Autonomic Trust Prediction for Pervasive Systems

Licia Capra and Mirco Musolesi
Dept. of Computer Science, University College London
Gower Street, London WC1E 6BT, UK
{L.Capra|M.Musolesi}@cs.ucl.ac.uk

Abstract

Pervasive computing is becoming a reality, thanks to advances in wireless networking and increased popularity of portable devices. Users of these devices will need support to decide who to interact with in a plethora of interconnected, self-interested peers. Various trust management models based on the human notion of trust have been proposed in recent years in order to support trust-aware decision making. However, the degree of subjectivity embedded in human trust often clashes with the requirements imposed by the target scenario: on one hand, pervasive computing calls for autonomic and light-weight systems that impose minimum burden on the user of the device (and on the device itself); on the other hand, computational models of human trust seem to demand a large amount of user input and physical resources. The result is often a computational trust model that actually does not 'compute': either the degree of subjectivity it offers is severely limited, or its complexity compromises its practical usability. In this paper, we present an accurate and efficient trust predictor *that is based on a basic Kalman filter. We discuss simulation results to demonstrate that the predictor is capable of capturing the natural disposition to trust of the user of the device, while being autonomic and light-weight.*

1 Introduction

Rapid advances in wireless networking and increased popularity of portable devices are quickly turning pervasive computing into a reality. It will not be long before the commercially exploitable potential of these technologies will be apparent, resulting in a huge number and variety of interconnected devices, services and information sources. Users of these devices will need support to decide who to interact with in this plethora of self-interested peers. We argue that a *human-tailored* trust management model could play a key role in the success of pervasive systems: integrated with a pervasive service discovery and selection framework, it would enable better informed decisions about what service provider to interact with in this potentially huge 'market'. This would improve the user's experience of the pervasive system, thus fostering its acceptance.

Various trust management models have been proposed in recent years that capture an increasingly wider spectrum of human trust facets (e.g., [1, 4, 12, 5]). Unfortunately, the degree of human trust they are able to represent comes along with a level of complexity that compromises their practical usability. In fact, these models depend on a large number of parameters that the user is required to set on her/his device; however, even an advanced user would have difficulties in understanding (let alone setting) them. Moreover, large amounts of data need to be locally kept and/or processed to improve the accuracy of trust prediction; however, pervasive computing devices are resource scarce (at least in terms of battery). The result is often a computational trust model that actually does not compute.

The realm of pervasive computing calls for novel trust models that are as *human-tailored* as possible, while being *autonomic* (in order to compute an accurate trust measure without the intervention of the user of the device) and *light-weight* (in order to minimise the overhead imposed on the device itself). In this paper, we propose a pervasive trust model that is capable of capturing many facets of human trust, while being both autonomic and light-weight. The model is grounded on the Kalman filter theory [10]: based on a set of observations (i.e., direct experiences), a trust prediction model is derived and used to foresee the state of the system. New observations are fed in by means of a set of recursive mathematical equations that can be efficiently computed in order to increase the accuracy of the prediction.

The reminder of the paper is structured as follows: Section 2 provides a brief introduction to the Kalman filter theory, and highlights the analogy between the filter and human trust. In Section 3 we describe how we have used the basic Kalman filter to build a trust predictor for the target scenario, and in Section 4 we discuss simulation results. Section 5 positions our work with respect to others in the field, and Section 6 concludes the paper.

2 Background

The Kalman filter is essentially a set of recursive mathematical equations that provide an optimal way to estimate the current state of a dynamic system, starting from observations that contain random errors. To ease presentation, let us consider a mono-dimensional system with state represented by vector $x \in \mathbb{R}^n$ ($n = 1$) and governed by equation:

$$x_{t+1} = x_t + V_t,\ t = 1, 2, \ldots \quad (1)$$

that is, the state of the system at time $t + 1$ depends on the state of the system at time t and a random process noise term V_t. Imagine we can make periodic observations y_t of the system, such that:

