Design and Validation of a Relative Trust Model[1]

Mark Hoogendoorn[*], Syed Waqar Jaffry[*,Ψ], Peter-Paul van Maanen,[*,+] and Jan Treur[*]

[*]Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{mhoogen, swjaffry, treur}@few.vu.nl
[Ψ]Punjab University College of Information Technology (PUCIT),
University of The Punjab, Shahrah-e-Quaid-e-Azam, Lahore, Pakistan
swjaffry@pucit.edu.pk
[+]Netherlands Organisation for Applied Scientific Research (TNO),
Department of Perceptual and Cognitive Systems
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
peter-paul.vanmaanen@tno.nl

Abstract

When considering intelligent agents that interact with humans, having an idea of the trust levels of the human, for example in other agents or services, can be of great importance. Most models of human trust that exist assume trust in one trustee is independent of trust in another trustee. The model introduced here addresses so-called relative trust. The idea of relative trust is that trust in a certain trustee not only depends on the experiences with that trustee, but also on trustees that are perceived competitors of that trustee. Such models for relative trust contain parameters to represent the specific dependency between trust for different trustees. In order to tailor the model towards a specific human, dedicated parameter estimation techniques are used. The validation shows that such a model for relative trust is able to predict human trust based behaviour significantly better compared to a benchmark model.

[1]Work presented in this paper is a significant extension of the works published in:

Hoogendoorn, M., Jaffry, S.W., and Maanen, P.-P. van., Validation and Verification of Agent Models for Trust: Independent compared to Relative Trust. Proc. of the International Conference on Trust Management, (TM'11). Advances in Information and Communication Technology, vol. 358, Springer Verlag, 2011, pp. 35-50.

Hoogendoorn, M., Jaffry, S.W., and Treur, J., An Adaptive Agent Model Estimating Human Trust in Information Sources. In: Baeza-Yates, R., Lang, J., Mitra, S., Parsons, S., Pasi, G. (eds.), Proceedings of the 9th International Conference on Intelligent Agent Technology, IAT'09. IEEE Computer Society Press, 2009, pp. 458-465.

1. Introduction

When considering relations and interaction between agents, the concept of trust is of utmost importance. Trust is being studied more and more in the areas of cognitive and social (neuro)science, often in relation to (social) decision making; see, for example ([1],[2],[3],[4],[5],[27],[28],[29],[30],[31],[32],[33],[34], [35],[36]). Within the domain of multi-agent systems, the concept of trust has been a topic of research for quite some years (e.g., [6],[7],[8],[9],[37],[38],[39],[40],[41],[42]). Within this research, the development of models expressing how agents form trust based upon direct experiences with a trustee or upon information obtained from parties other than the trustee is one of the central themes. Some of these models aim at creating trust models that can be utilized effectively within a software agent environment (e.g., [10]), whereas other models aim to present an accurate model of human trust (e.g., [11],[12],[13]). The latter type of model can be very useful when developing a personal assistant agent for a human with the awareness of the human's trust in different trustees (other agents, human or computer) and him or herself. This could for example avoid advising to use particular information sources that are not trusted by the human or could be used to enhance the trust relationship with the personal assistant agent itself.

In order for computational trust models to be usable in real life settings, the validity of these models should be proven first. As in such models parameters relating to personal characteristics play an important role, validation has to involve finding values for these parameters. For instance, in [14] an experiment has been conducted whereby the trends in human trust behaviour have been analyzed to verify qualitative properties underlying trust models developed in the domain of multi-agent systems. However, in that case no attempt was made to fit the model to the specific trusting behaviour of the human.

In this paper, a computational model for relative trust is presented, andthe results of a validation experiment for this trust model and an existing baseline trust model are reported. The existing trust model from [10], which was inspired on the trust model described in [13] has been taken as this baseline model. This model uses experiences with trustees in order to estimate the trust in different trustees and is an influential model in the domain of agent systems. The second model which was validated in this study is a model which also carries the notion of relative trust as described in [11]. The idea of relative trust is that trust in a certain trustee not solely depends on the experiences with that trustee, but also with trustees that are considered competitors of that trustee, or are related with it in another manner. A comparison between the two models is also made to see whether the notion of relative trust describes human trust behaviour in a more accurate way.The validation process includes a number of steps. First, an experiment with participants has been performed in which trust plays an important role. As a result, empirical data has been obtained, that is usable for validating the two models. One part of the dataset is used to learn the best parameter values for the two different trust models. The learning of these parameter values is not trivial, therefore dedicated parameter estimation techniques are introduced as well as an approach to apply them in the context of human trust models.

This paper is organized as follows. First, the two computational trust models are presented in Section 2. The human-based trust experiment used to collect data concerning human trust behaviour is explained in Section 3. Thereafter, in Section 4 parameter estimation techniques are introduced as well as an approach to tailor the parameters of the trust model towards the human behaviour as observed in the experiment described in Section 3. The verification and of properties underlying trust models against the human data is presented in Section 5. Finally, Section 6 is a discussion.

2. Models for Trust

In this section the two types of trust models are described. In Section 2.1 a model is explained that estimates human trust in one trustee independent of the trust in other trustees. In contrast, in Section 2.2 a model is described for which this relative dependency actually is important.

2.1 Models for Independent Trust

This section describes the independent trust model ([10],[13]), which was applied, for example, in ([15],[16],[17]). In this model trust is based on experiences and there is a certain decay of trust. Trustees are considered independent of each other.

For the present study, it is assumed that a set of trustees *{S_1, S_2, . . . , S_n}* is available that can be selected to give particular advice at each time step. Upon selection of one of the trustees S_i, an experience $E_i(t)$ is passed back indicating how well the trustee performed. This experience is a number on the interval [*-1, 1*]. Hereby, *-1* expresses a negative experience, *0* is a neutral experience and *1* a positive experience. There is also a decay parameter λ_i in the model, for which holds that $0 \leq \lambda_i \leq 1$. Given the above, for each trustee the next trust value $T_i(t)$ of trust now can be determined from the previous value $T_i(t-1)$ as follows:

$$T_i(t) = \lambda_i * T_i(t-1) + (1-\lambda_i) * \frac{E_i(t)+1}{2}$$

Note that the experience is mapped to the domain [*0, 1*] in this formula. Eventual reliance decisions are made by determining the maximum of the trust values over all trustees. Here it should be noted that $E_i(t)$ which resides in interval [*-1, 1*] is projected on the interval [*0, 1*] via transformation $(E_i(t)+1)/2$. This transformation is performed to keep $T_i(t)$ in [*0, 1*]. For more details on the rationale behind the formula, see ([10],[13]).

2.2 Relative Trust Model

In this section the relative trust model [11] is described. In this model trustees are considered to have some interrelation, for example, by being competitors, and the trust in a trustee depends on the experiences with the trustee relative to the experiences withthe other trustees. The model defines the total trust of the human as the difference between positive trust and negative trust (distrust) on the trustee. The model includes a number ofparameters representing human characteristics including trust flexibility β_i (measuring the change in trust on each new experience), decay γ_i (decay in trust when there is no experience) and autonomy η_i (dependence of the trust considering other options). The model parameters β_i , γ_i and η_i have values from the interval [*0, 1*].

As mentioned, the model is composed of two parts: one for positive trust, accumulating positive experiences, and one for negative trust, accumulating negative experiences. Both negative and positive trust are represented by a number between [*0, 1*]. The human's total trust $T_i(t)$ in S_i at time point *t* is the difference in positive trust $T^+_i(t)$ and negative trust $T^+_i(t)$ in S_i, which is a number between [*-1, 1*], where *-1* and *1* represent the minimum and maximum values of trust, respectively. The human's initial total trust in S_i at time point *0* is $T_i(0)$, which is the difference in initial trust $T^+_i(0)$ and distrust $T^-_i(0)$ in S_i at time point *0*.

As a differential equation the change in positive and negative trust over time is described in the following manner in [11]:

$$\frac{dT_i^+(t)}{dt} = \beta * \left(\eta * \left(1 - T_i^+(t)\right) + (1-\eta) * (\tau_i^+(t) - 1) * T_i^+(t) * \left(1 - T_i^+(t)\right)\right) * E_i(t) * (E_i(t)+1)/2 - \gamma * T_i^+(t) * \left(1 + E_i(t)\right) * \left(1 - E_i(t)\right)$$

$$\frac{dT_i^-(t)}{dt} = \beta * \left(\eta * \left(1 - T_i^-(t)\right) + (1-\eta) * (\tau_i^-(t) - 1) * T_i^-(t) * \left(1 - T_i^-(t)\right)\right) * E_i(t) * (E_i(t)-1)/2 - \gamma * T_i^-(t) * \left(1 + E_i(t)\right) * \left(1 - E_i(t)\right)$$

In these equations, $E_i(t)$ is the experience value given by S_i at time point *t*. Here the first equation represents the rate of change in the positive trust. The human's relative positive trust of S_i at time point *t* is the combination of two factors: the *autonomous*, and the *context-dependent* factor. For the *context dependent* factor an important indicator is the human's relative positive trust of S_i at time point *t* (denoted by $\tau_i^+(t)$): the ratio of the human's trust of S_i to the average human's trust on all options at time point *t*. Similarly an indicator for the human's relative negative trust of S_i at time point *t* (denoted by $\tau_i^-(t)$) is the ratio between the human's negative trust of the option S_i and the average human's negative trust on all options at time point *t*. These are calculated as follows:

$$\tau_i^+(t) = \frac{T_i^+(t)}{\sum_{j=1}^n T_j^+(t)/n} \text{ and } \tau_i^-(t) = \frac{T_i^-(t)}{\sum_{j=1}^n T_j^-(t)/n}$$

Here the denominators $\sum_{j=1}^{n} T_j^+(t)/n$ and $\sum_{j=1}^{n} T_j^-(t)/n$ express the average positive and negative trust of trustees over all options at time point *t* respectively. The context-dependent part was designed in such a way that when the positive trust is above the average, upon each positive experience the value is increased, and when it is below average it is decreased. This results in a form of competition between the different information agents. The principle used is a variant of a 'winner takes it all' principle, which for example is sometimes modelled by mutually inhibiting neurons representing the different options. In this case this is done by basing the change of trust when there is a positive experience on $\tau_i^+(t) - 1$, which is positive when the positive trust is above the average and negative when it is below. To normalise, this is multiplied by a factor $T_i^+(t) * \big(1 - T_i^+(t)\big)$. For the *autonomous* factor the change upon a positive experience is modelled by $1 - T_i^+(t)$. As η indicates to which extent the human is autonomous or context-dependent in trust attribution, the two factors are interpolated by using the weights η and *1-*η.

