Risk Probability Estimating Based on Clustering

[1] Yong Chen, [2] Christian Damsgaard Jensen, [1] Elizabeth Gray, [1] Jean-Marc Seigneur

Ubiquitous computing environments are highly dynamic, with new unforeseen circumstances and constantly changing environments, which introduces new risks that cannot be assessed through traditional means of risk analysis. Mobile entities in a ubiquitous computing environment require the ability to perform an autonomous assessment of the risk incurred by a specific interaction with another entity in a given context. This assessment will allow a mobile entity to decide whether sufficient evidence exists to mitigate the risk and allow the interaction to proceed. Such evidence might include records of prior experiences, recommendations from a trusted entity or the reputation of the other entity.

In this paper we propose a dynamic mechanism for estimating the risk probability of a certain interaction in a given environment using hybrid neural networks. We argue that traditional risk assessment models from the insurance industry do not directly apply to ubiquitous computing environments. Instead, we propose a dynamic mechanism for risk assessment, which is based on pattern matching, classification and prediction procedures. This mechanism uses an estimator of risk probability, which is based on the automatic clustering of defining features of the environment and the other entity, which helps avoid subjective judgments as much as possible.

Risk assessment, Risk probability, Cluster, Neural network, ART, BP

I. INTRODUCTION

In a global ubiquitous computing infrastructure, the number of autonomous interacting entities could be millions or even billions. It is therefore not possible to rely on a specific security infrastructure such as certificate authorities and authorization servers. The interactions between entities are very much like those faced by human beings confronted with unexpected or unknown interaction with each other. Human society has developed the concept of trust to overcome initial suspicion and gradually evolve privileges. The basic trust lifecycle is showed in Figure 1.

Recently, there has been an increased interest in security mechanisms based on the human notions of trust [2, 3, 4]. Entities in this infrastructure are both autonomous and mobile and must be capable of dealing with unforeseen circumstances ranging from unexpected interactions with other unknown entities to disconnected operation.

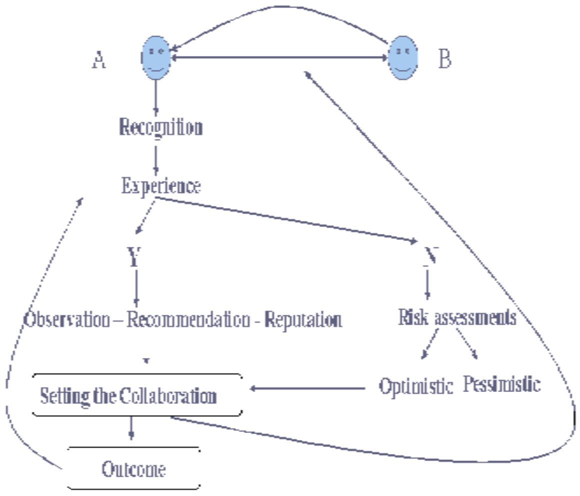

Figure1: Trust Lifecycle [1]

It has been recognized that an element of risk is part of the context of trust and a fundamental question is to characterize the extent to which risks are associated with the privileges that may be assigned to an unknown entity, so that the level of trust in an entity can be used to establish its level of privileges [5]. No one today can satisfactorily answer this question for computer-relative risks.

Risk is the possibility of something adverse happening, and risk management is the process of assessing risk, taking steps to reduce risk to an acceptable level and maintaining that level of risk. In her paper [6], Dr. Sharon Fletcher asserts that risk management has gone through two generations already, and that it needs to enter its third. Until now, there have been quite a few tools and methods proposed, but most of them still view risk assessment as a fairly static procedure [7].

Risk is commonly defined as the hazard level combined with [8]

- The likelihood of the hazard leading to an accident
- Hazard exposure or duration (latency).

Assessing risk is to assess the above two components. The latter factor in risk assessment is easier to determine as long as the accident can be confirmed. At least, the biggest loss can be estimated for a certain accident. Estimating the likelihood of the hazard is a much more complex issue. The insurance industry has done a lot of work to model the probability distribution functions for different contexts.

In this paper, we only focus on estimating risk probability for a certain interaction, i.e., the likelihood of the hazard leading to an accident. Section 2 investigates risk assessment in ubiquitous computing and analyzes the risk assessment used in the insurance industry, as well as defining the principles of risk assessment. Section 3 describes our design of a mechanism for the estimation of risk probability in the ubiquitous computing environment. Section 4 concludes and presents future work.

