A New Methodology for Identifying Unreliable Sensors in Data fusion

Anis Yazidi[a], Enrique Herrera-Viedma[b]

[a] *Department of Information Technology, University College of Oslo and Akershus, Oslo, Norway.*
[b] *Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain.*

Abstract

Sensor fusion is a fundamental research topic that has received significant attention in the literature. An important body of research has focused on assessing the reliability of a sensor or more generally an "information source" by comparing the readings with the ground truth in an online or offline manner. The Weighted Majority Voting algorithm [25], a well-known online learning algorithm, is a typical example of a class of approaches that assess the reliability of a sensor by comparing its readings to the ground truth in a online manner. Unlike the latter stream of research, in this article, we tackle the problem of identifying unreliable sensors *without* the knowledge of the ground truth– which is a novel research direction in its own right. We advocate that comparing the readings of a sensor to the rest of the sensors gives an invaluable information about its reliability. In this article, we present a solution to the problem based on the theory of S-Model Learning Automata (LA) [17]. Interestingly, the feedback to the S-Model environment LA is defined in an intuitive manner, namely, it is *proportional* to the number of sensors adhering to the chosen action. Our solution does not impose any constraint on the parity of the number of sensors and thus is *general* and can handle any arbitrary number of sensors. Apart from applying the classical S-Model LA, we develop a novel S-

Email address: anis.yazidi@hioa.no (Anis Yazidi)

Model based pursuit LA algorithm that achieves a faster convergence than the legacy solution by an order of magnitude of ten while still yielding high accuracy. The devised schemes have been subjected to comprehensive experiments including comparison to the state-of-the-art.

Keywords: Unreliable Sensors Identification, Learning Automata, S-Model Environment.

---

1. Introduction

Data fusion from uncertain sources of information is an important research topic that has gained an increasing research attention during recent years [10, 28, 35, 11]. Furthermore, it is known that fusioning information from a set of unreliable sensors can give a more robust information about the process being monitored [28, 10, 8, 31]. An important body of research has focused on assessing the reliability of a sensor or more generally an "information source" by comparing the readings with the ground truth. The Weighted Majority Voting algorithm [25], a well-known Machine Learning algorithms, is a typical example of a class of approaches that assess the reliability of a sensor by comparing its readings to the ground truth in a online manner. Once the reliability of the sensors is inferred, this information can be used as input to a fusioning process so that to mitigate the undermining effect of the unreliable sensors on the quality of the fusioned information. However, in many real life applications, access to the ground truth is simply impossible. This is particularly true in the field of "Softsensing" where the harsh nature of the environment prohibits accessing the ground truth [12]. In such settings where the ground truth is inaccessible, the question of assessing the reliability of the sensors is *apparently* impossible to solve. In [38], Yazidi et al. presented a counter-intuitive solution based on the idea of that the "agreement" between the sensors themselves can give invaluable knowledge about their respective reliabilities. The main tool used to solve that problem was the Linear Reward-Inaction Learning Automata (L_{RI}) [17]. Nevertheless, a major disadvantage with the previous solution was the constraint on the parity of the number of sensors so that to invoke the majority voting concept. In addition, the way by which reward and penalty were defined was quite complex and counter-intuitive. Another weakness is inherent to L_{RI}. The informed reader observes that since the L_{RI} is an LA of reward-inaction flavor, there are cases where updates do not take place, namely, if the chosen action is reliable and the sensor disagrees with the rest of the sensors, or if the chosen action is unreliable and the sensor agrees with the rest of the sensors. According to the L_{RI}, the action vector probability is not updated in case of penalty which definitively slows down convergence of the solution proposed in [38].

In this paper, we overcome those drawbacks by presenting a new methodology for identifying unreliable sensors in data fusion based on a LA which falls under the category of S-Model Learning Automata (S-LA) [17]. The feedback of our devised LA is more intuitive and non-binary. The LA attached to a given sensor reinforces its current action in a *proportional* manner to the number of sensors adhering with its chosen action. In addition, the solution is simpler than the original solution presented in [38]. In fact, we propose a more intuitive manner by which reward and penalty are defined. Furthermore, our solution is *general* and does not impose any constraint on the number of sensors [38].

Therefore, in contrast to the legacy solution presented in [38], we do not impose any extra constraint on the parity of the sensors. In fact, our current work is not based on the majority voting results for heterogeneous groups [6], and thus, there is no need for imposing the parity condition. As opposed to this, the current solution can cover cases which are not solvable by [38]. Additionally, we develop a new S-Model based pursuit LA algorithm that achieves faster convergence than legacy solution by order of magnitude 10. The latter algorithm is also compared to a baseline S-Model LA and exhibits clear superiority. The way we define the feedback of LA is intuitive and resorts to the concept of "proportionality". Furthermore, the applications of S-Model LA are sparse in the literature compared to P-Model LA. This article demonstrates that S-Model LA can be a powerful concept in the field of sensor type identification.