Finally, the total change in trust can be calculated as follows (using which the new trust value can easily be calculated):

$$\frac{dT_i(t)}{dt} = \frac{dT_i^+(t)}{dt} - \frac{dT_i^-(t)}{dt}$$

Similarly as for the independent trust model, in decision making the trustee with the highest trust value can be chosen.

2.3 Some Mathematical Analysis of the Relative Trust Model

To understand the model a bit better it may be useful to analyse some specific properties and special cases of it. This will be done here.

First consider special cases for the parameter η.

Case η = *1* (fully autonomous)

$$\frac{dT_i^+(t)}{dt} = \beta * \big(1 - T_i^+(t)\big) * E_i(t) * (E_i(t) + 1)/\,2 - \gamma * T_i^+(t) * \big(1 + E_i(t)\big) * \big(1 - E_i(t)\big)$$
$$\frac{dT_i^-(t)}{dt} = \beta * \big(1 - T_i^-(t)\big) * E_i(t) * (E_i(t) - 1)/\,2 - \gamma * T_i^-(t) * \big(1 + E_i(t)\big) * \big(1 - E_i(t)\big)$$

In this case no interaction takes place via the $\tau_i^+(t)$ and $\tau_i^-(t)$. This is a case of fully autonomous trust dynamics.

For constant experiences E_i over time in this case it can be determined what equilibrium value can occur for each of $T_i^+(t)$ and $T_i^-(t)$:

$$\frac{dT_i^+(t)}{dt} = \beta * \big(1 - T_i^+(t)\big) * E_i * (E_i + 1)/\,2 - \gamma * T_i^+(t) * (1 + E_i) * (1 - E_i) = 0$$
$$\frac{dT_i^-(t)}{dt} = \beta * \big(1 - T_i^-(t)\big) * E_i * (E_i - 1)/\,2 - \gamma * T_i^-(t) * (1 + E_i) * (1 - E_i) = 0$$

For example, for $T_i^+(t)$ this can be rewritten into $\beta * \big(1 - T_i^+(t)\big) * E_i * \frac{(E_i+1)}{2} = \gamma * T_i^+(t) * (1 + E_i) * (1 - E_i)$

$$\beta * E_i * \tfrac{(E_i+1)}{2} = \gamma * T_i^+(t) * (1 + E_i) * (1 - E_i) \; + \beta * T_i^+(t) * E_i * \tfrac{(E_i+1)}{2}$$
$$\beta * E_i * \tfrac{(E_i+1)}{2} = [\,\gamma(1 + E_i) * (1 - E_i) \; + \beta * E_i * \tfrac{(E_i+1)}{2}\,]\, T_i^+(t)$$
$$T_i^+(t) \;=\; \beta * E_i * \tfrac{(E_i+1)}{2} / [\,\gamma(1 + E_i) * (1 - E_i) \; + \beta * E_i * \tfrac{(E_i+1)}{2}\,]$$

A similar expression can be found for $T_i^-(t)$.

Caseη = *0* (dominant interaction)

$$\frac{dT_i^+(t)}{dt} = \beta * (\tau_i^+(t) - 1\,) * T_i^+(t) * \left(1 - T_i^+(t)\right) * E_i(t) * (E_i(t) + 1)/\,2 - \gamma * T_i^+(t) * \left(1 + E_i(t)\right) * \left(1 - E_i(t)\right)$$

$$\frac{dT_i^-(t)}{dt} = \beta * (\tau_i^-(t) - 1\,) * T_i^-(t) * \left(1 - T_i^-(t)\right) * E_i(t) * (E_i(t) - 1)/\,2 - \gamma * T_i^-(t) * \left(1 + E_i(t)\right) * \left(1 - E_i(t)\right)$$

In this case the interaction plays a dominant role.

Next, consider special cases for β:

Case $\beta = 0$ (fully inflexible)

$$\frac{dT_i^+(t)}{dt} = -\gamma * T_i^+(t) * \left(1 + E_i(t)\right) * \left(1 - E_i(t)\right)$$

$$\frac{dT_i^-(t)}{dt} = -\gamma * T_i^-(t) * \left(1 + E_i(t)\right) * \left(1 - E_i(t)\right)$$

In this case both will decrease over time due to the (autonomous) decay.

Furthermore, consider the simplification that can be made when for some *i* a constant positive experience $E_i(t) = 1$ occurs and constant negative experiences $E_j(t) = -1$ for all $j \neq i$. Then the model can be simplified into

$$\frac{dT_i^+(t)}{dt} = \beta * \left(\eta * \left(1 - T_i^+(t)\right) + (1 - \eta) * (\tau_i^+(t) - 1\,) * T_i^+(t) * \left(1 - T_i^+(t)\right)\right)$$

and for $j \neq i$:

$$\frac{dT_j^+(t)}{dt} = -\beta * \left(\eta * \left(1 - T_j^+(t)\right) + (1 - \eta) * \left(\tau_j^+(t) - 1\,\right) * T_j^+(t) * \left(1 - T_j^+(t)\right)\right)$$

Another special case is when $E_i(t) = 0$ for all *i* (only neutral experiences). In this case the model can be simplified into

$$\frac{dT_i^+(t)}{dt} = -\gamma * T_i^+(t)$$

$$\frac{dT_i^-(t)}{dt} = -\gamma * T_i^-(t)$$

In this case both will decrease over time exponentially due to the (autonomous) decay.

Finally it is addressed under which circumstances a trust of 1 can be maintained. That means that

$T_i^+(t) = 1$

$T_i^-(t) = 0$

$$\frac{dT_i^+(t)}{dt} = \frac{dT_i^-(t)}{dt} = 0$$

Filling this in the equations provides:

$-\gamma\left(1 + E_i(t)\right) * \left(1 - E_i(t)\right) = 0$
$\beta * \eta * E_i(t) * (E_i(t) - 1)/\,2 = 0$

The first can occur only when

$\gamma = 0$ or

$E_i(t) = 1$ or

$E_i(t) = -1$.

The second can only occur when

$\beta = 0$ or

$\eta = 0$ or

$E_i(t) = 0$ or

$E_i(t) = 1$.

In combination details are presented in Table 1:

Table 1. Combination Matrix

	$\boldsymbol{\gamma = 0}$	$\boldsymbol{E_i(t) = 1}$	$\boldsymbol{E_i(t) = -1}$
$\boldsymbol{\beta = 0}$	$\beta = 0$ and $\gamma = 0$ inflexible, no decay	$\beta = 0$ and $E_i(t) = 1$ inflexible, fully positive experience	$\beta = 0$ and $E_i(t) = -1$ inflexible, fully negative experience
$\boldsymbol{\eta = 0}$	$\eta = 0$ and $\gamma = 0$ fully autonomous, no decay	$\eta = 0$ and $E_i(t) = 1$ fully autonomous, fully positive experience	$\eta = 0$ and $E_i(t) = 1$ fully autonomous, fully negative experience
$\boldsymbol{E_i(t) = 0}$	$E_i(t) = 0$and $\gamma = 0$ no decay, neutral experience	*impossible*	*impossible*
$\boldsymbol{E_i(t) = 1}$	$E_i(t) = 1$ and $\gamma = 0$ no decay, fully positive experience	$E_i(t) = 1$ fully positive experience	*impossible*

The combination matrix shows a number of cases that are subsumed by another case. In particular the case of $E_i(t) = 1$ (fully positive experience) subsumes all other cases that also require $E_i(t) = 1$. This leads to the following simplification of the matrix as presented in Table 2:

Table 2. Simplified Combination Matrix

	$\boldsymbol{\gamma = 0}$	$\boldsymbol{E_i(t) = 1}$	$\boldsymbol{E_i(t) = -1}$
$\boldsymbol{\beta = 0}$	$\beta = 0$ and $\gamma = 0$ inflexible, no decay	$\beta = 0$ and $E_i(t) = 1$ inflexible, fully positive experience	$\beta = 0$ and $E_i(t) = -1$ inflexible, fully negative experience
$\boldsymbol{\eta = 0}$	$\eta = 0$ and $\gamma = 0$ fully autonomous, no decay	$\eta = 0$ and $E_i(t) = 1$ fully autonomous, fully positive experience	$\eta = 0$ and $E_i(t) = -1$ fully autonomous, fully negative experience
$\boldsymbol{E_i(t) = 0}$	$E_i(t) = 0$ and $\gamma = 0$ no decay, neutral experience	*Impossible*	*impossible*
$\boldsymbol{E_i(t) = 1}$	$E_i(t) = 1$ and $\gamma = 0$ no decay, fully positive experience	$E_i(t) = 1$ fully positive experience	*impossible*

Based on this the following summary can be made. There are the followingfour main cases (with subcases) to maintain trust 1:

$E_i(t) = 1$	(fully positive experience)	
$E_i(t) = 0$	(neutral experience)	and: $\gamma = 0$ (no decay)
$E_i(t) = -1$	(fully negative experience)	and: $\beta = 0$ (inflexibility) or $\eta = 0$ (nonautonomous)
$\gamma = 0$	(no decay)	and: $\beta = 0$ (inflexibility) or $\eta = 0$ (nonautonomous)

3. Human-based Trust Experiment

As the models described in Section 2 aim to describe the formation of human trust, it is essential to show that these models are able to accurately model human trust. Therefore, an experiment has been conducted to obtain empirical data that can be used to thoroughly evaluate the models. In this section the experimental methodology for this validation experiment is explained. In Section 3.1 the participants are described. In Section 3.2 an overview of the experimental environment used is given. Thereafter, the procedure of the experiment and data collection is explained in Sections 3.3.

3.1 Participants

Eighteen participants (eight male and ten female) with an average age of *23 (SD = 3.8)* participated in the experiment as paid volunteers. The participants were selected to be not colour blinded. All were experienced computer users, with an average of *16.2* hours of computer usage each week *(SD = 9.32)*.

3.2 Task

In Figure 1the setup is shown in which the experimental task was executed. This experimental task was a classification task in which two participants on two separate personal computers had to classify geographical areas according to specific criteria as areas that either needed to be attacked, helped or left alone by ground troops.