II. RISK ASSESSMENT IN UBIQUITOUS COMPUTING

In order to investigate risk assessment in ubiquitous computing, it is necessary to review how risk assessment takes place when an interaction with another entity happens in a specific context. In general, risk assessment involves 5 separate steps: identification of the context, identification of the other entity, recollection of any similar interactions with similar entities in similar contexts, risk assessment based on the recollected prior experiences and retention of evidence about the interaction for use in future risk assessments.

Identification of context involves extracting features from the environment that are relevant to the current interaction. Different features will be important for different types of interactions, which means that the extracted set of features will depend on the application. The only restriction on the set of features is that we need a mechanism for comparing two different values for the same feature.

Identification of the other entity is to associate the entity with his previous behavior to a certain level of confidence, regardless of whether the entity is fully anonymous or authenticated. The foundation of computer security has traditionally been built up on authentication. With this view, for initial collaboration to be possible, it is first necessary to know the identity of the entity with whom the interaction occurs. However, we believe that it is sometimes sufficient to determine whether an entity has behaved correctly or not in a previous collaboration rather than to get a precise identity without information about the likely behavior of the entity. [9]

Recollection of similar interactions involves searching for data in similar contexts, including entity properties, situations, similar observations of the entity, reputations, any recommendations from a third party and so on.

Risk assessment based on the evidence provided by the recollection involves a patterns classification and matching. Historical data, as well as new input, is categorized into a distinct subclass of patterns-cluster whose members are more similar to each other than they are to other patterns. According to the similarity to a given patterns, an estimation of risk value is determined in response to the trained historical data for the current interaction. Fresh input may be categorized into the cluster that matches most closely, leading to pessimistic estimation and generation of a new pattern after this result is observed.

Retention of evidence for future use in risk assessment involves recording the outcome of the interaction along with any recognizing features used to identify the context and the other entity.

From this, we find that assessing risk is a procedure that collects historical data to predict the outcome of the current situation. It is important to retain, classify, and remember the historical data and use new data to update historical data. In fact, traditional risk assessment in the insurance industry uses historical claims data to model the probability density function (PDF) of the frequency of claims and of loss. This is essentially a very similar procedure.

We reviewed the basic insurance models to determine if they provide techniques for managing risks in ubiquitous computing [10]. Essentially, in these models PDF must be confirmed. There are mainly two major classes of estimators, parametric and nonparametric. In parametric estimation, it is assumed that the distribution belongs to a known parametric family. In non-parametric estimation, the true distribution of the observations is not assumed to belong to a known parametric family, and the distribution is represented solely by the empirical distribution. For parametric estimation, it is very difficult to know what kind of distribution underlies some interaction. It is possible that there is no statistical distribution underlying an interaction. Moreover, selecting a distribution for a certain interaction is dependent on a third party or a server. It therefore may seem attractive to apply non-parametric estimation to risk assessment. However, non-parametric estimation requires large amounts of data to provide accurate results and cannot respond automatically to new situations. Thus, non-parametric estimator cannot be directly applied to the risk assessment in ubiquitous computing, although it is still very attractive and promising if it can dynamically self-update faced with completely fresh context.

Based on the analysis, we define the following characteristics of assessing risk in ubiquitous computing:

- Decentralized environment independent of special server.
- Retention of historical and current data for future use.
- Reasonable classification of data.
- Automatic risk assessment according to historical and current data.
- Ability to self-update in response to completely fresh dataset.

There is no doubt that risk probability estimation is the core problem in risk assessment in this type of environment. In the next section, we propose such an estimator for risk probability.

III. ESTIMATING THE RISK PROBABILITY

The probability of an unexpected result in a given interaction, i.e. the likelihood of that event occurring, represents the risk. For a given history of interactions, we know the results and can determine if an event is unexpected or not. The difficulty in estimating the risk probability lies in how to predict risk from the current context of the historical data. Estimating risk probability is one of the components in risk assessment, and all of the principles for risk assessment should apply when estimating the risk probability.

In our risk probability estimator, a feature vector represents an interaction. The feature vector is application-specific. Let us take, for example, a collaborative game such as playing Blackjack over a mobile ad hoc network. Suppose Alice want to join a game in which Bob is the dealer. To avoid the risk of losing money due to cheating, spoofing, or collusion in the game, Bob needs to estimate the risk of admitting Alice to his game. The feature vector in this case may contain elements such as: Alice, Time, Location, Payment Record, Win Rate, Playing Strategy, etc. For some non-continuous value of these elements, we need to use continuous numbers to denote them, such that the associated feature vector could be a computable one.

