Neural Comput & Applic (2017) 28 (Suppl 1):S461–S474
DOI 10.1007/s00521-016-2354-0

ORIGINAL ARTICLE

An efficient trust estimation model for multi-agent systems using temporal difference learning

G. Rishwaraj[1] · S. G. Ponnambalam[1] · Loo Chu Kiong[2]

Received: 3 November 2015 / Accepted: 17 May 2016 / Published online: 28 May 2016

Abstract In multi-agent system (MAS) applications, teamwork among the agents is essential as the agents are required to collaborate and pool resources to execute the given tasks and complete the objectives successfully. A vital part of the collaboration is sharing of information and resources in order to optimize their efforts in achieving the given objectives. Under such collaborative environment, trust among the agents plays a critical role to ensure efficient cooperation. This study looks into developing a trust evaluation model that can empirically evaluate the trust of one agent on the other. The proposed model is developed using temporal difference learning method, incorporating experience gained through interactions into trust evaluation. Simulation experiments are conducted to evaluate the performance of the developed model against some of the most recent models reported in the literature. The results of the simulation experiments indicate that the proposed model performs better than the comparison models in estimating trust more effectively.

Keywords Multi-agent system · Trust estimation · Temporal difference learning

✉ S. G. Ponnambalam
sgponnambalam@monash.edu

[1] Advanced Engineering Platform and School of Engineering, Monash University Malaysia, Jalan Lagoon Selatan, Bandar Sunway, 47500 Subang Jaya, Selangor, Malaysia

[2] Universiti Malaya, Jalan Universiti, 50603 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur, Malaysia

1 Introduction

Multi-agent system (MAS) is defined as a collective or a group of autonomously cooperating individuals or agents working in a common environment. Individually, the agents may have limited capabilities which are insufficient to achieve given objectives. To overcome this limitation, the agents could collaborate and work together forming a multi-agents system to share resources and accomplish their objectives [1–4]. MAS is widely applied as an effective system for various applications such as robotics, decision support systems, data mining, e-health, recommender systems, military and more [3, 5–9].

Cooperation among agents in the MAS is akin to cooperation among humans. Sometimes, the agents are required to cooperate with unknown agents for the first time in order to complete the mission objectives. The agents could also experience internal failures such as faulty sensors, motion error and imperfect state information which could hinder any successful mission undertakings. The agents may also be working in a highly dynamic and uncertain environment [10, 11]. Agents that are sharing information and resources under these conditions need to be able to trust each other in order to successfully complete the given mission objectives [12, 13].

This paper presents the development of a trust evaluation model for MAS to estimate the trust among agents to improve collaboration outcomes. The proposed trust estimation model incorporates reinforcement learning technique to estimate the trust on agents in MAS. The contributions of this paper are the development of a trust evaluation model framework using reward and observation-based learning that could empirically evaluate the trust among agents in MAS. This paper also highlights the simulation experiments conducted to examine the

performance of developed model against models reported in the literature.

1.1 Related work

In the current literature, there are several approaches studied to evaluate trust in MAS. Among the methods found in the literature that incorporates trust in MAS include direct trust evaluation model, indirect trust evaluation models, reputation-based trust evaluation models, organizational trust models, socio-cognitive trust models and information source [5, 14]. Two of the highly studied trust evaluation methods are direct/individual learning and indirect/social learning [15–18]. Direct learning is characterized as agents self-evaluating the trustworthiness of other agents through individual interaction while social learning is where the agents use the information available from its network to estimate trust of other agents.

One of the earlier works in trust evaluation modeling is presented by Jøsang [19], who proposed subjective logic as formulation trust using probability function of belief models represented by belief, disbelief and uncertainty. Due to the wide application of the belief models, subjective logic is extensively used by researchers in the study of trust model formulation. Zhou et al. [20] used the subjective logic approach to design a trust evaluation model in a deep learning framework that correlates interaction stereotypes and trustworthiness. Fan and Perros [21] presented a study on trust management in multi-cloud computing where a distributed Trust Service Providers (TSPs) calculated the trust of service providers using subjective logic from the collected evidence. Jin-Hee et al. [22] used subjective logic to formulate a trust evaluation model which is used as a filter in fusing information from multiple agents for decision making.

In recent studies, the concept of certainty and evidence are incorporated in trust models to ensure a more comprehensive trust computation. One of such works is by Basheer et al. [1], where the trust is modeled into a larger framework called confidence model using certainty and evidence as an evaluation agents. Another trust model that integrates certainty and evidence in trust evaluation is developed using a probabilistic approach to estimate trust values [23]. Wang and Singh [24] studied the mathematical bijection between trust and evidence and developed an evidence-based trust evaluation algorithm. A further improved version of this model is presented by Wang et al. [25], where additional parameters namely certainty and referral based third party trust evaluation are used as parameters.

Reputation, akin to social trust, is another framework of trust where the trustworthiness of an agent is evaluated through the information provided by other agents. Huynh et al. [17] proposed a comprehensive trust evaluation model that empirically integrates interaction trust, role-based trust, witness reputation and certified reputation. Sabater and Sierra [14] studied trust based on social learning and proposed a reward–pay off method to regulate the dependence on social network for trust information.

Mantel and Clark [26] presented a framework where agents are proficient in approximating the trustworthiness of other agents through a reputation management system consisting of a distributed trust evaluation model. In a study done by Mohammed et al. [3], a bucket brigade algorithm that uses hyberbox accuracy formulation is used in the trust evaluation system. Jøsang and Haller [27] employed mathematical formulation to estimate trust using a multinomial probability distribution function in describing a reputation-based trust system. Rosaci et al. [13] proposed a trust formulation that dynamically computes the weightage between reputation and reliability for trust evaluation. A trust evaluation method using basic cryptographic techniques which involves weight graph connecting to connect robots in the system is studied by Namin et al. [28].

Peng et al. [29] modeled a reputation and activity-based trust model where social relationship inspired concepts namely biological, social and business relationships is empirically used as weightage to model trust. Another sociological concept in trust modeling can be found in the works of Mui et al. [30] where the authors applied the concept of reciprocity to probabilistically model trust and reputation. Griffiths [31] proposed a multi-dimensional experience-based trust model where trustworthiness of an agent is estimated using criteria such as success rate, cost, timeliness and quality.

In the recent literature, reinforcement learning (RL)-based trust modeling is emerging as a new direction of study in trust evaluation. Fullam and Barber [32] investigated frequency of transmission, trustworthiness and accuracy of reputation as parameters for trust modeling and used reinforcement learning method to determine the best combination of the parameters to model trust. Yu et al. [18] proposed a trust aggregation model using actor-critic learning. The model uses both direct learning while it dynamically adjusts its indirect trust evaluation based on credible witness selection. Aref and Tran [7] proposed a trust evaluation method that combines Q-learning and fuzzy logic. Q-learning is used to calculate a trust value using direct trust evaluation, and then, this value together with two other parameters is used to compute the final trust value using fuzzy logic.