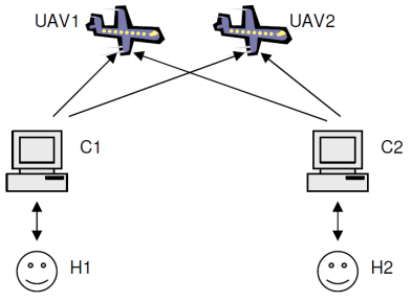


Figure 1. Experimental setup.

Per experiment, two participants needed to cooperate and base their classification on real-time computer generated video images that resembled video footage of real unmanned aerial vehicles (UAVs).There were two UAVs and the video footage of both UAVs was presented to each participant. On the UAV camera images, multiple objects were shown. There were four kinds of objects: civilians, rebels, tanks and cars (see Figure 2).

Name of object	Image	Score
Tank		+2
Rebel		+1
Civilian		-1
Person car		-2

Figure 2. Objects to be classified.

The identification of the number of each of these object types was needed to perform the classification. Each object type had a score (either *-2, -1, 0, 1* or *2*, respectively) and the total score within an area had to be determined. Based on this total score the participants could classify a geographical area: When the score was above *2*, then the area had to be attacked, since it contains enough enemy units as compared to friendly units. When the score was below -2, then the area had to be assisted, since it contains enough friendly units as compared to enemy units. When the score was between -2 and 2, then there are either too few units for any action or it is too difficult to assist or attack the area. Participants had to classify two

areas at the same time and in total *98* areas had to be classified. Both participants did the same areas with the same UAV video footage.

During the time a UAV flew over an area, three phases occurred.The first phase was the advice phase. In this phase both participants and a supporting software agent gave an advice about the proper classification (attack, help, or do nothing). This means that there were three advices at the end of this phase. It was also possible for the participants to refrain from giving an advice, but this hardly occurred. The second phase was the reliance phase. In this phase the advices of both the participants and that of the supporting software agent were communicated to each participant. Based on these advices the participants had to indicate which advice, and therefore which of the three trustees (self, other or software agent), they trusted the most. Participants were instructed to maximize the number of correct classifications at both phases (i.e., advice and reliance phase). The third phase was the feedback phase, in which the correct answer was given to both participants. Based on this feedback the participants could update their internal trust models for each trustee (self, other, software agent). In reality such feedback could be for instance the outcome of the mission related to the area after the decision was made to either attack, help or leave the area alone.

In Figure 3 the interface of the task is shown. The map is divided in *10 x 10* areas. These boxes are the areas that were classified. The first UAV started in the second area in the top left corner and the second one in the second area on left in the middle. The UAVs flew a predefined route, so participants did not have to pay attention to navigating the UAVs. The camera footage of the upper UAV is positioned top right and the other one bottom right.

The advice of the self, other and the software agent was communicated via dedicated boxes below the camera images. The advice to attack, help, or do nothing was communicated by red, green and yellow, respectively. On the overview screen on the left, feedback was communicated by the appearance of a green tick (correct answer in the reliance phase) or a red cross (wrong answer in the reliance phase). The reliance decision of the participant is also shown on the overview screen behind the feedback (feedback only shown in the feedback phase). The phase depicted in Figure 3 was the reliance phase before the participant indicated his reliance decisions for the two areas. The task was implemented using Gamemaker 8.0 (visit website at http://www.yoyogames.com/gamemaker).

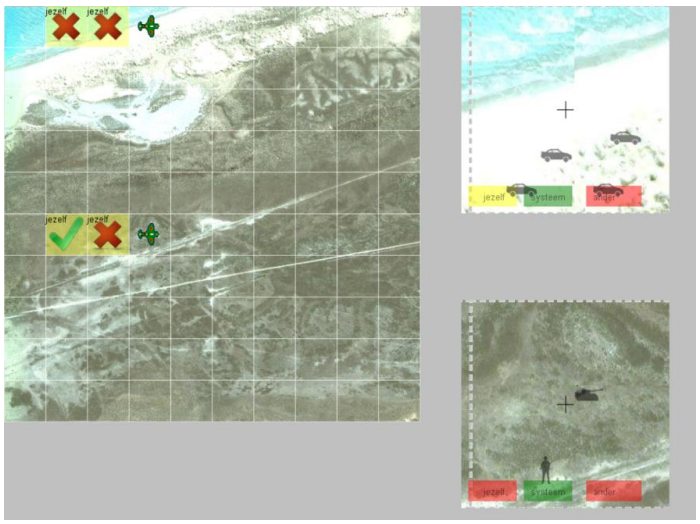


Figure 3. Interface of the task.

3.3 Data Collection

During the experiment described in section 3.2, input and output were logged using a client-server application. The interface of this application is shown in Figure 4. Two other client machines, that were responsible for executing the task as described in the previous subsection, were able to connect via a local area network to the server, which was responsible for logging all data and communication between the clients. The interface shown in Figure 4 could be used to set the client's IP-addresses and ports, as well as

several experimental settings, such as how to log the data. In total the experiment lasted approximately *15* minutes per participant couple.

Experienced performance feedback of each trustee and reliance decisions of each participant were logged in temporal order for later analysis. During the feedback phase the given feedback was translated to a penalty of either *0*, *0.5* or *1*, representing a good, neutral or poor experience of performance, respectively. This directly maps to the value $(E_i(t)+1) / 2$ in the trust models. During the reliance phase the reliance decisions were translated to either *0* or *1* for each trustee S_i, which represented that one relied or did not rely on S_i.

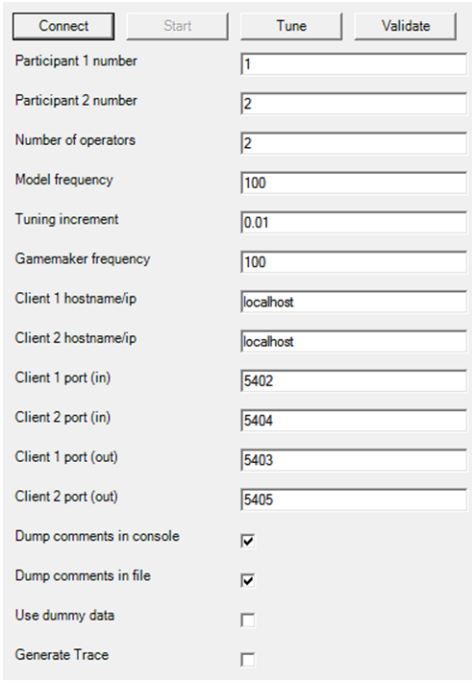
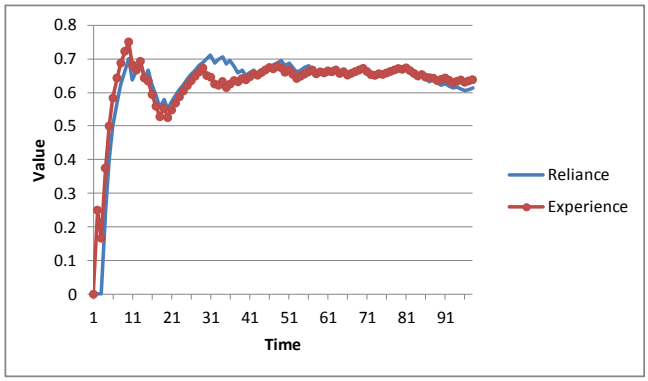
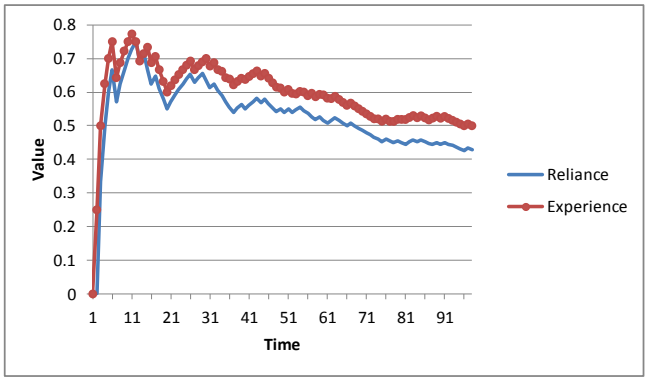


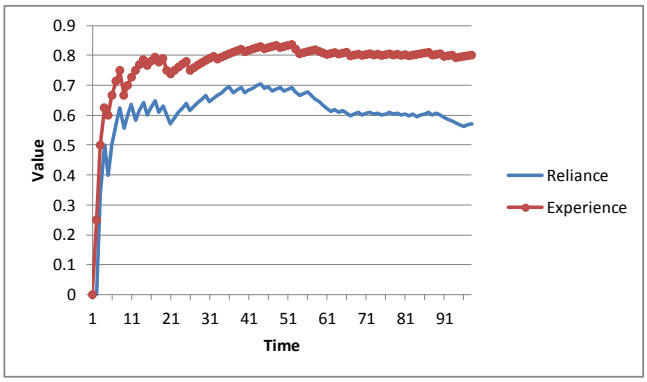
Figure 4. Interface of the application used for gathering validation data (Connect), for parameter adaptation (Tune) and validation of the trust models (Validate).



a)



b)



c)

Figure 5: Example of averaged experiences and reliances over time of an arbitrary participant for a) the self, b) the other, c) the software agent.

Figure 5 shows an example of the averaged experiences and reliances over time of an arbitrary human participant for the three trustees: Figure 5a the self, Figure 5b the other, and Figure 5c the software agent. Here the self means the reliance decision and respective experience received by the participant (the human subject under study) on herself, the other means the human peer in game which provides her advice on current game scenario and participant received experience afterwards while the software agent is the program which also provides advice to the participant based on a statistical distribution. It should be noted that if more than one trustee gave the same advice and the participant chooses that option, then it is assumed that the participant has relied on all of those trustees. As in Figure 5a, 5b and 5c one can see, that in these case the values of experiences and reliances often go in a similar direction over time, but do not necessarily overlap.

4. Tailoring the Models Towards Human Behaviour

In order to be able to perform a validation of the model against the human trust data obtained in the experiment described in Section 3 an additional step is needed. The trust models presented in Section 2 include a number of parameters representing personal characteristics and in order to judge how accurate the model is able to describe human trust behavior, these parameters should be set to appropriate values that fit the particular human that generated the data. Hence, the parameters need to be tuned. In order to do so, dedicated parameter estimation techniques have been developed. The scenario describing how these techniques are used to find appropriate parameter settings of the model is shown in Figure 6.

