Comparing Reputation Schemes for Detecting Malicious Nodes in Sensor Networks

Partha Mukherjee
University of Tulsa
partha-mukherjee@utulsa.edu

Sandip Sen
University of Tulsa
sandip@utulsa.edu

Abstract

Remotely deployed sensor networks are vulnerable to both physical and electronic security breaches. The sensor nodes, once compromised, can send erroneous data to the base station, thereby possibly compromising network effectiveness. We assume that sensor nodes are organized in a hierarchy and use offline neural network based learning technique to predict the data sensed at any node given the data reported by its siblings. This allows us to detect malicious nodes even when the siblings are not sensing data from the same distribution. The speed of detection of compromised nodes, however, critically depends on the mechanism used to update the reputation of the sensor nodes over time. We compare and contrast the relative strengths of a statistically grounded scheme and a reinforcement learning based scheme both for their robustness to noise and responsiveness to change in sensor behavior. We first extend an existing mechanism to improve detection capability for smaller errors. Next we analyze the influence of different discount factors, including unweighted, exponential, and linear discounts, on the trade-off between responsiveness and robustness. We both develop a theoretical analysis to understand the trade-off and perform experimental verification of our predictions by varying the patterns in sensed data.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Experimentation

Keywords

Sensors, Sensor Networks, Q-Learning, Beta-reputation

1. INTRODUCTION

Wireless sensor networks consist of spatially distributed autonomous devices using sensors to cooperatively monitor physical and environmental conditions specially in regions where human access is limited or carries potential risk. The size and cost of the sensor nodes impose limitations on the network capabilities such as speed and bandwidth conditions and they are susceptible to compromise by the intruders physically and electronically. A sensor network has at least one base station (BS), considered as the data sink, where all other nodes report their sensed and aggregated data.

In traditional data aggregation protocols, a tree hierarchy is built where the sensor nodes reside at the leaf level and the non-leaf nodes act as aggregator nodes, aggregating the data received from their child nodes. Now more than one node may be damaged or compromised by some unauthorized third party to alter the data it sensed or aggregated and would be transmitting upstream. If a large number of nodes become anomalous then the entire sensor network may be compromised and data integrity would be lost. To detect the faulty nodes and to protect the integrity of the data two reputation update mechanisms are proposed in [11]. One is based on reinforcement learning and the other is statistically grounded (RFSN scheme). These two approaches incrementally update the reputations of the sensor and aggregator nodes at each time step. Patterns in the sensed data are learnt using the neural net learning approach on sufficient data where each trained net predicts the data sensed by a sensor node. The data for training the neural nets is collected offline simulating environments where there is a wide variation of the data with time and over the spatial expanse of the network. The likelihood of a reported data being faulty is calculated as a function of the difference between the actual output it measured and value predicted by the neural net that corresponds to the node given the data reported by sibling nodes in the sensor net hierarchy. The error likelihoods are used to update the reputations of the nodes. Nodes are identified to be malicious if their reputations fall below a specified limit.

Whereas our previous work presented experimental results showing faster detection with Q-learning approach over basic RFSN technique, it did not provide detailed analysis of the underlying cause for such a difference. In this extension, we analyze the RFSN framework in detail and identify a basic weakness of the approach if used without time discounting of evidence. We then study the effect of different time discounting schemes, namely unweighted, linear and exponential discounting within the RFSN framework. These categories of the RFSN approach are derived by introducing a discount factor which could be viewed as the weights that are applied to the past experiences while computing the node reputations. We vary the initial data reporting interval with an error-free network and study the effects of the delay before malicious node are introduced on the performance of different reputation schemes. The evaluation criteria is how fast the nodes get detected when they are compromised.

2. RELATED WORK

Sensor networks, constrained with limited power supply,

memory and computation ability, renders traditional security techniques inadequate and requires radically different power-aware solutions. Recently a lot of work has been done in securing sensor network applications like key establishment, secrecy, authentication, robustness to denial-of-service attacks, secure routing and node capture.

There has been some research in probabilistic key sharing to establish a secure path between neighboring nodes[10, 2]. From a large pool of symmetric keys, a random subset of keys is loaded to each sensor nodes, where the number of keys loaded per node depend on the desired probability of two nodes having a common key. Neighboring nodes can establish a secure path if they have a common key. This scheme has a scalability issue because it consumes memory to store keys which keeps increasing with growing network size and also presents the risk that the original key pool can be constructed if the attacker compromise sufficient number of nodes.

