observation, which actually provides a concrete clue of trust. Regarding mobile application usage, we posit that credible information is gained only after a mobile user has both engaged in trust behaviors (e.g., acting on using a mobile application) and assessed the trustworthiness of the application by observing the consequences of its performance and depending on it in his/her routine life.

However, few existing trust models explore trust in the view of human trust behaviors [6]. Thus, little work in the literature generates reputation and provides recommendations based on trust behaviors. In this paper, we propose TruBeRepec, a trust-behavior-based reputation and recommender system for mobile applications. We explore a model of trust behavior for mobile applications through a large-scale user survey with more than 1,500 participants. Its construct has been examined and proved with sound validity and reliability by principal components analysis, reliability analysis, and confirmatory factor analysis [3]. We further formalize this model in order to evaluate individual mobile user's trust in a mobile application through trust behavior observation. Thereafter, we design several algorithms to generate an application's reputation by aggregating individual trust and provide application recommendations based on the correlation of trust behaviors. The contributions of this paper are:

- TruBeRepec achieves auto-data collection for an individual user's trust evaluation through trust behavior observation and provides application recommendation based on trust behavior correlation;
- TruBeRepec supports both voting and non-voting; it has sound usability by reducing the need of user-device interaction and at the same time providing a convinced explanation on trust, which can be easily accepted by users since the trust explanation follows the model achieved from a large-scale user study;
- TruBeRepec's reputation scheme is robust according to our simulation results. It applies the device auto-generated individual trust as the credibility of user's voting, thus overcomes the unfair rating attack. Meanwhile, TruBeRepec adopts recommendation trust in reputation generation with the concern of recommendation quality and time decay in order to punish on–off attackers and conflict behavior attackers, as well as attackers on trust behaviors.
- TruBeRepec preserves user privacy since it does not require users to share and specify personal details, e.g., usage statistics and personal interests.

The rest of the paper is organized as follows. Section 2 gives a brief overview of related work in the literature. Section 3 introduces the trust behavior model for mobile applications. This is followed by TruBeRepec system design in Sect. 4. The algorithms used for individual trust evaluation, application reputation, and recommendation generation are described in Sect. 5. We further evaluate TruBeRepec through simulations and analysis in Sect. 6. Thereafter, we discuss some additional issues such as the practical significance and user privacy preservation in Sect. 7. Finally, conclusions and future work are presented in the last section.

2 Background and related work

2.1 Trust model

The method to specify, evaluate, setup, and ensure trust relationships among entities is the trust model while trust modeling is the technical approach used to represent trust [6]. One of the earliest formalizations of trust in computing systems was done by Marsh [5]. He integrated the various facets of trust from the disciplines of economics, psychology, philosophy, and sociology. Since then, many trust models have been constructed for various computing paradigms including ubiquitous computing, peer-to-peer systems, ad hoc networks, GRID virtual organizations, multi-agent systems, web services, e-commerce, and component software [6, 7]. In almost all of these studies, trust is accepted as a subjective notion, which brings us to the question: how to measure trust? Translation of this subjective concept into a machine-readable language is the main objective of trust modeling.

In computer science, a trust model aids the digital processing and/or management of trust. Most existing trust models are based on the understanding of trust characteristics, accounting for factors influencing trust. A common approach in the literature is with regard to computational trust [8–11]. Despite the availability of various trust models, the fundamental criteria of trust models are still not well understood. Current work focuses on concrete

solutions in specific systems. Additional examination is required before applying an existing solution into another domain.

One promising approach of trust modeling aims to conceptualize trust based on user studies through a psychological or sociological approach (e.g., using a measurement scale, i.e., measure). This kind of research aims to recognize the complicated relationships among trust and other multiple factors in different facets. However, the achieved trust model using this method is conceptual and semantic, thus cannot be directly applied into computer systems. Two examples are the initial trust model proposed by McKnight et al. [4] and the technology trust formation model (TTFM) studied by Li et al. [12]. *Initial trust* refers to trust in an unfamiliar trustee, a relationship in which the involved entities do not yet have credible, meaningful information about, or affective bonds with, each other [13]. These two models used the framework of the theory of reasoned action (TRA) to explain how people form initial trust in an unfamiliar entity [14]. Since the objective of both models was to predict initial trust (i.e., trusting intention) before any actual interaction with the trusting object, trust behavior was excluded from them.

On the other hand, *short-term trust* is built up over the first interactions with an entity and *long-term trust* is developed over the continuous interactions with an entity for a longer period of time. *On-going trust* concerns the short-term trust and the long-term trust. In our study, we mainly focus on the on-going trust evaluation based on the user's usage behaviors. In particular, the on-going trust could contribute to the trusting object's reputation and thus greatly help other entities generate their initial trust.

2.2 Trust behavior study

TRA theory posits that beliefs lead to attitudes, which lead to behavioral intentions, which lead to the behavior itself [14]. Applying this theory, we propose that trusting beliefs (e.g., perceptions of specific mobile application attributes) lead to trusting intentions (e.g., intention to engage in trust behaviors of using a mobile application through user-device interaction), which in turn result in trust behaviors (e.g., using the application in various context). Additionally, numerous researchers have conceptualized trust as a behavior which has been validated in work collaboration and social communications [15–17]. Prior research has also confirmed a strong correlation between behavioral intentions and actual behavior, especially for software system usage [18, 19]. However, still very few studies examined trust from the view of trust behaviors [20]. Some work studies the trust behavior in e-banking [20]. To our knowledge, no existing work explores trust behavior of mobile application usage, which is a different context from the above research domains with regard to running environment and user interface. Due to the above differences and the challenges caused by small device interface, the design of reputation systems for mobile applications has additional challenges considering usability and performance.

Muir found a positive correlation between trust and use [21, 22]. The relationship between trust and interaction behavior is obvious since usage through human–device interaction implies trust. Lee and Moray [23] found that trust in a system partially explained system use, but other factors (such as the user's own ability to provide manual control) also influenced the system use. All above studies serve as the foundation of our work: a user's trust in a mobile application can be evaluated based on the user's trust behaviors. It actually plays as our hypothesis to explore and confirm the structure of trust behavior model for mobile application usage through a large-scale user experiment. However, these studies do not provide any implications on the design and development of a reputation and recommender system for mobile applications. In our work, based on the explored and verified trust behavior structural model, we formalize it in a mathematic measure and further design a reputation and recommender system for mobile applications that can be applied in practice with sound effectiveness.

Existing trust behavior studies focus on human's trust in an automation and intelligent machine [21–23]. A number of trust models have been proposed in the context of e-commerce [4, 20] while little work has been done in the context of mobile applications. Prior arts also lack study on the influence of recommendations, personality and usage context on human–computer trust. With the rapid development of mobile computing technologies, a mobile device has become a multi-application system for multipurpose and multi-usage. A mobile device is an open platform with always network connection that allows deploying new or upgraded applications at anytime and anywhere. Therefore, such a dynamically changed system introduces new challenges for human–computer trust behavior study.

2.3 Trust management, reputation, and recommender systems

Trust management is concerned with: collecting the information required to make a trust relationship decision; evaluating the criteria related to the trust relationship, monitoring, and reevaluating the existing trust relationships, as well as ensuring it dynamically; and automating the process [6, 7, 24]. Recently, trust management has emerged as a promising technology to facilitate collaboration among

entities in a distributed and uncertain environment [25]. However, prior arts generally lack considerations on the means to gather experiential evidence for effective trust evaluation. Many systems rely on a user to provide feedback [6]. Sometimes, it may not be appropriate or convenient to request him/her feedback, especially for a mobile user. This is because the user may be bothered by such a request during usage. His/her usage experience could be negatively influenced. Moreover, user interface design for feedback requests extra design efforts, which may cause additional challenges for mobile devices with small displayers. Another issue is different users may apply different scales in the feedback, which may cause confusion, even attacks. All above introduces a requirement of automating the experiential evidence in a uniformed norm. In our opinion, observing trust behaviors directly during mobile application usage could be a good way to automatically collect evidence for trust evaluation in a uniformed norm. On the other hand, we found that most methods applied in trust management are not user-centric or user-driven [26], lacking considerations on or support from users in system design. Few of them study trust based on the trusting subject's behaviors, thus trust management in these systems is mostly based on trust evaluation on the trusting object's behavior or performance [6]. The work presented in this paper explores users' trust in a mobile application based on their trust behaviors (i.e., the trusting subject's behaviors). Our study supports automatic evidence collection for trust evaluation and management.

There are various trust management systems in the literature and practice [25]. However, it still lacks common criteria to evaluate these systems. System context diversity complicates the situation. Most literature results are difficult to be directly applied in practice because the assumed conditions are actually hard to be satisfied and the system design cannot fulfill practical requirements, e.g., usability and privacy [26, 27].

2.3.1 *Reputation and recommender systems*

A category of large practical importance is reputation-based trust management system, in short, reputation system. Reputation is a measure that is derived from direct or indirect knowledge on earlier interactions of entities and is used to assess the level of trust put into an entity [28]. Thus, the reputation system is a specific approach to evaluate and manage trust. Recommender systems generally apply information filtering technique that attempts to recommend information items (e.g., films, books, web pages, etc.) that are likely to be of interest to users [29]. Typically, a recommender system compares a user profile to some reference characteristics, and seeks to predict the "rating" that a user would give to an item they had not yet considered or experienced [30]. These characteristics may be from the information item (a content-based approach) or the user's social environment (a collaborative filtering approach) [31]. In [32], the authors introduced using trust as both weighting and filtering in recommendations. The recommendation partners should have similar tastes of preferences and they should be trustworthy with a history of making reliable recommendations. This trust information can be incorporated into the recommendation process. But to our knowledge, most characteristics used for recommendations are not based on trust behaviors, which however is an important clue to imply users' preferences.

Jøsang et al. [33] classified reputation network architecture into two main types: centralized and distributed. The network architecture determines how ratings and reputation scores are communicated between participants in a reputation system. In the literature, distributed trust evaluations have been studied for distributed systems, e.g., ad hoc networks and peer-to-peer systems [8–11]. On the other hand, practical reputation systems generally apply a centralized server to collect feedback for reputation generation. However, many existing systems (e.g., Amazon, eBay [34], Yahoo auctions [35]) lack considerations on the credibility of a user's rating. This greatly influences the quality of produced reputations. Moreover, the centralized reputation network architecture may not be suitable and flexible to be applied in the context of mobile applications. TruBeRepec adopts hybrid reputation network architecture. The algorithms designed to generate reputation and recommendation can be applied in both the mobile devices and a centralized reputation service provider.

Reputation and recommender systems still face several challenges. Firstly, incentives are required in order to encourage users to provide their feedback on interactions and their personal interests and profiles (due to privacy issue, some users are hesitate to provide their details) [29, 35]. This may raise some usability issues, especially for mobile users. Secondly, reputation systems may face the problem of unfair ratings to artificially inflate or deflate reputations [34–37]. They are vulnerable to a number of potential attacks, such as Sybil attack, on–off attack, independent bad mouthing attack, collaborative bad mouthing attack, and conflict behavior attack [37–39]. The usage of pseudonyms introduces new challenges by making it hard to trace malicious behaviors. This also influences the accuracy of reputation. Sun et al. [39, 40] proposed a number of schemes to overcome some of the above attacks, but they did not consider the additional challenges caused by usability and privacy preservation. In addition, collecting the reference characteristics for recommendation sometimes is not easy due to the users' concern of privacy. Thirdly, the existing reputation and recommender systems based on user rating generally lack uniform criteria, which makes the rating a totally subjective behavior. Meanwhile,

different users could treat and consider the reputation and recommendation information in different ways. These further complicate the users' decision and could negatively influence their usage behaviors. However, credibility is a positive signal of the trustworthiness of an object [41] since it provides a reason to trust. Feedback credibility is essential to generate a reliable reputation value in order to overcome the above challenges. Particularly, trust behaviors provide an important clue to indicate feedback credibility and users' preference.

Obviously, the success of a practical reputation and recommender system requests sound usability with regard to user-device interaction. It should be robust to overcome various potential attacks. Meanwhile, a mechanism is expected to uniform the user's voting with trustworthy credibility. Finally, the system should preserve the user's privacy to a certain level at the same time when it collects user data for reputation and recommendation generation.

3 Trust behavior model

In order to achieve the trust behavior model, we design a questionnaire survey that asks for user opinion about trust behaviors regarding mobile application usage. Based on the data collected from 1,575 participants, we get a conceptual trust behavior model, as shown in Fig. 1 [3]. The construct of the model and the relations among all factors are analyzed and validated using principal component analysis (PCA), confirmatory factor analysis (CFA), correlation analysis, and reliability analysis with positive psychometric properties and sound validity and reliability [3]. The relationships of different components (i.e., the edge values in Fig. 1) are set based on the correlation analysis with the values in the scope of [0, 1]. The model relates the trust behavior to three types of usage behaviors: using behavior (UB), reflection behavior (RB), and correlation behavior (CB). These behaviors can be automatically monitored by a mobile device during application consumption. They also relate to a number of external factors: personal motivation, brand impact, perceived device quality, and personality. They are further delineated into twelve measurable sub-constructs. Figure 2 illustrates the sub-construct of the UB, RB, and CB according to the PCA, CFA, and correlation analysis.

