Design and Implementation of User Authentication using Dynamic Trust Model
A. MADHAVI¹, K. SHARATH KUMAR², DEEPTHI JANAGAMA³

¹PG Scholar, Dept of CSE, Sphoorthy Engineering College, Hyderabad, TS, India, Email: madhavireddy386@gmail.com.
²Associate Professor, Dept of CSE, Sphoorthy Engineering College, Hyderabad, TS, India.
³Assoc Prof & HOD, Dept of CSE, Sphoorthy Engineering College, Hyderabad, TS, India, Email: deepthi.janagama@gmail.com.

Abstract: Improvement of authorization process for protected information access by a large society of users in an open environment is an important problem in today’s Internet world. In this paper we propose a computational dynamic trust model for user authorization, rooted in findings from social science. Unlike most existing computational trust models, this model distinguishes trusting belief in integrity from that in capability in different contexts and accounts for subjectivity in the evaluation of a particular trustee by different trusters. Many Model studies were conducted to evaluate the presentation of the proposed integrity belief model with other trust models from the creative writing for different user behavior patterns. Results showed that the proposed model resulted in higher performance than other models especially in predicting the behavior of unbalanced users.

Keywords: Authorization, Human Factors, Security, Trust.

I. INTRODUCTION

Growing wealth of information available in online have made more secure by obtaining mechanisms on systems today’s world. The user authorization mechanisms in today’s environment are mostly centre on role-based access control (RBAC). It is a mechanism where it divides the authorization process in to the role-permission and user-role assignment. RBAC in modern systems uses digital identity as facts about a user to allow access to resources which the user is allowed. On the other hand, holding evidence does not necessarily certify a user’s good behavior. For example, when a bank is deciding whether to issue a loan to a customer, it does not only required proof such as social security number and home address, but also checks the belief about the applicant, formed based on previous behavior. Such belief, which we call dynamic trusting belief, can be used to calculate the possibility that a user will not perform risky actions. In this effort, we propose a computational dynamic trust model for user authorization. Mechanisms for building trusting belief by means of the direct experience which we can also call first-hand information as well as recommendation and reputation process which is also called as second-hand information are integrated in this model. The hand-outs of the model are:

1. The model is embedded in findings from social science i.e. it provides automated trust management that mimics trusting behaviors in the public, bringing trust computation for the society closer to estimate of trust in the real world.
2. Dissimilar to other trust models, the proposed model will have records for different types of trust. Particularly, this model distinguishes trusting belief in integrity from other models.
3. The proposed model takes into consideration about the prejudice of trust ratings by different entities, and set up a mechanism to take away the impact of subjectivity in reputation aggregation. Observed evaluation supports that the difference between competence and integrity trust is necessary in decision-making.

Distinguishing between integrity and competence permits the model to make more informed and fine-grained authorization decisions in different circumstances. Let us consider some examples:

1. Consider an example of real estate consultancy site, competence consists of elements such as finding the best plot area, the best construction, the Interior facilities etc., whereas integrity trust is based on factors like whether the site puts fraudulent charges on the customer. In a context where better deals are valued higher than the potential fraud risks, an agency with lower integrity trust could be preferred due to higher competence.
2. Consider an online site which is providing seasonal offers for customers to attract, the capability trust of a seller can be determined by how fast the seller ships the product or product quality etc., each being a different competence type. The integrity trust can be determined by whether he/she sells buyers’ information to other parties without buyer permission. In the case of an urgent purchase, a seller with low integrity trust can be allowed if he/she has high competence trust.
3. In support of a web service, the competence trust can include factors such as response time, quality of results etc., whereas integrity trust can depend on whether the service outsources requests to untrusted parties. Tentative evaluation of the proposed integrity belief model in a simulated environment of entities with different behavior patterns propose that the model is able to give better estimations of integrity trust behavior than other major trust computation models, especially in the case of trustees with changing behavior.
A. McKnight’s Trust Model
The social trust model, which guide the design of the computational model in this paper, was proposed by McKnight et al., after analyzing many papers across a wide range of disciplines. It has been validated via empirical study. This model describes five conceptual trust types: trusting behavior, trusting intention, trusting belief, institution-based trust, and disposition to trust. Trusting behavior is an action that increases a trustee’s risk or makes the trustee expose to the trustor. Trusting intention specifies that a trustor is willing to connect in trusting behaviors with the trustee. A trusting intention involves a trust decision and leads to a trusting behavior. Trusting belief is a trustor’s subjective faith in the fact that a trustee has attributes beneficial to the trustor. Two subtypes of institution-based trust are:
1. Structural pledge: The faith that structures organize promote positive outcomes. Structures include guarantees, policies, assurance etc.
2. Situational normality: The belief that the properly ordered environments facilitate success outcomes. Disposition to trust characterizes a thruster’s general propensity to depend on others across a broad spectrum of situations. Institution-based trust depends on situation. Disposition to trust is independent of situation and trustee. Trusting belief positively relates to trusting intention, which in turn results in the trusting behavior. Institution-based trust positively influence on trusting belief and trusting intention. Structural pledge is more related to trusting intention while situational normality affects both. Disposition to trust positively manipulate institution-based trust, trusting belief and trusting intention. Confidence in humanity impact trusting belief. Trusting stance influences trusting intention.