Cryptography has been adopted as a standard solution to protect confidentiality, integrity and availability[13, 18, 12]. Faced with the limited power resource problem, symmetric key encryption is preferable over asymmetric versions. However SPINS [13] implements symmetric key cryptography with delayed key disclosure to achieve asymmetric key cryptography. SPINS incorporate both Secure Network Encryption Protocol (SNEP) which provides data confidentiality, authentication, integrity freshness and μTESLA [13] which provides authentication to broadcasts.

Secure routing protocols have been used to maintain normal operation of sensor networks even when some nodes have been compromised [3, 9, 1]. Intrusion Tolerant routing in wireless Sensor Networks (INSENS) [3] works by creating routing tables at each node, thereby improving communication between nodes and the base station. INSENS tries to bypass malicious nodes and nullifies the effect of compromised nodes in the vicinity of malicious nodes.

Most work on data aggregation[4, 7, 6, 15] assumes no node is malicious. SEF [17] and SDAP [16] has proposed a secure aggregation approach in presence of malicious nodes, which can detect and drop false reports. They use a pre key distribution technique along with cryptography to detect false data injection. However such a scheme has considerable overhead considering the resource constrains of sensor nodes. Sampling techniques has been used by Secure Information Aggregation in sensor networks (SIA) system [14] to calculate summary report even when a fraction of the nodes are malicious.

The work proposed in [8] uses beta probability density functions to combine feedback and derive ratings. Here it is considered that highly reputed agents should carry more weight than feedback from agents with low reputation rating. To take care of this discounting of the feedback is introduced as a reputation function of the agent who provided the feedback. This acts as a forgetting factor to weight the relevance of the feedback. Old feedbacks are given lesser weight than the more recent ones. The proposed technique in [8] is used here to compare the performance of different beta reputation schemes along with a reinforcement learning scheme. The performance metric is the number of cycles required to detect the anomalous nodes in the network.

3. APPROACH

We assume that the sensor nodes are deployed in a terrain where the data being sensed follows a pattern over the entire sensed area, e.g., there is a temperature gradient over the side of a hill where the sensors are deployed. We further take into consideration the reality of this pattern varying over time, e.g., the actual temperature sensed will be higher at daytime than during the night. We assume that the nodes and the network will function without error for an initial period of time after deployment. We take this window of opportunity to gather error-free data from which the pattern, over the sensor field, of the physical parameter being sensed would be learned from sufficient number of observations [11].

We proposed a framework in the previous work [11] where each sensor's reported value is to be predicted based on the values reported by its neighbors within close proximity. This means training of as many predictors as there are sensor nodes. While the training process can be computationally expensive, depending on the learning algorithm used, the entire computation is performed offline and hence is not constrained by the processing limitations of sensor nodes. Online computation involves use of this learned patterns to predict sensor values, a straightforward computation with very little computational cost.

We could have used a number of different learning schemes to learn the patterns over the sensor field. For this paper, we have used a backpropagation based neural network learning scheme as proposed in [11] for its robustness and accuracy properties.

We expect that data values reported by correctly operating nodes to be close to their predicted values but it is likely to have some errors due to environmental variations and physical characteristics of the sensor nodes. Rather than making compromise or fault detection decisions based on just one reported sample, it is imperative that we observe multiple reportings of data by suspected nodes before making any conclusions. Hence we use different incremental reputation update schemes that take a sequence of errors between predicted and reported values from a given node. A node is identified to be malicious when the updated reputation falls below a specified threshold.

Here our research goal is to compare the performance metrics of the different reputation schemes. In this paper we incorporate different discount or forgetting factors e.g., unweighted, linear and exponential with different exponent values to be used with the statistically grounded approach (RFSN). The discount factors are used as weights on the present and the past experiences. We compare the performance of different RFSN based schemes along with the Q_learning approach in terms of the number of cycles required to detect the malicious nodes. We are interested in reducing both false positive (the number of correctly functioning nodes which are detected as malicious nodes) and false negatives (undetected malicious nodes).