The rest of the paper is organized as follows. Section 2 briefly reviews the theory of LA which is the main tool used in this paper. Section 3 gives a formal statement of the problem. In Section 4, we present our solution, which is based on S-LA scheme for identifying unreliable sensors in a stochastic environment in the absence of knowledge of the ground truth. Some experimental results that validate the theoretical results are presented in Section 5. Section 6 concludes the paper.

2. Stochastic Learning Automata

Learning Automata (LA) is a decision making mechanism designed for decision making under uncertainty [1, 18, 26, 30].

From a historical perspective, the first work on LA is due to Tsetlin [32] who pioneered learning mechanisms that attempt to mimic biological learning mechanisms.

Generally speaking, the LA chooses a random action according to a probability vector. Based on the feedback from the environment, the probability vector is updated over time. The LA interacts with the environment according to a feedback loop.

The introduction of the term "Learning Automata" is due to Narendra and Thathachar [18].

The work on LA consists of two main threads: Fixed Structure Stochastic Automata (FSSA) and Variable Structure Stochastic Automata (VSSA). It is worth mentioning that FFSA [32] design was the de facto standard before the discovery of the first instances of VSSA later by Vorontsova and Varshavskii [18]. According to the FFSA design, the input (usually the feedback from the environment) and output (usually the action) of the LA are connected according to a deterministic mapping. In simple terms, the choice of the next action is deterministic as a function of the feedback from the environment.

In this article, we base our work on the family of VSSA. The VSSA is defined using a set of actions representing the output of the LA, a set of inputs representing the feedback from the random environment and the learning algorithm T by which the so called action probability vector is updated. The behavior of the LA is determined by the mapping T. VSSA falls under two main families: absorbing VSSA and ergodic VSSA. In the case of absorbing barriers LA, the probability action vector converges to unit vector, and thus an exclusive choice of one action.

Ergodic VSSA [18, 22] are usually modelled as non-absorbing Markov Chain. In this case, the probability action vector converges in distribution to a non-unit vector. By virtue of the ergodicity propriety, VSSA are adequate for non-stationary environments while absorbing VSSA are suitable for stationary environments.

In order to boost the convergence speed of the LA algorithms, the concept of discretizing the probability space into a finite set of values was proposed [22, 29]. The discretization is called linear whenever the values are equi-space, and non-linear in the opposite case [22].

A breakthrough in the field of the LA is the advent of pursuit LA. The idea behind pursuit LA is to maintain estimates of the reward probability of each action and to pursue the action with the highest estimate, i.e, increase the reward of the action with the highest probability. The *reward-estimate* vector is updated each time an action is chosen and the corresponding feedback from the environment is returned.

LA are also broadly classified into P-Model and S-Model [14]. In P-Model, the feedback from the environment is binary, i.e, either 0 or 1, where 0 denotes a favorable feedback while 1 denotes an unfavorable feedback by definition. However, in S-Model the feedback from the environment admits a continuous value in the interval $[0, 1]$ where values approaching 0 represent a favorable feedback, while values approaching 1 represent unfavorable feedback. Usually, normalization is applied in order to obtain a feedback between 0 and 1. Despite the importance of the S-Model and its applicability to a large set of real-life problems where the feedback from the environment admits continuous values, S-Model have unfortunately received little attention in the field of

LA compared to P-Model.

LA has found the large set of applications. Those applications include routing problems [16, 24, 3, 4, 34], image processing [5, 7], recommendation systems [36, 15, 37], priority assignment in queueing systems [33], adaptive polling protocols [19, 20, 21], resource allocation under uncertainty [9], to mention a few.

3. Modeling the Problem

We consider a population of N sensors, $\mathcal{S} = \{s_1, s_2, \dots, s_N\}$. Let the unknown ground truth at the time instant t be modeled by a binary variable $T(t)$, which can take one of two possible values, 0 and 1. The value of T is unknown and can only be inferred through measurements from sensors. The output from the sensor s_i is referred to as x_i. Let π be the probability of the state of the ground truth, i.e., $T = 0$ with probability π.

We suppose that the probability of the sensor reporting a value erroneously is symmetric. Formally, this reduces to:

$$Prob(x_i = 0|T = 1) = Prob(x_i = 1|T = 0). \tag{1}$$

Further, let p_i denote the Correctness Probability (CP) of sensor s_i, where: $p_i = Prob(x_i = 0|T = 0) = Prob(x_i = 1|T = 1)$.

It is easy to prove $Prob(x_i = T)$ is, indeed, p_i.