B. Computational Trust Models
The problem of launching and maintaining dynamic trust has fascinated much research hard work. One of the first efforts trying to celebrate trust in computer science was made by Marsh. The model introduced the concepts extensively used by other researchers such as context and situational trust. Many existing reputation models and security mechanisms rely on a social network structure. Pujol et al. propose an approach to mine reputation from the social network topology that encodes reputation information. Lang proposes a trust model for access control in P2P networks, based on the assumption of transitivity of trust in social networks, where a simple mathematical model based on fuzzy set membership is used to calculate the trustworthiness of each node in a trust graph symbolizing interactions between network nodes. FCTrust utilises the transaction density and similarity to calculate a measure of reliability of each recommender in a P2P network. Its main disadvantages are that it has to regain all transactions within a certain time period to estimate trust, which imposes a big performance penalty, and that it does not distinguish between recent and old transactions. Matt et al., introduced a method for modeling the trust of a given agent in a multiagent system by joining statistical information regarding the past behavior of the agent with the agent’s usual upcoming behavior.

Zhu et al., introduces a dynamic role based access control model for grid computing. The model determines authorization for a specific user based on its role, task and the context, where the authorization decision is updated dynamically by a monitoring module keeping track of user attributes, service attributes and the environment. Fan et al., proposed a similar trust model for grid computing, which focuses on the dynamic change of roles of services. Nagarajan et al., propose a security model for trusted platform based services based on evaluation of past evidence with an exponential time decay function. The model evaluates trust separately for each property of each component of a platform, similar to the consideration of competence trust in our proposed model. Although these approaches integrate context into trust computation, their application is limited to specific domains different from the one considered in our work. Walter et al., proposed a dynamic trust model for social networks, based on the concept of feedback centrality. The model, which enables computing trust between two disconnected nodes in the network through their neighbor nodes, is suitable for application to recommender systems.

III. EXISTING AND PROPOSED SYSTEMS
A. Existing System
The everyday increasing wealth of information available online has made secure information access mechanisms an indispensable part of information systems today. The mainstream research efforts for user authorization mechanisms in environments where a potential user’s permission set is not predefined, mostly focus on role-based access control (RBAC), which divides the authorization process into the role-permission and user-role assignment. RBAC in modern systems uses digital identity as evidence about a user to grant access to resources the user is entitled to.