$$y_t = x_t + W_t,\ t = 1, 2, \ldots \quad (2)$$

that is, the observation depends on the current state of the system and a random measurement noise term W_t. The question we are trying to address is the following: how can we determine the best estimate of the state variable x, given our knowledge about the system behaviour and (noisy) measurements y? Under the assumptions that the process noise V_t is a white gaussian noise with covariance Q_t, the measurement noise W_t is a white gaussian noise with covariance R_t, and the two noises are not correlated, the Kalman filter provides an *optimal* prediction algorithm in that it minimises the estimation error. Even though these assumptions (which are necessary for optimality) rarely holds, yet the filter works well for many applications. The basic Kalman filter takes the following form:

$$x_{t+1} = x_t + \frac{\Omega_t}{\Omega_t + R_t} * (y_t - x_t), \quad (3)$$

$$\Omega_{t+1} = \Omega_t + Q_t - \frac{\Omega_t^2}{\Omega_t + R_t} \quad (4)$$

with $\Omega_0 = E[(y_0 - x_0)^2]$. Given the (noisy) observation y_t of the state of the system at time t, and the prediction x_t computed after the $t - 1^{th}$ observation, the next best estimate of the state of the system is equal to the previous state plus a term that is proportional to the distance between the last observation and the prediction. Intuitively, the higher the measurement noise R_t, the lower the impact of the observation in computing the next estimate. Viceversa, the higher the process noise Q_t, the higher the importance assigned to the latest observation. The Kalman equations thus both project the current state forward in time (*prediction*) and incorporate new measurements in order to improve the estimate (*correction*). A more accurate description of the Kalman filter can be found in [3].

Let us now re-phrase the whole problem in terms of trust for pervasive systems. Client device A is willing to assess the trustworthiness of server device B before deciding whether to interact with (e.g., request a service from) B or not. It does so by means of a basic Kalman filter that predicts B's trustworthiness at time $t + 1$ based on t previous observations of B's behaviour (direct experiences). After each observation, the filter updates its inner state, so to make a more accurate estimate the next time. The Kalman filter is particularly appealing to pervasive systems as it is extremely light-weight, both in terms of memory requirements (only one vector x_t is maintained per service provider, thus collapsing an arbitrary long history of interactions onto a single tuple) and computational load (the recursive Kalman equations can be efficiently computed, adding a negligible overhead on the device). Moreover, even in its simplest formulation, the Kalman filter is able to capture many facets of human trust: it makes a prediction based on an arbitrary long history of interactions; it implicitly represents the concept of confidence in the trust prediction, as the more frequently A interacts with B, the more quickly the filter stabilises and reduces the distance between prediction and actual state; finally, it enables simple yet effective modeling of the subjective nature of trust by means of the measurement and system errors. In particular, we use R_t to model *cautiousness* of behaviours (i.e., high values to R_t indicate a cautious attitude, with higher importance assigned to history than to the latest measure); we use Q_t to model *confidence* instead (i.e., a confident behaviour will assign high values to Q_t, thus giving higher importance to the latest experience).

3 Pervasive Trust Model

Given our rationale to use Kalman filters in the pervasive trust scenario, we now formulate rigorously the trust prediction problem in terms of a state space model, and illustrate how to use the basic Kalman filter to derive an autonomic, light-weight, and yet accurate trust predictor.

The first step is to define trust in terms of *observable* state variables that a device can *autonomically* measure without the user input. To achieve this goal, we must analyse the target scenario. A pervasive system can be thought of as a market, where a potentially large number of devices (the servers) are offering their services to an even larger number of users (the clients). For any given service s, there will be many different providers that a client may rely on, at any given time and location. Each provider will advertise service s by promising a certain quality of service, which we can assume is represented in terms of attribute/value pairs (a_i, v_i). We call this advertisement a *service specification*. The client cannot solely rely on the information contained in the various service specifications to decide which provider to rely on, as some providers will inflate their specifications, to secure the client's request. To make a well-informed decision about whether to interact with B or not, A would like to *predict* B's actual behaviour, based on the

previously observed interactions. Because of the high dynamicity of context in pervasive settings, it is not feasible to train a Kalman filter so to predict the actual values of the attributes listed in the service specification. What we can do instead is to predict the *discrepancy* between the advertised attribute values and what the client's measurements will be, with respect to the attribute range of values (i.e., we predict the variation from the service specification in percentage terms). We can then define A's trust in B in terms of these discrepancies, in a way that trust decreases as a result of high discrepancies, and viceversa.