One of the commonly observed drawbacks in the current models is the lack of temporal analysis on the trust evidence obtained from the complex behavior of agents especially in a real-world scenario [5]. Table 1 highlights a

Table 1 A brief summary of some of the recently reported trust estimation models in the literature

References	Model and critical feature	Limitation
Fullam and Barber [32]	Uses a weight parameter to select between reputation model and experience model RL is used as the basis for experience-based trust modeling. Uses the best of three parameters to model trust	The weight parameter only selects either experience-based trust modeling or reputation-based modeling. The model does not consider the combination of the two into a single Trust model
Yu et al. [18]	Uses actor-critic method to build the trust evaluation system The learning rate is dynamically computed to weight the direct trust and indirect trust values	When only one known witness is available for the indirect trust evaluation, bootstrapping error occurs as the credibility is based only on unweighted witness testimony
Aref and Tran [7]	Model uses Q-learning and fuzzy logic Q-learning is used to compute direct trust estimation which is then used with to compute the defuzzied output The defuzzied output used in fuzzy logic to compute final trust estimate	Q-learning directly bootstraps previous experience to current outcome using simple minus function which maybe be insufficient for direct trust estimation
Das and Islam [8]	Mathematical framework considers multi-dimensional factor of MAS for trust estimate Some of the factors incorporated in trust evaluation are direct trust, indirect trust, historical trust, expected trust, credibility, decaying trust, deviation reliability	The highly multi-dimensional factors in the model make it computationally complex and taxing to the evaluation system
Basheer et al. [1]	Defined confidence for decision making based on evidence and trust Model is based on degree of certainty regarding opinion of other agents, agent's trust and evidence for certainty and uncertainty	Evaluating agent does not observe the environment directly, but instead gets the observation results from another agent The error in this communication is not properly highlighted in the model
Peng et al. [29]	Trust model considers different aspect of social relationship among the agent The model considers biological/kin, social ties and business activity	Lack of consideration and weightage on direct trust estimation in the trust evaluation
Mantel and Clark [26]	Distributed trust estimation model that incorporates reputation management system Incorporates indirect trust estimates into direct trust evaluation using constant proportional gain as weightage	The direct trust estimation only considers averaging the difference between observed outcome and outcomes observed by other. Direct trust estimate does not properly considered the duration of interaction in trust estimate
Zhou et al. [20]	Deep learning framework to learn correlations between context-aware stereotypes and trustworthiness to estimate updated trust Proposed seven context-aware stereotypes to use when direct historical evidences are not available	Although the proposed context-aware stereotypes are robust, it still does not cover agents that fall outside those seven categories

brief summary on some of the recently reported trust estimation models in the literature and the limitation of these models.

This paper presents a direct trust evaluation model framework that incorporates the concept of learning via interactions with agents using temporal difference learning. The performance of the proposed model is evaluated against some of the recently reported trust models to evaluate its performance, and the potential application of the proposed model in real-world scenarios is highlighted.

The paper is organized as follow. Section 2 defines the problem studied in this work. Section 3 presents the model framework and development. Section 4 details the simulation experiments conducted and the results. Finally, Sect. 5 highlights the potential application of the proposed model in real-world scenarios and Sect. 6 summarizes the conclusion of the study and the future work.

2 Problem definition

To understand the problem investigated in this work, consider a simple goal collecting example involving multiple agents. Three agents (A_1, A_2 and A_3) are required to explore an environment where each agent is required to collect a certain number of specific colored balls respective to each agent, i.e., A_1 is required to collect ten blue balls, A_2 is required to collect ten red balls, and A_3 is required to collect ten yellow balls. Although the agents know the size of the environment, they do not know the location of the

balls. The agents are tasked to randomly explore the environment, looking for their respective balls. If an agent find a ball of another, e.g., A_1 found a red ball (belong to A_2), that agent shares the information with the other agent, e.g., A_1 shares the location information with A_2.

In this example, agent A_1 may end up sharing the wrong information with A_2 due to various reasons such as:

1. Faulty or damaged sensors.
2. High degree of uncertainty in the environment.
3. Highly dynamic environment that defers from initial observation.

The illustrative example of this situation is shown in Fig. 1 below.

When agent A_1 shares this information with A_2, agent A_2 would use its time and resources to arrive at target location. If A_2 arrives at the target location and did not find any ball there (due to misinformation or false data communication), the entire endeavor would be a waste of precious time and resources to achieve nothing. In order to avoid this outcome, the agents need to have a certain measure of trust in each other to help in the decision-making process. The interest of this research is to address these issues of trust:

1. Determine the optimal interactions required to determine trustworthiness of an agent.
2. Develop an empirical model to evaluate the trust of an agent.
3. Iteratively learn and update the trust placed on an agent through direct interaction and subsequently gained experience.
4. Determine non-value-added agents or untrustworthy agents in a MAS.

3 Model development

3.1 Model structure

In this study, direct trust evaluation approach is considered in formulating the trust evaluation model. Trust evaluation is considered as a decision-making process where trust of an agent is evaluated through experience gained via interactions with other agents. In a decision-making process, each action takes an agent from one state to the next (state or location or position), resulting in a utility (or reward) gain from taking that action. The objective of the decision-making process is to maximize the expected utility an agent would receive when moving to the next state after taking a specific action. The expected utility is the summation of discounted rewards obtained through the policy or action execution, which is shown in Eq. (1). In Eq. (1), '$U^{\pi}(s)$' is the expected utility when executing action or policy 'π,' 'γ' is the discount factor that weights the rewards received, '$R(s_t)$' is the reward received in state 's' at time 't.'

$$\textit{Expected Utility}, U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi, s_0 = s\right] \quad (1)$$

In a decision-making process, the agent is evaluating the consequences of taking an action when in a particular state. The important outcome is to compute the value (utility) of a state 's' when taking executing an action or policy. The expected utility of a state is represented by '$V(s)$.' However, the execution of the policy or action is often probabilistic in nature, i.e., there is a probability that the action taken leads the agent to a different state than predicted. Therefore, this probability consideration has to be included