B. Proposed System
We propose a computational dynamic trust model for user authorization. Mechanisms for building trusting belief using the first-hand (direct experience) as well as second-hand information (recommendation and reputation) are integrated into the model. The contributions of the model to computational trust literature are:
1. The model is rooted in findings from social science, i.e. it provides automated trust management that mimics trusting behaviors in the society, bringing trust computation For the digital world closer to the evaluation of trust in the real world.
2. Unlike other trust models in the literature, the proposed model accounts for different types of trust. Specifically, it distinguishes trusting belief in integrity from that in competence.
3. The model takes into account the subjectivity of trust ratings by different entities, and introduces a mechanism to eliminate the impact of subjectivity in reputation aggregation.
C. Implementation Modules

1. Context and Trusting Belief: Context: Trust is environment-specific. Both trusters concern and trustees' behavior vary from one situation to another. These situations are called contexts. A truster can specify the minimum trusting belief needed for a specific context. Direct experience information is maintained for each individual context to hasten belief updating. In this model, a truster has one integrity trust per trustee in all contexts. If a trustee disappoints a truster, the misbehavior lowers the trustee's integrity belief in him. For integrity trust, contexts do not need to be distinguished. Competence trust is context-dependent. The fact that Bob is an excellent professor does not support to trust him as a chief. A representation is devised to identify the competence type and level needed in a context.

2. Belief information and reputation Aggregation methods: Belief about a trustee's competence is context specific. A trustee's competence changes relatively slowly with time. Therefore, competence ratings assigned to her are viewed as samples drawn from a distribution with a steady mean and variance. Competence belief formation is formulated as a parameter estimation problem. Statistic methods are applied on the rating sequence to estimate the steady mean and variance, which are used as the belief value about the trustee's competence and the associated predictability.

Fig.1. System Architecture

IV. EXPERIMENTAL STUDY OF TRUST MODEL

Experimental studies were conducted to evaluate the integrity belief model proposed. The objective is to identify the suitable approaches for various scenarios (different types of trustees) and obtain guidelines to determine the appropriate values of parameters for the algorithms. Sections 4.1, 4.2 and 4.3 evaluate the approaches to build integrity belief based on direct experience. We also conducted experiments to evaluate the competence belief model introduced. The CRE-A and CRE-K methods were evaluated under different scenarios, with trustee behavior generated using a normal distribution. Experiments were conducted to compare the true mean and variance with the estimated mean and variance of competence reputation for different number of trusters. The relative error (RE) of CRE-A was found to be around 5 percent, and that of CRE-K was less than 3.5 percent, which are promising results. We omit detailed experiment results due to space constraints.

A. Study on Integrity Belief Building Methods

In this section, the BDES algorithm is compared with three other algorithms for five trustee behavior patterns. Algorithms compared. The algorithms compared are BDES, simple average, single exponential smoothing, and the time-weighted average, called REGRET, proposed. Let \( t_i \) denote the trusting belief after observing rating sequence \( r_1, r_2, \ldots, r_i \). Table 1 summarizes how \( t_i \) is evaluated under the four algorithms. \( w(k, i) \) in REGRET is a time dependent function giving higher values to ratings temporally close to \( r_i \). Table 2 shows the initial values of the parameters of BDES and SES. A function linearly decreasing with \((i - k)\) is used as \( w(k, i) \) in REGRET.

### TABLE 1. Algorithms to Build Integrity Belief

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Equation</th>
<th>Initial condition</th>
<th>Boundary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>( t_i = \frac{1}{i} \sum_{k=1}^{i} r_k )</td>
<td>( t_i = c )</td>
<td>( t_i = c )</td>
</tr>
<tr>
<td>SES</td>
<td>( t_i = \alpha r_i + (1 - \alpha) t_{i-1} )</td>
<td>( t_i = c )</td>
<td>( t_i = c )</td>
</tr>
<tr>
<td>Regret</td>
<td>( t_i = \frac{1}{\sum_{k=1}^{i} w(k, i)} \sum_{k=1}^{i} w(k, i) r_k )</td>
<td>( t_i = c )</td>
<td>( t_i = c )</td>
</tr>
<tr>
<td>BDES</td>
<td>Equation 32</td>
<td>( t_i = c )</td>
<td>( t_i = c )</td>
</tr>
</tbody>
</table>

### TABLE 2. Parameters Used in Experiments

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>( \alpha = 0.3 ) (initial value)</th>
<th>( w(k,i) ) is a function linearly decreasing with ((i-k))</th>
<th>( \alpha = 0.3 ) (initial value)</th>
<th>( \beta = 0.7 ) (initial value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SES</td>
<td>( \alpha = 0.3 ) (initial value)</td>
<td>( w(k,i) ) is a function linearly decreasing with ((i-k))</td>
<td>( \alpha = 0.3 ) (initial value)</td>
<td>( \beta = 0.7 ) (initial value)</td>
</tr>
</tbody>
</table>