More formally, let us indicate with $a_i, i \in [1, n]$ the observable attributes that constitute the specification of a certain category of services, with $p_{[a_i,B]_t}$ the value of attribute a_i promised by server B for interaction t, and with $m_{[a_i,A]_t}$ the value of attribute a_i measured by client A for interaction t. The state of the system for service s at time t can be described by $x_t^s \in \mathbb{R}^n$, $x_t^s = [x_{1_t}, \ldots, x_{n_t}]^T$, with:

$$x_{i_t} = \begin{cases} \frac{|m_{[a_i,A]_t} - p_{[a_i,B]_t}|}{\max\{m_{[a_i,A]_t}, p_{[a_i,B]_t}\}} & \text{if } (m_{[a_i,A]_t} < p_{[a_i,B]_t} \wedge a_i \text{ is IU}) \\ & \text{or } (m_{[a_i,A]_t} > p_{[a_i,B]_t} \wedge a_i \text{ is DU}), \\ 0 \quad \text{otherwise.} \end{cases} \tag{5}$$

Let us analyse the above state variable definition[1]. The inner state of the system x_t^s is composed of the absolute values of the differences between promised and observed values of service attributes, weighted by the range of values (i.e., $x_{i_t} \in [0, 1]$). These differences are computed in different ways depending on the nature of the attribute in examination. More precisely, from the client's perspective, each attribute a_i belongs to one of these two categories: *Increasing Utility* (IU), that is, the higher the measured value of the attribute, the higher the utility experienced by the client; and *Decreasing Utility* (DU), that is, the higher the measured value of the attribute, the lower the utility. Attributes are known to belong to one of these two classes based on the ontology in use. For example, bandwidth is an IU attribute (the higher the experienced bandwidth, the better for the client), while service time is a DU attribute (the higher the experienced service time, the worse for the client). For IU attributes (e.g., bandwidth), A's trust in B should be negatively affected if the experienced value $m_{[a_i,A]_t}$ is lower than the promised value $p_{[a_i,B]_t}$; for DU attributes (e.g., service time), A's trust in B should be negatively affected if the experienced value $m_{[a_i,A]_t}$ is higher the promised value $p_{[a_i,B]_t}$ instead. The discrepancy predictor we are trying to build for state variable x_t^s must thus be able to accurately estimate those discrepancies that have a *negative* impact on trust (equation 5 - first case). Discrepancies may also result in higher client's utility (e.g., higher available bandwidth than the promised amount); in this case, we simply consider the service specification as being fulfilled, and thus set the discrepancy to 0 (equation 5 - second case).

The server trustworthiness can then be described by state variable $l_t^s \in \mathbb{R}^n$, $l_t^s = [l_{1_t}, \ldots, l_{n_t}]^T$, with $l_{i_t} = 1 - x_{i_t}$, $l_{i_t} \in [0, 1]$. The higher the discrepancy x_t^s with negative impact on A's utility in intercourse t, the lower the experienced trust l_t^s, and viceversa. For a discrepancy with positive impact on A's utility, the experienced trust in intercourse t is maximum (i.e., 1). By using state variable l_t^s in equations 1 and 2, we obtain a formal representation of the trust prediction problem using a state space model. The basic Kalman filter, illustrated in equations 3 and 4, can now be used to make an accurate *prediction* of how much the user's experience of a service will deviate from what the service provider has promised (described by state variable x_{t+1}^s) and, consequently, of how much the server can be considered trustworthy (described by state variable l_{t+1}^s). At any time t, the predicted trustworthiness l_t^s of service providers can be used by a client device to make better informed decisions about who to interact with.