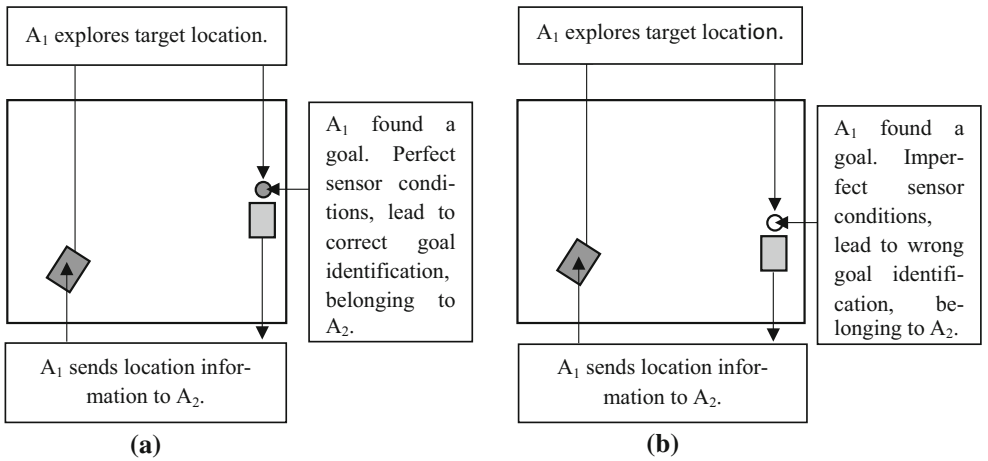


Fig. 1 An illustrative example of the problem definition. **a** Shows an example of agent A_1 sending the correct information to agent A_2. **b** Illustrates an example when agent A_1 sends the wrong information to agent A_2

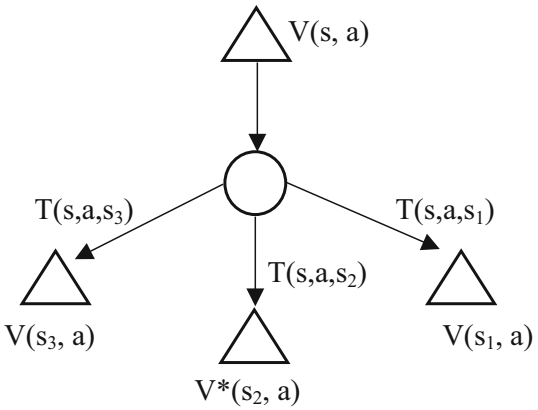


Fig. 2 Illustration of the decision-making process

in the calculation of the expected utility of the state, taking into consideration all the possible states that the agent might land in. The example of this is shown in Fig. 2. When an agent is in state 's,' the action that it takes can lead the agent to three possible states denoted by s_1, s_2 and s_3. The transition probability of reaching the states of s_1, s_2 and s_3 is $T(s, a, s_1)$, $T(s, a, s_2)$ and $T(s, a, s_3)$, respectively.

Under the probabilistic condition, the decision-making process considers the optimal action which results in highest return as shown in Eq. (2).

$$V^*(s) = \max_a Q^*(s', a) \tag{2}$$

In Eq. (2), '$V^*(s)$' is the expected value of a state when executing the optimal action and the '$Q^*(s', a)$' is the possible resulting states utility values that the agent receive. Examining Eqs. (1) and (2), the expected utility of a state is defined as the immediate reward received when taking the action plus the expected discounted utility of the next state landed if an optimal action is chosen as shown in Eq. (3). The reward of taking that action is given by '$R(s, a)$.'

$$V^*(s) = \max_a \sum_{s'} T(s, a, s')[R(s, a) + \gamma V^*(s')] \tag{3}$$

Equation (3) is called the Bellman equation, which returns the highest expected value of a state when an optimal action is taken. This decision-making process computation is called Markov decision process (MDP). Further details and explanation on MDP can be found in [33].

In real world, the state transition probability and the reward are often unpredictable if not unknown. In the proposed work, temporal difference leaning from reinforcement learning is adopted to address this issue. Temporal difference (TD) learning uses weighted average approach in calculating the expected gain in trust by taking samples of outcomes from action taken. The samples of outcomes act as the sample experience for the trust evaluation using MDP as shown in Eq. (4).

$$T(s) \leftarrow T'(s) + \alpha(sample\ experience - T'(s)) \tag{4}$$

In Eq. (4), '$T(s)$' is the value of the state which in this case is the trust and 'α' is the learning rate that factors the sample experience received to the initial trust '$T'(s)$.' The proposed model evaluates the trust placed on an agent by iteratively bootstrapping new experience to the previous trust estimate using observations obtained through interaction with a target agent. The model is built on the concept of TD learning which combines the characteristic of dynamic programming and Monte Carlo method [34]. The underlying trust in the agent based on observation or evidence is calculated using the beta reputation system (BRS) in Eq. (5) [35].

$$T_{i,j} = \left(\frac{r+1}{r+s+2}\right) \tag{5}$$

Equation (5) evaluates the trust '$T_{i,\ j}$' of agent 'i' on agent 'j.' Here, 'r' denotes an observed positive outcome and 's' is observed negative outcome obtained through interactions with agent 'j.' As the agents collaborate to complete given objectives, the subsequent interactions and the observation that follow serve as new sample experience, '$f(exp)$.' This sample experience is a function of both reinforcement reward and observational trust.

With each experience, the agent receives a reward that could be positive or negative reinforcement based on the outcome. The outcome is essentially a verification on the accuracy and truthfulness of the information given to the agent based on the observation it perceived after interaction. In the proposed experience evaluation model, BRS is used to compute the trust gained from the interaction outcome. The sample experience '$f(exp)$' can be modeled as Eq. (6).

$$f(exp) = R + \gamma\left(\frac{r+1}{r+s+2}\right) \tag{6}$$

The $f(exp)$ value is then added to the previous trust value to get an updated trust. This iterative process is called bootstrapping and results in temporal difference learning function. The gamma represents the discount factor which indicates the importance factor of the current experience gained. In this case, the discount factor is set as '1.' The update function for the trust combining the sample experience and the bootstrapping process is shown in Eq. (7).

$$f(exp) = \begin{cases} (0.333) + 1\left(\frac{r+1}{r+s+2}\right), & \textit{if positive outcome} \\ (-0.333) + 1\left(\frac{r+1}{r+s+2}\right), & \textit{if negative outcome} \end{cases} \tag{7}$$

Table 2 Brief summary of the cases for evaluating the probabilistic nature of the developed model

Case	Model	Observation
i	$T_{i,j} \leftarrow T'_{i,j} + \alpha\left(0.5 - T'_{i,j}\right)$	The trust value $T_{i,j}$ at time zero is 0.5 which relates to a neutral trust
ii	$T_{i,j} \leftarrow T'_{i,j} + \alpha\left(1 - T'_{i,j}\right)$	With continuous update under positive interaction outcomes, the trust value will not go higher than '1'
iii	$T_{i,j} \leftarrow T'_{i,j} + \alpha\left(0 - T'_{i,j}\right)$	With continuous update under negative interactions outcome, the trust value will not go below '0'

The reward values are established as 0.333 and −0.333 for positive and negative reward reinforcement, respectively. These values are established after conducting preliminary simulation experiments. Equation (8) shows the complete trust value update equation.

$$T_{i,j} \leftarrow T'_{i,j} + \alpha\left(f(exp) - T'_{i,j}\right) \tag{8}$$

In Eq. (8), 'α' is the learning rate which decides how much that the new experience is weighted during the trust value and '$T'_{i,\,j}$' is the previous trust value. The proposed Eq. (8) is the developed trust model called TD Trust model. This equation represents a probabilistic approach of evaluating the trust value where the trust lies between [0, 1]. This is proofed with these cases explained below.