4. EXPERIMENTAL FRAMEWORK

The sensor network with n nodes are arranged in a tree hierarchy with the base station as the root node. Each non-leaf node in the L-level [1] hierarchy aggregates data reported to it by its k children and forwards it to its own parent in turn. The initial error-free data reporting interval is assumed as D and the predefined threshold is considered as a

[1]There are k^{l-1} nodes in level l and $n = \sum_{l=1}^{L} k^{l-1}$ where k^{L-1} leaf-level nodes are sensing nodes.

fraction $p = 0.03$ of the maximum reputation a sibling possesses at a particular iteration. E stands for the entire data set including training and real time data. For our physical sensing environment, we assume that the sensors are distributed over a region with (x_i, y_i) representing the physical location of the sensing node i.

We model fluctuations of the sensed data in the environment by adding Gaussian noise to the function value $f(x_i, y_i, t)$ for the $i-$th node at time interval t. So, the sensed value at position (x, y) at time t is computed as: $f(x, y, t) = g(x, y) + h(t) + N(0, \sigma)$, where $N(0, \sigma)$ represents a 0 mean, σ standard deviation Gaussian noise. We have used two different g functions, $e^{-(x^2+y^2)}$ and $\frac{(x+y)}{2}$. We will refer to these two environments as E1 and E2 respectively. We use a simple function $h : T \rightarrow [l, h]$ that maps a time to the range $[l, h]$.

In our experiments, we assume that each sensor node adds a randomly generated offset in the range $[0, \epsilon]$ to the data value it senses and vary the number of compromised nodes only at the leaf level,though our mechanism, is capable of detecting faulty nodes at any position in the hierarchy except the root node, assumed as base station.

4.1 Learning Technique

To form the predictor for a given node i in the sensor network, we use a three-level feed-forward neural network with $k - 1$ nodes in the input layer which receives data reported by the siblings of this node. Each such neural network has one hidden layer with H nodes and the output layer has one node that corresponds to the predicted value to be reported by this sensor node. A back-propagation training algorithm, of learning rate η and momentum term δ is used with sigmoid activation function $f(y) = \frac{1}{1+e^{-y}}$ (where y is the linear combination of the inputs) for the neural network computing units, whose outputs are restricted to the $[0, 1]$ range. To evaluate the strengths and weaknesses of the schemes we focus on eighty five node network organised in a tree hierarchy with four children for each internal node and four levels, resulting in 64 sensing nodes at the leaf level. The prediction success rate for the neural net, after offline training with data from environments E1 and E2 are 93.3% and 90.6% respectively. The neural networks used for learning node predictors have the following parameters: $\eta = 0.8, \delta = 0.7$, 3 nodes in the input layer, $H = 8$, and training set size of 4500.

4.2 Reputation management schemes

Algorithm 1 is used online to update the reputation for each node i at each data reporting time interval based on relative error $\varepsilon_i^t = \left|1 - \frac{reported_i^t}{predicted_i^t}\right|$, where $predicted_i^t$ and $reported_i^t$ are the values predicted for and the actual output by the node i at time t respectively. From this relative error, an error statistic $\aleph_i^t$ is computed for updating node reputation. μ_i and σ_i are the mean and standard deviation of the predicted errors computed offline for node i over the training set. During online reputation calculation, $\aleph_i^t = 1$ if $\varepsilon_i^t \leq \mu_{\varepsilon_i}$, otherwise it is evaluated as

$$\aleph_i^t = e^{-\frac{\varepsilon_i^t - \mu_i}{2\sigma^2 \sigma_i^2}}$$

Reputation updates are performed by the Q-learning and Beta-reputation schemes as follows.

Q-Learning Framework: The reputation of every node i is updated as follows:

$$Reputation_{QL_i}^t \leftarrow (1 - \alpha) * Reputation_{QL_i}^{t-1} + \alpha * \aleph_i^t.$$

We use a learning rate, α, of 0.2 and an initial reputation, $Reputation_{QL_i}^0 = 1, \forall i$.

RFSN Framework: In Reputation Based Framework for Sensor Networks (RFSN) [5] framework the corresponding reputation update equation is given by

$$Reputation_{\beta_i}^t = \frac{\gamma_i^t + 1}{\gamma_i^t + \beta_i^t + 2}$$

where γ_i^t and β_i^t are the cumulative cooperative and non-cooperative responses received from node i until time t. We assume $\gamma_i^0 = \beta_i^0 = 0$ and these values are subsequently updated as $\gamma_i^t \leftarrow \gamma_i^{t-1} + \aleph_i^t$ and $\beta_i^t \leftarrow \beta_i^{t-1} + (1 - \aleph_i^t)$.