Experiment setup. For the experiments discussed in Sections 4.2 and 4.3 below, trustee behavior was simulated using the five different integrity rating generation functions detailed below. A rating for trustee \( u \) generated by a behavior pattern function at time \( i \) is considered to be the true integrity rating submitted for \( u \) by a truster \( t \) at time point \( i \). For each behavior pattern experiment, a sequence of 100 ratings for each trustee were generated using the pattern function and the performances of the four integrity belief building methods listed above were evaluated by measuring the difference between the true rating and the rating output by the integrity belief method at each point in the sequence. Note that the identity of the trusters is not relevant in this case: The 100 ratings for a trustee could be submitted by a single truster or by 100 different trusters. Generate ratings based on trustee behavior patterns. The true values of integrity trusting belief about a trustee can be viewed as the range of a time
dependent function \( f(i) \). A pattern is a family of \( f(i) \)s with the same form. It is impossible and unnecessary to enumerate all possible forms of \( f(i) \)s. We are interested in meaningful patterns revealing the trend and intention of a trustee’s behavior. Five types of patterns, random trustee, stable trustee, trend trustee, jumping trustee and two-phase trustee, are identified and used in the experiments. The random pattern shows that the trustee’s behavior is variable. Prediction based on previous knowledge may not lead to good results. On the other hand, we can expect to precisely predict the next performance of a trustee with a stable pattern. The trend pattern captures the improving or deteriorating behavior pattern. The jumping and two-phase patterns indicate a sudden shapely change. Usually, they imply misbehaving of trust builders.

### TABLE 3. Trustee Behavior Patterns

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Form of ( f(i) )</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>( f(i)=U(0,1) ) for ( N(i) )</td>
<td>Fig. 5</td>
</tr>
<tr>
<td>Stable</td>
<td>( f(i)=c_i ) for ( N(i) )</td>
<td>Fig. 6</td>
</tr>
<tr>
<td>Trend</td>
<td>( f(i)=c_i + c_{i+1} ) for ( N(i) )</td>
<td>Fig. 7</td>
</tr>
<tr>
<td>Jumping</td>
<td>( f(i)=c_i ) if ( i \leq n_0 ) &lt;br&gt;( f(i)=c_{i+1} ) otherwise</td>
<td>Fig. 8</td>
</tr>
<tr>
<td>Two-phase</td>
<td>( f(i)=c_i ) if ( n_0 \leq i \leq n_1 ) &lt;br&gt;( f(i)=c_{i-n_0} ) if ( n_0 &lt; i &lt; n_1 ) &lt;br&gt;( f(i)=c_i ) if ( n_1 &lt; i )</td>
<td>Fig. 9</td>
</tr>
</tbody>
</table>

Table 3 shows the form of \( f(i) \) for each pattern. In this work, the independent variable of \( f(i) \) is the number of interactions. \( n_0 \) and \( n_1 \) are constants. Based on the behavior patterns, we can systematically evaluate a belief formation algorithm. The effectiveness of an algorithm in an environment is determined by (1) how the algorithm performs for each type of trustee, and (2) what is the distribution of trustees belonging to each type. In this section, we study the first issue. Algorithms are evaluated against the interaction sequences representing different trustee behaviors. Each interaction sequence is generated to reflect certain trustee behavior patterns. A trustee’s behavior is determined by her trustworthiness and is influenced by some unpredictable factors. Therefore, the ith rating is generated using (1). The ith rating falls into \( [f(i) - 0.1, f(i) + 0.1] \) with probability 90 percent. The interval is interpreted as the region where relative error is smaller than 10 percent:

\[
N(f(i), (0.1/1.645)^2)
\]  

(1)

### B. Distribution of Errors

The first set of experiments compares absolute error (AE) and relative error, as defined in (2a) and (2b) respectively, of the four algorithms. We choose this measurement because the purpose of evaluating \( t_i \) is to forecast \( r_{i+1} \), i.e., a good trust building algorithm shall output good predictions. Absolute and relative errors characterize how close one prediction is to the true value:

\[
AE = |t_i - r_{i+1}|
\]  

(2a)

\[
RE = |t_i - r_{i+1}|/r_{i+1}|
\]  

(2b)

We generate 100 ratings for each type of behavior pattern. The parameters are summarized in Table 4. Four algorithms are applied on each trustee. The absolute and relative error for each predication is computed. The distribution of errors generated by each algorithm is plotted using cumulative frequency figures.