Case i When agent '*i*' collaborates with agent '*j*' to complete a set of objectives, it receives a series of observations during the collaboration. The observations are used to assist in evaluating the trust agent '*i*' has on agent '*j*.' A positive outcome means that the collaboration successfully completed an objective and vice versa. At the initiation of the system, the values of '$T_{i,\,j}$,' '$T'_{i,\,j}$,' 'α,' '*r*' and '*s*' are '0.' As such, the initial trust value is computed as follow.

$$f(exp) = 0 + 1 \cdot \left(\frac{0+1}{0+0+2}\right)$$

$$T_{i,j} \leftarrow T'_{i,j} + \alpha\left(0.5 - T'_{i,j}\right)$$

The trust value $T_{i,j}$ at time zero is 0.5 which relates to a neutral trust.

Case ii When the interaction returns a positive outcome, the value of '*r*' and '*s*' is set to '1' and '0,' respectively. The trust value '$T_{i,j}$' is updated by adding the new experience, '*f*(*exp*)' and the reinforcement reward received to the previous trust value:

$$f(exp) = 0.333 + 1 \cdot \left(\frac{1+1}{1+0+2}\right)$$

$$T_{i,j} \leftarrow T'_{i,j} + \alpha\left(1 - T'_{i,j}\right)$$

Assuming that every collaboration results in a positive outcome, the '*f*(*exp*)' value results in one. With continuous update under positive interaction outcomes, the trust value will not go higher than '1.'

Case iii When a negative outcome is observed from the interaction, the values of r and s are set to '0' and '1,' respectively. The trust value '$T_{i,j}$' is updated by adding the new experience, '*f*(*exp*)' and the reinforcement reward received to the previous trust value:

$$f(exp) = -0.333 + 1 \cdot \left(\frac{0+1}{0+1+2}\right)$$

$$T_{i,j} \leftarrow T'_{i,j} + \alpha\left(0 - T'_{i,j}\right)$$

Assuming that every collaboration results in a negative outcome, the '*f*(*exp*)' value results in zero. With continuous update under negative interactions outcome, the trust value will not go below '0.'

Table 2 summarizes the three cases presented above highlighting the importance of the case study.

3.2 Algorithm structure

The pseudocode for the proposed trust evaluation model in a MAS exploration environment is shown in Fig. 3 highlighting the application of the proposed model in an exploration scenario when the agents are given a set of objectives to complete. When information or data related to the objectives are shared by an agent in the system, other agents explore that accuracy of that information and in turn update the trust placed on the agent sharing the information. Once the agents are able to verify the accuracy of information by observing the environment, the return of the observation provides a sample experience for trust evaluation. A positive sample experience is indicated by the presence of a goal or objective and vice versa. Figure 4 shows the pseudocode for the trust update algorithm.

4 Simulation

In order to examine the performance of the proposed model, the model is tested using simulation experiments in Visual Studio 2013 C ++ platform. The test-bed design proposed in this experiment is adapted from the works of

Algorithm 1: Exploring the Environment

```
Initiate the system;
while objectives not achieved do
    if information available from other agents then
        verify the information;
        update the trust;
    else
        explore the environment;
```

Fig. 3 Proposed trust evaluation model in exploration environment

Algorithm 2: Trust Update

Data: V(s), Trust value update
Result: Calculate the sample experience and then update trust value

```
if goal is observed then
    r = 1;
    s = 0;
else
    r = 0;
    s= 1;
end
Update sample experience;
```

$$f(exp) = r + \gamma\left(\frac{r+1}{r+s+2}\right) \quad (1)$$

Update trust value;

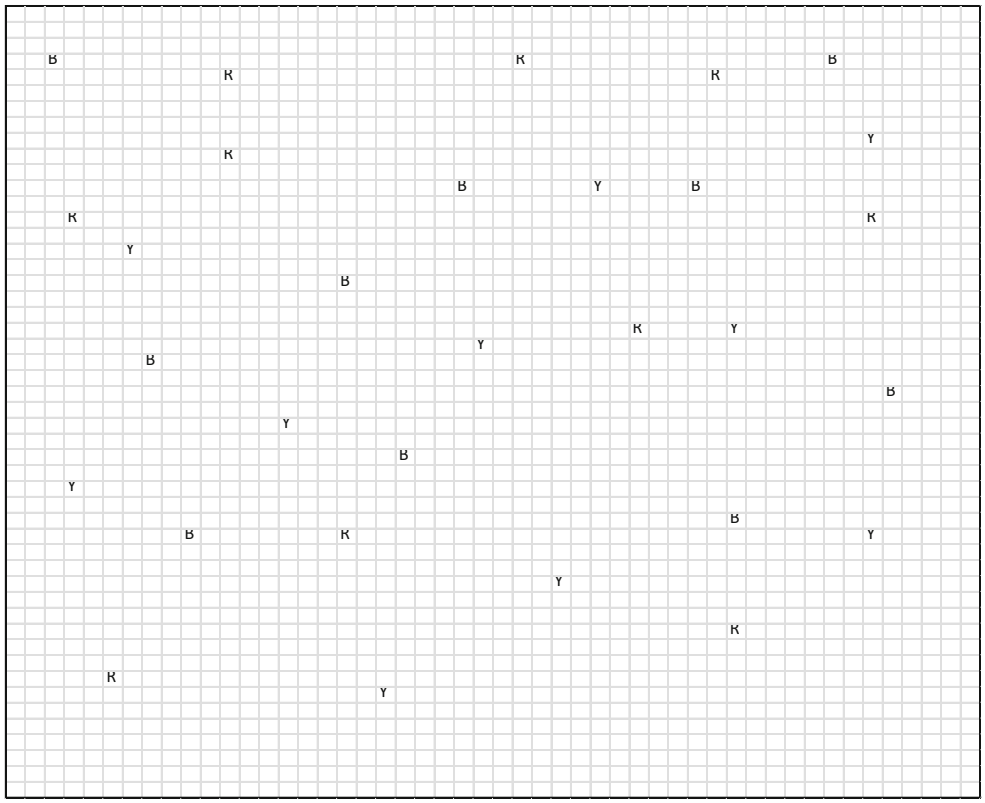
$$T(s) \leftarrow T(s) + \alpha[f(exp) - T'(s)] \quad (2)$$

Fig. 4 Trust update algorithm

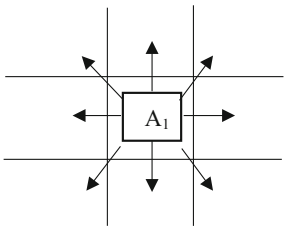
Mantel and Clark [26]. The simulation environment is divided into different grids cell with some of the cells having different colored balls in them. In the simulation experiment, each agent is assigned a specific colored ball and is tasked to explore the environment to find their respective colored ball found among the empty cells. The probability values used to reflect the chances of lying are 20 %. 40, 60 and 80 %, i.e., 20 % of lying or chance of information shared being false and so on. The objective of the simulation is to collect all the goals, and the simulation terminates once all the goals have been collected.