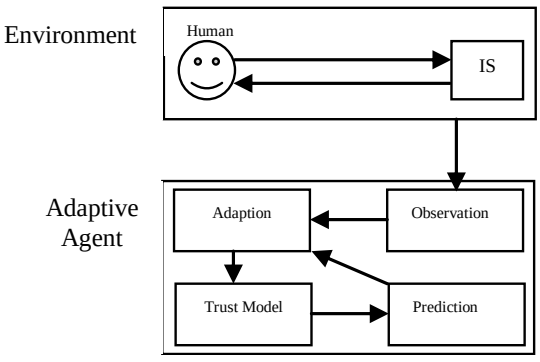


Figure 6. Adaptive Agent Model

In this approach it is assumed that the agent observes the behavior of the human (on what advice did the human rely) and the results of the advice (was the answer provided correct or not) over time. This scenario completely complies to the scenario within the experiment sketched in Section 3. At each time step the agent adapts the values of the model parameters using the available information. The agent starts with an initial vector of parameter values, determines human trust on information sources over time and predicts which reliance choices will be made by the human. Then it observes the actual human reliance decision. If the human places request to the same information source as predicted by the agent then the agent does not change the parameter vector (considering this prediction being correct), otherwise the parameter values are adopted accordingly. This approach allows both for the searching of parameters in an offline manner (the entire log of data is simply passed, resulting in an accurate set of parameters

describing the data), but it can also be used online in case new information continuously comes in. Note that the latter is not addressed further in this paper. To measure how accurately a parameter vector is representing human personality attributes, the accuracy of the parameter vector is calculated using the number of correct predictions made by the agent using it as follows:

Accuracy=Correct Predictions / Observed Behaviors

Four different parameter tuning methods have been used; they are described below.

4.1. Exhaustive Search

Using this method the entire attribute search space is explored to find the vector of parameter settings with maximum accuracy. This method guarantees to find the optimal solution. It is described as follows:

ALGORITHM – I:
for each observed behavior B
  for each vector P of parameter values
    calculate the accuracy of P
  end for
output the vector of parameter values with maximal accuracy
end for

In the algorithm – I, the calculation of the accuracy of a vector of parameter values P is done using the equation for calculating the accuracy described above. Here if α is the number of parameters to be estimatedwith precision τ, $n$is the number of information sources, and $b$the number of observed behaviors (i.e., number of time points), then the worst case complexity of the method can be expressed as $O((10)^{\alpha\tau}nb^2)$, which is exponential in the number of parameters and the precision. In particular, when $\alpha=3$ (parameters β, γ, and η), $\tau=2$ (i.e., precision 0.01), $n=3$ and $b=100$ then the use of this method takes (at most) 3×10^{10} steps.

4.2. Bisection Search

In the bisection method the attribute search space is reduced by halving the intervals for the parameter values at each step, and maintaining an exponentially shrinking focus range of parameter values between value vector P1 (lower bound) and value vector B2 (upper bound). It is described as follows:

ALGORITHM – II:
for each observed behavior B
  P1 = vector of lowest possible values of parameters
  P2 = vector of highest possible values of parameters
    while P2 – P1 > required precision
      d1 = calculate accuracy of P1
      d2 = calculate accuracy of P2
      take P3 the value vector determined as (P1 + P2)/2
        if d1 > d2
        then assign P3 to P2
        else assign P3 to P1
    end while
    if d1 > d2
    then output P1
    else output P2
end for

The operations in the algorithm – II should be considered as vector operations, for example P2 – P1 > precision in this algorithm means that it should hold for all components. The worst case complexity of this method is $O(\alpha\tau nb^2)$.

4.3. Extended Bisection Search Method

The extended bisection method is an extension to the bisection method. In this method, after finding a vector of parameter values against a new observed human behavior, this vector is kept in a list for future

use. On each next observed human behavior, bisection search finds a new vector of parameter values which is then compared with accuracies of known vectors at the current point in time. The vector with maximum accuracy is outputted. The method is described as follows:

ALGORITHM – III:

```
Solution-Parameter-List L is Empty
for each observed behavior B
    P1= vector of lowest possible values of parameters
    P2= vector of highest possible values of parameters
        while P2–P1 > required precision
            d1 = calculate accuracy of P1
            d2 = calculate accuracy of P2
            P3 = (P1 + P2)/2
              If d1 > d2
              then P2 = P3
              else P1 = P3
        end while
        if d1 > d2
        then add P1 to L
        else add P2 to L
        for all parameter value vectors P in L
          recalculate the accuracy of P
        end for
    output P with maximal accuracy from list L
end for
```

The worst case complexity of the algorithm – III can be expressed as $O(\alpha\tau nb^2)$.

4.4. Simulated Annealing Method

Simulated Annealing [18] uses a probabilistic technique to find a vector of parameter settings that best corresponds to human personality characteristics. In this method a random vector of parameter values is chosen as the best available parameter setting at the start. Then a displacement is introduced into this vector to generate a neighbor of the current parameter values in the search space. If this neighboring vector is found a more appropriate representation of the observed human behavior then it is marked as the best known vector of parameter values, otherwise a new neighbor is selected to evaluate its appropriateness. The number of neighbors that could be tried is limited by the computational budget available to the algorithm. The displacement in the parameter values vector to find a new neighbor depends on the temperature of the algorithm. In case the temperature is higher, the steps will become larger. The temperature of the algorithm at a certain time point is defined as follows

Temperature = computational_budget_left * (1 – accuracy)

In this expression the accuracy is the accuracy of the currently known best vector of parameter values. The displacement of a specific parameter (say γ) in the vector can be derived from the following two equations by selecting one at random:

new γ = γ + Temperature * (1–γ) * random_between [0,1]
new γ = γ – Temperature * γ * random_between [0,1]

The algorithm is described as follows:

ALGORITHM – VI:

```
for each observed behavior B
     chose a random parameter values vector R
     while computational-budget-remains
            find neighbor R1 of parameter values vector R
            if accuracy of R1 > R
            then R=R1
     decrease computational-budget
     end while
     output R
end for
```

If c is the computational budget, then the worst case complexity of the algorithm – IV can be expressed as $O(cnb^2)$. Note that the computational complexity of this method is independent of the number of parameters and precision.

4.5. Theoretical Analysis

Considering a case where an adaptive agent is to be designed for the trust model that has α number of parameters to be estimated with τ digits of precision in the estimated values then the worst case complexities of the algorithms I, II, III and IV are shown in Table 3.

Table 3. The rate of the growth of time complexity

Methods	**Complexity**
Exhaustive	$O((10)^{\alpha\tau}nb^2)$
Bisection	$O(\alpha\tau nb^2)$
Extended Bisection	$O(\alpha\tau nb^2)$
Simulated Annealing	$O(cnb^2)$

From this table it is obvious that the exhaustive search method, being exponential in the number of parameters to be estimated and the precision required in the values of parameters, is only practical for use in an adaptive agent when the number of parameters of the trust model under consideration and the required precision are modest. For instance, if the initial trust values for each of the information sources are to be taken as parameters as well, the computation time would severely increase. However, it is the only method with guaranteed success in finding the optimal solution.

4.6. Simulation-Based Experiments forthe Parameter Estimation Approaches

Before applying the approaches to the actual human data, a number of experiments with synthetic data have been conducted to characterize the behavior of the various parameter adaptation methods. In order to generate the synthetic data, simulated human behaviors were generated by the relative trust model as explained in Section 2.2 for specific values of human personality attributes (namely β, γ and η), and then the methods described in section 4 were used by the adaptive agent against these behaviors to predict the human characteristics β, γ and η. Configurations for generating the simulated human behavior are described in Table 4.

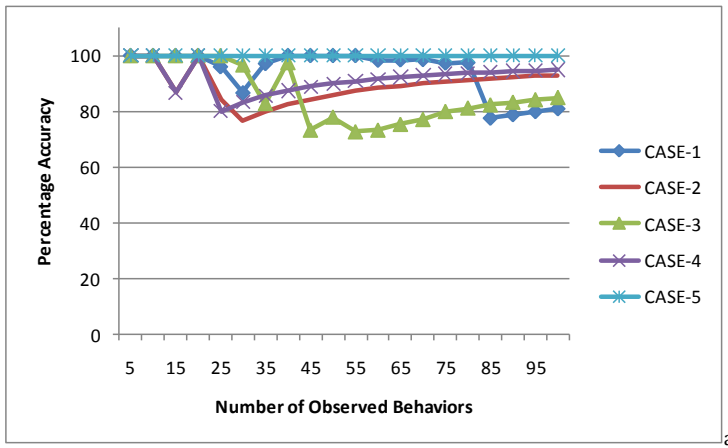
Within these simulation-based experiments, the parameters β, γ, and η are estimated. The desired precision of the estimated parameters, the number of information sources, and initial trust on the different information sources were kept constant. Furthermore, it is assumed that one of the three information sources (IS_1) gives positive responses while the other two (IS_2 and IS_3) give negative responses over each human request for information. Five cases have been studied, each representing different human personalities (i.e., different values of the parameters β, γ, and η).The experimental configurations used for the adaptive agent are the same as those shown in Table 4, except that the agent does not know the values of β, γ, and η in advance (as these are to be estimated).