For example, let us imagine Alice being an independent financial adviser. As part of her work, she visits various clients' sites daily; while not in the office, she must still be able to access financial information using a wireless networked device (e.g., a PDA). Pervasive services she requires include: (1) access to historical information of various financial products; (2) real-time updates of stock quotes; (3) financial transactions (buy/sell). Several providers exist that can offer these services with different Quality-of-Service (QoS). Let us suppose that the QoS of each of these services is measured using a single attribute, being respectively: (1) a_1 =bandwidth availability (the higher the bandwidth, the quicker the download of historical information); (2) a_2 =network delay (the smaller the delay, the more quickly stock quote updates are received); (3) a_3 =reliability, measured as the percentage of transactions committed successfully (the higher the percentage of successful transaction, the more reliable the service that is provided). Note that all these attributes can be autonomically computed by Alice's device, without any user input. In order to decide which service provider B is best to use, Alice's device runs a service discovery and selection mechanism that has been enhanced with our trust predictor. The very first time Alice has to choose what provider to rely on (time $t = 0$), the predicted discrepancy x_{i_0}, for each attribute a_i, is solely based on her natural disposition to trust with respect to the on-going service s. We assume that Alice chooses a profile that describes her disposition; this is the *only* user input required, and can be as simple as a binary choice between 'tendency to trust' and 'tendency not to trust'. Based on this disposition, the initial values of x_{i_0} are set and the corresponding trust expectations $l_{0_i} = 1 - x_{i_0}$ computed. At any point

[1] In this description of the problem, we are making the assumption that the state variables, that is, the service attributes, are linearly independent.

in time t, Alice expects a service delivery from provider B that is worse than the advertised quality of an amount proportional to the predicted discrepancies x_{i_t}. More precisely, for each advertised attribute value $p_{[a_i,B]_t}$, Alice expects to perceive $p_{[a_i,B]_t} \pm p_{[a_i,B]_t} * x_{i_t}$ ($+$ is used for DU attributes, while $-$ is used for IU attributes); using trust levels $l_{i_t} = 1 - x_{i_t}$, the client expects $p_{[a_i,B]_t} * (2 - l_{i_t})$ (for DU attributes), and $p_{[a_i,B]_t} * l_{i_t}$ (for IU attributes), that is, the predicted quality of service is proportional to the client's trust in the server. Based on this prediction, Alice decides whether to interact with B or not (this is an application-specific choice and it may depend, for example, on whether the expected values overcome a certain threshold); in case of interaction, Alice's device A autonomically measures $m_{[a_i,A]_t}$ (i.e., the perceived value of each service attribute), and then computes y_t^s (i.e., the actual measure of the discrepancy for intercourse t). This information is then fed into the filter according to equation 3, so that next time Alice has to reason about B's trustworthiness, a better prediction of the deviation in B's behaviour (x_{t+1}^s), and consequently of B's trustworthiness (l_{t+1}^s), will be available. In the next section, we will demonstrate that the predictor quickly converges (i.e., it predicts the correct discrepancy) in a short number of interactions, thus making the initial guess x_0^s not critical.

Besides individual levels of trust l_{i_t}, a single value $\tau_{s,t} \in [0, 1]$, that expresses the overall level of trustworthiness of a service provider, can be computed by composing the utilities derived from the individual attributes [11]:

$$\tau_{s,t} = \frac{\sum_{i=1}^{n} w_i * l_{i_t}}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} w_i * (1 - x_{i_t})}{\sum_{i=1}^{n} w_i},$$

w_i being the weights given to the service attributes. As shown, $\tau_{s,t}$ approaches 0 for untrustworthy service providers (i.e., providers that have repeatedly inflated their service specifications), while it approaches 1 for providers that have realistically estimated the quality of their services. This overall trust level is useful, for example, in reputation systems to disseminate B's reputation as $\tau_{s,t}(B)$.