The simulation experiments framework is similar to the goal collecting example explained in Sect. 2: When agent A_2 or A_3 encounters a potential ball location of A_1, the location information is shared with to A_1. When A_1 arrives at the target location, it verifies the presence of the ball through observation and updates the trust value of the agent that shared the information based on the observation outcomes. The term goal is used to refer to the ball in the subsequent sections. Three agents (A_1, A_2 and A_3) are set at the same initial position and are given the task to locate their respective targets goals, scattered in the given environment. If an agent encountered the target of another agent, e.g., A_1 encountered the goal of A_2, that agent could share the information about the location with the other agent, e.g., A_1 shares the location information with A_2. Each of the agents is subjected to probability or chances of lying when sharing the information.

Experiments are conducted under three different lying conditions, and for each condition thirty different environment maps with different goal locations in each environment are used. In this instance, the goals are the different colored balls scattered in the environment. Figure 5 shows an example of the simulation environment map used in the simulation experiment. Each color



(a)



(b)

Fig. 5 **a** An example map of the simulation environment used in the experiment. Each alphabet (*R* red, *Y* yellow, *B* blue) represents the goal or objective of the respective agents. **b** An example of the agent's possible movement directions in the simulation environment. The agent can move in eight possible directions from the current cell it is in

represents the goal of the respective agents, and the simulation ends once all the goal locations are identified.

The purpose of this simulation is to study how the trust among the agents changes or fluctuates when other agents lying. The measures used to evaluate the proposed model are:

1. Time (time steps) taken to complete objective.
2. Fluctuation in the trust values.
3. Optimal interactions required to identify trustworthy robot.

The performance of the proposed trust-based exploration algorithm is evaluated against the trust models from the literature namely Secure Trust [8], ACT Trust [18] and Distributed Trust [26] which are some of the recent trust evaluation models.

4.1 Performance: time step

The total time taken to complete given simulation objectives, i.e., collecting the goals is one the measure investigated in this study is. However, the simulation experiments execute at different speeds in different computers due to the varying hardware processing capabilities. To address this issue, a more generalized approach is considered to measure the simulation completion time. The proposed approach is by evaluating the time steps taken.

Each time the agent moves from one unit location to the next, it is considered as taking one step. In the proposed evaluation approach, each step taken by the agent corresponds to one time step. Additionally, each time the agents goes to a potential goal location shared by other agents, the agent is increasing the total time steps it took to complete its objective. The smaller the time steps the robot took to complete the objective reflects faster completion time, denoting an efficient system.

Having high trust on a particular information provider means the information shared is always true thus reducing the overall time taken to complete the objective. This is due to the reduction in random wandering in the environment and exploring the wrong locations. However, if the information is coming from an untrustworthy agent, following the information only leads to increase in time steps and overall time as the agent is constantly misled to wrong locations.

Figure 6 shows the time steps performance comparison for the three-agent goal collecting simulation experiment when agents A_1, A_2 and A_3 are subjected to 0, 20 and 40 % probability of lying, respectively. The simulation is repeated under this condition for thirty different environment map. Under these conditions, the ACT Trust model and the proposed TD Trust model performed the best with the most minimal time steps to achieve the given objectives. However, the proposed TD model performed 3.32 % better than the ACT Trust model.

The simulation experiments are continued by changing the probability of lying where agents A_1, A_2 and A_3 are subjected to 0, 40 and 60 % probability of lying, respectively. Once again, the simulation is carried out in thirty different simulation maps to obtain an average result. The results obtained are shown in Fig. 7. Again, the developed TD Trust model and the ACT Trust model show the lowest time steps indicating fastest objective completion rate. Of the two models, the proposed model performed slightly better by 1.13 %.

For the subsequent simulation experiments, the noise levels are escalated further where agents A_1, A_2 and A_3 are subjected to 0, 60 and 80 % probability of lying, respectively. The simulation experiments are conducted thirty times in different environments to investigate how the trust fluctuates under high chances of lying. Under these conditions, the Secure Trust and the proposed TD Trust models show the lowest time steps for objectives completion, i.e.,

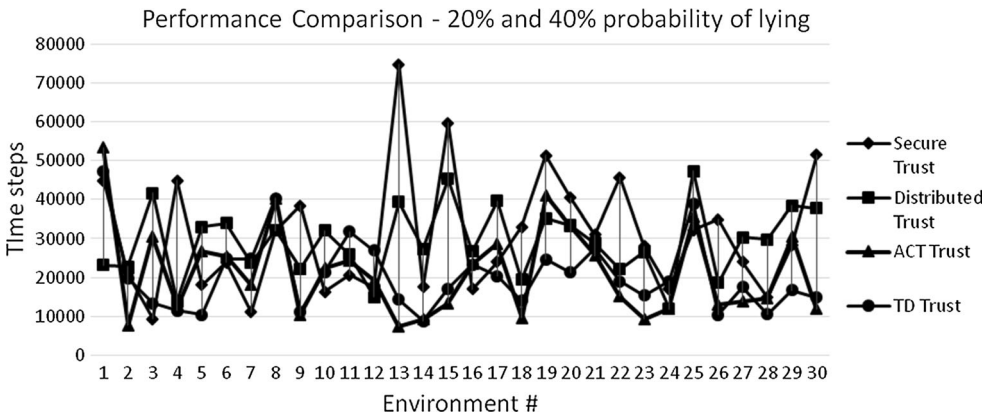


Fig. 6 Comparing the time steps performance between models at 20 and 40 % probability of lying by A_2 and A_3, respectively

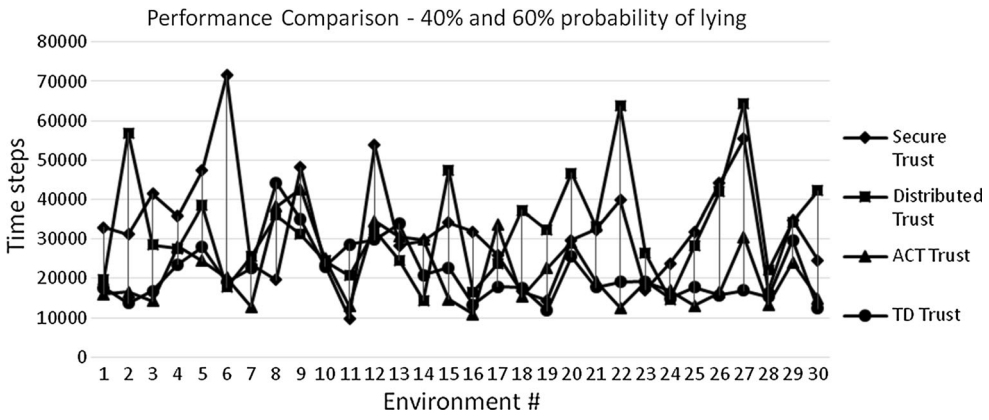


Fig. 7 Comparing the time steps performance between models at 40 and 60 % probability of lying by A_2 and A_3, respectively