Algorithm 1: DetectMalicious(n, N)

```
Data: The trained neural net N with set of given
      parameters, number of nodes n
Result: Detection of malicious nodes
initialization: Reputation_Threshold = 0.4,
∀i, Reputation^0_{QL_i} = 1, Reputation^0_{β_i} = 0;
for t=0;;t++ do
    for each sensor node node_i do
        Compute relative_error: ε;
        Compute error_statistic: f(ε);
        Update Reputation^t_{QL_i};
        Update Reputation^t_{β_i};
        if Reputation^t_{QL_i} ≤ Reputation_Threshold
        then
            node_i is malicious according to Q-learning
            based reputation mechanism;
        end
        if Reputation_{β_i} ≤ Reputation_Threshold then
            node_i is malicious according
            Beta-Reputation mechanism;
        end
    end
end
```

For Beta reputation we introduce different types of discount factors (unweighted, linear and exponential) and study their effect on reputation calculation while detecting malicious sensor nodes. The discount factor is used as weights on past positive and negative experiences and enables time discounting of past interaction experience. So the positive and negative experiences at time t can be expressed as

$$\beta_i^t = \sum_{j=1}^{t} (1 - \aleph_i^j) \cdot \lambda^{t-j-1},$$

$$\gamma_i^t = \sum_{j=1}^{t} \aleph_i^j \cdot \lambda^{t-j-1}$$

where $\aleph_i^t$ is the likelihood $0 \leq lambda \leq 1$. For $\lambda > 1$ the above equations correspond to *exponential* discounting. If $\lambda = 1$, the Beta reputation updates are *unweighted* and the

resulting equations will be

$$\beta_i^t = \sum_{j=1}^{t} (1 - \aleph_i^j)$$

and

$$\gamma_i^t = \sum_{j=1}^{t} \aleph_i^j.$$

In our experiments we analyze the characteristics of Beta reputation by varying λ over the set $\{1.0, 0.8, 0.6, 0.4, 0.2\}$.

We also evaluate reputation updates with *linear* weights on past experiences:

$$\beta_i^t = \sum_{j=1}^{t} (1 - \aleph_i^j) \cdot \frac{1}{t - j + 1},$$

and

$$\gamma_i^t = \sum_{j=1}^{t} \aleph_i^j \cdot \frac{1}{t - j + 1}$$

A node is considered to be malicious if the updated reputation of any node becomes less than some predefined fraction, p, of the initial reputation (we have used $p = 0.4$ in our experiments). We actually use an m-of-n approach where a node is marked as malicious only if its reputation falls below the threshold in any m of the last n data reporting time intervals. The m-of-n approach balances number of all false positives and false negatives. We have used m and n values of 5 and 7 respectively.

4.3 Results

We carry out experiments to show the relative strengths of reinforcement learning and Beta reputation schemes with different discount factors(λ). As performance metric, we use the iterations taken by these mechanisms to detect the first and last erroneous nodes. So each experiment returns the minimum and the maximum cycle time taken for detecting the malicious nodes. The latter value corresponds to the time taken to detect all faulty nodes.

The experiments are carried out on the eighty-five node sensor network and in two different environments E_1 and E_2. We run the experiments taking 15 malicious nodes at sensor level, which is around 25% of the 64 nodes sensing the data online. In this work we are interested in examining the influence of the discount factors on the Beta reputation schemes and compare the performance of the statistically grounded (RFSN) scheme and the reinforcement learning Q-learning approach.

For each set of experiments we analyze the outcomes as follows:

- Compare the performances of unweighted and exponential Beta-reputation schemes, with various discount factors, in terms of the mean value of the maximum (minimum) cycle time to detect all the malicious nodes.
- Relative comparison between Q_learning approach and Beta reputation schemes with linear and two extreme exponential discount factors of $\lambda = 0.8$ and 0.2.

We vary the number initial data reporting interval, D when all nodes are reporting data accurately, i.e., no node is malicious. We report the maximum, minimum and mean of maximum (minimum) cycle time taken after the malicious nodes are introduced.