We have so far dwelled on the autonomic nature of our trust prediction model. We now analyse how the predictor captures the natural disposition to trust of the user of the device and uses it in order to 'correct' the filter and compute 'better' predictions when new observations are available. As shown in formulae 3 and 4, the corrective-predictive behaviour of the filter depends on the values of parameters R_t and Q_t. By tuning the values of these parameters, we can assign a different weight to the new measure (direct experience), with respect to the current state of the system (history of interactions). This allows us to capture the subjectivity of human trust within our autonomic model; for example, if a user selects a profile that describes her/him-self as a risk-averse being, parameter R_t will be set to a high value, so that higher relevance is given to the history with respect to the latest experience (i.e., cautious behaviour with slow change of opinion). Even better, this value may be set to vary with the number of interactions, so to fine tune the prediction during the system lifetime; for example, we may assign lower values of R_t at bootstrap (when no historical information is present), and then gradually increase them.

Note that, by varying the values of R_t and Q_t over time, we are able to capture, in a very simple yet effective way, another characteristic of human trust, that is, its dependency on time. The level of confidence in a trust prediction not only depends on the number of interactions occurred, but also on the frequency of interaction. Our Kalman predictor implicitly models the dependency on the number of interactions, but not on the frequency (we use parameter t simply as a counter of past interactions, with no reference to physical time). As such, it correctly models interactions that happen at regular intervals of time T, while it does not cater for confidence fading due to lack of intercourses for periods of time $n*T$, $n \in \mathbb{N}^+$. We can model the uncertainty that time brings in by varying the values of R_t and Q_t with respect to the frequency of interactions. Let us assume that A interacts with B every T time units; as long as A and B interact with this frequency, we can simply increase the value of R_t every m time units (so that the system stabilises). However, if the frequency of interaction falls well below T, we can start decreasing R_t again, so to give higher importance to freshly available information. A unique choice for the values of R_t and Q_t does not exist. However, we can use simulation to study how to set these parameters so that human trust is best mimicked. The study is simple and we will discuss some results we have already obtained in the following section.

4 Simulation Results

In this section, we study the behaviour of the predictor for different values of the parameters R_t and Q_t. As we made the assumption that all state variables are independent, we can, without loss of generality, consider a monodimensional state space problem ($x \in \mathbb{R}^1$). Figure 1 illustrates the behaviour of the discrepancy predictor x_t^s (top) and of the associated trust predictor $l_t^s = 1 - x_t^s$ (bottom) over 100 interactions when using fixed parameter values. We illustrate here a particular server behavioural model, characterised by long periods of trustworthy service delivery (e.g., the measured difference between promised and observed service time is fairly low), alternated by sporadic peaks of misbehaviours. The initial guess of the Kalman filter x_0 has been set to 0.5 (mid range value); the expected server trustworthiness, before any interaction takes place, is thus $l_0 = 1 - x_0 = 0.5$. We have set the error Q_t to 0.01 (i.e., low error in the process); different values of R_t are plotted. As shown (top chart), the lower the value of R_t,

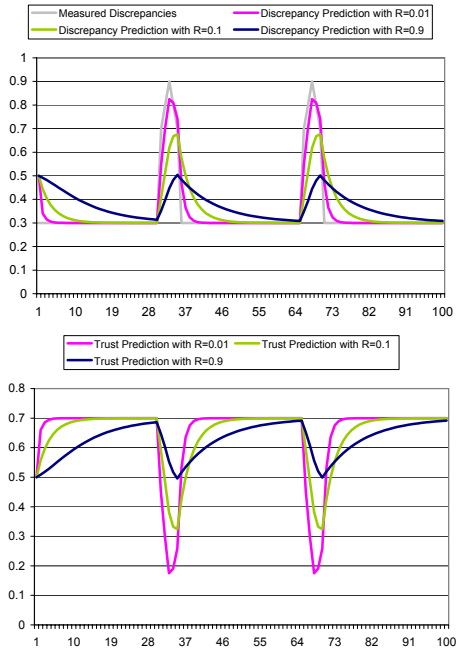

Figure 1. Predictors' Behaviour. Top: prediction of discrepancies. Bottom: prediction of trust.

the more quickly the predicted discrepancy converges to the actual measured data (i.e., the lower the error in the measure, the higher the weight assigned to the measure than to history), and viceversa. In terms of trust (bottom chart) the lower the value of R_t, the more quickly the client updates its trust opinion of the server (i.e., the server gains and looses trust mainly depending on the latest interaction). Viceversa, the higher the value of R_t (high error in the measure), the more importance is given to past interactions. The opposite behaviour can be observed when choosing different values of Q_t - the higher the noise in the process, the higher the importance assigned to the latest measure.