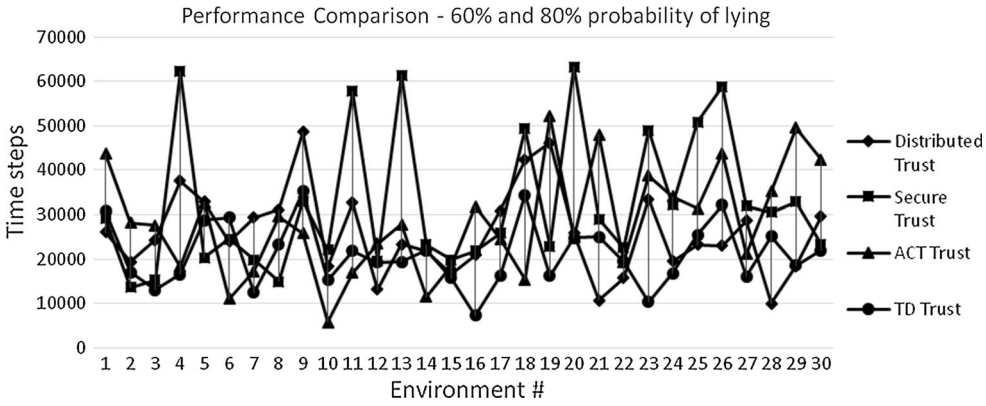


Fig. 8 Comparing the time steps performance between models at 60 and 80 % probability of lying by A_2 and A_3, respectively

Table 3 Summary of time steps taken to complete simulation objectives

	Sending false information (%)	Secure trust	Distributed trust	ACT trust	TD trust (proposed)	Comparison: TD and ACT (%)
Time steps	20–40	30,779	29,261	21,289	**20,582**	**3.32**
	40–60	32,298	32,420	21,707	**21,462**	**1.13**
	60–80	25,904	32,691	28,352	**20,987**	**25.60**

Bold values indicate that the proposed TD Trust model performs better than the models from literature

fastest to achieve the objectives. Between the two models, the proposed TD Trust model performed 18.98 % faster than Secure Trust model. Figure 8 shows the results of this simulation study.

Using trust evaluation method, an agent can choose to follow information coming from agents that are trustworthy only. This way, the agents are always guaranteed to find the goal at the target locations and only value-added time steps are included to its total time steps count. As the distrust increases, the agent spends more time steps exploration location that do not contain the goal. This is the theory behind the simulation study. When comparing the performance of the three trust models from the literature, the ACT Trust model is the best performing among the three. The developed TD Trust performs just as good as, in some case better than the best model from the literature especially under high probability of lying. The results of the simulations (average time steps of 30 environments) are shown in Table 3.

4.2 Performance: interactions and trust value

4.2.1 Interactions

The second parameter investigated in this simulation experiments is the number of interactions required to distinguish between trustworthy and untrustworthy agents as well as the fluctuation of trust values. Since trust is a highly subjective and abstract notion, there are certain factors to be considered before adopting a trust evaluation model. The two factors considered in this parameters investigation are minimal fluctuation of trust value during updates and optimal interaction to identify trustworthy robots.

For the first factor, minimal trust fluctuation refers to a situation where when the trust value placed on a particular agent does not change dramatically upon encountering a positive or negative interaction. A certain degree of benefit of doubt has to be given to an agent before updating the trust even when positive outcome is observed. As for the second factor, a trust model should take an optimal number of interactions before the trustworthiness of an agent can be concluded. Having too little interaction before concluding trustworthiness could lead to trusting the wrong agent while taking too long to trust could lead to a longer time steps before completing objectives.

In this investigation, the evaluation index named successful transaction rate (STR) adopted from [8] is used to evaluate the performance of the models. STR measures the percentage of successful transactions or interactions by calculating the ratio of number of successful interactions over the total number of interactions. A successful interaction refers to accurate information shared by an agent with a positive outcome observed during interaction, i.e., goal presence at the target location. In almost all the simulation experiments, the proposed TD trust evaluation model outperforms all the other trust models. Table 4 shows the STR performance comparison between the TD Trust model and the benchmarking models.

Table 4 STR performance deviation (in %) between TD Trust and other trust models

Models	Probability of lying			
	20 %	40 %	60 %	80 %
Distributed trust	0.25	16.52	26.01	38.97
Secure trust	7.89	20.60	21.74	157.76
ACT trust	−16.93	3.19	17.85	28.79

Positive percentage indicates that the TD Trust model outperforms the other models and vice versa

Table 4 shows that at 20 % probability of lying, the STR of TD Trust model performs marginally better than Distributed Trust and Secure Trust model with 0.25 and 7.89 % higher in STR, respectively. However, under the same condition TD Trust model showed 16.93 % lower in STR than ACT Trust model. With the exception of this low performance, TD Trust model continuously showed increasingly higher STR as the probability of lying increases.

The STR is determined by the interaction outcomes and directly refers to the capability of the trust model to identity a trustworthy agent correctly and quickly. A higher number of interactions with an untrustworthy agent result in a lower STR. The simulation results from Table 4 show that the proposed TD Trust achieved higher STR which implies that the model is faster in distinguishing between trustworthy and untrustworthy agents.

Since determining trustworthiness is a subjective matter, another criteria investigated in this study are the optimal number of interactions required before trustworthiness of an agent can be determined. Once trustworthiness is determined, an agent that received information could choose to ignore it if the agent sharing the information is deemed untrustworthy. Table 5 highlights the number of interaction required by the models before the trustworthiness can determine by each of them in all the simulations.

It is observed that Distributed Trust and Secure Trust models took the least number of interactions to determine trustworthiness while ACT Trust and the proposed TD Trust models took almost twice the number of interaction. Closer observation showed that the proposed TD Trust model took slightly less interactions compared to ACT Trust before the trust worthiness is determined. Under optimal interactions, the agent should not take too many or too few interaction to determine the trustworthiness of an agent as this would leads to increase in time taken to complete mission objectives. Based on this observation, it can be objectively concluded that TD Trust model performs optimal interactions before concluding trustworthiness of an agent.

Table 5 Number of interactions required to determine trustworthiness

Models	Probability of lying			
	20 %	40 %	60 %	80 %
Distributed trust	3	2	2	1
Secure trust	3	3	2	2
ACT trust	6	6	5	4
TD trust	*5*	*4*	*4*	*3*

Italic values indicate that the proposed TD Trust model exhibits the most optimal number of interaction when compared against the models from the literature

4.2.2 Trust value

This study also investigates how the trust value fluctuates when an outcome of an interaction is observed. One critical point raised here is that when a positive or negative outcome is observed through interaction, the increase or decrease in the trust value should not be drastic. Instead, it is objectively arguable that the increase or decrease in the trust should be gradual or incremental.