4.4 Observations

4.4.1 Malicious nodes introduced immediately

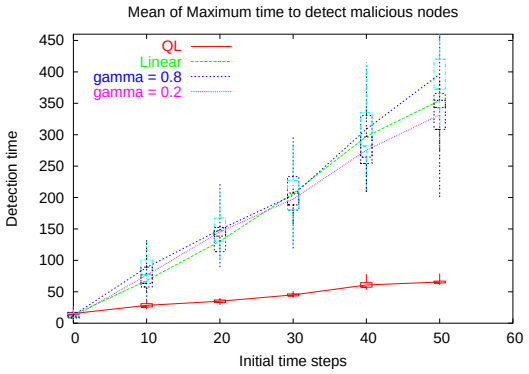
We first discuss the situation when the malicious nodes are introduced as soon as the reputation management system starts calculating reputations ($D = 0$). We observe the following in such situations.

- It is observed from the figures 1(b) and 2(b) that mean of minimum cycle time for Q_learning scheme is slightly more than that required for the Beta reputation schemes (with linear and exponential discounting with $\lambda = 0.2, 0.8$) for both environments. The first malicious node is identified within 5 to 6 time steps for all Beta reputation schemes whereas Q_learning takes around 10 iterations.
- From the figures 1(a) and 2(a), it is seen that the mean of the maximum cycle time for Q_learning algorithm is almost the same ($\approx$ 15 iterations) as that required for different Beta reputation schemes (Linear and exponential with λ values 0.8 and 0.2). The time interval ($mean_{Max}$ - $mean_{Min}$) to capture all the erroneous nodes, however, is less for Q_learning than that of Beta-reputation schemes.

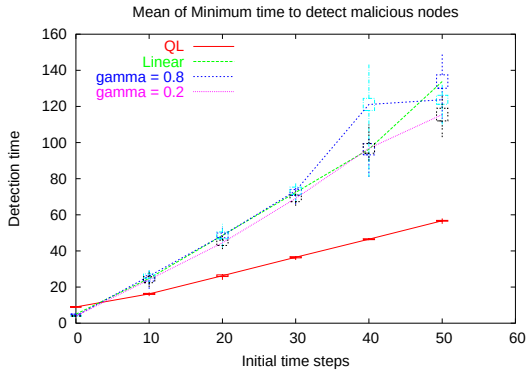
4.4.2 Malicious node introduced after some delay

Now we analyze the situation when the malicious nodes are introduced some time after invoking the reputation management scheme, i.e., $D > 0$. We expect that higher D values will correspond to longer time taken to detect malicious nodes for all reputation schemes. This is due to the fact that before becoming malicious nodes will have longer history of normal performance, and this reputation will have to be superseded with sufficient number and amount of deviations after the nodes become malicious. Even with time discounting, therefore, longer periods of error-free performance will necessitate longer periods of erroneous behavior before a node is accurately identified to have turned malicious. The actual number of periods taken before such detection will, however, be a factor of the repuation management scheme and the learning rate and discount factors used therein.

- It is interesting to see from the graphs 1(b) and 2(b) that when the nodes become malicious after some data reporting intervals, the performance of the Beta-reputation schemes, irrespective of the value of discount factor, become gradually worse in comparison to that of the Q-learning scheme. We have examined the performances of both the schemes with the initial error-free time interval set $D = \{10, 20, 30, 40, 50\}$. Such delay of onset of malicious data affects the performance of the Beta-reputation schemes and the mean of minimum cycle time increases significantly. In general, lower values of λ produces slightly better detection times (we discuss this in more detail later). When $D = 10$, the means of minimum cycle for detecting a malicious node for different Beta reputation schemes take around 25 iterations in both E_1 and E_2. When $D = 50$, these values range from 110 to 140 cycles, for different λ values, and 100 to 120 cycles for distributions E_1 and E_2 respectively. On the other hand,