What follows refers to the notations used in Fig. 2. The first type of trust behavior—using behavior relates to normal application usage, which can be reflected mainly by elapsed usage time, number of usages, and usage frequency. We found that trust is reflected by UB1, normal usage behavior. Meanwhile, usage context such as risk, importance, and urgency could also influence the trust behavior (i.e., UB2: behavior related to context). Generally, a mobile application provides a number of functionalities, i.e., features. The more features experienced by the user, the more proficient he/she has in the application usage (i.e., UB3: feature related usage behavior). What is more, frequent usage can somehow indicate trust. This is also reflected from the user data collected in our survey [3]. Herein, we pay more attention to public trust, i.e., reputation, which aggregates many users' trust opinions on an application.

The second type of trust behavior is reflection behavior. It concerns the usage behaviors after the user confronts application problems/errors or has good/bad usage experiences. It contains six sub-constructs: RB1: bad performance reflection behavior; RB2: bad performance reflection behavior related to context; RB3: good performance reflection behavior; RB4: good performance reflection behavior related to context; RB5: bad experience reflection to context; RB6: good experience reflection to context. The difference of the reflection behavior and the using behavior lies in the fact that the first is a type of

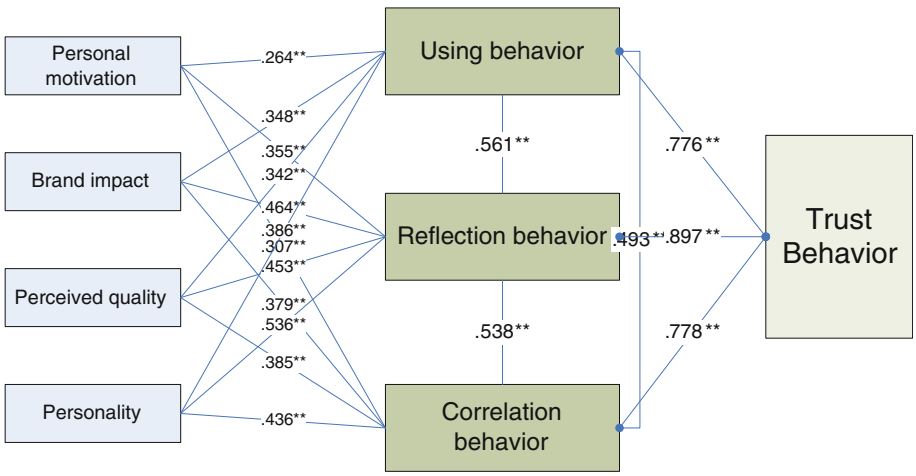

Fig. 1 Trust behavior construct of mobile applications

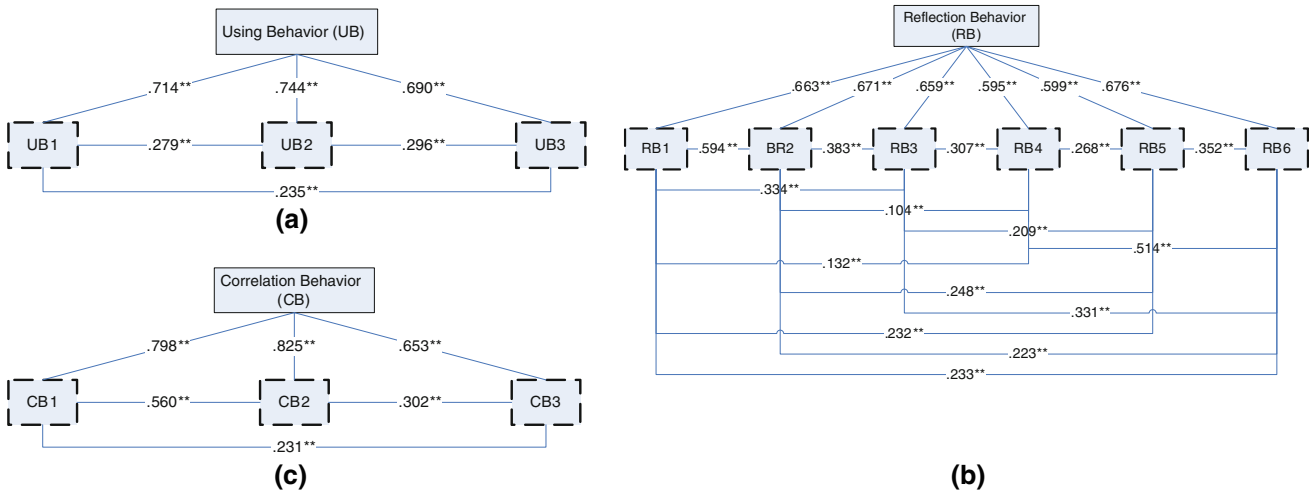

Fig. 2 Internal relationships of **a** UB; **b** RB; and **c** CB

event-related behavior while the second is about general usage statistics. Their contributions to trust evaluation could be different. For example, one type of the reflection behavior is the usage behavior when the user confronts an application error, whether he/she would like to continue using the application or not in such a situation. The using behavior only reflects normal usage information, not indicates the change of usage.

Future mobile market could be very competitive. A number of similar functioned mobile applications developed by different vendors would be available at the same time for consumption. The third type of behavior is correlation behavior, which concerns the usage behaviors correlated to a number of similar functioned mobile applications. Since trust is obviously correlated to use [21–23], the usage could imply trust. Meanwhile, it is also influenced by various contexts [5, 15]. The correlation behavior has 3 sub-constructs: CB1: comparison of normal usage behavior; CB2: comparison related to context; CB3: recommendation behavior (i.e., a behavior to suggest other people using a mobile application).

The trust behavior construct (i.e., the trust behavior model) for mobile applications is achieved with sound reliability (UB: alpha = 0.71; RB: alpha = 0.85; CB: alpha = 0.79; overall trust behavior: alpha = 0.90) [3]. Reliability is reflected by alpha, a value between 0 and 1, with a larger value indicating better reliability. Generally, alpha above 0.7 implies sound reliability [42]. We found that UB, RB, and CB have significant correlation (as shown in Fig. 1) with the trust behavior, which indicates that these three factors can represent it. We also found that these factors have lower correlations with each other than their correlations with the trust behavior. This indicates that these three factors can measure not only the general aspects but also the specific aspects of the trust behavior. Notably, their mutual correlations are around 0.5, which implies that these factors may influence or impact with each other. However, the assumed relationships cannot be well proved by internal nomological validity of our experiment and in literature theory. This means that these factors could be correspondingly in parallel, without any causal relationships. We also found the influence of a number of external variables (i.e., personal motivation, brand impact, perceived device quality, and personality) on UB, RB, and CB; their correlations are shown in Fig. 1. Note that ** indicates correlation is significant at the 0.01 level (2-tailed); * indicates correlation is significant at the 0.05 level (2-tailed). Herein, the level of correlation significance indicates the error probability of correlation. Thus, the lower the level, the more significant correlation holds.

As can be seen from Fig. 2, the correlation between each internal sub-factor (e.g., UB1, UB2, and UB3) and its corresponding principal factor (construct) (e.g., UB) is almost in the same level (except CB3's correlation with CB is a bit lower than CB1-CB's and CB2-CB's). This correlation is also higher than the correlations among the sub-factors. This indicates that the sub-factors belonging to a concrete principal factor can measure not only the general aspects but also the specific aspects of the represented type of trust behavior.

Our work in [3] only explores the conceptual structure of trust behaviors for mobile application usage based on a user survey. This paper designs TruBeRepec by formalizing the trust behavior model and developing a number of algorithms that can be adopted by TruBeRepec to provide application reputation and recommendation according to trust behaviors.

4 System design

We design a distributed client–server system structure for TruBeRepec, see Fig. 3. Its client software "Trust Manager" can be installed in a number of mobile devices (MD_k, $k = 1,\ldots,K$). The trust manager contains Trust Behavior Monitor that monitors trust behaviors and inputs statistical data about UB, RB and CB into a secure storage (Trust Data), which is located inside the device platform and has a secure channel to communicate with the behavior monitor and Trust/Reputation Information Presenter. The statistical data can be accessed by Data Interpreter for (a) individual trust evaluation regarding a specific application by Trust Evaluator; (b) data dissemination to send local trust information and vote applications to reputation service provider (RSP) or other devices by Trust Value Disseminator; (c) reputation/recommendation extraction to get mobile applications reputation information and/or application recommendations from the RSP or other devices by Reputation/Recommendation Extractor. Particularly, the Trust Evaluator can also generate application reputation and recommendation at the user's device based on collected information from other mobile devices, e.g., through an ad hoc network. The Data Interpreter is a secure mechanism to access the user's usage statistical data from the Trust Data since these data are private information. We design the Data Interpreter based on the trusted computing technology [25, 43]. Only authorized data interpreter mechanisms can access and unseal the protected usage information. The reputation/recommendation extraction can be tailored based on the mobile user's preference, either a reputation extraction policy or a recommendation extraction policy, or both. In addition, Trust/Reputation Information Presenter is applied to show trust/reputation information to the user in order to aid his/her application usage [44, 45].

In the RSP, Trust Value Receiver receives individual trust information and votes automatically or by request from the mobile devices. Reputation/Recommendation Generator generates application reputations and recommendations for mobile users. Herein, the reputation could be generated based on all users' usage statistics. But due to a privacy concern, we apply another approach to aggregate the individual trust values based on UB, RB, and CB (calculated in each mobile device). The reputation/recommendation information about each mobile application is saved in a secure storage (Reputation Data) in the RSP. This information can be retrieved and distributed to the mobile devices through Reputation/Recommendation Distributor. It receives reputation retrieve requests and provides application reputations and recommendations to the requestors.

Assuming a mobile application, a user receives a recommendation from the RSP that indicates its high public reputation and high personalized recommendation to install it since most people using it consume other applications in a similar way as he does regarding trust behaviors. Further initiated by some external factors (e.g., personal motivation, brand impact, perceived device quality, and personality), the user installs the application and starts consuming it. The Trust Behavior Monitor monitors his trust behaviors regarding UB, RB, and CB and inputs collected statistical data into the Trust Data. Based on these data, the Trust Evaluator evaluates the user's individual trust in the

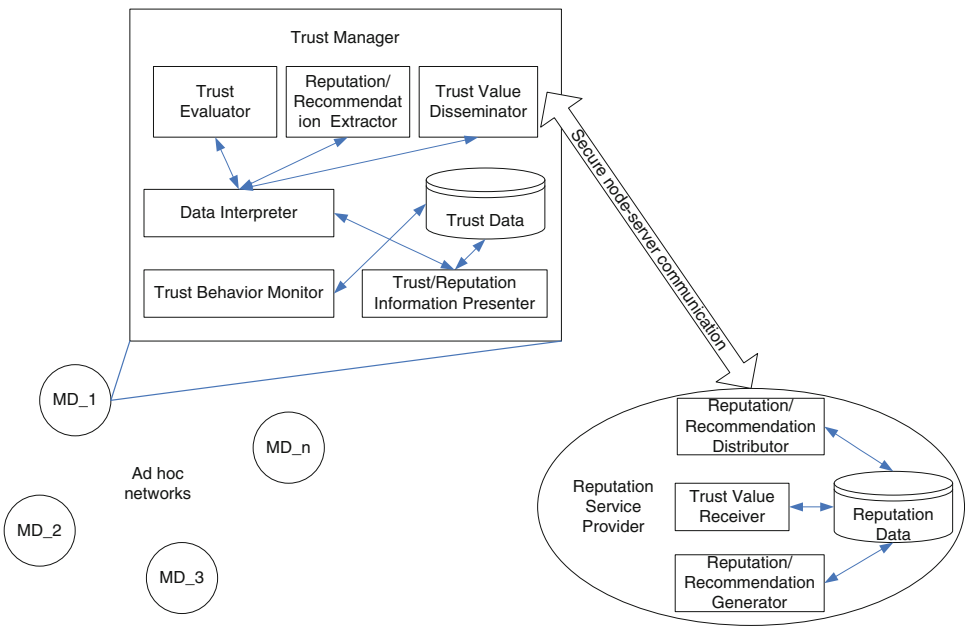

Fig. 3 TruBeRepec system structure

application. The Trust/Reputation Information Presenter shows the application's individual trust value, its reputation value and recommendation value to the user. Periodically or by request, the user sends local trust information (e.g., individual trust value, the trust values, respectively, contributed by UB, RB, and CB) and/or application votes to the RSP. The RSP then regenerates the application's reputation and reprovides the application's recommendations based on newly collected data. If the recalculated recommendation suggests the user not trusting in the application that he/she has already installed, the device could automatically inquire the user if uninstall is needed or warning is expected at the application start-up. The user could configure his/her personal settings in the mobile device to handle this kind of situations.