### C. Results and Observations

A trustee with random behavior pattern. For a trustee who has the random behavior pattern, the next behavior has no relation to the previous behaviors. The rating can increase or decrease sharply at any time. Because the behavior of the trustee is completely unpredictable, none of the evaluated algorithms is able to provide a good prediction of how the next behavior will be. The Average, SES, and REGRET algorithms have almost the same performance in terms of absolute error, as shown in Fig. 2a. The Average algorithm performs slightly better than the other two. About 88 percent of its results have an absolute error less than 0.4, while the percentages of the SES and REGRET algorithms are 85 and 81 percent respectively. Nearly all results of these three algorithms have an absolute error less than 0.6. The BDES algorithm fails to achieve low error rate in this experiment. Only 70 percent of its results have an absolute error less than 0.4. The upper bound of the error is 0.8 instead of 0.6. Fig. 2b shows that all algorithms generate large relative errors. For the Average, SES, REGRET, and BDES algorithms, the percentages of the results that have a relative error less than 100 percent are respectively, 80, 78, 80, and 77 percent. The percentages of the results that have a relative error greater than 200 percent are 12, 14, 12, and 15 percent respectively. The Average and REGRET algorithms perform the best.
rating has a greater probability of being closer to the mean of
the previous ratings. All algorithms are able to produce very
good results in terms of absolute error and relative error as
shown in Figs. 3a and 3b. The REGRET algorithm performs
the best, slightly better than the BDES algorithm. For these
two algorithms, around 98 percent of the results have an
absolute error that is less than 0.2.

![Fig 3. (a) Absolute error, and (b) Relative error for a trustee with the stable behavior pattern.](image)

![Fig 4. (a) Absolute error, and (b) Relative error for a trustee with the trend behavior pattern.](image)

The corresponding percentage for the SES algorithm is 94
percent. The percentages of the ratings that have less than 0.1
absolute error are 86 percent for the Average and REGRET
algorithms, and 78 percent for the SES and BDES algorithms.
As shown in Fig. 3b, for every algorithm, almost all results
have a relative error less than 60 percent. Ninety percent of
the results generated by the Average and REGRET algorithms
have a relative error less than 20 percent. The corresponding
percentages for the SES and BDES algorithms are 82 and 87
percent respectively. A trustee with a trend behavior pattern.
For a trustee who has the trend behavior pattern, the behavior
becomes better and better (or worse and worse depending on
the trend) as time passes, i.e., the number of interactions
increases. The BDES algorithm outperforms the other
algorithms in terms of absolute and relative error when the
trustee has a trend behavior pattern. As shown in Fig. 4a, 88
percent of its results have an absolute error less than 0.2 and
83 percent of its results have an absolute error less than 0.1.
The corresponding percentages are 72 and 42 percent for the
Average algorithm, 89 and 67 percent for the SES algorithm,
and 87 and 50 percent for the REGRET algorithm. As shown
in Fig. 4b, although the percentage of the results that have less
than 40 percent relative error is around 98 percent for the
BDES,