As previously said, the three parameters x_0, R_t and Q_t represent the only input required by our model. Different sets of parameters can be associated to different profiles from which the user chooses the one that best describes her/his disposition to trust. For example, a profile 'natural disposition to trust' could correspond to $x_0 = 0.2$ (small divergence between measure and actual data - thus high trust), $Q = 0.1$ (relative confidence in the process), and $R = 0.05$ (quick change of opinion following the experience); 'natural disposition to distrust' could correspond to $x_0 = 0.7$ (high distance between promised and measured behaviour - thus low trust), $Q = 0.01$ (high confidence in the prediction process), and $R = 0.5$ (cautious change of opinion). These are examples of static client profiles, with clients' reactions to intercourses that do not change in magnitude over time

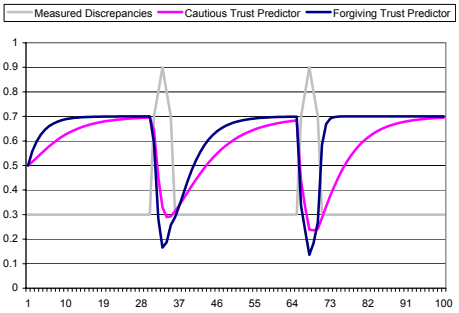

Figure 2. Advanced Trust Predictor.

(constant values of R_t and Q_t). More advanced profiles can be obtained by autonomically changing the values of R_t and Q_t with the *number* and *result* of experiences. We have built two trust predictors with varying values of R_t and compared their estimations with respect to the same server behavioural model shown before. The results are plotted in Figure 2 (we have plotted the measured discrepancies as well, to better understand the behaviour of the predictor with respect to actual data). The Cautious Trust Predictor (CTP) is set so to give higher importance to history than to the latest interaction; viceversa, the Forgiving Trust Predictor (FTP) is more willing to change opinion based on the result of the latest interaction. As a result, trust increases more quickly for FTP than CTP during the first set of successful interactions. As soon as the server starts misbehaving (i.e., there is a big distance between the prediction and the measured value), both predictors autonomically lower their value of R_t, as a protection measure against malicious service providers; the weight of the last interaction over history in the prediction process increases, and consequently the predicted trust decreases. When the server starts behaving properly again, it takes longer to regain trust than when constant values of R_t where used (see Figure 1); note also that more interactions are needed when using CTP to rebuild trust than when using FTP, as expected. When the server starts misbehaving again, the same corrective measure is taken; this time, R_t is set to even lower values, so that the predicted trust decreases more deeply and more quickly (as this is the second time a misbehaviour occurs). However, forgiveness is quicker as well, as the history of good interactions is much longer (thus stronger) than it was when the first misbehaviour happened. Determining what client's behaviour is best (what values of R_t and Q_t should be chosen) highly depends on the situation (i.e., what is at stake in these transactions) and on the user (i.e., what is the natural user disposition). Simulation studies can be conducted at application development time to create a portfolio of profiles (with corresponding parameters values) from which a user can choose.

5 Related Work

Research in the area of trust modeling has gained momentum, and various approaches have been proposed in recent years that fall in one of these two categories: formal trust models and computational trust models.

Formal trust models aim at providing a rigorous framework to represent trust and analyse its dynamics. While being grounded on solid mathematical theory, these approaches often lack practical relevance. For example, in [9] an opinion model based on subjective logic is discussed that can assign trust values in the face of uncertainty; however, the approach does not describe how to compute these values. In [7], a formal model for trust formation, evolution and propagation is presented; however, the algorithms for dynamically re-evaluating trust are not provided. Similar considerations hold for the formal trust models described in [2] (based on probability theory), [8] (founded on set theory), [13] (based on lattice, denotational semantics and fixed point theory) and [6] (founded on fuzzy logic).