Particularly, the user can recommend this application to his friend directly, for example via an ad hoc network. In this case, the user's trust information is attached to the recommendation message. His friend's device can generate the application's reputation and recommendation value (at Trust Evaluator) based on the recommender's individual trust and their trust behavior correlations with regard to commonly consumed applications.

5 Algorithms

Based on the above system design and the trust behavior model, we propose a number of algorithms to implement individual trust evaluation in a mobile device, and application reputation and recommendation generation. For easy of reference, Table 1 summarizes the notations used in this section.

5.1 Individual trust evaluation

We formalize the conceptual trust behavior model in a computational measure. It is a coherent adaptive trust model for quantifying the individual user's trust in a mobile application based on trust behavior observation.

5.1.1 Formalizing using behavior

The PCA assumes that the extracted factors are based on linear combinations. Formalizing the using behavior, we consider the influence of the number of usages, elapsed usage time, usage frequency, and experienced features on trust based on the trust behavior model. Their influence is scaled by the total number of usages and elapsed usage time of all applications, the total number of the application features and the usage frequency of all mobile applications in the underlying device (refer to UB1 and UB3). Meanwhile, we further tailor trust based on the index of importance, urgency and risk, i.e., the context index (CI), refer to UB2. Denoting the importance index (ii), urgency index (ui), and risk index (ri) of the nth mobile application i usage as ii (i, n), ui (i, n) and ri (i, n), then

$$\begin{aligned}\mathrm{CI}_i(t) &= \frac{1}{N_i(t)}\sum_{n=1}^{N_i(t)} \mathrm{ci}(i,n) \\ &= \frac{1}{N_i(t)}\sum_{n=1}^{N_i(t)} (\alpha * \mathrm{ii}(i,n) + \beta * \mathrm{ui}(i,n) + \gamma * \mathrm{ri}(i,n))\end{aligned} \tag{1}$$

where, α, β, and γ are parameters to weight the importance of different context indices. Thus, the individual trust contributed by the using behavior $T_i(t)_{\mathrm{UB}}$ can be calculated as:

$$T_i(t)_{\mathrm{UB}} = \left(\frac{N_i(t)}{N(t)} * \frac{\mathrm{UT}_i(t)}{\mathrm{UT}(t)} * \frac{\mathrm{EF}(i)}{F(i)} * \frac{\mathrm{FE}_i(t)}{\mathrm{FE}(t)}\right) * \mathrm{CI}_i(t) \tag{2}$$

In our user study [45], we found that the importance rate is highly related to the elapsed usage time, frequency, and the number of usages. If the weights of urgency index and risk index are 0 (i.e., $\beta = 0$, and $\gamma = 0$), we can simplify formula (2) as:

$$\mathrm{CI}_i(t) = \mu * \left(\frac{N_i(t)}{N(t)} * \frac{\mathrm{UT}_i(t)}{\mathrm{UT}(t)} * \frac{\mathrm{FE}_i(t)}{\mathrm{FE}(t)}\right) \tag{1'}$$

$$T_i(t)_{\mathrm{UB}} = \mu * \left(\frac{N_i(t)}{N(t)} * \frac{\mathrm{UT}_i(t)}{\mathrm{UT}(t)} * \frac{\mathrm{FE}_i(t)}{\mathrm{FE}(t)}\right)^2 \left(\frac{\mathrm{EF}(i)}{F(i)}\right) \tag{2'}$$

where μ is the parameter used to adjust the context index.

5.1.2 Formalizing reflection behavior

Our user study on the reflection behavior showed that the change of the elapsed usage time, the number of usages, the usage frequency and the change caused by the CI have influence on trust (refer to RB1–RB6). We introduce a parameter called performance index (PI) that can be used to reflect application performance

$$\mathrm{PI}_i(t) = d_t\{N_i(t) + \mathrm{UT}_i(t) + \mathrm{FE}_i(t)\} + d_t\{\mathrm{CI}_i(t)\} \tag{3}$$

where, $d_t\{g(t)\} = \frac{g(t)-g(t-\tau)}{\tau}$, $(\tau \to 0)$; $g(t)$ is a function of t; and τ is a time interval applied to measure the changes of usage behavior and context. For the same reason mentioned above, we simplify the formula (3) as

$$\mathrm{PI}_i(t) = 2(d_t\{N_i(t) + \mathrm{UT}_i(t) + \mathrm{FE}_i(t)\}) \tag{4}$$

The contribution of the reflection behavior to individual trust generation $T_i(t)_{\mathrm{RB}}$ can be specified as

$$T_i(t)_{\mathrm{RB}} = \mathrm{PI}_i(t) \tag{5}$$

Table 1 Notations

Notation	Description
t	The time variable
$T_i(t)$	The individual user's trust in application i at time t
$T_i(t)_{\mathrm{UB}}$	The individual trust in application i at time t contributed by UB
$T_i(t)_{\mathrm{RB}}$	The individual trust in application i at time t contributed by RB
$T_i(t)_{\mathrm{CB}}$	The individual trust in application i at time t contributed by CB
I	The total number of applications in a user's device
W	The time window applied to collect usage information
$N_i(t)$	The total number of usages of application i within W at time t
$N(t)$	The total number of usages of all applications in a device within W at time t
$\mathrm{UT}_i(t)$	The total elapsed usage time of applications i within W at time t
$\mathrm{UT}(t)$	The total elapsed usage time of all applications in a device within W at time t
$\mathrm{FE}(t)$	The usage frequency of all applications in a device within W at time t
$\mathrm{FE}_i(t)$	The usage frequency of application i within W at time t
$\mathrm{NR}(t)$	The total number of recommendations on all applications within W at time t
$\mathrm{NR}_i(t)$	The number of recommendations on applications i within W at time t
$F(i)$	The total number of features of application i
$\mathrm{EF}_i(t)$	The user experienced number of features of application i at time t
$\mathrm{ci}(i, n)$	The context index of application i regarding the nth usage
$\mathrm{ii}(i, n)$	The importance index of application i regarding the nth usage
$\mathrm{ui}(i, n)$	The urgency index of application i regarding the nth usage
$\mathrm{ri}(i, n)$	The risk index of application i regarding the nth usage
$\mathrm{CI}_i(t)$	The context index representing the importance, urgency and risk factors of application i usage in W
$\mathrm{ac}(i, k)$	The correlation factor indicating the similarity of application i and application k, $ac(i, k)$ ranges in the real interval [0,1]
$f(x)$	The Sigmoid function $f(x) = \frac{1}{1+e^{-x}}$ used to normalize the trust value into (0, 1)
$\mathrm{PI}_i(t)$	The performance index that indicates an application i's performance change
$\rho, \vartheta, \varsigma$	The normalized weight factors for using behavior evaluation, reflection behavior evaluation, and correlation behavior evaluation, respectively
u_k	The user k
V_i^k	The user k's vote on application i;
$V_i^k(t_p)$	The user k's vote on application i at time t_p
$\overline{R^k(i)}$	The aggregated reputation of application i based on user k's experiences
τ	The parameter to control time decaying
$T_i^k(t_p)$	The individual trust of user k in application i reported at time t_p
$R(i)$	The application i's public reputation
K	The number of users who consume application i
K'	The total number of users in the TruBeRepec system
$\theta(K)$	The Rayleigh cumulative distribution function to model the impact of K
ε	$\varepsilon = -\mathrm{K}/\mathrm{K}'$, the factor to indicate the popularity of an application
s^k	The user k's recommendation trust s^k;
δ	The parameter to control the adjustment of s^k;
γ	The warning flag to record the number of bad input into reputation generation
thr	The threshold to indicate the on–off attack or conflict behavior attack
μ	The parameter to control bad input punishment
max (V_i^k)	The maximum reputation input value
min (V_i^k)	The minimum reputation input value
φ	The parameter to decide a bad input
A	The set of applications: $A = \{a_1, a_2, \ldots, a_i, \ldots, a_I\}$
U	The set of users: $U = \{u_1, u_2, \ldots\ldots, u_K\}$
$D(u_k)$	The metric to present u_k's trust behaviors regarding applications A
$D(U)$	The metric that expresses all users' trust behaviors
R_i^k	The recommendation vector at time t for u_k regarding application i
σ	The parameter that inversely controls how fast the number of recommender's impact on R_i^k
$\mathrm{Rel}(u_j, u_k)$	The correlation between u_j and u_k with regard to trust behaviors
N_K	The population K's influence on R_i^k
$*$	The multiply operator

5.1.3 Formalizing correlation behavior

Trust based on the correlation behavior contains two parts. The first part reflects the comparison of normal usage behavior and the level of context index to similar applications (refer to CB1 and CB2). The second part reflects the recommendation behavior (refer to CB3). Herein, we deduct the contribution of the recommendation behavior according to current trust value $T_i(t)$ and context index $\mathrm{CI}_i(t)$. We have

$$T_i(t)_{\mathrm{CB}} = \sum_{k=1,k\neq i}^{I} ac(i,k) * \left\{ \begin{array}{l} \dfrac{N_i(t)-N_k(t)}{N(t)} + \dfrac{\mathrm{UT}_i(t)-\mathrm{UT}_k(t)}{\mathrm{NT}(t)} + \dfrac{\mathrm{FE}_i(t)-\mathrm{FE}_k(t)}{\mathrm{FE}(t)} \\ + (\mathrm{CI}_i(t)-\mathrm{CI}_k(t)) + \dfrac{\mathrm{NR}_i(t)-\mathrm{NR}_k(t)}{\mathrm{NR}(t)} \end{array} \right\} + \lambda \frac{\mathrm{NR}_i(t)}{\mathrm{NR}(t)} * \mathrm{CI}_i(t) * T_i(t) \quad (6)$$

where λ is a parameter that weights the contribution of the recommendation behavior. An important reason to introduce λ is the correlation of CB3 to CB is lower than CB1's and CB2's correlation to CB, as shown in Fig. 2. We use *ac(i, k)* to indicate the similarity of application *i* and *k*.

5.1.4 General metric of individual trust

The PCA assumes that the observed data set is linear combinations of certain basis. Aggregating all the above together, we get the following uniformed formula for individual trust evaluation.

$$T_i(t) = T_i(t)_{\mathrm{o}} + \rho T_i(t)_{\mathrm{UB}} + \vartheta T_i(t)_{\mathrm{RB}} + \varsigma T_i(t)_{\mathrm{CB}} \quad (7)$$

where parameters ρ, ϑ, and ς denote the normalized weight factors for using behavior evaluation, reflection behavior evaluation, and correlation behavior evaluation.

This metric consists of four parts. The first part is the original trust value, which could be an initial trust value at the beginning of the application usage or a trust value generated in the previous time window. The initial trust value could be negative since the usage could go down or a user could prefer using another similar application. The second part is a pure usage experience based trust evaluation according to the using behavior. We consider the influence of elapsed usage time, frequency and the number of usages, and experienced application features, as well as context influence. The third part is contributed by the reflection behavior according to the application's performance, which is reflected by usage changes and context index change. The last part is a weighted evaluation contribution about the correlation and recommendation behaviors. It takes the current trust value into account to counter dishonest recommendations, and capture the context influence on the recommendations. This history-based evaluation can be seen as a prediction for the recommendation behaviors regarding its contribution to the trust evaluation. Inside the last part, there is an application-comparison based contribution. It adjusts the trust value based on the difference of usage number/time/frequency, recommendations, and the context index with regard to similar functioned applications. In order to uniform the trust value into the scope of (0, 1), we apply a sigmoid function on the trust value

$$T_i(t) = f\{T_i(t)_{\mathrm{o}} + \rho T_i(t)_{\mathrm{UB}} + \vartheta T_i(t)_{\mathrm{RB}} + \varsigma T_i(t)_{\mathrm{CB}}\} \quad (8)$$

Important to note is that this general metric of individual trust may have different appearances depending on which of the parameters are switched on and how the parameters and weight factors are set. The setting of ρ, ϑ, and ς could be based on the correlation of UB, RB, and CB to trust behavior as 0.776, 0.897 and 0.778. Algorithm 1 is applied to evaluate individual trust at the Trust Evaluator (refer to Fig. 3).