Average and REGRET algorithms, only the BDES
algorithm is able to make 85 percent of its results having a
relative error less than 20 percent. The Average and REGRET
algorithms can achieve 42 and 61 percent respectively. 87
percent of the results obtained using the SES algorithm have
less than 40 percent relative error, 78 percent of the results
have less than 20 percent relative error. A trustee with
jumping behavior pattern. A trustee with the jumping
behavior pattern behaves as if he had the stable behavior
pattern, and suddenly changes his behaviors. Comparing the
results of this experiment with those of the previous two
experiments, we can see that the performance downgrades for
all, especially the Average and REGRET algorithms. As
shown in Fig. 5a, the BDES and SES algorithms still make,
respectively, 93 and 88 percent of the results have less than
0.2 absolute error. The corresponding percentage is 48
percent for the Average algorithm and 61 percent for the
REGRET algorithm. The upper bound of the absolute error is
0.6 for the BDES and SES, and 0.7 and 0.9 for the Average
and REGRET algorithms respectively. Fig. 5b shows that the
BDES algorithm has the highest percentage of the results with
less than 100 percent relative error, which is 96 percent. For
the Average, REGRET, and SES algorithms, the percentages
are, respectively, 63, 78 and 90 percent. From another
perspective, 90 percent of the results obtained using the
BDES algorithm have less than 47 percent relative error. The
same percentage of results obtained using the Average,
REGRET, and SES algorithms have a relative error less than
190, 170, and 100 percent respectively.

The BDES algorithm has the best performance among the
evaluated algorithms. A trustee with two-phase behavior
pattern. A trustee who has the two-phase behavior pattern has
similar behaviors as compared to the trustee with the jumping
behavior pattern, except that he changes his behaviors
gradually instead of suddenly. In terms of absolute errors, the
BDES and SES algorithms perform a little better, while the
Average and REGRET algorithms perform slightly worse as
compared to the jumping behavior pattern. As shown in Fig.
6a, the percentages of the results with less than 0.2 absolute
errors are 85, 90, 62 and 47 percent for the BDES, SES,
REGRET, and Average algorithms, respectively. The
percentages of the results with less than 0.1 absolute error are
82, 69, 37, and 37 percent correspondingly. Fig. 6b shows
that all algorithms have a better performance in terms of
relative errors compared to what they achieve in the previous
experiment. All the results obtained using the BDES
algorithm have less than 100 percent relative error. For the
SES, REGRET, and Average algorithms, 98, 83, and 71
percent of the results, respectively, have a relative error less
than 100 percent. Ninety percent of the results obtained from
the BDES, SES, REGRET, and Average algorithms have less
than 38, 58, 140, and 170 percent relative errors respectively.
D. Distribution of Mean Squared Errors

Previous experiments studied the errors generated by a single user per type. The second set of experiments explores the distribution of mean squared errors, as defined in (3).

\[
MSE = \frac{(t_i - r_i)^2}{n}
\]  

We generate the 100 trustees per behavior pattern using the parameters in Table 4. Please note the trustees are not the same. Four algorithms are applied on each trustee. For each trustee, the MSE generated by each algorithm is computed. For each type of trustee, the distribution of MSE generated by each algorithm is plotted using cumulative frequency figures. Also, average MSE is computed.

E. Results and Observations

When the trustee has the stable behavior pattern, the average algorithm outperforms the other algorithms in terms of MSE as shown in Fig 7a. Its MSEs range from 0.002 to 0.01.

TABLE 5. Average MSE for Each Behavior Pattern

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Random</th>
<th>Stable</th>
<th>Trend</th>
<th>Jumping</th>
<th>Two-Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0.086069</td>
<td>0.0037669</td>
<td>0.026811</td>
<td>0.072545</td>
<td>0.063554</td>
</tr>
<tr>
<td>SES</td>
<td>0.093301</td>
<td>0.007613</td>
<td>0.014744</td>
<td>0.021522</td>
<td>0.014272</td>
</tr>
<tr>
<td>REGRET</td>
<td>0.08932</td>
<td>0.0075901</td>
<td>0.016907</td>
<td>0.055423</td>
<td>0.046255</td>
</tr>
<tr>
<td>BDES</td>
<td>0.12558</td>
<td>0.0056795</td>
<td>0.0062433</td>
<td>0.012282</td>
<td>0.0074293</td>
</tr>
</tbody>
</table>