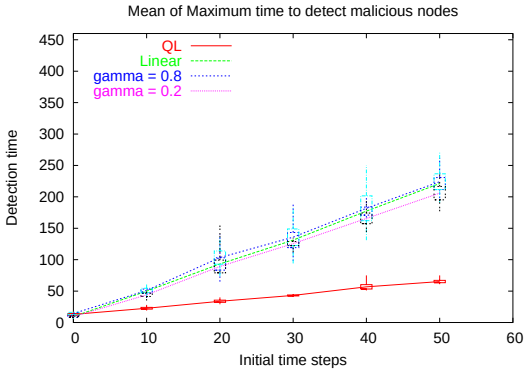


(a) Maximum detection time for distribution $e^{-(x^2+y^2)}$

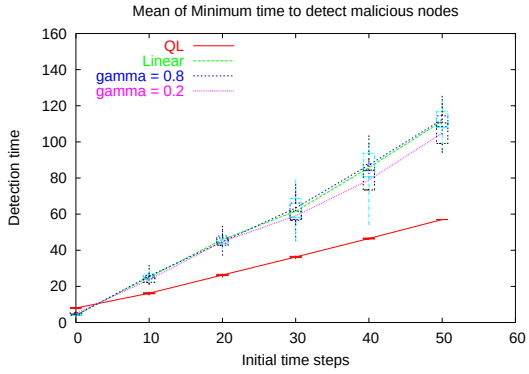


(b) Minimum detection time for distribution $e^{-(x^2+y^2)}$

Figure 1: Maximum and minimum cycle of anomaly detection for eighty-five node network for environment E1.



(a) Maximum detection time for distribution $\frac{x+y}{2}$



(b) Minimum detection time for distribution $\frac{x+y}{2}$

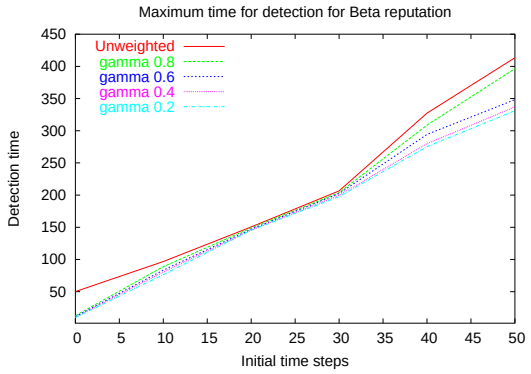
Figure 2: Maximum and minimum cycle of anomaly detection for eighty-five node network for environment E2.

the Q_learning scheme takes the corresponding values around 18 ($D = 10$) and 55 ($D = 50$) cycles respectively in both environments E_1 and E_2.

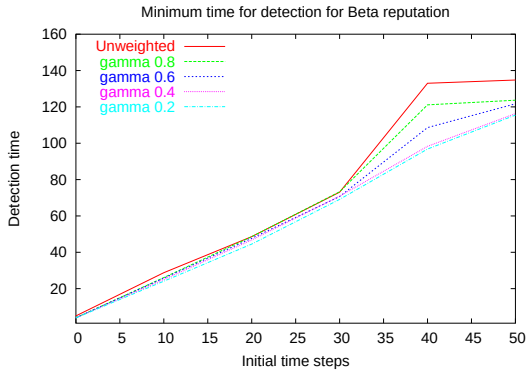
- Similar phenomena are observed for the mean values of the maximum cycle time for both the distributions as shown in the Figures 1(a) and 2(a). For different Beta reputation schemes, we observe that the means of maximum cycle time lie between 50 and 90 for environment E_1 and 45 and 50 for E_2 when $D = 10$. The corresponding values for Q_learning scheme are 30 and 25 cycles for E_1 and E_2 respectively. For $D = 50$ the corresponding values for Beta reputation schemes lies between 325 to 400 cycles for environment E_1 and between 200 and 225 iterations for environment E_2. The Q_learning scheme takes almost the same mean maximum detection times (≈ 70 cycles) for both the environments for $D = 50$. The relative advantage of the Q-learning scheme is likely to only increase with further delays in onset of malicious nodes, i.e., for higher values of D.
- The performance of the Beta-reputation scheme with linear discount factor lies between that of the Beta-reputation scheme with two limiting discount factors $\lambda = 0.8$ and $\lambda = 0.2$ (see Figures 1(a), 2(a), 1(b) and 2(b)). The scheme with linear discount factor performs slightly better than that with $\lambda = 0.8$ but is somewhat slower to detect malicious nodes when compared with the Beta-repuation scheme with $\lambda = 0.2$.