Algorithm 1 Individual trust evaluation

1. **Input**:
2. - t: the time to calculate individual trust;
3. - i $(i = 1, \ldots, I)$: the identity of mobile application;
4. - I W, t, $N_i(t)$, $N(t)$, $\mathrm{UT}_i(t)$, $\mathrm{UT}(t)$, $\mathrm{FE}(t)$, $\mathrm{FE}_i(t)$;
5. - $R(t)$, $R_i(t)$, $F(i)$, $\mathrm{EF}_i(t)$, ci(i, n), $\mathrm{CI}_i(t)$, $\mathrm{CI}(i)$.
6. **For** each mobile application i, **do**
7. Calculate $T_i(t)_{\mathrm{UB}}$ based on (1) or (1');
8. Calculate $T_i(t)_{\mathrm{RB}}$ based on (4), (5);
9. Calculate $T_i(t)_{\mathrm{CB}}$ based on (6);
10. Generate individual trust in application i based on (8).
11. **Output**: $T_i(t)$ (i = 1, …, I).

Algorithm 2 Application reputation generation

1. **Input**:
2. - $V_i = \{V_i^1, V_i^2, \ldots, V_i^K\}$: K reports regarding application i;
3. - $T_i^k(t_p)(k = 1, \ldots, K)$: individual trust of user k in application i at time t_p;
4. - K: the total number of individual trust reports and votes;
5. - K': the number of registered system users.
6. **For** each application i, **do**
7. **For** each user k who consumes application i, **do**
8. Aggregate the reputation of application i based on user k's experiences according to (9);
9. Generate application i's reputation based on (10).
10. Adjust s^k, **For** each user k, **do**
11. **For** each application consumed or voted by user k, **do**
12. s^k adjustment based on (11);
13. **Output**: $R(i)$ (i = 1, …, I).

5.2 Application reputation generation

The RSP collects individual trust in various mobile applications. During the individual trust sharing, some mobile users may like to vote the applications directly based on our user study [45]. TruBeRepec considers the individual trust automatically generated by the mobile device and/or direct votes (i.e., the user's subjective opinion on the application). Based on the votes and individual trust values, we generate the application reputation at the RSP's Reputation/Recommendation Generator (see Fig. 3) by applying Algorithm 2. We apply weighted aggregation using the individual trust as the credibility of voting and also consider the influence of time and the number of reputation contributors. Note that Algorithm 2 can also be applied by the Trust Evaluator (see Fig. 3) to generate application reputation based on locally collected trust behavior information.

Obviously, user k, u_k ($u_k \in U$, $k = 1,\ldots,K$) could vote application i many times and at different time

$t_p : \{V_i^k\} = \{V_i^k(t_p)\}$. Considering the time influence and potential on–off attack, we pay more attention to the user's recent voting. $\overline{R^k(i)}$ is the aggregated reputation of application i based on user k's experiences.

$$\overline{R^k(i)} = \frac{1}{O}\sum_p V_i^k(t_p) * T_i^k(t_p) * e^{-\frac{|t-t_p|^2}{\tau}} \tag{9}$$

where $O = \sum_p T_i^k(t_p) * e^{-\frac{|t-t_p|^2}{\tau}}$, $V_i^k(t_p)$ is user k's vote on application i at time t_p, t is the reputation generation time, τ is the parameter to control the time decaying, ($\tau = 2$ in our simulations). $T_i^k(t_p)$ is the individual trust of user k reported at time t_p, with vote $V_i^k(t_p)$ attached. If $V_i^k(t_p)$ is not provided by the user, we set $V_i^k(t_p) = T_i^k(t_p)$ automatically (note that $V_i^k(t_p) \in [0, 1]$).

We consider the users' experiences on application i to generate its public reputation, denoted as $R(i)$ based on the following function by considering also recommendation trust s^k:

$$R(i) = \frac{\theta(K)}{W}\sum_{k=1}^{K} s^k * \overline{R^k(i)} \tag{10}$$

where K is the number of users who consume application i. Herein, we apply the Rayleigh cumulative distribution $\theta(K) = \left\{1 - \exp\left(\frac{-K^2}{2(\sigma+\varepsilon)^2}\right)\right\}$ to model the impact of K on application reputation. The percentage of usage, $\varepsilon = -K/K'$ is the factor to indicate the popularity of an application. K' is the total number of users in the TruBeRepec system. $W = \sum_{k=1}^{K} s^k$. We introduce user k's recommendation trust s^k in order to overcome potential attacks in TruBeRepec. At the user registration time, s^k is set to an initial value (e.g., 0.5 in our simulations) at the RSP. Then, it is further evolved based on user k's performance regarding application reputation generation. We have

$$\rho = \frac{1}{2}\{\max(V_i^k) - \min(V_i^k)\}$$

$$y = \rho - |R(i) - V_i^k(i)|$$

$$\text{If } y < \varphi \quad (\varphi = 0), \quad \gamma + +;$$

$$s^k = \begin{cases} s^k + \delta y & (\gamma < \text{thr}) \\ s^k + \delta y - \mu\gamma & (\gamma \geq \text{thr}) \end{cases} = \begin{cases} 1 & (s^k > 1) \\ 0 & (s^k < 0) \end{cases} \tag{11}$$

where $\delta > 0$ is a parameter to control the adjustment of s^k. In order to detect on–off attackers and conflict behavior attackers, we further introduce a warning flag γ to record the number of bad input into reputation generation. γ's initial value is 0. It is increased by 1 each time when a bad input happens. Parameter thr is a threshold to indicate the on–off attack or conflict behavior attack (thr = 3 in our simulation). Parameter $\mu > 0$ controls bad input punishment. $\max(V_i^k)$ is the maximum voting value, while $\min(V_i^k)$ the minimum voting value. φ is a parameter to decide the bad input. We set $\delta = 0.05$, $\mu = 0.1$, and $\varphi = 0$ in our simulations.

5.3 Application recommendation

Except for $T_i(t)$, the RSP also collects $T_i(t)_{\text{UB}}$, $T_i(t)_{\text{RB}}$, and $T_i(t)_{\text{CB}}$ in order to provide appropriate recommendations based on the correlation of trust behaviors. In addition, TruBeRepec also provides the public reputation of recommended applications for the reference of mobile users. Algorithm 3 is used to generate application recommendation vector that contains the recommendation value of application i for each TruBeRepec user at the RSP's Reputation/Recommendation Generator (see Fig. 3). Herein, we only consider good users as recommendation contributors. Note that Algorithm 3 can also be applied by the Trust Evaluator (see Fig. 3) to generate application recommendation vector based on locally collected trust behavior information.

Suppose a set of applications: $A = \{a_1, a_2, \ldots, a_i, \ldots, a_I\}$ considered in the system. For each application i, a user could have $T_i(t)_{\text{UB}}$, $T_i(t)_{\text{RB}}$, and $T_i(t)_{\text{CB}}$, where t is the recommendation time. We have K users $U = \{u_1, u_2, \ldots, u_K\}$ contribute to the application recommendation in TruBeRepec.

For kth user u_k, we have the following metric $D(u_k)$ to present his/her trust behaviors regarding applications $A = \{a_1, a_2, \ldots, a_I\}$ based on past experiences:

$$D(u_k) = \left\{ \begin{Bmatrix} T_1^k(t)_{\text{UB}} \\ T_1^k(t)_{\text{RB}} \\ T_1^k(t)_{\text{CB}} \end{Bmatrix} \cdots \begin{Bmatrix} T_I^k(t)_{\text{UB}} \\ T_I^k(t)_{\text{RB}} \\ T_I^k(t)_{\text{CB}} \end{Bmatrix} \right\} \tag{12}$$

The metric that expresses all users' trust behaviors is:

$$D(U) = \left\{ \begin{array}{c} \begin{Bmatrix} T_1^1(t)_{\text{UB}} \\ T_1^1(t)_{\text{RB}} \\ T_1^1(t)_{\text{CB}} \end{Bmatrix} \cdots \begin{Bmatrix} T_I^1(t)_{\text{UB}} \\ T_I^1(t)_{\text{RB}} \\ T_I^1(t)_{\text{CB}} \end{Bmatrix} \\ \cdots \\ \begin{Bmatrix} T_1^k(t)_{\text{UB}} \\ T_1^k(t)_{\text{RB}} \\ T_1^k(t)_{\text{CB}} \end{Bmatrix} \cdots \begin{Bmatrix} T_I^k(t)_{\text{UB}} \\ T_I^k(t)_{\text{RB}} \\ T_I^k(t)_{\text{CB}} \end{Bmatrix} \\ \cdots \\ \begin{Bmatrix} T_1^K(t)_{\text{UB}} \\ T_1^K(t)_{\text{RB}} \\ T_1^K(t)_{\text{CB}} \end{Bmatrix} \cdots \begin{Bmatrix} T_I^K(t)_{\text{UB}} \\ T_I^K(t)_{\text{RB}} \\ T_I^K(t)_{\text{CB}} \end{Bmatrix} \end{array} \right\} \tag{13}$$

Recommendation vector R_i^k at time t for u_k regarding application i can be calculated based on the following

Algorithm 3 Application recommendation vector generation

1. **Input**:
2. – K, $D(U)$;a_i; $U = \{u_1, u_2, \ldots, u_K\}$.
3. **For** each user u_k, **do**
4. Calculate R_i^k based on (14)–(17).
5. **Output**: R_i^k ($k = 1, \ldots, K$).

formula to provide personalized recommendations according to the correlation of trust behaviors:

$$R_i^k = \left\{ \begin{array}{l} R_i^k(t)_{\mathrm{UB}} \\ R_i^k(t)_{\mathrm{RB}} \\ R_i^k(t)_{\mathrm{CB}} \end{array} \right\} = \frac{\sum_{j \neq k} \left(\left\{ \begin{array}{l} T_i^j(t)_{\mathrm{UB}} \\ T_i^j(t)_{\mathrm{RB}} \\ T_i^j(t)_{\mathrm{CB}} \end{array} \right\} * \mathrm{Rel}(u_j, u_k) \right)}{\sum_{k \neq j} \mathrm{Rel}(u_j, u_k)}, \quad (i = 1, \ldots, I) \tag{14}$$

see "Appendix 1". Herein, we focus on the evaluation of the reputation and recommender algorithms. In TruBeRepec, malicious users could provide dishonest votes on applications in order to frame good ones and/or boost bad ones. This attack, referred to as the bad mouthing or unfair rating attack is the most straightforward attack [39, 40]. Malicious users could also behave well and badly alternatively, hoping that they can remain undetected while causing damage [39, 40]. This attack is called as the on–off attack. In particular, they can perform differently to different applications in order to impair good users' recommendation trust. This attack is referred to as the conflict behavior attack. TruBeRepec aims at overcoming the above attacks caused by user subjective voting. Herein, we assume that a malicious user is the user whose opinion on an application is obviously different from the public reputation of the application regarding his/her voting or his/her trust behavior is obviously different from others in

$$\mathrm{Rel}(u_j, u_k) = \frac{1}{I-1} \sum_{i' \neq i} \left(1 - \sqrt{\frac{\left(T_{i'}^k(t)_{\mathrm{UB}} - T_{i'}^j(t)_{\mathrm{UB}}\right)^2 + \left(T_{i'}^k(t)_{\mathrm{RB}} - T_{i'}^j(t)_{\mathrm{RB}}\right)^2 + \left(T_{i'}^k(t)_{\mathrm{CB}} - T_{i'}^j(t)_{\mathrm{CB}}\right)^2}{3}} \right) \tag{15}$$

Considering the influence of the number of recommenders, we set

$$N_K = \left\{ 1 - \exp\left(\frac{-K^2}{2\sigma^2} \right) \right\} \tag{16}$$

where $\sigma > 0$, is a parameter that inversely controls how fast the number of recommender's impact on R_i^k, it increases as K increases. The parameter σ can be set from 0 to theoretically ∞, to capture the characteristics of different scenarios.

We use N_K to adjust the recommendation vector by considering population K's influence. The final recommendation vector R_i^k is:

$$R_i^k = \frac{\sum_{j \neq k} \left(\left\{ \begin{array}{l} T_i^j(t)_{\mathrm{UB}} \\ T_i^j(t)_{\mathrm{RB}} \\ T_i^j(t)_{\mathrm{CB}} \end{array} \right\} * \mathrm{Rel}(\mathrm{u_j}, \mathrm{u_k}) \right)}{\sum_{k \neq j} \mathrm{Rel}(u_j, u_k)} * N_K \quad (i = 1, \ldots, I) \tag{17}$$

6 Analysis and evaluation

We have evaluated the trust behavior formalization (i.e., Algorithm 1) based on a number of usage models in [46], terms of application usage. This generally accords with the reality. In this section, we designed a number of experimental simulations to investigate the effectiveness and robustness of the reputation algorithm. Meanwhile, we also evaluate the recommender algorithm based on a number of usage examples.