Table 4. Model Configurations used for Experiments

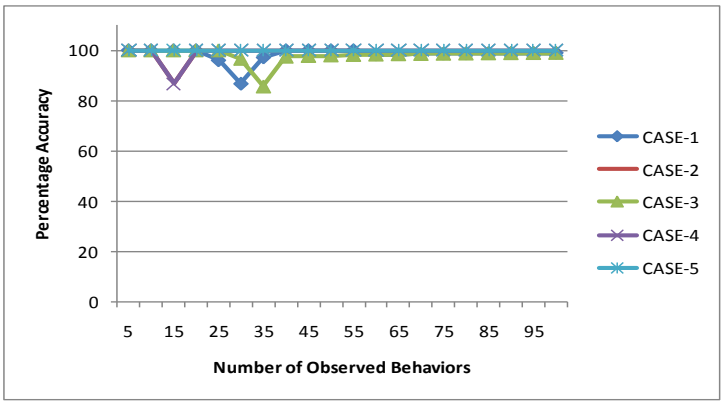
Case	**1**	**2**	**3**	**4**	**5**
Number of Parameters	*3*	*3*	*3*	*3*	*3*
Precision (digits)	*2*	*2*	*2*	*2*	*2*
Information Sources	*3*	*3*	*3*	*3*	*3*
Response of IS_1, IS_2, IS_3	*1,-1,-1*	*1,-1,-1*	*1,-1,-1*	*1,-1,-1*	*1,-1,-1*
Observed Behaviors	*100*	*100*	*100*	*100*	*100*
Trust Decay γ	*0.01*	*0.01*	*0.01*	*0.25*	*0.01*
Trust Flexibility β	*0.75*	*0.75*	*0.25*	*0.75*	*0.75*
Trust Autonomy η	*0.25*	*0.25*	*0.25*	*0.25*	*0.75*
Human Initial Trust on IS_1, IS_2, IS_3	*0.00, 0.15, 0.30*	*0.00, 0.05, 0.10*	*0.00, 0.05, 0.10*	*0.00, 0.05, 0.10*	*0.00, 0.05, 0.10*
Computational Budget (for S.A.)	*1000*	*1000*	*1000*	*1000*	*1000*

4.6.1 Results for the Simulation-Based Parameter Estimation Experiments

The adaptive agent as described has been implemented in C++. The graphs depicted in Figure 7 show the percentage accuracy of the parameter estimation for the bisection and extended bisection methods against the number of human behaviors observed. In Figure 7a it can be noted that initially for a smaller number of observed behaviors the accuracy of the estimated parameters is much higher. This is due to the fact that initially, the human behavior is slightly disclosed so there are many possible parameter settings that correspond to the observed human behavior hence the bisection method can find that with good accuracy. As the human behavior reveals itself more extensively over time, the set of the possible parameter settings that correspond to this behavior becomes smaller that makes good accuracy harder to achieve.



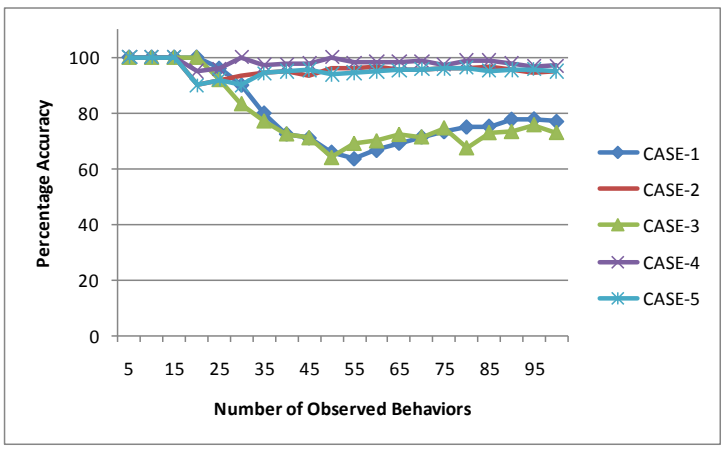
a)



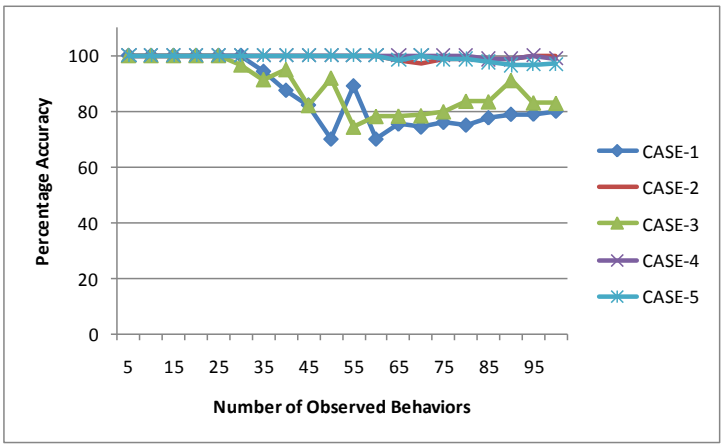
b)

Figure 7. Percentage accuracy of the adaptive agent using the a) Bisection and b) Extended Bisection Method

In Figure 7b it can be noted that the extended bisection method gives much better accuracy than the original bisection method as it keeps all previously known solutions in memory for future use.



a)



b)

Figure 8. a) Maximum and b) Minimum percentage accuracy using Simulated Annealing.

As Simulated Annealing is a probabilistic method hence several simulations were conducted to find the behavior of this method. Figures 8a and 8b show the maximum and minimum percentage accuracy of the estimated parameter settings for the simulated annealing method in ten simulation runs. The graph in Figure 9 shows the percentage accuracy of the estimated parameter settings for different methods after observing 100 human behaviors.

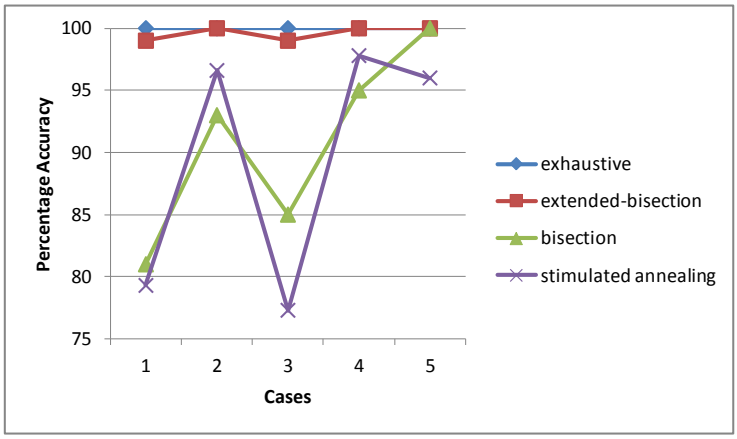


Figure 9. Accuracy of the parameter estimation methods in a simulation-based experiment

Here it can be observed that as expected the exhaustive search method gives 100 percent accuracy while the extended bisection outperforms the bisection method in all cases. Note that the accuracy mentioned here for the simulated annealing is the average accuracy for ten sample runs. It can be seen that bisection and simulated annealing are competitive in different cases.

Note that three human personality attributes (β, γ, η) with two digit precision generates one million human personalities, hence different values of human personality attributes may generate the same behavior trace particularly when only few human behaviors are observed. The number of human personality attributes vectors exactly generating the behaviors of cases 1, 2, 3, 4 and 5 of Table 4are found to be respectively 2, 1343, 3, 2387 and 46742 by the exhaustive search method. It can be noted that the performance of all methods (except exhaustive search) in cases 1 and 3 is below average compared to the other cases. This happened because behavior generated by a human under the configurations of these two cases matches to a small number of human personality attributes vectors in the entire human personality attributes vector set. This makes it harder for search methods to locate a vector corresponding to the human behavior.

Finally, during the runs the execution time has been measured. The bisection method was found most efficient taking *0.013375* seconds for parameter estimation against 100 human behaviors while the extended bisection took *0.014070*. Simulated Annealing for a computational budget 1000 took *0.022190* seconds that is approximately twice the time of the bisection method while the exhaustive search completed the task in *13.375* seconds. It can be observed that as expected, exhaustive search is much

more expensive than the other approaches, while the extended bisection consumes almost the same computation time as the bisection method. The computational time of the simulated annealing depends on the computational budget assigned.

5. Verification and Validation of Relative and Absolute Trust Models

In this section the verification and validation processesfor the relative and absolute trust models described in Section 2 are presented upon the data presented in Section 3 thereby utilizing the parameter adaptation techniques expressed in Section 4. In Section 5.1 the application of the parameter adaption techniquesfor is explained for the specific dataset, Section 5.2 explains the validation whereas 5.3 addressed the verification of the model.

5.1 Parameter Adaptation

The number of parameters of the models presented in Section 2 to be adapted for each model and each participant suggest that for the context considered here an exhaustive search as described in Section 4.1 for the optimal parameters is feasible (as also confirmed by the experiments in Section 4.6). This means that the entire parameter search space is explored to find a vector of parameter settings resulting in the maximum accuracy (i.e., the amount of overlap between the model's predicted reliance decisions and the actual human reliance decisions) for each of the models and each participant.

The data collection described in Section 3.3 was repeated twice on each group of two participants, called condition 1 and condition 2, respectively. The data from one of the conditions was used for parameter adaptation purposes for both models, and the data from the other condition for model validation. This process of parameter adaptation and validation was balanced over conditions, which means that condition 1 and condition 2 switch roles, so condition 1 is initially used for parameter adaptation and condition 2 for model validation, and thereafter condition 2 is used for parameter adaptation and condition 1 for model validation (i.e. cross-validation). Both the parameter adaptation and model validation procedure was done using the same application as was used for gathering the empirical data. The interface shown in Figure 2 can also be used to alter validation and adaptation settings, such as the granularity of the adaptation.

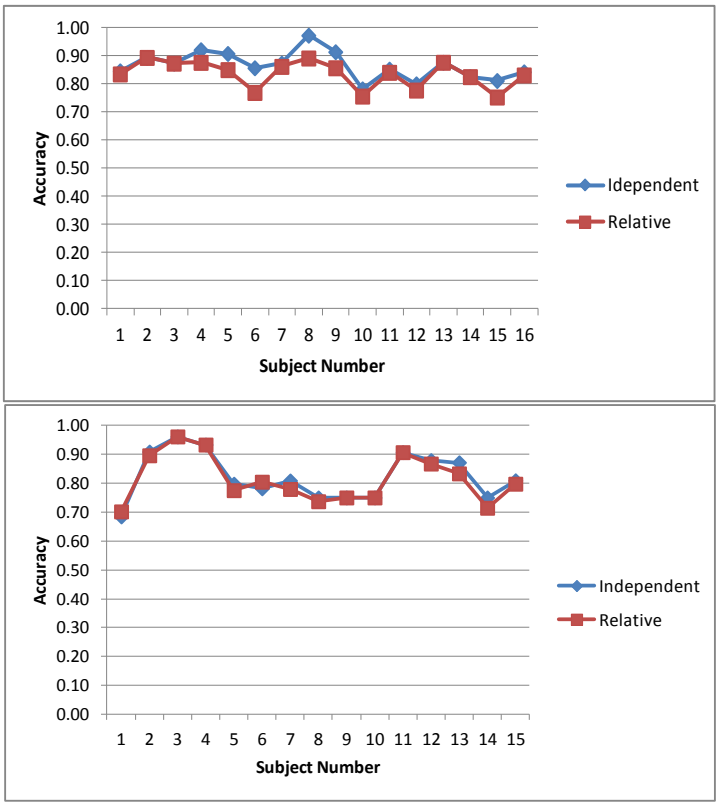


Figure10: Accuracy of parameter tuning of exhaustive search algorithm for both trust models for all subjects for condition1 (upper graph) and condition 2 (lower graph)

Using the approach put forward in the previous section, $2.94x10^4$ computation steps are needed for the independent trust model and $2.94x10^8$ for the relative trust model, which took on average 31 milli-seconds for the first, and 3 minutes and 20 seconds computation time for the second model[2].