4.4.3 *Effect of discount factors*

We now analyze the behavior of Beta reputation schemes with different exponential discount factors (λ). From the Figures 3(a), 4(a), 3(b) and 4(b), it is observed that the performance of the Beta-reputation with the least discount factor $\lambda = 0.2$ gives the minimum mean value for both the

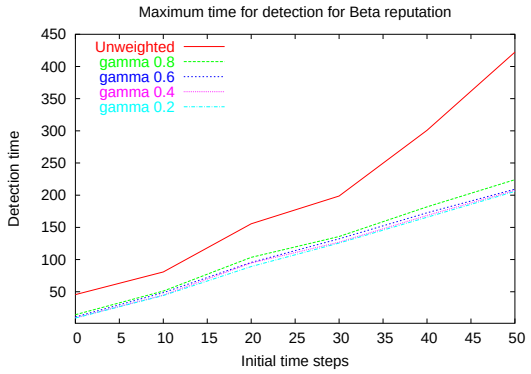


(a) Influence of exponential discount factor on maximum detection time

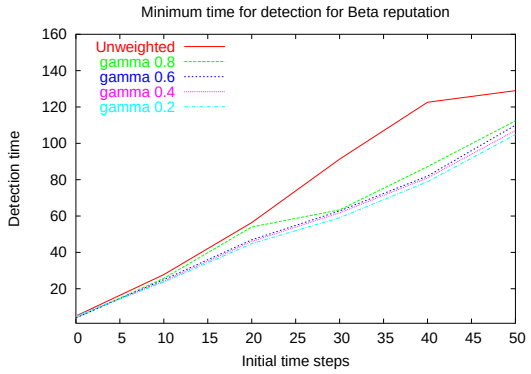


(b) Influence of exponential discount factor on minimum detection time

Figure 3: Cycle of anomaly detection for different Beta-reputation schemes for distribution $e^{-(x^2+y^2)}$ in eighty-five node network.



(a) Influence of exponential discount factor on maximum detection time



(b) Influence of exponential discount factor on minimum detection time

Figure 4: Cycle of anomaly detection for different Beta-reputation schemes for distribution $\frac{x+y}{2}$ in eighty-five node network.

minimum and maximum detection time for both the environments irrespective of error-onset delays, i.e., different D values. The number of cycles required for detecting the malicious nodes increase with the increasing λ values. This is because higher λ values correspond to longer-lasting memories and hence more time is needed for errors reported after introduction of malicious behavior to supersede the previous correct data reported by those nodes for the first D data intervals. When the discount factor becomes 1, the reputation update is unweighted. So it takes the maximum amount of time to detect the malicious nodes and results in the worst performance.

4.4.4 *Effect of environment*

From the Figures 3(b) and 4(b) we observe that, for the RFSN based approach with different exponential discount factors, the mean of both the minimum and maximum time required to detect a malicious node for environment E1 is greater than that for environment E2 for different values of D.

4.4.5 *Additional observations*

We further note that for our experimental settings, we have not observed any false positives or false negatives. As observed this depends on the values of n and m used in the m-of-n detection procedure. While we provide no guarantees, it is encouraging to observe such robust performance of these approaches.

A pertinent question is what makes the Q-learning based reputation management schemes a faster mechanism compared to the Beta-reputation approach. Note that when the discount factor in the Beta-reputation system is complementary to the learning rate in the Q-learning approach, i.e., $\lambda = (1-\alpha)$, the $Reputation_i$ and γ_i updates put equal weight on the past experiences. The new experience is weighted by α in the Q-learning approach, and is unweighted in the up-

date of γ_i. Additionally the actual Beta-reputation calculation computes a ratio that includes the β_i factor. We believe this combined effect affects the responsiveness of the Beta reputation scheme. We plan to do more detailed experimentation and analysis to shed further light on this issue.

5. CONCLUSION

In this paper we compared different, time-discounting variations of the Beta-reputation scheme and a repuation scheme based on standard reinforcement learning approach, to manage the repuations of nodes in a sensor network organized in a hierarchy. We assume the availability of data from an error-free network which is used by a neural network based learning technique [11] to learn the correlations in data sensed by different nodes. Different environments are used to represent different correlation patterns. Subsequently, online reputation management schemes update node reputations based on differences between the data they report and that they are predicted to report based on data reported by their siblings in the sensor net hierarchy. The goal is to quickly detect all malicious nodes that consistently or periodically report errors with small offsets to influence the aggregate data reported to the base station of the sensor network. We study the responsiveness of different reputation management schemes by varying the time delays before introducing malicious nodes.