In our simulations, we assume $K = 50$ users who commonly consume one application in the TruBeRepec system that has totally 100 registered users (i.e., $K' = 100$). There are a number of applications $A = \{a_1, a_2, \ldots, a_I\}$ that can be selected by the users to consume. In the experiment, honest vote means that the voting value matches the user's individual trust while dishonest vote means that the voting value mismatches the user's individual trust. Obviously, TruBeRepec can work without user voting, thus automatically avoid the above attacks. We apply Algorithm 2 to generate the reputation of an application and Algorithm 3 to generate the application recommendation vector. We simply use R to denote the application's reputation value, while $R = 0.1$ indicates low reputation and $R = 0.9$ indicates high reputation. Ri is the reputation of application a_i.

We adopt commonly used metrics in information retrieval, Recall (E), Precision (P), and F-measure (F) to describe the malicious user detection performance [38]. For the RSP, the number of users that belong to Malicious User

(MU) and are indeed detected as MU, denoted as x; the number of users that don't belong to MU but are detected as MU, denoted as y; the number of nodes that belong to MU but are not detected as MU, denoted as z. With these data, we do a precision-recall evaluation. Define

$$E = \frac{x}{x+z} \tag{18}$$

$$P = \frac{x}{x+y} \tag{19}$$

$$F = \frac{2PE}{P+E} \tag{20}$$

Using F-measure, we express TruBeRepec's robustness against various attacks. In the simulation, if user k's recommendation trust $s^k \leq 0.1$, TruBeRepc treats u_k as a malicious user.

In the experiment, we assume that the reputation of one application is not related to another. One user could behave honestly in using one application while badly in consumption of another. The F-measure is used to indicate the performance of TruBeRepec according to the recommendation trust values of users. The recommendation trust value is evaluated according to the user's contribution to applications' reputation generation, no matter which application the user contributes. Herein, we focus on voting on one application commonly used by 50 users and voting on two applications in case of conflict behavior attack investigation.

The simulation result is similar if 50 users vote different applications at the same time. TruBeRepec can detect the malicious users faster if it can accumulate more information. Due to paper size limitation, we only report the simulation results in the hardest detection cases. That is the users contribute to the reputation of one or two applications.

6.1 Effects of TruBeRepec reputation mechanism

We test the performance of Algorithm 2 with the following scenario: 50 users consume one application. Each user's initial recommendation trust is 0.5. They recommend the application honestly (e.g., no votes provided) at different time periods while their individual trust is (1) fixed (e.g., 0.1 or 0.9); (2) increasing from 0.1 to 0.9; and (3) decreasing from 0.9 to 0.1. We try to evaluate how effective the TruBeRepec is. Figure 4 shows this simulation result. We observe that TruBeRepec performs very well in

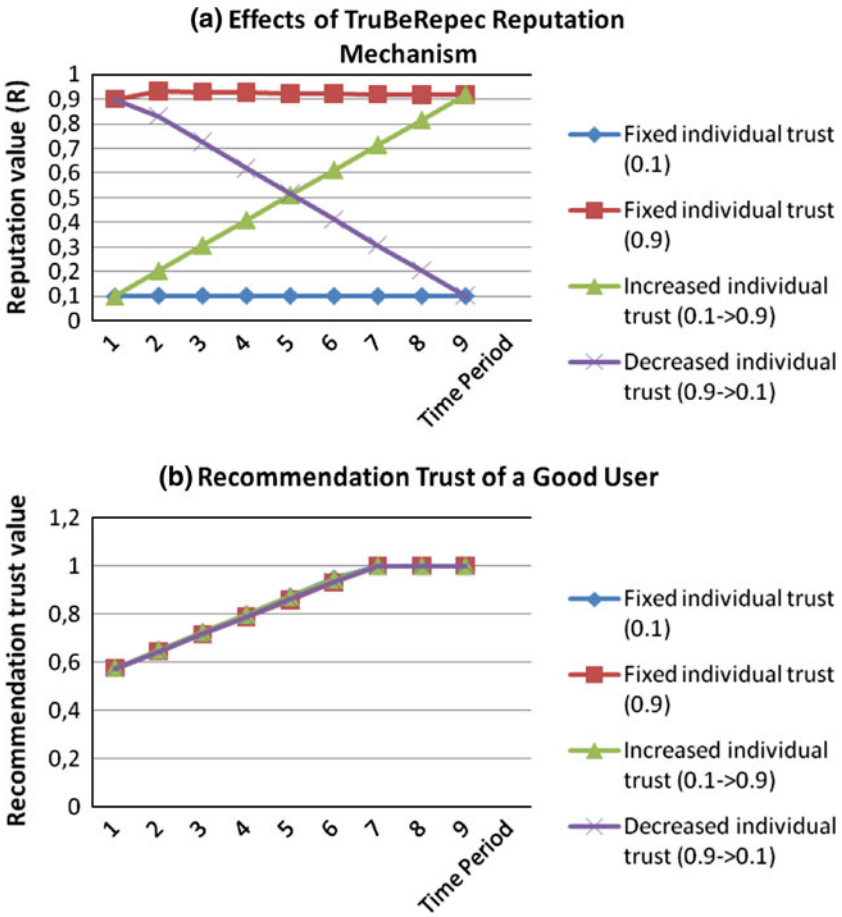

Fig. 4 Effects of TruBeRepec reputation mechanism **a** Application reputation in the case that all users are honest (1) all users have fixed individual trust (0.1); (2) all users have fixed individual trust (0.9); (3) all users' individual trust increases gradually from 0.1 to 0.9; (4) all users' individual trust decreases gradually from 0.9 to 0.1. **b** A good user's recommendation trust in above situations

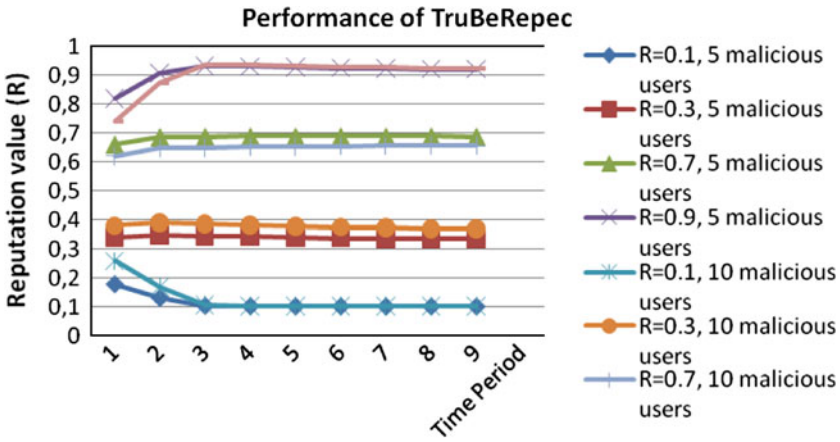

Fig. 5 Performance of TruBeRepec reputation mechanism with 10 and 20% malicious users

these situations (see Fig. 4a). Since the users are honest, thus their recommendation trust is gradually increased until reaching full trust (see Fig. 4b).

We further test TruBeRepec in the scenarios with malicious users who intentionally use the application differently from others. That is their individual trust is different from others even though their votes seem honest. We try to evaluate how robust the TruBeRepec is regarding this attack on the trust behavior model. Figure 5 shows this simulation result. We observe that TruBeRepec can evaluate the application's real reputation efficiently even though some users' usage behaviors are malicious.

6.2 Unfair rating attack

Unfair rating could influence the TruBeRepec system in the situation when users are allowed to vote the application. The influence of the unfair rating attack is demonstrated in Fig. 6. In the simulation, we assume that attackers consume the application in a normal way, but with unfair voting. We test four scenarios with 5, 10, 15, 20 unfair rating attackers, respectively, while other users vote the application (with $R = 0.1$ or $R = 0.9$) honestly. We observe that TruBeRepec can overcome the unfair rating attack in a very efficient way, mostly it can find the unfair rating attackers immediately if the percentage of attackers is below 30%. Even though the percentage of attackers is up to 40%, TruBeRepec can still find the attackers within 10 time periods if the attackers continuously vote unfairly.

6.3 On–off attack

The influence of the on–off attack due to malicious voting is demonstrated in Fig. 7. We test four scenarios: 5, 10, 15, and 20 attackers vote the application (with $R = 0.1$ or $R = 0.9$) with honest and dishonest recommendations alternatively, while other users vote the application honestly. We can see that TruBeRepec can efficiently overcome the on–off attack when the percentage of attackers is below 40%. Even though half of the users are on–off attackers, TruBeRepec can still detect them, but need more time periods.

6.4 Conflict behavior attack

The influence of the conflict behavior attack is demonstrated in Fig. 8. We test five scenarios: 5, 10, 15, 20, and 25 attackers vote one application a_1 (with $R1 = 0.9$) dishonestly while another application a_2 (with $R2 = 0.1$) honestly at the same time, while other users vote both applications honestly. We observe that TruBeRepec performs very well against this attack, even though the attackers occupy 50% of the users. It can detect the attackers mostly in the first time period, within 2nd time period if the attackers are 40% of the users, and within 3rd time period if the attackers reach 50% of the users.

6.5 Recommendation accuracy

We illustrate the accuracy of TruBeRepec recommendation mechanism with the following example: 10 users use three applications, with simulated $T_i(t)_{\mathrm{UB}}$, $T_i(t)_{\mathrm{RB}}$, and $T_i(t)_{\mathrm{CB}}$. For the 11th user who only consumes two applications a_0 and a_1 of the three, we calculate the recommendation vector for this user regarding the third application a_2. Our simulation result is shown in Table 2. The random number generated in the simulation is given in "Appendix 2", where Table 4 provides random $T_i^k(t)_{\mathrm{UB}}$ generated for the third test and Table 5 provides random $T_i^k(t)_{\mathrm{UB}}$, $T_i^k(t)_{\mathrm{RB}}$ and $T_i^k(t)_{\mathrm{CB}}$ generated for the forth test. We can see that TruBeRepec can provide personalized recommendations on the basis of the correlation of trust behavior, which is a concrete clue of interest similarity and preferences. For a simple example shown in Table 2, if all eleven users have

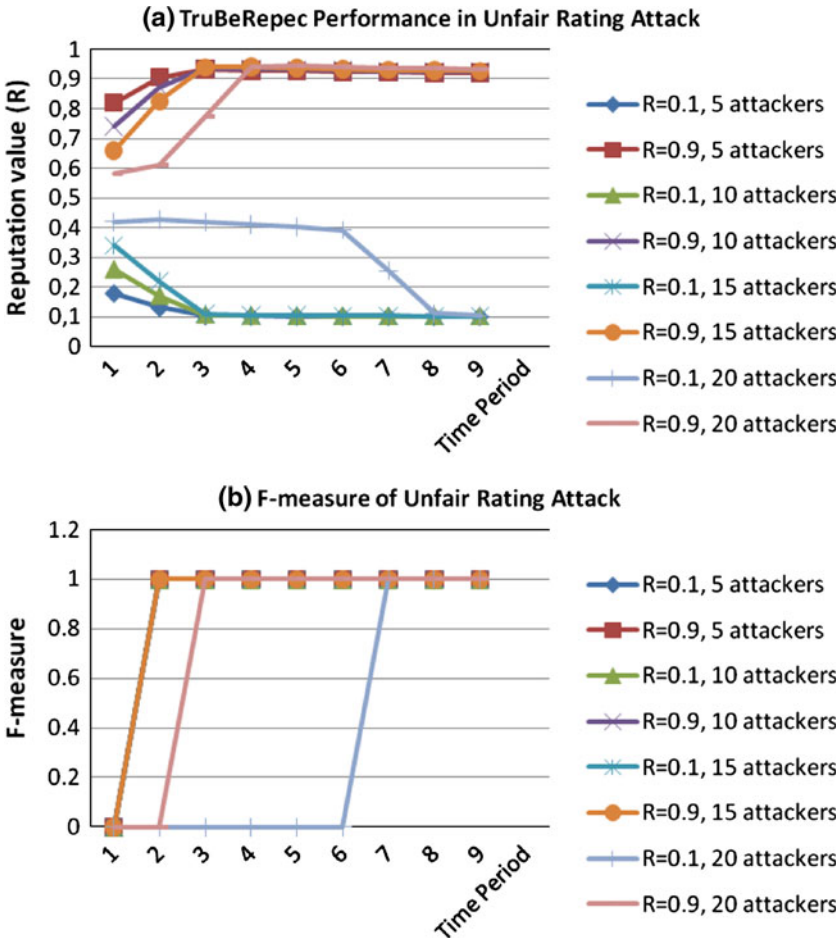

Fig. 6 **a** Performance of TruBeRepec reputation mechanism with 10, 20, 30, and 40% unfair rating attackers; **b** *F*-measure with 10, 20, 30, and 40% unfair rating attackers

the same trust behavior values [e.g., (0.6, 0.6, 0.6)] in terms of a_0 and a_1, and the first 10 users have the same trust behavior values [e.g., (0.6, 0.6, 0.6)] regarding a_2, i.e.,

$$\left\{\begin{array}{l} T_i^k(t)_{\mathrm{UB}} \\ T_i^k(t)_{\mathrm{RB}} \\ T_i^k(t)_{\mathrm{CB}} \end{array}\right\} = \left\{\begin{array}{c} 0.6 \\ 0.6 \\ 0.6 \end{array}\right\}$$
$$(i = 0, 1, 2;\ k = 0, \ldots, 10 (i = 0, 1);\ k = 0, \ldots, 9 (i = 2)),$$

the 11th user gets a recommendation vector (0.6, 0.6, 0.6) for a_2 based on Algorithm 3. This is obviously correct. For another example,

$$\left\{\begin{array}{l} T_i^k(t)_{\mathrm{UB}} \\ T_i^k(t)_{\mathrm{RB}} \\ T_i^k(t)_{\mathrm{CB}} \end{array}\right\} = \left\{\begin{array}{c} 0.7 \\ 1 - 0.1 * k \\ 0.1 * k \end{array}\right\}$$
$$(i = 0, 1, 2;\ k = 0, \ldots, 10 (i = 0, 1);\ k = 0, \ldots, 9 (i = 2)),$$

the 11th user's behavior correlation with other users is increasing with k's increase for RB and CB and reaches the highest for UB due to the same $T_i^k(t)_{\mathrm{UB}}$ values (0.7), thus the recommendation vector for the 11th user regarding a_2 holds the same value 0.7 for UB, and close to average value for RB and CB, i.e., (0.7, 0.43, 0.57).