When an entity meets another entity in some interaction, such as in the above example, he tries to search for previous similar interactions, e.g. Bob would search for a record of previous interactions playing Blackjack with Alice. This is a pattern-matching procedure. Obviously, not all historical interaction details will be relevant. For example, details of Bob's interaction with Alice in a non-gaming interaction may prove less relevant when he is trying to determine the risk of collaborating with her in a Blackjack game. Therefore, clustering historical interactions according to patterns is necessary as a kind of pre-processing in which distinct subclasses of patterns are discovered whose members are more similar to each other than they are to other patterns. Pre-processing of historical interaction details saves searching time and improves the precision of pattern matching.

For each cluster, the rate between the number of unexpected results of some interactions and the numbers of elements in the cluster is defined as the Average Loss Rate (ALR). Each element itself has a risk probability associated with the ALR. We propose a hybrid neural network to implement this procedure of estimating risk probability.

There are three components in our risk probability estimator. The first is a clustering component. It takes a set of input vectors as input and gives a set of clusters as output, as well as a mapping of each input vector to a cluster. Input vectors that are close to each other according to a specific similarity measure should be mapped to the same cluster. Adaptive Resonance Theory (ART)[1] [11] implements the procedure. Besides some features of the clustering algorithm, ART remains adaptive in response to significant input, yet remains stable in response to irrelevant input. It can retain previously learned information while continuing to acquire new information.

The second component consists of a modification procedure that collects the output vectors from the first component and makes some modification to the vectors to derive input vectors for the third component. If we view the risk probability as the response of the feature vector, the commitment of Back Propagation (BP) neural network is to predict the risk probability using the feature vector. Neural networks often expose the implicit principle that is underlying a large dataset that might be difficult to be represented in any explicit way. To train the BP network, the risk probability of previous data associated with an unexpected result as well as ALR for some cluster is calculated here. This component is only needed for the training procedure.

The third component is the BP prediction procedure which trains the history vector with its risk probability and uses neural network parameters trained to predict the risk probability associated with new input of a certain cluster.

The diagram presented in Figure 2 shows the three components in the risk estimator.

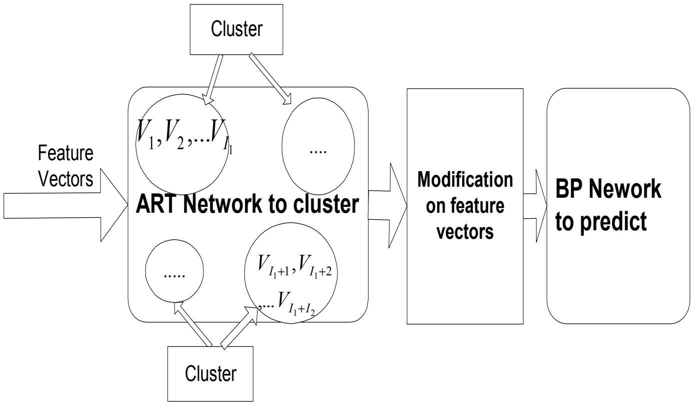
Cluster
Feature Vectors
$V_1, V_2, \ldots V_{I_1}$
....
ART Network to cluster
.....
$V_{I_1+1}, V_{I_1+2}, \ldots V_{I_1+I_2}$
Cluster
Modification on feature vectors
BP Nework to predict

Figure 2: Risk Estimator

Before calculating the risk probability of a new input, it is necessary to train the network with historical data. Training processing works as follows.

1. **Clustering Procedure**

- Abstract the features from the historical data to generate the features vectors;
- Feature vectors as the input vectors for ART neural network. Start clustering procedure;
- After finishing clustering, select any one cluster set of the clustering result, for example:

  $I_k = \{V_1, V_2, \ldots V_k\}$,

  Here $V_i = (v_{1i}, v_{2i}, \ldots, v_{mi})^T, i = 1,2\ldots, k$

  I_k is a cluster, V_i is the feature vector and K is the number of the features. From next step to the end, calculate risk probabilities for every vector in this cluster. Do the same for other clusters.

[1] We use ART2.

2. **Modification procedure**

- Calculate ALR:

$$ALR = \frac{\sum_{i \in I_k} E(V_i)}{|I_k|}$$

Here $|I_k|$ means the number of elements in I_k $E(V_i) = 1$, if the result of interaction associated with V_i is unexpected. Otherwise, $E(V_i) = 0$.