In Figure 10 the x-axis represents subject number while the y-axis shows parameter tuning accuracy for subjects. Figure 10shows that the exhaustive search parameter adaption algorithm has tuned parameters that gives on average more than 80 percent accuracy on the training data. It canalso be noted that the Independent Trust Model has shown approximately 2.29 percent better tuning results than the Relative Trust Model.

5.2 Validation

In order to validate the two models described in Section 2, the measurements of the experienced performance feedback were used as input for the models and the output (predicted reliance decisions) of the models was compared with the actual reliance decisions of the participant. The overlap of the predicted and the actual reliance decisions was a measure for the accuracy of the models. The results are in the form of dynamic accuracies over time, average accuracy per condition (1 or 2) and per trust model (independent or relative).

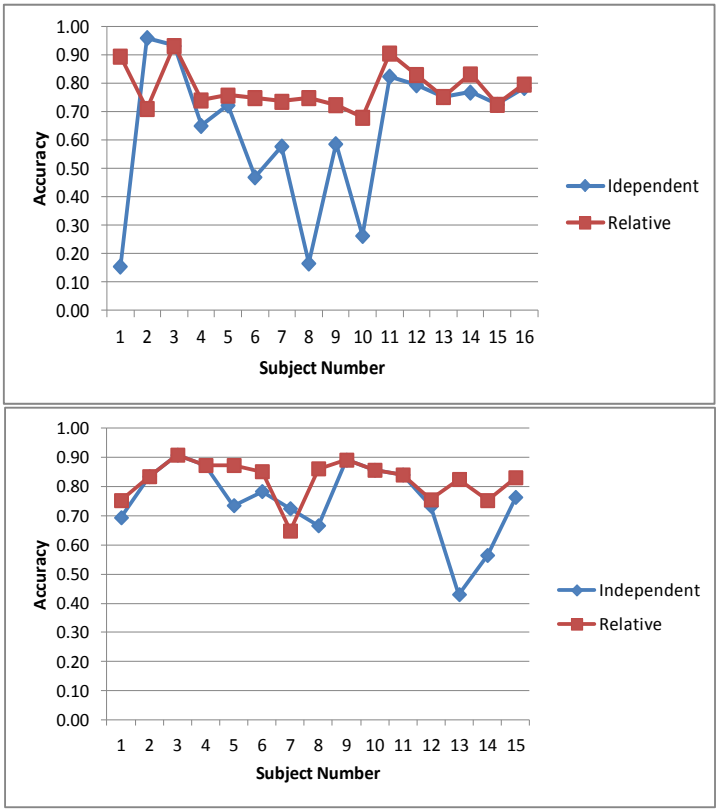


Figure11: Accuracy of human behavior prediction of both models for all subjects, prediction of condition1 (upper graph) and prediction of condition 2 (lower graph); outliers were excluded.

Figure 11 shows that relative trust model has a much better predictive capability then the independent trust model. It gives on average approximately 80 percent accuracy while the independent trust model gives approximately 72 percent. It canalso be noted that the trend of the independent trust model is not consistent overall subjects as compared to the relative trust model.

[2] This was on an ordinary PC with an Intel(R) Core(TM)^2 Quad CPU @2.40 GHz inside. Note that 31 x (2 x 94 x 10^8)/ (2 x 94 x 10^4) milli-seconds = *5.17* minutes *≠3.33* minutes computation time. This is due to a fixed initialization time of on average 11 milli-seconds for both models.

To perform a statistical analysis, from the data of 18 participants, one dataset has been removed due to an error while gathering data. This means that there are 34 data pairs (accuracies for 2 models): 2 (condition role allocations, i.e., parameter adaptation either in condition 1 or 2) times 17 (participants). Using Grubbs' test for outliers, from these pairs 3 outliers were removed. Hence in total 31 pairs were used for the data analysis. In Figure 12 the main effect of model type (either independent or relative trust) for accuracy is shown. A repeated measures analysis of variance (ANOVA) showed a significant main effect: $F(1, 29) = 7.60$, $p < .01$. This means that indeed the relative trust model had a higher accuracy $(M = .7968, SD = .0819)$ than the independent trust model $(M = .7185, SD = .1642)$.

Figure 13 shows the possible interaction effect between condition role allocation (parameter adaptation in condition 1 is referred to as adaptation 1 and parameter adaptation in condition 2 is referred to as adaptation 2) and model type (either independent or relative trust) on accuracy. No significant interaction effect was found $(F(1, 29) = .01, p = .93)$. Hence, no significant learning effect between conditions was found. Cross validation was not needed to balance the data, but the procedure still produced twice as much data pairs.

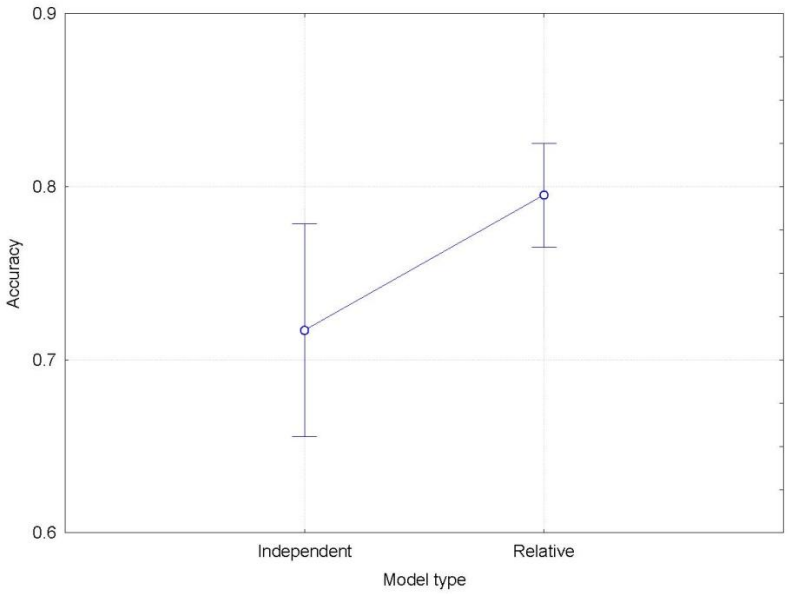


Figure 12. Main effect of model type for accuracy.

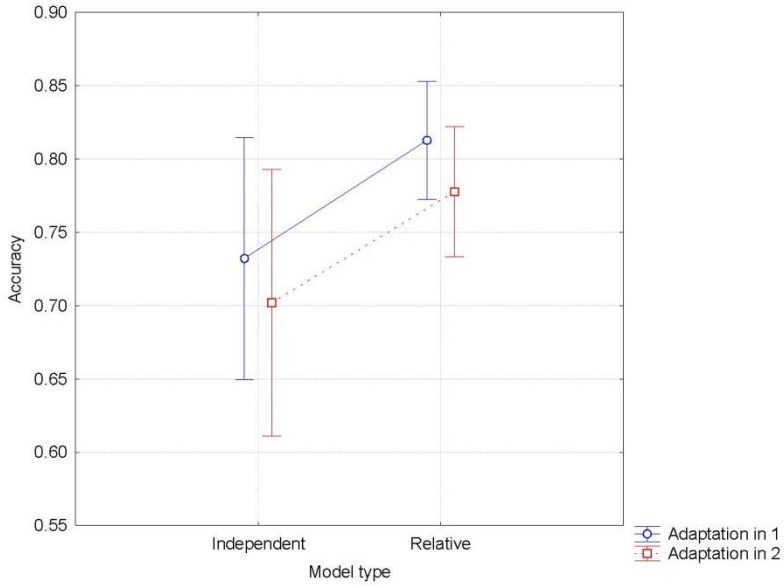


Figure 13. Interaction effect between condition role allocation and model type on accuracy.

5.3 Verification

Next to a validation using the accuracy of prediction using the models, another approach has been used to validate the assumptions underlying existing trust models. The idea is that properties that form the basis of trust models are verified against the empirical results obtained within the experiment. In order to conduct such an automated verification, the properties have been specified in a language called Temporal Trace Language (TTL) ([19],[20]) that features a dedicated editor and an automated checker. The language TTL is explained first, followed by an expression of the desired properties related to trust.

5.3.1 Temporal Trace Language (TTL)

The hybrid temporal language TTL supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. TTL is built on atoms referring to states of the world, time points and traces, i.e., trajectories of states over time. In addition, dynamic properties are temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γover state ontology Ont, the state in γ at time point tis denoted by state(γ, t). These states can be related to state properties via the formally defined satisfaction relation denoted by the infix predicate|=, i.e., state(γ, t) |= p denotes that state property p holds in trace γat time t. Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as $\neg, \wedge, \vee, \Rightarrow, \forall, \exists$. As a built-in construct in TTL, summations can be expressed, indexed by elements X of a sort S:

$$\Sigma_{X:S}\ \mathrm{case}(\varphi(X), V1, V2)$$

Here for any formula φ(X), the expression

$$\mathrm{case}(\varphi(X), V1, V2)$$

indicates the valueV1 if φ(X)is true, and V2otherwise. For example,

$$\Sigma_{X:S}\ \mathrm{case}(\varphi(X), 1, 0)$$

simply denotes the number of elements X in S for which φ(X) is true. As expressing counting and summation in a logical format in an elementary manner in general leads to rather complex formulae, this built-in construct is very convenient in use. For more details on TTL, see ([19],[20]).