Figure 9 shows an example of the trust value fluctuation observed during a simulation experiment where agent A_1 is evaluating the trust on agent A_2 based on a series of interaction. The value 0.30 is the minimum trust value required in order to initiate any form of interaction. In Fig. 9, the fluctuation of trust value during the five interactions between A_1 and A_2 is shown when A_2 is subjected to 20 % probability of lying. During the first and third interactions, a positive outcome is observed so the trust value increases from the previous trust values. During interaction number two, four and five on the other hand, negative outcome is observed; thus, the trust value decreases.

When the first negative interaction is observed during the second interaction, the Secure Trust and Distributed Trust models calculated a drastic drop in the trust value, where it almost goes below the minimum value. When a

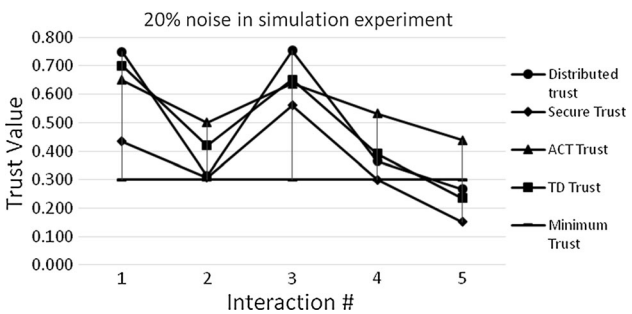


Fig. 9 Trust values fluctuation with A_2 having 20 % probability of lying

positive outcome is observed immediately after, there is a sharp increase in trust under Distributed Trust model. The Secure Trust model on the other hand makes a very cautious trust update. Both ACT Trust model and the proposed TD Trust model show an intermediate increase in trust, falling between Secure Trust and Distributed Trust. When the subsequent interactions reveal continuous negative outcomes, all the trust models except for ACT Trust model deemed that agent A_2 is no longer trustworthy as the trust value dropped below the minimum value. ACT Trust model, however, still gives room for further update on the trust even when this may lead to non-value-added time steps to its completion time. The proposed TD model exhibits a subjective evaluation of trust while the benchmarking models do not.

5 Application of the proposed approach

The application of the proposed trust evaluation model is suitable for any multi-agents system that works in the same environment to accomplish common objective. This section highlights some of the potential applications of the proposed trust evaluation mode in the field of multi-robot systems, MRS. The research in the field of robotics has exponentially grown in the past decade due to the vast applicability of robotics in various fields and industries. Some examples of such application used in this section are search and rescue, navigation and surveillance. It is hypothesized that the application of the proposed trust evaluation model could increase the success rate of the robotics system by objectively eliminating non-value-added robots from the mission.

5.1 Search and rescue

One of the potential applications of multi-robot system is to aid and assist in the search and rescue operations. The use of robots in such operations is more apparent when the mission involves dangerous environment such as in unstable building structures, structures under fires, radioactive area or any environment that is dangerous for human to operate [36, 37]. The robots could be given critical roles to assist the human rescue workers in the operations such as identifying trapped victims, spot potential hazardous zones, investigate radioactive leaks, mapping the environment.

While exploring the environment, the robots could share sensor data among each other and relay the necessary information to human search and rescue counterparts for further actions. However, due to the highly dangerous environment that the robots are working in, the robots could experience failures and damages. Some of the potential outcomes are damaged sensors, breakdown, trapped under obstacles, etc. If the robots continue to broadcast information contradicting to their actual situation, i.e., the robot are lying, the entire time sensitive mission could be end up in failure with the possibility of lost in human life. Therefore, it is critical that the robots work together with the human teams to accomplish the mission without interference from such untrustworthy or faulty robots.

The proposed trust evaluation model could be beneficial in such application to help in identifying the faulty or untrustworthy robots. The individual robot is evaluated each time it shares information, and if the trust value drops below a minimum trust level, the robot is presumed untrustworthy and possibly damaged. With the removal of non-value-added robots, the efficiency of the system should improve and the rescue of the victims could be accomplished faster.

5.2 Navigation

Robotics are widely used for exploring and mapping of a new environment. One of the most well-known uses of robotics for exploration is the Mars rover, a mobile robot launched by NASA to the planet Mars for research. During exploration, navigation is critical to ensure the robots avoid dangerous areas or obstacles, able to localize itself and map the environment correctly. In a multi-robot exploration scenario, the robots could share such information to explore a larger area, avoid dangerous zones and share the map to ensure same places are not repeatedly explored.

However, since the robots are exploring new environment, the robots could be subjected to various issues such as damaged sensors, collision with obstacles, noisy sensor reading due to high noise in environment. If a robot is sharing incorrect information under such condition, the robot might be endangering the progress of the entire mission. For example, a robot may wrongly map a danger zone as safe zone due to incorrect readings from its damaged sensors. When other robots use this information, those robots may end up getting damaged or destroyed, putting the mission at risk.

Under such circumstances, a trust evaluation model could prove helpful in identifying faulty or lying robots. As each robot shares their respective information, their trust value is also update to reflect accuracy in the information. If a robot persistently sends false data, the trust on the robot would eventually drop below a minimum level. The robot would be recalled back from the mission for checking and maintenance. With the removal of faulty robot, the other robots could carry on the exploration with higher efficiency.

5.3 Surveillance

The application of robotics in military for surveillance and intelligence gathering is highly discussed for the past few years [36, 38]. The field of robotics in the art of covert warfare has evolved over the years with the advancements in robots especially with the introduction of nano drones, insect drones, high-altitude drones and more. Together with the advancements in communication technology, these drones could be automated or operated remotely in order to ensure secrecy of the mission.

In a surveillance operation, the drones are used for various purposes such as stalking a target, surveying a target location or building and mostly for collecting intelligence useful for subsequent missions by human soldiers. To ensure a thorough and accurate information gathering, multiple drones could be used as surveillance teams. During the operation, the drones may have to share and transmit intelligence as a part of its mission objectives. However, it is highly risky to trust the intelligence with ensuring that the drones are not damaged or faulty or worst hijacked by the enemy due to unpredicted reason. The drones could be sending false intelligence such as incorrect sensor reading, incorrect map layout, tracking the wrong target and so on. Any follow-up mission by the soldiers based on this information could result in failure or worst, lost in life.

Therefore, the proposed trust evaluation framework could be used in the surveillance teams to ensure the drones are trustworthy. As the drones share intelligence with each other, the drone is accessed for its accuracy and quality of shared intelligence. If a drone often sends incorrect or worst, damaging intelligence, the drone is deemed untrustworthy for the operation and further intelligence from the drone can be ignored. Such drone can be recalled for inspection and maintenance.