Around 99 percent of them are less than 0.005. The REGRET and SES algorithms have almost the same performance, which is worse than that of the BDES algorithm. Ninety percent of the MSEs of the REGRET and SES algorithms are less than 0.009, while the same percentage of MSEs of the BDES algorithm are less than 0.0078. As shown in Fig. 7b the BDES algorithm introduces larger MSE than the other three algorithms when the trustee has the random behavior pattern. The MSEs range from 0.07 to 0.20. Ninety percent of them are less than 0.16. The MSEs of the other three algorithms are very close. All of them are in the range of 0.06 to 0.12. Fig. 7c shows that the BDES performs better than the other algorithms in terms of introducing less MSE when the trustee has the trend behavior pattern. Its smallest MSE is about 0.005. Ninety nine percent of its MSEs are less than 0.012, which is the smallest one among all the MSEs introduced by the other algorithms. The Average algorithm has the worst performance. Its MSEs are in the range of 0.02 to 0.04, 94 percent of them are less than 0.03. The SES algorithm performs slightly better than the REGRET algorithm. Its MSEs range from 0.012 to 0.018, while 99 percent of the MSEs of REGRET are in the range of 0.014 to 0.02.

As shown in Figs. 7d and 7e, when the trustee has the jumping or two-phase behavior pattern, the BDES algorithm has much better performance than the other algorithms. Even its largest MSE is smaller than the smallest one introduced by the other algorithms. For a trustee with the jumping behavior pattern, the ranges of the MSEs are 0.009 to 0.017 for the BDES algorithm, 0.018 to 0.03 for the SES algorithm, 0.04 to 0.07 for the REGRET algorithm, and 0.06 to 0.09 for the Average algorithm. For a trustee with the two-phase behavior pattern...
pattern, the corresponding ranges are 0.004 to 0.001, 0.012 to 0.017, 0.04 to 0.06, and 0.05 to 0.08, respectively. Table 5 shows the average MSE for each behavior pattern obtained in the experiment. For a completely unpredictable trustee, i.e., the one with random behavior, no algorithm is able to provide practically useful integrity trusting belief. For a stable trustee, all algorithms can provide satisfactory information, with the Average algorithm being the best. When a trustee has the trend to change his behavior, e.g., the trend, jumping, and two-phase behavior pattern, only the BDES algorithm is able to catch this trend. The accuracy of the integrity trusting belief computed using the BDES algorithm is not affected much by the change of behavior, as seen in the last row.

V. CONCLUSION

In this paper we presented a dynamic computational trust model for user authorization. This model is ingrained in answering from social science, and is not restricted to trusting belief as most computational methods are. We presented a demonstration of context and functions that relate dissimilar contexts, enabling Building and testing initial competence trust. The proposed dynamic trust model enables automated trust management that mimics trusting behaviors in the public, such as selecting a community partner, forming an association, or choosing conciliation protocols in e-commerce. The formalization of trust helps in scheming algorithms to choose dependable resources in peer-to-peer systems, budding secure protocols for ad hoc networks and detecting unreliable agents in a virtual community. Experiments in a virtual trust environment show that the proposed integrity trust model carries out better than other major trust models in calculating the behavior of users whose behaviour transform based on certain patterns over time. The Future enhancement for this paper will be not only allocating dynamic computational trust model for user authorization but also distributing a Dynamic Trust Computation Model for safe Communication in Multi-Agent Systems.

VI. REFERENCES


Author’s Profile:

A Madhavi is doing her M.Tech in Computer Science Engineering department from Sphoorthy Engineering College, Hyderabad. Email: madhavireddy386@gmail.com

K. Sharath Kumar, received the Master of Technology degree (CSE). He has published 10 papers in various international journals. He has a life membership in ISTE and CSI. He is currently working as Associate Professor in the Department of CSE at Sphoorthy Engineering College, Hyderabad. Email: sravanthigopidi@gmail.com

Deepthi Janagama has done Master of Science in Texas A&M University Commerce, Texas, USA and received the Bachelor of Technology degree from Kamala Institute of Technology and Science. She is currently working as Associate Professor and Head of the Department of CSE with Sphoorthy Engineering College, Nadergul. Her interested subjects are Database management system, Data mining and Data Warehousing, Digital logic design, computer programming. Email: deepthi.janagama@gmail.com