5.3.2 Properties for Trust Models

Within the literature on trust, a variety of properties have been expressed concerning the desired behaviour of trust models. In many of these properties, the trust values are explicitly referred to. For instance in the work of [13] a number of characteristics of trust models have been defined (e.g., monotonicity, and positive experienceslead to higher trust). In this paperhowever, the trust function is subject of validation in experiments where no trust levels were reported. Therefore properties are needed that are expressed based on the available empirical informationon decision behaviour, to see whether these behaviours indeed comply to the desired behaviour of the trust models, while solely using the information which has been observed within the experiment. This information is then limited to the experiences that are received as an input and the decision choices that are made by the human that are generated as output. The properties from [21] are taken as a basis. Essentially, the properties indicate the following desired behaviour of human trust:

1. Positive experiences lead to higher trust
2. Negative experiences lead to lower trust
3. Most trusted trustee is selected

As can be seen, the properties also use the intermediate state of trust. In order to avoid this, it is however possible to combine these properties into a single property that expresses a relation between the experiences and the selection (i.e., the above items 1 &3 and 2 &3). Two of these properties are shown below. In addition, a property is expressed which specifies the notion of relativity in the experiences and the resulting selection of a trustee. The first property expresses that a trustee that gives the absolute best

experiences during a certain period is eventually selected at least once within, or just after that particular period, and is shown below.

P1(min_duration, max_duration, max_time): absolute more positive experiences results in selection
If a trustee *t1* always gives more positive experiences than all other trustees during a certain period with minimal duration *min_duration* and maximum duration *max_duration*, then this trustee *t1* is selected at least once during the period [*min_duration, max_duration+max_time*].
Formally:

$\forall\gamma$:TRACE, t_start, t_end:TIME, a:TRUSTEE
[[t_end - t_start$\geq$min_duration&t_end - t_start$\leq$max_duration&
absolute_highest_experiences(γ, a, t_start, t_end)]
$\Rightarrow$ selected(γ, a, t_start, t_end, max_time)]

Here

absolute_highest_experiences(γ:TRACE, a:TRUSTEE, t_start:TIME, t_end:TIME) $\equiv$
$\forall$t:TIME, r1, r2 :REAL, a2:TRUSTEE $\neq$ a
[[t $\geq$t_start& t <t_end&
state(γ, t) |= trustee_gives_experience(a, r1) &
state(γ, t) |= trustee_gives_experience(a2, r2)] $\Rightarrow$ r2 < r1]

selected(γ:TRACE, a:TRUSTEE, t_start:TIME, t_end:TIME, z:duration) $\equiv$
$\exists$t:TIME
[t $\geq$t_start& t <t_end + z & state(γ, t) |= trustee_selected(a)]

The second property, P2, specifies that the trustee which gives more positive experiences on average is at least selected once within or just after that period.

P2(min_duration, max_duration, max_time, higher_exp): average more positive experiences result in selection
If a trustee *t1* on average gives the most positive experiences (on average more than *higher_exp* better than the second best) during a period with minimal duration *min_duration* and maximum duration *max_duration*, then this trustee *t1* is selected at least once during the period [*min_duration, max_duration+max_time*].
Formally:

$\forall\gamma$:TRACE, t_start, t_end:TIME, a:TRUSTEE
[[t_end - t_start$\geq$min_duration&t_end - t_start$\leq$max_duration&
average_highest_experiences(γ, a, t_start, t_end, higher_exp)]
$\Rightarrow$ selected(γ, a, t_start, t_end, max_time)]

Here

average_highest_experiences(γ:TRACE, a:TRUSTEE, t_start:TIME, t_end:TIME, higher_exp:REAL) $\equiv$
$\forall$t:TIME, r1, r2 :REAL, a2:TRUSTEE $\neq$ a
[t $\geq$t_start& t <t_end&
[$\sum_{t:TIME}$ case(experience_received(γ, a, t, t_start, t_end, e), e, 0) >
($\sum_{t:TIME}$ case(experience_received(γ, a, t, t_start, t_end, e), e, 0) + higher_exp * t_end-t_start)]]

In the formula above

experience_received(γ:TRACE, a:TRUSTEE, t:TIME, t_start:TIME, t_end:TIME, r:REAL) $\equiv$
$\exists$r:REAL, t $\geq$t_start& t <t_end& state(γ, t) |= trustee_gives_experience(a, r)

The final property concerns the notion of relativeness which plays a key role in the models verified throughout this paper. The property expresses that the frequency of selection of a trustee that gives an identical experience pattern during two periods is not identical in case the other trustees give different experiences.

P3(interval_length, min_difference, max_time): Relative trust
If a trustee *t1* gives an identical experience pattern during two periods [t1, t1+interval_length] and [t2, t2+interval_length] and the experiences of at least one other trustee is not identical (i.e. more than min_difference different at each time point), then the selection frequency of *t1* will be different in the two periods.
Formally:

$\forall\gamma$:TRACE, t1, t2:TIME, a:TRUSTEE
[[same_experience_sequence(γ, a, t1, t2, interval_length) &
$\exists$a2 :TRUSTEE $\neq$ a [different_experience_sequence(γ, a, t1, t2, min_difference)]
$\Rightarrow\exists$i:DURATION <max_time
$\sum_{t:TIME}$ case(selected_option(γ, a, t, t1+i, t1+i+interval_length), 1, 0) /
(1+$\sum_{t:TIME}$ case(trustee_selected(γ, t, t1, t1+i+interval_length), 1, 0)) $\neq$
$\sum_{t:TIME}$ case(selected_option(γ, a, t, t2+i, t2+i+interval_length), 1, 0) /
(1+$\sum_{t:TIME}$ case(trustee_selected(γ, t, t2+i, t2+i+interval_length), 1, 0))]

Here

same_experience_sequence(γ:TRACE, a:TRUSTEE, t1:TIME, t2:TIME, x:DURATION) ≡
∀y:DURATION
[y ≥ x & y ≤ x &∃r :REAL
[state(γ, t1+y) |= trustee_gives_experience(a, r) &
state(γ, t2+y) |= trustee_gives_experience(a, r)]]

different_experience_sequence(γ:TRACE, a:TRUSTEE, t1:TIME, t2:TIME, x:DURATION,
min_difference:REAL) ≡
∀y:DURATION
[y ≥ x & y ≤ x &∃r1, r2 :REAL
[state(γ, t1+y) |= trustee_gives_experience(a, r1) &
state(γ, t2+y) |= trustee_gives_experience(a, r2) & |r1-r2| > y]]

trustee_selected(γ:TRACE, t:TIME, t_start:TIME, t_end:TIME) ≡
∃a:TRUSTEE [t ≥t_start& t <t_end& state(γ, t) |= trustee_selected(a)]

5.3.3 Verification Results

The results of the verification of the properties against the empirical traces (i.e., formalized logs of human behaviour observed during the experiment) are shown in Table 5. First, the results for properties P1 and P2 are shown. Hereby, the value of *max_duration* has been kept constant at 30 and the *max_time* after which the trustee should be consulted is set to *5*. The minimal interval time (*min_duration*) has been varied. Finally, for property P2 the variable higher exp indicating how much higher the experience should be on average compared to the other trustees is set to *0.5*. The results in Table 5 indicate the percentage of traces in which the property holds out of all traces in which the antecedent at least holds once (i.e., at least one sequence with the min duration occurs in the trace). This has been done to avoid a high percentage of satisfaction due to the fact that in some of the traces the antecedent never holds, and hence, the property is always satisfied in a trivial manner. The table shows that the percentage of traces satisfying P1 goes up as the minimum duration of the interval during which a trustee gives the highest experience increases. This clearly complies to the ideas underlying trust models as the longer a trustee gives the highest experiences, the higher his trust will be (also compared to the other trustees), and the more likely it is that the trustee will be selected. The second property, counting the average experience and its implication upon the selection behaviour of the human, also shows an increasing trend in satisfaction of the property with the duration of the interval during which the trustee on average gives better experiences.

The percentages are lower compared to P1 which can be explained by the fact that they might also give some negative experiences compared to the alternatives (whereas they are giving better experiences on average). This could then result in a decrease in the trust value, and hence, a lower probability of being selected.

Table 5. Results of verification of property P1 and P2

Setting for min_duration	% traces satisfying P	% traces satisfying P
1	64.70	29.40
2	64.70	29.40
3	86.70	52.90
4	92.30	55.90
5	100.00	58.80
6	100.00	70.60

The third property, regarding the relativity of trust has also been verified and the results of this verification are shown in Table 6. Here, the traces of the participants have been verified with a setting of min difference to *0.5* and max time to *5* and the variable interval length during which at least one trustee shows identical experiences whereas another shows different experiences has been varied. It can be seen that property P3 holds more frequently as the length of the interval increases, which makes sense as the human has more time to perceive the relative difference between the two. Hence, this shows that the notion of relative trust can be seen in the human trustee selection behaviour in almost *70%* of the cases.

Table 6. Results of verification of property P3

Setting for interval_length	% traces satisfying P
1	00.00
2	41.10
3	55.90
4	67.60
5	66.70
6	68.40

6. Discussion and Conclusions

In this paper, two models for trust dynamics have been presented and empirically analysed. An extensive validation study has been performed to show that human trust behaviour can be accurately described and predicted using such computational trust models. The steps involved in the validation process of these models have been presented in detail:

1) design and execution of experiment keeping the human in loop,
2) tuning or personalization of trust models against the traces of human behaviour, and
3) prediction of future human behaviour based on personalized human personalized model.

In order to get the empirical data, first an experiment has been designed that places humans in a setting where they have to make decisions based upon the trust they have in other themselves, other humans or software agents. In total 18 participants took part in the experiment. The results show that both an independent trust model ([10],[13]) as well as a relative trust model as described in [11] can predict this behaviour with a high accuracy (72% and 80%, respectively) by learning on one dataset and predicting the trust behaviour for another (cross-validation). Furthermore, it has also been shown that the underlying assumptions of these trust models (and many other trust models) are found in the data of the participants. Future research will be aimed at further testing assumptions on larger and more diverse groups of participants.The latter is needed to account for the variability related to individual differences in cognitive processing. Also more effort should be put on the investigation of different factors affecting cognition during task execution. Especially when tasks are time consuming and cognitively strenuous, factors such as fatigue, stress and anchoring heuristics can have a great effect on trust and the eventual reliance decisions made. The current study reports on the cumulative performance of models, but it would be interesting to see what would happen over time, given the previously mentioned factors.