6 Conclusion and future work

In a MAS, trust plays a vital role in determining the success or failure of a mission. This paper highlights the work done to incorporate the human concept of trust into a MAS where agents are capable of evaluating and updating the trust they have on each other. The proposed TD Trust model is developed using temporal difference learning via reward and observation conditioning. The proposed model is tested in simulation experiments, and its performance is compared against some of the recent trust models found in the literature. The results indicated that the proposed model is capable of empirically evaluating trust of an agent with greater accuracy and takes optimal number of interactions to determine the trustworthiness of an agent. Future work includes improving the model to include indirect trust learning and implementing the proposed model in real-world experimentation.

Acknowledgments This research is funded by e-Science grant provided by Ministry of Science, Technology and Innovation (MOSTI), Malaysia, Project Number: 03-02-10-SF0200.

References

1. Basheer GS, Ahmad MS, Tang AY, Graf S (2015) Certainty, trust and evidence: towards an integrative model of confidence in multi-agent systems. Comput Hum Behav 45:307–315
2. Yugang L, Nejat G, Vilela J (2013) 'Learning to cooperate together: a semi-autonomous control architecture for multi-robot teams in urban search and rescue', Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, pp 1–6
3. Mohammed M, Lim C, Quteishat A (2014) A novel trust measurement method based on certified belief in strength for a multi-agent classifier system. Neural Comput and Appl 24(2):421–429
4. Fullam KK, Klos TB, Muller G, Sabater J, Schlosser A, Topol Z, Barber KS, Rosenschein JS, Vercouter L, Voss M (2005) 'A specification of the Agent Reputation and Trust (ART) testbed: experimentation and competition for trust in agent societies', Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, pp 512–518
5. Han Y, Zhiqi S, Leung C, Chunyan M, Lesser VR (2013) A survey of multi-agent trust management systems. IEEE Access 1:35–50
6. Busoniu L, Babuska R, De Schutter B (2008) A comprehensive survey of multiagent reinforcement learning. IEEE Trans Syst Man Cybern Part C Appl Rev 38(2):156–172
7. Aref A, Tran T (2014) 'Using fuzzy logic and Q-learning for trust modeling in multi-agent systems', Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on, pp 59–66
8. Das A, Islam MM (2012) SecuredTrust: a dynamic trust computation model for secured communication in multiagent systems. IEEE Trans Dependable Secure Comput 9(2):261–274
9. Miao G, Ma Q, Liu Q (2015) Consensus problems for multi-agent systems with nonlinear algorithms. Neural Comput Appl 1–10. doi:10.1007/s00521-015-1936-6
10. Kurniawati H, Yanzhu D, Hsu D, Lee WS (2010) Motion planning under uncertainty for robotic tasks with long time horizons. Int J Robot Res 30(3):308–323
11. Kurniawati H, Bandyopadhyay T, Patrikalakis NM (2012) Global motion planning under uncertain motion, sensing, and environment map. Auton Robots 33(3):255–272
12. Gorbunov R, Barakova E, Rauterberg M (2013) Design of social agents. Neurocomputing 114:92–97
13. Rosaci D, Sarné GML, Garruzzo S (2012) Integrating trust measures in multiagent systems. Int J Intell Syst 27(1):1–15
14. Sabater J, Sierra C (2005) Review on computational trust and reputation models. Artif Intell Rev 24(1):33–60
15. Vanderelst D, Ahn RM, Barakova EI (2008) Simulated Trust-Towards robust social learning. In: ALIFE, pp 632–639
16. Vanderelst D, Ahn RM, Barakova EI (2009) Simulated trust: a cheap social learning strategy. Theor Popul Biol 76(3):189–196
17. Huynh TD, Jennings NR, Shadbolt NR (2006) An integrated trust and reputation model for open multi-agent systems. Auton Agent Multi-Agent Syst 13(2):119–154
18. Yu H, Shen Z, Miao C, An B, Leung C (2014) Filtering trust opinions through reinforcement learning. Decis Support Syst 66:102–113

19. Jøsang A (2001) A logic for uncertain probabilities. Int J Uncertain Fuzziness Knowl-Based Syst 9(3):279–311
20. Zhou P, Gu X, Zhang J, Fei M (2015) A priori trust inference with context-aware stereotypical deep learning. Knowl-Based Syst 88:97–106
21. Fan W, Perros H (2014) A novel trust management framework for multi-cloud environments based on trust service providers. Knowl-Based Syst 70:392–406
22. Jin-Hee C, Chan K, Mikulski D (2014) Trust-based information and decision fusion for military convoy operations, Military Communications Conference (MILCOM), 2014 IEEE, pp 1387–1392
23. Wang Y, Singh MP (2007) Formal trust model for multiagent systems, In Proceedings of the 20th international joint conference on Artifical intelligence, pp 1551–1556
24. Wang Y, Singh MP (2010) Evidence-based trust: a mathematical model geared for multiagent systems. ACM Trans Auton Adapt Syst 5(4):1–28
25. Wang Y, Hang C-W, Singh MP (2011) A probabilistic approach for maintaining Trust.pdf. J Artif Intell Res 40:47
26. Mantel KT, Clark CM (2012) Trust networks in multi-robot communities, In Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on, pp 2114–2119
27. Jøsang A, Haller J (2007) 'Dirichlet Reputation Systems', Availability, Reliability and Security, ARES 2007. The Second International Conference on, pp 112–119
28. Namin AS, Ruizhong W, Weiming S, Ghenniwa H (2006) An efficient trust model for multi-agent systems, Computer Supported Cooperative Work in Design, 2006. CSCWD '06. 10th International Conference on, pp 1–6
29. Peng M, Xu Z, Pan S, Li R, Mao T (2012) AgentTMS: a MAS trust model based on agent social relationship. J Comput 7(6):1535–1542
30. Mui L, Mohtashemi M, Halberstadt A (2002) 'A computational model of trust and reputation', System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on, pp 2431–2439
31. Griffiths N (2005) Task delegation using experience-based multi-dimensional trust, In Proceedings of the fourth international joint conference on autonomous agents and multiagent systems, pp 489-496
32. Fullam KK, Barber KS (2007) Dynamically learning sources of trust information: experience versus reputation, In Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, pp 1–8
33. Russell SJ, Norvig P (2003) Artificial intelligence a modern approach (Pearson Education, Inc., 2003, Second edn. 2003)
34. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT press, Cambridge
35. Jsang A, Ismail R (2002) The beta reputation system, In Proceedings of the 15th bled electronic commerce conference, vol 5, pp 2502–2511
36. Gautam A, Mohan S (2012) A review of research in multi-robot systems', Industrial and Information Systems (ICIIS), 2012 7th IEEE International Conference on, pp 1–5
37. Nagatani K, Kiribayashi S, Okada Y, Otake K, Yoshida K, Tadokoro S, Nishimura T, Yoshida T, Koyanagi E, Fukushima M, Kawatsuma S (2013) Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. J Field Robot 30(1):44–63
38. Burdakov O, Doherty P, Holmberg K, Kvarnstrom J, Olsson PM (2010) Relay positioning for unmanned aerial vehicle surveillance. Int J Robot Res 29(8):1069–1087