- Calculate similarity rate of between any vector associated with unexpected result and the average vector

$$Sim(V_i, \overline{V}) = \frac{\sum_{n=1}^{m} v_{in} \times \overline{v_n}}{\sqrt{(\sum_{n=1}^{m} v_{in}^2)(\sum_{n=1}^{m} \overline{v_n}^2)}}$$

Here $\overline{V} = \frac{\sum_{i \in I_k} V_i}{|I_k|}$

- Calculate risk probability (RP) for every feature vector associated with unexpected result

$$RP_i = Sim(V_i, \overline{V}) \times ALR$$

3. **BP prediction procedure**

- Look at the pairs
  (V_i, RP_i) $i = 1,2,...k$
  as pair (input vector, response) with proper network behavior for BP neural network.
  (RP_i is the response of vector V_i)
- Start BP training processing to confirm parameters of network.

Then we could use this network to predict or test a new input. Test processing is as follows.

1. **Clustering Procedure**

- Abstract the features from the current data to create a new vector V_l;
- New feature vector as the input vector for ART neural network. Start clustering procedure;
  After finishing clustering, V_l should be in one of clusters, such as $V_l \in I_k$.

2. **BP prediction procedure**

- Vector V_l in pair (V_l, RP_i) as input vector for BP neural network which has been trained by the proper vectors;
- Start BP processing and get RP_l as risk probability of current interaction.

This estimator only focuses on assessing the risk probability of a certain interaction. A monitor system collects relevant evidences, observing the behavior of entities and results of the interaction. Then it updates an evidence store with interaction details for future use.

IV. Conclusion and future work

Most of traditional risk assessments follow a fairly static procedure and cannot satisfy the requirements of the ubiquitous computing environment. A more flexible, dynamic risk assessment, without subjective evaluation, is needed. Models from the insurance industry fail to cope with an unknown interaction as PDF is very difficult to confirm and serve to a third party dependent. PDF models could not apply to ubiquitous computing directly.

In this paper we described the principles of risk assessment and proposed an estimator of risk probability that is the core component in risk assessment. This estimator is based on clustering which is implemented by a neural network. We are developing a framework in which this risk probability estimator is embedded. This project is still in progress. There are many parameters to confirm for the neural networks before evaluation of the project result will be presented. Our aim is not to give a universal solution estimating risk probability. However, for some scenarios we believe it is a novel way of estimation that can respond to new interactions dynamically and avoid subjective assessment in the ubiquitous computing environment.

V. Acknowledgements

The authors thank Professor Padraig Cunningham and his students for their helpful comments and suggestions about neural network.

This work is funded by the SECURE: Secure Environments for Collaboration among Ubiquitous Roaming Entities (IST-2001-32486), a part of the EU FET Global Computing initiative.

VI. References

[1] Waleed Wagealla, Presentation in 2nd of SECURE workshop in Glasgow

[2] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized Trust Management," in Proceedings of the 1996 IEEE Symposium on Security and Privacy, pp. 164-173.

[3] Y. H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss, “REFEREE: Trust management for Web applications.” Computer Networks and ISDN Systems, 29(8–13): 953–964, Sept. 1997.

[4] S. Marsh, “Formalising Trust as a Computational Concept.”, Ph.D. Thesis, University of Stirling, 1994.

[5] Kevin J. Soo Hoo,"How Much Is Enough? A Risk-Management Approach to Computer Security", Working Paper, http://citeseer.nj.nec.com/505332.html

[6] S. Fletcher, R. Jansma, J. Lim, R. Halbgewaches, M. Murphy, G. Wyss, "Software system risk management and assurance", Proceedings of the 1995 New Security Paradigms Workshop, August 22-25, 1995, San Diego, CA

[7] Craft, R., Vandewart, R., Wyss, G., Funkhouser, D., "An Open Framework for Risk Management", Proceedings of the 1998 21st National information Systems security Conference.
http://csrc.nist.gov/nissc/1998/proceedings/paperE6.pdf

[8] B. Jean, D. Nathan, I. David, M. Ken, S. Brian, T. Andrain. "Definition of Risk Model", SECURE Deliverable.

[9] J.-M. Seigneur, S. Farrell, C. D. Jensen, E. Gray, and Y. Chen, "End-to-end Trust Starts with Recognition", Proceedings of the First International Conference on Security in Pervasive Computing, 2003

[10] O. Raz, M. Shaw, "Software risk management and insurance", Proceedings of the 23rd International Conference on Software Engineering
http://www.cs.virginia.edu/~sullivan/edser3/raz.pdf

[11] Stephen Grossberg, Editor. Neural Networks and Natural Intelligence. MIT Press, Cambridge, Mass., USA, 1988