7 Further discussions

7.1 Practical significance and limitations

Developing TruBeRepec based on the trust behavior model has practical significance. First, the model provides a valuable guideline on what kind of user data should be monitored and collected for the purpose of user trust evaluation. In practice, it is hard to directly evaluate user perceived trust, which actually reflects the technical trustworthiness of mobile applications. Second, applying the trust behavior model helps us ease the load of extra human–device interaction that may be required by some existing trust management solutions [25]. This is because the trust behaviors can be monitored through an auto-observation mechanism located at the mobile device. There is no need for extra usability study if TruBeRepec is employed. Through auto-monitoring users' trust behaviors via user–device interactions during application

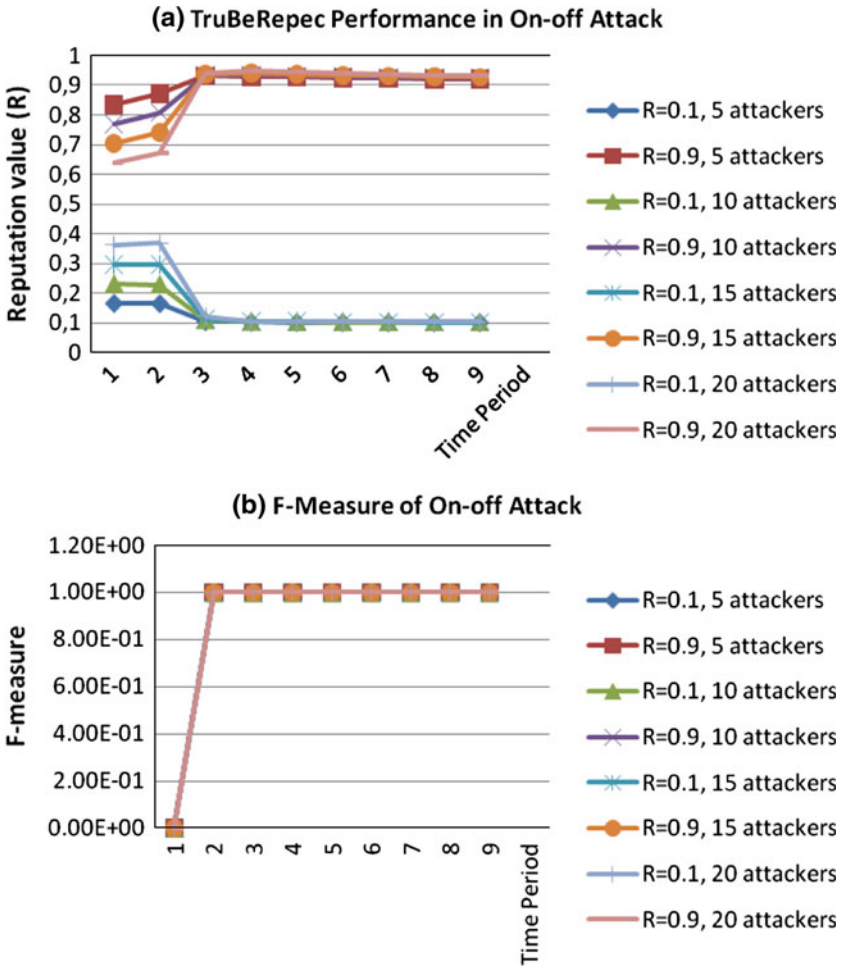

Fig. 7 **a** Performance of TruBeRepec reputation mechanism with 10, 20 30, and 40% on–off attackers; **b** *F*-measure with 10, 20, 30, and 40% on–off attackers

consumption, we can automatically extract useful information for trust evaluation, reputation generation, and recommendation provision. Thereby, TruBeRepec can provide sound usability. Third, the trust behavior model is examined through user study. The trust explanation mechanism based on this model could be easily understood and accepted by the users [26]. Meanwhile, a recommendation from a user or the RSP can be further assessed and explained with the trust behavior model in order to help other users selecting a trustworthy mobile application. Therefore, TruBeRepec supports usable trust management.

TruBeRepec design assumes that malicious or dishonest users occupy a small proportion (<50%) of the total number of mobile application users. Although this generally accords with the reality, TruBeRepec cannot afford large-scale collaborative attacks, e.g., the malicious users are more than half of total users and they attempt to collaboratively attack the reputation of an application at the same time. This is what TruBeRepec should further improve in the future.

7.2 User data privacy preservation

Based on the interviews with about 180 participants [45], we found that people pay special attention to user data privacy (e.g., usage statistics). In TruBeRepec, the RSP only collects trust values $T_i(t)$, $T_i(t)_{\mathrm{UB}}$, $T_i(t)_{\mathrm{RB}}$, and $T_i(t)_{\mathrm{CB}}$ in order to generate application reputations and provide recommendations. There is no need for the users to share their detailed application usage information and personal interests or preferences. In Privacy Enhancement Technology (PET), data minimization, i.e., minimizing personal data collected and used by service providers and merchants is one important technique to preserve privacy. Our method falls into this PET category, although it may not be a perfect one. Thus, TruBeRepec can preserve user data privacy to a certain level.

7.3 Attack on trust behavior

Trust is a subjective concept, trust or not trust is a user's personal opinion. Meanwhile, reputation published by the reputation service provider can be referred by a malicious

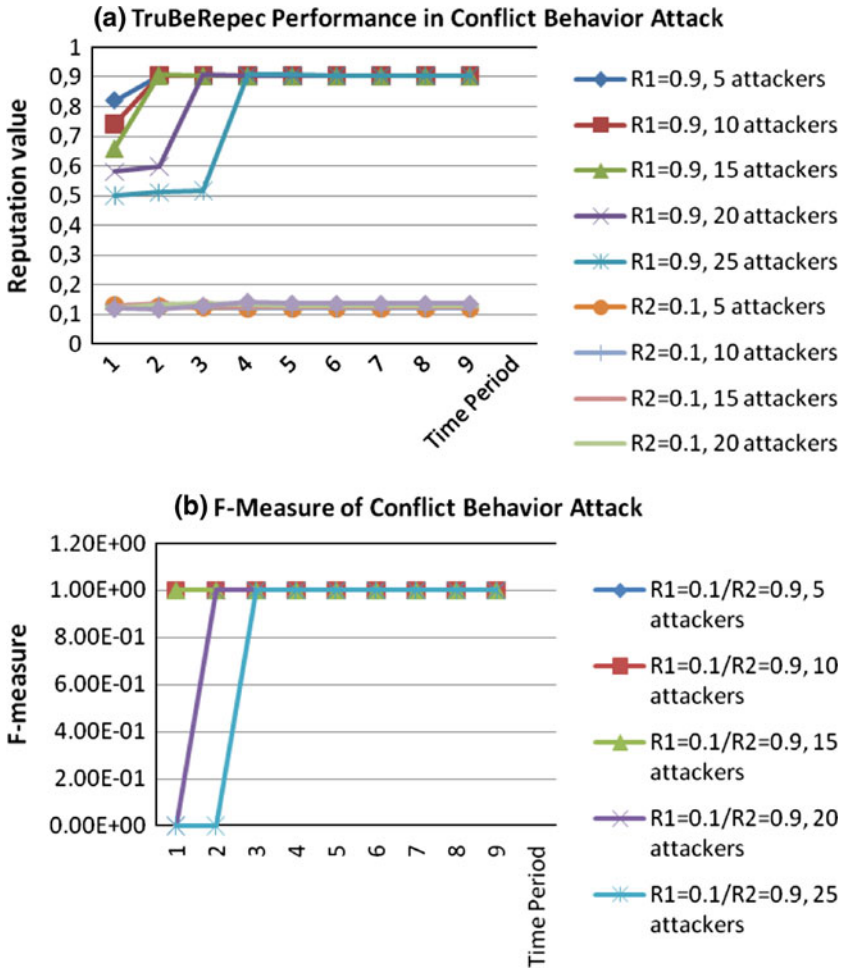

Fig. 8 **a** Performance of TruBeRepec reputation mechanism with 10, 20, 30, 40, and 50% conflict behavior attackers; **b** *F*-measure with 10, 20, 30, 40 and 50% conflict behavior attackers

user. If a user intensively uses a dislike or distrusted application, which is, for example, full of bugs and errors based on his own experiences, his/her behavior is malicious. If a number of users try to increase the reputation of an application with very low reputation through intensive usage, their behaviors are malicious.

Thereby, except for the attacks raised by voting mentioned in Sect. 6, malicious users could intend to attack the proposed trust evaluation mechanism in a way by frequently using a bad application for a long time, by continuing consume it even though the application has many problems and by always recommending it to other people, and meanwhile voting it positively. We argue that this attack could not influence much on the accuracy of application reputation and recommendation in the case that most users are normal users. It is also a big pain for malicious users to use a bad application in a trustworthy way. In addition, this kind of malicious users could be easily detected by the RSP by introducing the recommendation trust s^k in the reputation generation, as shown in Fig. 9 ($R = 0.1$). This is because a user's recommendation trust s^k is negatively influenced if the user's behaviors and/or votes are different from most of other users, i.e., having a big deviation from the reputation value. In this test, we set $V_i^k(t_p) = T_i^k(t_p)$ automatically since $V_i^k(t_p)$ is not provided by the user. We can see that TruBeRepec can also overcome the attack caused by malicious behaviors without user voting. Thus, it can support both voting and non-voting.

Herein, we hold an assumption that the mobile device computing platform applies for example trusted computing technology to reject malware installation or has a good detection mechanism to find and remove malwares [25, 43, 47, 48].