Some more work on the validation of trust models has been performed. In [14] an experiment has been presented to investigate human trust behavior; here the observational data do not concern decisions made but reported scores of trust values over time. Moreover, although the underlying assumptions of trust models have to some extent been verified in that paper, no attempt has been made to fit a trust model to the data. Other papers describing the validation of trust models for instance validate the accuracy of trust models describing the propagation of trust through a network (e.g., [22]). In [23] a multidisciplinary, multidimensional model of trust in e-commerce is validated. The model includes four high-level constructs: disposition to trust, institution-based trust, trusting beliefs, and trusting intentions. The proposed model itself does however not describe the formation of trust on such a detailed level as the models used inthis paper, it presents general relationships between trust measures and these relationships are subject to validation. [24] validated a four-dimensional scale of trust in the context of e-Products and revalidates it in the context of e-Services which shows the influence of social presence on these dimensions of trust, especially benevolence, and its ultimate contribution to online purchase intentions. Again, correlations are found between the concepts of trust that have been distinguished, but no computational model for the formation of trust and the precise prediction thereof is proposed. Finally, in [25] a development-based trust measurement model for buyer-seller relationships is presented and validated against a characteristic-based trust measurement model in terms of its ability to explain certain variables of interest in buyer-seller relationships (long-term relationship orientation, information sharing, behavioral loyalty and future intentions).

The research in trust discussed above is mostly practically oriented. In recent years a more fundamental type of study of trust is developing moreand more, especially within the area of cognitive and social neuroscience. Among the topics addressed are, for example, the role of trust in social decision making, and the trust in faces, in relation to brain activity that can be measured, but also neurologically grounded computational models examples of this can be found in ([1],[2],[3],[4],[5],[21],[26]). For future research, the further integration of this more fundamental line of research with the more practically oriented line of research is an interesting challenge.

References

1. Delgado, M.R., Frank, R.H., Phelps, E.A. (2005). Perceptions of moral character modulate the neural systems of reward during the trust game. Nature Neuroscience, vol. 8, pp. 1611-1618.
2. Kang, Y., Williams, L., Clark, M., Gray, J., and Bargh, J., (2011). Physical temperature effects on trust behavior: The role of insula. Social Cognitive and Affective Neuroscience, vol. 6, 507-515.
3. Todorov, A., Baron, S.G., Oosterhof, N.N. (2008). Evaluating face trustworthiness: A model based approach. Social cognitive and Affective Neuroscience, vol. 3, pp. 119-27.
4. Winston, J.S., Strange, B.A., O'Doherty, J., Dolan, R.J. (2002). Automatic and intentional brain responses during evaluation of trustworthiness of faces. Natural Neuroscience, vol. 5, pp. 277–83.
5. Rilling, J.K., Sanfey, A.G., (2011). The Neuroscience of Social Decision-Making. Annu. Rev. Psychol, vol. 62, pp. 23-48.
6. Ramchurn, S., Huynh, D. and Jennings, N., (2004). Trust in multi-agent systems. The Knowledge Engineering Review, vol. 19, pp.1-25.
7. Sabater, J. and Sierra, C., (2005). Review on computational trust and reputation models. Artificial Intelligence Review, vol. 24, pp. 33-60.
8. Taddeo, M., (2010). Modelling trust in artificial agents, a first step toward the analysis of e-trust. Minds and Machines, vol. 20, issue 2, pp. 243-257.
9. Lai, C.-H., Liu, D.-R., and Lin, C.-S., (2013). Novel personal and group-based trust models in collaborative filtering for document recommendation. Information Sciences, vol. 239, pp. 31-49.
10. Maanen, P.-P. v., Klos, T. and Dongen, K. v., (2007). Aiding human reliance decision making using computational models of trust. In Proceedings of the Workshop on Communication between Human and Artificial Agents (CHAA • F07), pp. 372-376, Fremont, California, USA, 2007. IEEE Computer Society Press. Co-located with The 2007 IEEE IAT/WIC/ACM International Conference on Intelligent Agent Technology.
11. Hoogendoorn, M., Jaffry, S. W., and Treur, J., (2012). Cognitive and neural modeling of dynamics of trust in competitive trustees. Cognitive Systems Research,vol. 14, issue 1, pp. 60-83.
12. Falcone, R. and Castelfranchi, C., (2004). Trust dynamics: how trust is influenced by direct experiences and by trust itself. In Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04), pp. 740-747.
13. Jonker, C. M. and Treur, J., (1999). Formal analysis of models for the dynamics of trust based on experiences. In F. J. Garijo and M. Boman, editors, Multi-Agent System Engineering, Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, (MAAMAW'99).Springer Verlag vol. 1647, pp. 221-232.
14. Jonker, C. M., Schalken, J. J. P., Theeuwes, J. and Treur, J., (2004). Human experiments in trust dynamics. In Proceedings of the Second International Conference on Trust Management (iTrust 2004), vol. 2995 of LNCS, pp. 206-220. Springer Verlag.
15. Singh, S.I., and Sinha, S.K., (2010). A new trust model based on time series prediction and markov model. In: Das, V.V., and Vijaykumar, R. (eds.), Proceedings of the Information and Communication Technologies International Conference, ICT 2010, Springer Verlag,Communications in Computer and Information Science, vol. 101, pp. 148-156.
16. Skopik, F., Schall, D. and Dustdar, S., (2010). Modeling and mining of dynamic trust in complex service-oriented systems, Information Systems, vol. 35, pp. 735-757.
17. Walter, F.E., Battiston, S. and Schweitzer. F., (2009). Personalised and dynamic trust in social networks. In: Bergman, L, Tuzhilin, A., Burke, R., Felfernig, A., Schmidt-Thieme, L., Proceedings of the Third ACM conference on Recommender systems, RecSys'09, ACM, pp. 197-204.
18. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., (1983). Optimization by simulated annealing, science, New Series, vol. 220, pp. 671-680.

19. Bosse, T., Jonker, C., Meij, L. v. d., Sharpanskykh, A., and Treur, J., (2009). Specification and verification of dynamics in agent models. International Journal of Cooperative Information Systems, vol. 18, pp. 167-193.
20. Sharpanskykh, A., and Treur, J., (2010). A temporal trace language for formal modelling and analysis of agent systems. In: Dastani, M., Hindriks, K.V., and Meyer, J.J.Ch. (eds.), Specification and Verification of Multi-Agent Systems. Springer Verlag, pp. 317-352.
21. Hoogendoorn, M., Jaffry, S.W., and Treur, J., (2011). Modelling trust dynamics from a neurological perspective. In: Wang, R., Gu, F. (eds.), Advances in Cognitive Neurodynamics II, Proceedings of the 2nd International Conference on Cognitive Neurodynamics. (ICCN'09). Springer Verlag, 2011, pp. 523-536.
22. Guha, R., Kumar, R., Raghavan, P. and Tomkins, A., (2004). Propagation of trust and distrust. In Proceedings of the 13th international conference on World Wide Web (WWW'04), pp. 403-412, New York, NY, ACM.
23. McKnight, D. H., Choudhury, V. and Kacmar, C., (2001). Developing and validating trust measures for e-commerce: an integrative topology. Information Systems Research, vol. 13, no. 3, pp. 334-359.
24. Gefen, D. and Straub, D. W., (2004). Consumer trust in b2c e-commerce and the importance of social presence: experiments in e-products and e-services. Omega, vol. 32, pp.407.424.
25. Hernandez, J. M. da C. and Santos, C. C. dos., (2010). Development-based trust: Proposing and validating a new trust measurement model for buyer-seller relationships. Brazilian Administration Review, vol. 7, pp. 172-197.
26. Jaffry, S.W., and Treur, J., (2009). Comparing a cognitive and a neural model for relative trust dynamics. In: Leung, C.S., Lee, M., and Chan, J.H. (eds.), Proceedings of 16th International Conference on Neural Information Processing. (ICONIP'09). Lecture Notes in Computer Science, vol. 5863 Part I, Springer Verlag, 2009, pp. 72-83.
27. Corritore, C.L., B. Kracher, and S. Wiedenbeck, (2003). On-line trust: concepts, evolving themes, a model. International Journal of Human-Computer Studies, vol. 58, no. 6, pp. 737-758.
28. Johnson, D. and K. Grayson, (2005). Cognitive and affective trust in service relationships. Journal of Business Research, vol. 58, no. 4, pp. 500-507.
29. Morrow, J.L., Jr., M.H. Hansen, and W.P. Allison, (2004). The cognitive and affective antecedents of general trust within cooperative organizations. Journal of Managerial Issues, vol. 16, no. 1, pp. 48-64.
30. Newton, K., (2001). Trust, social capital, civil society, and democracy. International Political Science Review, vol. 22, no. 2, pp. 201-214.
31. Gilson, L., (2003). Trust and the development of health care as a social institution. Social Science & Medicine, vol. 56, no. 7, pp. 1453-1468.
32. Rothstein, B. and E.M. Uslaner, (2005). All for All: Equality, corruption, and social trust. World Politics, vol. 58, no. 01, pp. 41-72.
33. Warren, M.E., (1999). Democracy and trust, Cambridge University Press.
34. Robinson, R.V. and E.F. Jackson, (2001). Is trust in others declining in america? An age–period–cohort analysis. Social Science Research, vol. 30, no. 1, pp. 117-145.
35. Lewis, J.D. and A. Weigert, (1985). Trust as a social reality. Social forces, vol. 63, no. 4, pp. 967-985.
36. Rothstein, B., (2000). Trust, social dilemmas and collective memories. Journal of Theoretical Politics, vol. 12, no. 4, pp. 477-501.
37. Liau, C. J., (2003). Belief, information acquisition, and trust in multi-agent systems - a modal logic formulation. Artificial Intelligence, vol. 149, no. 1, pp. 31-60.
38. Schmidt, S., et al., (2007). Fuzzy trust evaluation and credibility development in multi-agent systems. Applied Soft Computing, vol. 7, no. 2, pp. 492-505.
39. Prietula, M., (1999). Exploring the effects of agent trust and benevolence in a simulated organizational task. Applied Artificial Intelligence, vol. 13, no. 3, pp. 321-338.
40. Ramchurn, S. D., et al., (2004). Devising a trust model for multi-agent interactions using confidence and reputation. Applied Artificial Intelligence, vol. 18, no. 9, pp. 833-852.
41. Huynh, T., N. Jennings, and N. Shadbolt, (2006). An integrated trust and reputation model for open multi-agent systems. Autonomous Agents and Multi-Agent Systems, vol. 13, pp. 2, pp. 119-154.
42. Breban, S. and J. Vassileva, (2002). Using inter-agent trust relationships for efficient coalition formation. Advances in Artificial Intelligence, vol. 2338, pp. 221-236.