Computational trust models focus more on modeling various facets of human trust with machine understandable structures that can be updated by means of well-defined algorithms. For example, [1] defines algorithms to combine direct experiences with recommendations; [4] discusses a model that makes explicit the distinction between trust and knowledge; [12] introduces the notion of 'recommendation reputation' (i.e., peers are judged based on the recommendations they have given in the past); [5] adds customisable functions to tailor the framework to the individual disposition to trust of the user of the device. A common limitation of these approaches is that the degree of subjectivity they capture comes at the expense of usability (the amount of user input required to customise the system is overbearing) and performance (the memory and processing overhead cannot be sustained by portable devices).

We believe our trust predictor makes a significant contribution to the field of trusted autonomic computing, as it brings together thorough mathematical results to derive a truly *computational*, human-tailored trust model.

6 Conclusion

In this paper, we have described an autonomic and lightweight computational trust model for pervasive systems based on a Kalman filter. When a service delivery occurs, a number of attributes describing the quality of the service are measured and compared against the promised values; these discrepancies are used to train a Kalman filter to assess the trustworthiness of a service provider. Simulation studies have proven the accuracy of the model on synthesised data; we now intend to analyse its behaviour on real data. We are also refining our trust predictor to improve both its accuracy and its human-facet: in terms of accuracy, we are studying more complex filters (i.e., Kalman filters with trend and seasonal components) that provide better predictions, without compromising on autonomicity; in terms of human trust, we are incorporating recommendations in the filter training, so to make accurate predictions even in situations where direct experiences are lacking.

References

[1] A. Abdul-Rahman and S. Hailes. Using Recommendations for Managing Trust in Distributed Systems. In *Proc. of IEEE Malaysia International Conference on Communication (MICC'97)*, Kuala Lumpur, Malaysia, Nov. 1997.

[2] T. Beth, M. Borcherding, and B. Klein. Valuation of Trust in Open Networks. In *Proc. of the 3^{rd} European Symposium on Research in Computer Security (ESORICS '94)*, pages 3–18, Brighton, UK, Nov. 1994.

[3] P. Brockwell and R. Davis. *Introduction to Time Series and Forecasting*. Springer, 1996.

[4] V. Cahill et. al. Using Trust for Secure Collaboration in Uncertain Environments. *IEEE Pervasive Computing Mobile And Ubiquitous Computing*, 2(3):52–61, Aug. 2003.

[5] L. Capra. Engineering Human Trust in Mobile System Collaborations. In *Proc. of the 12^{th} International Symposium on the Foundations of Software Engineering (SIGSOFT 2004/FSE-12)*, pages 107–116, Newport Beach, CA, USA, Nov. 2004. ACM Press.

[6] J. Carbo, J. Garcìa, and J. Molina. Subjective Trust Inferred by Kalman Filtering vs. a Fuzzy Reputation. In *Proc. of 1^{st} International Workshop on Conceptual Modeling for Agents (CoMoA 2004)*, volume 3289 of *LNCS*, pages 496–505, Shanghai, China, Nov. 2004.

[7] M. Carbone, M. Nielsen, and V. Sassone. A Formal Model for Trust in Dynamic Networks. In *Proc. of First International Conference on Software Engineering and Formal Methods (SEFM'03)*, pages 54–63, Brisbane, Australia, Sept. 2003.

[8] C. Jonker and J. Treur. Formal Analysis of Models for the Dynamics of Trust Based on Experiences. In *MAAMAW '99: Proceedings of the 9^{th} European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, pages 221–231, London, UK, 1999. Springer-Verlag.

[9] A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.

[10] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[11] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preference and Value Tradeoffs*. Wiley & Sons, 1976.

[12] J. Liu and V. Issarny. Enhanced Reputation Mechanism for Mobile Ad Hoc Networks. In *Proc. of the 2^{nd} International Conference on Trust Management (iTrust)*, volume 2995, pages 48–62, Oxford, UK, Mar. 2004. LNCS.

[13] S. Weeks. Understanding Trust Management Systems. In *Proc. IEEE Symposium on Security and Privacy*, pages 94–105, Oakland, CA, May 2001.