7.4 Performance impact

TruBeRepec can run as an independent mobile application for the user to check usage information and trust/reputation values of installed applications. In this case, it won't impact other application's performance. It can also run as a backend application to collect usage information and display both the reputation and detected individual trust values during the application usage. To avoid its impact on

Table 2 Recommendation vectors

$D(U)$	Recommendation vector for the 11th user R_2^{10} regarding a_2
1. $\begin{Bmatrix} T_i^k(t)_{\mathrm{UB}} \\ T_i^k(t)_{\mathrm{RB}} \\ T_i^k(t)_{\mathrm{CB}} \end{Bmatrix} = \begin{Bmatrix} 0.6 \\ 0.6 \\ 0.6 \end{Bmatrix} (i = 0, 1, 2;\ k = 0, \ldots, 10(i = 0, 1);\ k = 0, \ldots, 9(i = 2))$	$\begin{Bmatrix} 0.6 \\ 0.6 \\ 0.6 \end{Bmatrix}$
2. $\begin{Bmatrix} T_i^k(t)_{\mathrm{UB}} \\ T_i^k(t)_{\mathrm{RB}} \\ T_i^k(t)_{\mathrm{CB}} \end{Bmatrix} = \begin{Bmatrix} 0.7 \\ 1 - 0.1 * k \\ 0.1 * k \end{Bmatrix} (i = 0, 1, 2;\ k = 0, \ldots, 10(i = 0, 1);\ k = 0, \ldots, 9(i = 2))$	$\begin{Bmatrix} 0.7 \\ 0.43 \\ 0.57 \end{Bmatrix}$
3. $\begin{Bmatrix} T_i^k(t)_{\mathrm{UB}} \\ T_i^k(t)_{\mathrm{RB}} \\ T_i^k(t)_{\mathrm{CB}} \end{Bmatrix} = \begin{Bmatrix} \mathrm{random}() \\ 1 - 0.1 * k \\ 0.1 * k \end{Bmatrix} (i = 0, 1, 2;\ k = 0, \ldots, 10(i = 0, 1);\ k = 0, \ldots, 9(i = 2))$ (see Table 4)	$\begin{Bmatrix} 0.60 \\ 0.43 \\ 0.57 \end{Bmatrix}$
4. $\begin{Bmatrix} T_i^k(t)_{\mathrm{UB}} \\ T_i^k(t)_{\mathrm{RB}} \\ T_i^k(t)_{\mathrm{CB}} \end{Bmatrix} = \begin{Bmatrix} \mathrm{random}() \\ \mathrm{random}() \\ \mathrm{random}() \end{Bmatrix} (i = 0, 1, 2;\ k = 0, \ldots, 10(i = 0, 1);\ k = 0, \ldots, 9(i = 2))$ (see Table 5)	$\begin{Bmatrix} 0.47 \\ 0.57 \\ 0.59 \end{Bmatrix}$

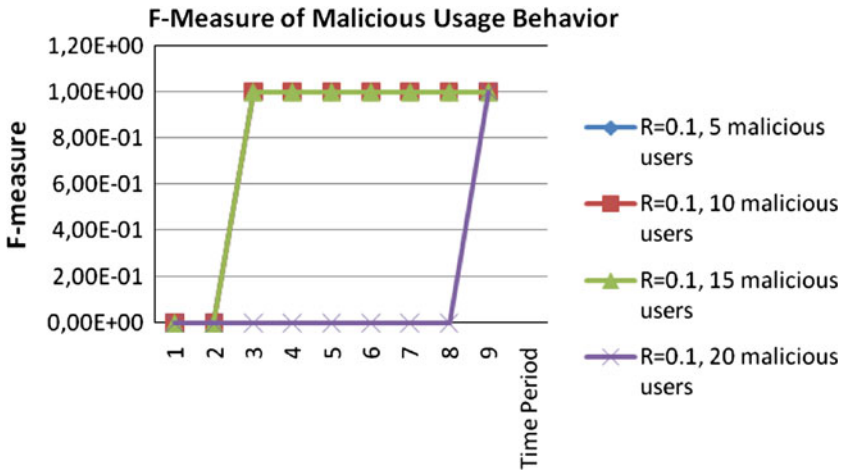

Fig. 9 Performance of TruBeRepec reputation mechanism regarding malicious usage behavior

other applications' performance, individual trust evaluation and reputation extraction from the RSP are conducted when the application is starting up or ending up.

In addition, we also explored the effects of trust information's visualization on mobile application usage in Finland and China. Although the user experiment results achieved in above two countries showed differences, both positively indicated that displaying an application's reputation value and/or an individual user's trust value could assist in the usage of mobile applications. Detailed results and discussions are reported in [44].

7.5 Synchronization of s^k

In TruBeRepec, we can synchronize local s^k (i.e., s_l^k) generated at a user's device and global s^k (i.e., s_g^k) generated at the RSP based on different policies in practice.

s_l^k is updated to s_g^k once it is issued by the RSP in the case that the RSP collects much more trust information than the local device. That is

$$s_l^k = s_g^k$$

s_l^k is further evolved each time when s_g^k is issued if the user would like to consider personally accumulated information. For example, $s_l^k = \omega_l * s_l^k + \omega_g * s_g^k$, where ω_l and ω_g are weighting factors to aggregate s_l^k and s_g^k. Suppose the number of trust-related information collected locally about user k is N_l^k and the number of trust information collected globally about user k is N_g^k, we have $\omega_l = \frac{N_l^k}{N_l^k + N_g^k}$ and $\omega_g = \frac{N_g^k}{N_l^k + N_g^k}$.

8 Conclusions and future work

This paper proposed TruBeRepec, a trust-behavior-based reputation and recommender system for mobile applications. Based on the trust behavior model explored through a large-scale user survey and validated using PCA, CFA, reliability analysis, and correlation analysis, we developed a number of algorithms to evaluate individual user's trust in a mobile application, generate application reputations and provide application recommendations based on trust behaviors. We showed the practical significance of TruBeRepec through simulations and analysis with regard to effectiveness, robustness, and usability, as well as privacy.

Regarding the future work, we will further improve the TruBeRepec system to eventually develop it toward a product quality implementation. Meanwhile, we will attempt to embed the system into a pervasive social networking platform [49] as part of its trust solution for mobile application services.

Acknowledgments The authors would like to thank Dr. Yan Dong, Prof. Rong Yan, Prof. Guoliang Yu, Dr. Valtteri Niemi and Dr. N. Asokan for user experiments, comments and their support to this work.

Appendix

A. Evaluation on individual trust calculation

From Nokia SmartPhone 360 usage statistics [50], we can figure out one usage model that is periodically changed, e.g., mobile email usage. We use function $|\sin(\omega t)|$, $(\omega = 1)$ to model it in our simulation with regard to usage frequency. The second usage model could be a logistic function, also known as Richards' curve, which is widely used for growth modeling. We use a modified logistic function$(1 - e^{-\gamma t})/(1 + e^{-\gamma t})$, $(\gamma = 1/2)$in our simulation in order to make the growth start from 0 at $t = 0$. The third usage model is a growth curve at the beginning and then reducing to a stale level (including 0, which can be controlled by the function parameters). Herein, we use a $\Gamma(\alpha, \beta)$ distribution $t^{\alpha-1}e^{-\beta t}(\alpha = 2;\ \beta = 0.5)$ to model it. We also propose a linear increase model $\eta t(\eta = 0.1,\ \eta t < 1)$ to roughly model, for example, recommendation percentage, elapsed usage time, and the number of usages. The above usage models can be applied in usage time, the number of usage, frequency, or context index. The user-experienced feature $\mathrm{EF}(i)/F(i)$ could be increased quickly and then gradually stay in a stable level. We use the logistic function to model it.

Figures 10a, b, and c show the simulation results of usage behavior formalization, reflection behavior formalization, and correlation behavior formalization, respectively. The usage models (or functions) applied in the simulations are listed in Table 3. For simplification, we apply function (2') and (4) in our simulation. Figure 10d shows the aggregated trust value $(T_i(t)_0 = 0.5)$ based on function (8) and the data of T(UB)_3; T(RB)_3; and T(CB)_1, T(CB)_2, and T(CB)_3 in Table 3, respectively.

From the simulation, we can see that the individual trust value calculated based on the proposed formalization reflects usage change no mater it is periodically up and

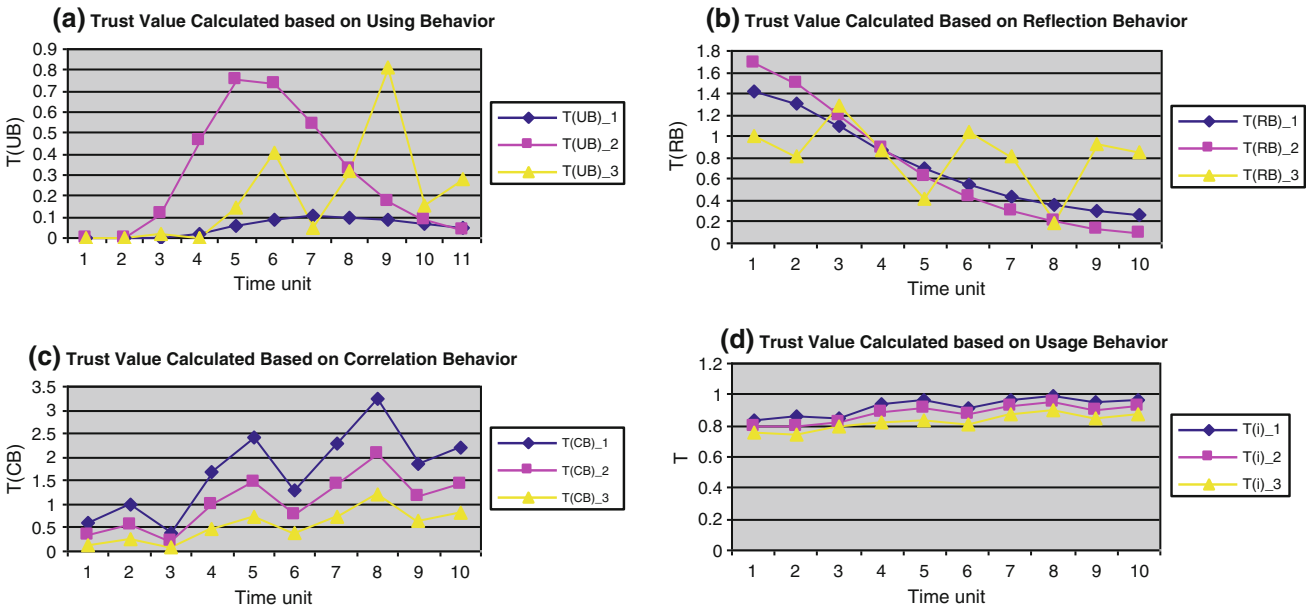

Fig. 10 Individual trust value calculated based on usage behavior

Table 3 Usage models applied in simulations ($\eta = 0.1$; $\gamma = 1/2$; $\alpha = 2$; $\beta = 0.5$)

UB	$\frac{N_i(t)}{N(t)}$	$\frac{\mathrm{UT}_i(t)}{\mathrm{UT}(t)}$	$\frac{\mathrm{EF}(i)}{F(i)}$	$\frac{\mathrm{FE}_i(t)}{\mathrm{FE}(t)}$	μ	
T(UB)_1; T(RB)_1	ηt	ηt	$(1-e^{-\gamma t})/(1+e^{-\gamma t})$	$t^{\alpha-1}e^{-\beta t}$	10	
T(UB)_2; T(RB)_2	$(1-e^{-\gamma t})/(1+e^{-\gamma t})$	$(1-e^{-\gamma t})/(1+e^{-\gamma t})$	$(1-e^{-\gamma t})/(1+e^{-\gamma t})$	$t^{\alpha-1}e^{-\beta t}$	10	
T(UB)_3; T(RB)_3	$(1-e^{-\gamma t})/(1+e^{-\gamma t})$	$(1-e^{-\gamma t})/(1+e^{-\gamma t})$	$(1-e^{-\gamma t})/(1+e^{-\gamma t})$	$\lvert\sin(t)\rvert$	1	
CB	$\frac{N_i(t)-N_k(t)}{N(t)}$	$\frac{\mathrm{UT}_i(t)-\mathrm{UT}_k(t)}{\mathrm{NT}(t)}$	$\frac{\mathrm{FE}_i(t)-\mathrm{FE}_k(t)}{\mathrm{FE}(t)}$	$\mathrm{CI}_i(t)-\mathrm{CI}_k(t)$	$\frac{R_i(t)-R_k(t)}{R(t)}$	
Applied functions	$(1-e^{-\gamma t})/(1+e^{-\gamma t})-\eta t$	$(1-e^{-\gamma t})/(1+e^{-\gamma t})-\eta t$	$\lvert\sin(t)\rvert - t^{\alpha-1}e^{-\beta t}$	Apply function (1.1)	ηt	$\lambda = 1$; $R_i(t)/R(t) = \eta t$
T(CB)_1 $T(i)$_1	$(ac = 0.9; T_i(t)_o = 0.5)$; $\rho = \sigma = \varsigma = 1$					
T(CB)_2 $T(i)$_2	$(ac = 0.5; T_i(t)_o = 0.5)$; $\rho = \sigma = \varsigma = 1$					
T(CB)_3 $T(i)$_3	$(ac = 0.2; T_i(t)_o = 0.5)$; $\rho = \sigma = \varsigma = 1$					

Table 4 Simulated random data 1

D [i: UserID][j: applicationID][0: UB]	D [i: UserID][j: applicationID][0: UB]
D[0][0][0] = 0.9064958856499105	D[5][1][0] = 0.23348992101415078
D[0][1][0] = 0.06961319859619164	D[5][2][0] = 0.6013253576376123
D[0][2][0] = 0.2183471174240742	D[6][0][0] = 0.8739642199277439
D[1][0][0] = 0.8513795090980439	D[6][1][0] = 0.890709927426842
D[1][1][0] = 0.3993033705716592	D[6][2][0] = 0.9248748344848168
D[1][2][0] = 0.6133334246777308	D[7][0][0] = 0.1866548693205774
D[2][0][0] = 0.2752691512656219	D[7][1][0] = 0.6062966873370097
D[2][1][0] = 0.5289455839507496	D[7][2][0] = 0.1872759257992369
D[2][2][0] = 0.9774920583183733	D[8][0][0] = 0.7279913387055578
D[3][0][0] = 0.1503207684552016	D[8][1][0] = 0.872639432334287
D[3][1][0] = 0.4616927399505547	D[8][2][0] = 0.5636699013168454
D[3][2][0] = 0.8237619523454625	D[9][0][0] = 0.2705792752234426
D[4][0][0] = 0.43638695873038813	D[9][1][0] = 0.9742742901953039
D[4][1][0] = 0.04546481458154461	D[9][2][0] = 0.42057674674649537
D[4][2][0] = 0.8914774859448656	D[10][0][0] = 0.36896848348032407
D[5][0][0] = 0.6595432741162669	D[10][1][0] = 0.2171094418169347

down or increased or decreased. It also implies the context's influence on trust. The trust value contributed by the correlation trust behavior indicates the impact of application similarity and usage difference on trust. To uniform the result, we apply a sigmoid function to map final trust value into (0, 1). We can also use this function to map different part of trust contribution into (0, 1) and then aggregate them together. In this case, the general metric becomes:

$$T_i(t) = f\{T_i(t)_0 + \rho f\{T_i(t)_{\mathrm{UB}}\} + \vartheta f\{T_i(t)_{\mathrm{RB}}\} + \varsigma f\{T_i(t)_{\mathrm{CB}}\}\} \quad (\rho + \vartheta + \varsigma = 1)$$

B. Random data generated for simulations in 6.5

The random data generated for simulations in Sect. 6.5 are provided in Tables 4 and 5.