A malicious node is recognized when its reputation consistently falls below a pre-specified threshold. We have considered the computation of incremental reputation based on Q_learning based approach and Beta-reputation approach with different discount factors represented as the weights over the past experiences. We experimented with a eighty-five node network arranged in a four-level hierarchy. We found that for two different environmental patterns all the randomly generated malicious nodes are detected even when upto to 25% of the sensor nodes are malicious. We have compared the results of the reputation schemes mentioned in terms of number of cycles to detect the first and all malicious nodes. We vary the length of the initial error-free data reporting interval before introducing malicious nodes. As this period increases all schemes take more time to detect malicious nodes. Lower discount factors for exponential weighting in Beta-reputation, that uses lower weights for past experiences, are found to be more responsive. Linear weighting schemes are found to be better than unweighted schemes and compare better than some weighted schemes while performing worse than others. The Q-learning algorithm, however, performs increasingly better in comparison to all the Beta-reputation scheme variants.

We are working on studying, in more detail, the changes in the reputation values, as calculated by the different reputation management schemes presented here, for a given sequence of data. The goal is to understand what makes some of these schemes more responsive than others. By clearly identifying the strengths and weaknesses of these approaches, we hope to develop a hybrid scheme that will be more robust than any individual scheme discussed above.

We also intend to extend our work to analysis the performance of these repuation management schemes on more sophisticated collusion, where malicious nodes may take turns to report higher errors.

Acknowledgement: This work has been supported in part by a DOD-ARO Grant # W911NF-05-1-0285.

6. REFERENCES

[1] B. Awerbuch, D. Holmer, and H. Rubens. Provably secure competitive routing against proactive byzantine adversaries via reinforcement learning.

[2] H. Chan, A. Perrig, and D. Song. " random key predistribution schemes for sensor networks". In *Security and Privacy: Proceedings of the 2003 Symposium*, pages 197–213, May 2003.

[3] J. Deng, R. Han, and S. Mishra. Insens: Intrusion-tolerant routing in wireless sensor networks, 2002.

[4] A. Deshpande, S. Nath, P. B. Gibbons, and S. Seshan. Cache-and-query for wide area sensor databases. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 503–514, New York, NY, USA, 2003. ACM Press.

[5] S. Ganeriwal and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 66–77, New York, NY, USA, 2004. ACM Press.

[6] T. He, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Aida: Adaptive application-independent data aggregation in wireless sensor networks. *Trans. on Embedded Computing Sys.*, 3(2):426–457, 2004.

[7] C. Intanagonwiwat. Impact of network density on data aggregation in wireless sensor networks, Nov 2001.

[8] A. Josang and R. Ismail. The beta reputation system. In *e-Reality: Constructing the e-Economy : proceedings of 15th Bled conference of Electronic Commerce*, Bled, Slovenia, June 2002.

[9] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2–3):293–315, September 2003.

[10] L.Eschenauer and V.D.Gligor. A key-management scheme for distributed sensor networks. In *"Proceedings of the 9th ACM conference on Computer and communications security"*, pages 41–47, Nov 2002.

[11] P. Mukherjee and S. Sen. Detecting malicious nodes fron learned data patterns. In *In proceedings of workshop on Agent Technology for SensorNetwork (ATSN'07), AAMAS*, pages 11–17, Hawaii, USA, May 2007.

[12] T. Park and K. G. Shin. Lisp: A lightweight security protocol for wireless sensor networks. *Trans. on Embedded Computing Sys.*, 3(3):634–660, 2004.

[13] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. Spins: security protocols for sensor networks. *Wirel. Netw.*, 8(5):521–534, 2002.

[14] B. Przydatek, D. Song, and A. Perrig. Sia: Secure information aggregation in sensor networks, 2003.

[15] M. Sharifzadeh and C. Shahabi. Supporting spatial aggregation in sensor network databases. In *GIS '04: Proceedings of the 12th annual ACM international*

workshop on Geographic information systems, pages 166–175, New York, NY, USA, 2004. ACM Press.

[16] Y. Yang, X. Wang, S. Zhu, and G. Cao. Sdap:: a secure hop-by-hop data aggregation protocol for sensor networks. In *MobiHoc '06: Proceedings of the 7th international symposium on Mobile ad hoc networking and computing*, pages 356–367, New York, NY, USA, 2006. ACM Press.

[17] F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical en-route filtering of injected false data in sensor networks. *IEEE Jounal on Selected Areas in Communications, Special Issue on Self-organizing Distributed Collaborative Sensor Networks*, 23(4):839–850, April 2005.

[18] S. Zhu, S. Setia, and S. Jajodia. Leap: efficient security mechanisms for large-scale distributed sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 62–72, New York, NY, USA, 2003. ACM Press.