Table 5 Simulated random data 2

D [i: UserID][j: applicationID] [k: trust behavior type]	D [i: UserID][j: applicationID] [k: trust behavior type]	D [i: UserID][j: applicationID] [k: trust behavior type]	D [i: UserID][j: applicationID] [k: trust behavior type]
D[0][0][0] = 0.216	D[2][2][0] = 0.895	D[5][1][0] = 0.955	D[8][0][0] = 0.169
D[0][0][1] = 0.747	D[2][2][1] = 0.952	D[5][1][1] = 0.081	D[8][0][1] = 0.769
D[0][0][2] = 0.926	D[2][2][2] = 0.506	D[5][1][2] = 0.701	D[8][0][2] = 0.081
D[0][1][0] = 0.636	D[3][0][0] = 0.699	D[5][2][0] = 0.082	D[8][1][0] = 0.821
D[0][1][1] = 0.850	D[3][0][1] = 0.203	D[5][2][1] = 0.768	D[8][1][1] = 0.535
D[0][1][2] = 0.746	D[3][0][2] = 0.307	D[5][2][2] = 0.036	D[8][1][2] = 0.188
D[0][2][0] = 0.780	D[3][1][0] = 0.409	D[6][0][0] = 0.717	D[8][2][0] = 0.360
D[0][2][1] = 0.479	D[3][1][1] = 0.765	D[6][0][1] = 0.956	D[8][2][1] = 0.027
D[0][2][2] = 0.940	D[3][1][2] = 0.871	D[6][0][2] = 0.563	D[8][2][2] = 0.707
D[1][0][0] = 0.804	D[3][2][0] = 0.716	D[6][1][0] = 0.174	D[9][0][0] = 0.318
D[1][0][1] = 0.352	D[3][2][1] = 0.567	D[6][1][1] = 0.711	D[9][0][1] = 0.717
D[1][0][2] = 0.340	D[3][2][2] = 0.638	D[6][1][2] = 0.078	D[9][0][2] = 0.349
D[1][1][0] = 0.191	D[4][0][0] = 0.568	D[6][2][0] = 0.097	D[9][1][0] = 0.402
D[1][1][1] = 0.892	D[4][0][1] = 0.972	D[6][2][1] = 0.320	D[9][1][1] = 0.407
D[1][1][2] = 0.776	D[4][0][2] = 0.615	D[6][2][2] = 0.615	D[9][1][2] = 0.500
D[1][2][0] = 0.095	D[4][1][0] = 0.329	D[7][0][0] = 0.600	D[9][2][0] = 0.926
D[1][2][1] = 0.636	D[4][1][1] = 0.050	D[7][0][1] = 0.483	D[9][2][1] = 0.967
D[1][2][2] = 0.283	D[4][1][2] = 0.792	D[7][0][2] = 0.670	D[9][2][2] = 0.746
D[2][0][0] = 0.166	D[4][2][0] = 0.506	D[7][1][0] = 0.546	D[10][0][0] = 0.636
D[2][0][1] = 0.666	D[4][2][1] = 0.196	D[7][1][1] = 0.471	D[10][0][1] = 0.426
D[2][0][2] = 0.499	D[4][2][2] = 0.856	D[7][1][2] = 0.748	D[10][0][2] = 0.007
D[2][1][0] = 0.624	D[5][0][0] = 0.685	D[7][2][0] = 0.241	D[10][1][0] = 0.632
D[2][1][1] = 0.772	D[5][0][1] = 0.412	D[7][2][1] = 0.830	D[10][1][1] = 0.733
D[2][1][2] = 0.730	D[5][0][2] = 0.314	D[7][2][2] = 0.535	D[10][1][2] = 0.476

References

1. Yan Z (2007) Trust management for mobile computing platforms. Dissertation, Helsinki University of Technology
2. Avizienis A, Laprie JC, Randell B, Landwehr C (2004) Basic concepts and taxonomy of dependable and secure computing. IEEE Trans Dependable Secur Comput 1(1):11–33
3. Yan Z, Dong Y, Niemi V, Yu G (2009) Exploring trust of mobile applications based on user behaviors. InTrust 2009, LNCS, pp 212–226
4. McKnight DH, Choudhury V, Kacmar C (2002) Developing and validating trust measures for e-commerce: an integrative typology. Inf Syst Res 13(3):334–359
5. Marsh S (1994) Formalising trust as a computational concept. Dissertation, University of Stirling
6. Yan Z, Holtmanns S (2008) Trust modeling and management: from social trust to digital trust. In: Subramanian R (ed) Computer security, privacy and politics: current issues, challenges and solutions. Idea Group Inc, USA, pp 290–323
7. Yan Z, Prehofer C (2010) Autonomic trust management for a component based software system. IEEE Trans Dependable Secur Comput.doi:10.1109/TDSC.2010.47
8. Xiong L, Liu L (2004) PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities. IEEE Tran Knowl Data Eng 16(7):843–857
9. Song S, Hwang K, Zhou R, Kwok YK (2005) Trusted P2P transactions with fuzzy reputation aggregation. IEEE Intern Comput 9(6):24–34
10. Theodorakopoulos G, Baras JS (2006) On trust models and trust evaluation metrics for ad hoc networks. IEEE J Sel Areas Commun 24(2):318–328
11. Sun Y, Yu W, Han Z, Liu KJR (2006) Information theoretic tramework of trust modeling and evaluation for ad hoc networks. IEEE J Sel Area Commun 24(2):305–317
12. Li X, Valacich JS, Hess TJ (2004) Predicting user trust in information systems: a comparison of competing trust models. In: Proceedings of 37th annual Hawaii international conference on system sciences, 10 pp
13. Bigley GA, Pearce JL (1998) Straining for shared meaning in organization science: problems of trust and distrust. Acad Manag Rev 23(3):405–421
14. Fishbein M, Ajzen I (1975) Beliefs, attitude, intention and behavior: an introduction to theory and research. Addison-Wesley, Reading
15. Anderson JC, Narus JA (1990) A model of distributor firm and manufacturer firm working partnerships. Marketing 54(1):42–58
16. Fox A (1974) Beyond contract: work, power, and trust relations. Faber, London
17. Deutsch M (1973) The resolution of conflict: constructive and destructive processes. Yale University Press, New Haven
18. Sheppard BH, Hartwick J, Warshaw PR (1988) The theory of reasoned action: a meta analysis of past research with recommendations for modifications in future research. Consumer Res 15(3):325–343
19. Venkatesh V, Davis FD (2000) A theoretical extension of the technology acceptance model: four longitudinal field studies. Manag Sci 46(2):186–204

20. Grabner-Kräuter S, Kaluscha EA (2003) Empirical research in on-line trust: a review and critical assessment. Int J Hum Comput Stud 58(6):783–812
21. Muir BM (1994) Trust in automation part I: theoretical issues in the study of trust and human intervention in automated systems. Ergonomics 37(11):1905–1922
22. Muir BM (1996) Trust in automation part II: experimental studies of trust and human intervention in a process control simulation. Ergonomics 39(3):429–469
23. Lee J, Moray N (1992) Trust, control strategies and allocation of function in human-machine systems. Ergonomics 35(10):1243–1270
24. Grandison T, Sloman M (2000) A survey of trust in internet applications. IEEE Commun Surv 3(4):2–16
25. Yan Z (2010) Trust modeling and management in digital environments: from social concept to system development. In: IGI Global, pp 20–57
26. Yan Z, Niemi V (2009) A methodology towards usable trust management. In: ATC09, LNCS, vol 5586, pp 179–193
27. Yan Z, Chen Y (2010) AdContRep: a privacy enhanced reputation system for MANET content services. UIC 2010, LNCS, vol 6406, pp 414–429
28. Aberer K, Despotovic Z (2001) Managing trust in a peer-to-peer information system. In: Proceedings of the ACM conference on information and knowledge management (CIKM), pp 310–317
29. Resnick P, Varian HR (1997) Recommender systems. Commun ACM 40(3):56–58
30. Hancock JT, Toma C, Ellison N (2007) The truth about lying in online dating profiles. In: Proceedings of the ACM conference on human factors in computing systems (CHI 2007), ACM, pp 449–452
31. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intell. doi:10.1155/2009/421425
32. O'Donovan J, Smyth B (2005) Trust in recommender systems, IUI'05, pp 167–174
33. Jøsang A, Ismail R, Boyd C (2007) A survey of trust and reputation systems for online service provision. Decis Support Syst 43(2):618–644
34. Resnick P, Zeckhauser R (2002) Trust among strangers in Internet transactions: empirical analysis of eBay's reputation system. In: Baye M (ed) Advances in applied microeconomics: the economics of the internet and e-commerce, vol 11. Elsevier, Amsterdam, pp 127–157
35. Resnick P, Kuwabara K, Zeckhauser R, Friedman E (2000) Reputation systems. Commun ACM 43(12):45–48
36. Corritore CL, Kracher B, Wiedenbeck S (2003) On-line trust: concepts, evolving themes, a model. Int J Hum Comput Stud Trust Technol 58(6):737–758
37. Yang Y, Sun Y, Kay S, Yang Q (2009) Defending online reputation systems against collaborative unfair raters through signal modeling and trust. In: SAC'09, pp 1308–1315
38. Douceur JR (2002) The sybil attack. In: IPTPS'02, LNCS, vol 2429, pp 251–260
39. Sun Y, Han Z, Liu KJR (2008) Defense of trust management vulnerabilities in distributed networks. IEEE Commun Mag 46(2):112–119
40. Sun Y, Han Z, Yu W, Liu KJR (2006) A trust evaluation framework in distributed networks: vulnerability analysis and defense against attacks. In: IEEE INFOCOM, pp 1–13
41. Fogg BJ, Tseng H (1999) The elements of computer credibility. In: Proceedings of the CHI'99, ACM Press, New York, pp 80–87
42. Crocker L, Algina J (1986) Introduction to classical and modern test theory. Thomson Leaning, Belmont
43. TCG TPM Specification v1.2, http://www.trustedcomputinggroup.org/resources/tpm_main_specification. Accessed 8 Sep 2010
44. Yan Z, Liu C, Niemi V, Yu G (2010) Effects of displaying trust information on mobile application usage. In: ATC'10, LNCS, vol 6407, pp 107–121
45. Yan Z, Liu C, Niemi V, Yu G (2010) Trust information indication: effects of displaying trust information on mobile application usage. Technical Report NRC-TR-2009-004, Nokia Research Center. http://www.research.nokia.com/files/NRCTR2009004.pdf. Accessed 8 Sep 2010
46. Yan Z, Yan R (2009) Formalizing trust based on usage behaviours for mobile applications. ATC09, LNCS 5586:194–208
47. Schiffman J, Moyer T, Jaeger T, McDaniel P (2011) Network-based root of trust for installation. IEEE Secur Priv 9(1):40–48
48. Wu J, Fang M, Yu P, Zhang X (2009) A secure software download framework based on mobile trusted computing. WCSE '09, pp 171–176
49. Ahtiainen A, Kalliojarvi K, Kasslin M, Leppanen K, Richter A, Ruuska P, Wijting C (2009) Awareness networking in wireless environments: means of exchanging information. IEEE Veh Technol Mag 4(3):48–54
50. Nokia SmartPhone 360 panel survey results: http://www.nwiki.nokia.com/Smartphone360/WebHome. Accessed 31 Jan 2009