We can define a reliable sensor to be one that has a CP $p_i > 0.5$ and an unreliable sensor as one that has a CP $p_i < 0.5$.

In addition, we assume that every p_i can have one of two possible values from the set $\{p_R, p_U\}$, where $p_R > 0.5$ and $p_U < 0.5$. Then, a sensor s_i is said to be reliable if $p_i = p_R$, and is said be unreliable if $p_i = p_U$. We assume that p_R and p_U are unknown to the algorithm.

Based on the above, the set of reliable sensors is $\mathcal{S}_R = \{s_i|p_i = p_R\}$, and the set of unreliable sensors is $\mathcal{S}_U = \{s_i|p_i = p_U\}$. Furthermore, let $N_R = |\mathcal{S}_R|$ and $N_U = |\mathcal{S}_U|$.

Throughout this paper, we will resort to the following assumption [38]: $(N_R - 1)p_R + N_U p_U > (N_R + N_U)/2$. The above mild condition that we formulate in this paper rests on the philosophical fundament found in the society where the truth is a virtue among the individuals, and that the truth prevails over lies.

4. The Solution

4.1. Overview of Our Solution

In this section, we provide a novel solution to the identifying unreliable sensors in data fusion based on the field of S-LA. Our solution involves a team of LA where each LA is uniquely attached to a specific sensor. Each automaton attached to sensor s_i, has two actions. The aim of LA is to infer the identity of the sensor in question by exploiting a type of proportional feedback where an action is reinforced in proportional manner to the number of sensors adhering to it. It is important to note that in [38] we assumed a parity condition according to which the total number of sensors $N_R + N_U$ must be an even number. This condition is not required in the current article.

First, we will present two main theorems that we will use in the design of our LA in Section 4.2.

Theorem 1. *Consider the scenario when $(N_R - 1)p_R + N_U p_U > (N_R + N_U)/2$ and when $N_R + N_U \geq 2$. Let $s_i \in \mathcal{S}_R$. Consider $\beta_0^i(t)$ the normalized ratio of the number of sensors agreeing with reliable sensor s_i. This number is proportional to the number of sensors agreeing with reliable sensor s_i.*

$$\beta_0^i(t) = \frac{\sum_{\substack{k=1 \\ k \neq i}}^{N_R+N_U} I\{x_k(t) = x_i(t)\}}{N_R + N_U - 1} \quad (2)$$

Let $\beta_1^i(t)$ as a normalized ratio of the number of sensors disagreeing with reliable *sensor s_i.*

$$\beta_1^i(t) = \frac{N_R + N_U - 1 - \sum\limits_{\substack{k=1 \\ k \neq i}}^{N_R+N_U} I\{x_k(t) = x_i(t)\}}{N_R + N_U - 1} \tag{3}$$

Then, $E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U) > E(\beta_1^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U)$.

Proof:

Let $\zeta_k^{(N_R,N_U)}$ be the probability of a sensor i adheres with the mixture of exactly k sensors from N_R reliable and N_U unreliable.

$\zeta_k^{(N_R,N_U)} = Prob(\beta_0^i(t) = k)$

We define:

$$\beta_0^i(t) = \frac{\sum\limits_{\substack{k=1 \\ k \neq i}}^{N_R+N_U} I\{x_k(t) = x_i(t)\}}{N_R + N_U - 1} \tag{4}$$

We define too:

$$\beta_1^i(t) = \frac{N_R + N_U - 1 - \sum\limits_{\substack{k=1 \\ k \neq i}}^{N_R+N_U} I\{x_k(t) = x_i(t)\}}{N_R + N_U - 1} \tag{5}$$

Let us prove this by recurrence that the following expression holds true:

$$\begin{gathered} E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U + 1) = \\ \frac{(N_R - 1)\left(p_R^2 + q_R^2\right) + (N_U + 1)\left(p_U p_R + q_U q_R\right)}{N_R + N_U} \end{gathered} \tag{6}$$

We suppose that N_R is fixed, while the proof by recurrence is performed for N_U.

We first consider the case where $N_U = 0$ and N_R is an arbitrarily integer.

We know that $\sum_{k=0}^{N_R-1} k\zeta_k^{(N_R,0)}$ is the average of a binomially distributed random variable, where $N_R - 1$ being the total number of experiments and

$p_R^2 + q_R^2$ the probability of agreement between sensor i and another reliable sensor. Therefore, we can write:

$$\sum_{k=0}^{N_R-1} k\zeta_k^{(N_R,0)} = (N_R - 1)(p_R^2 + q_R^2) \tag{7}$$

Thus, we obtain:

$$\begin{aligned} E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = 0) = \\ \sum_{k=0}^{N_R-1} \frac{k}{N_R - 1}\zeta_k^{(N_R,0)} &\quad (8) \\ = \frac{(N_R - 1)(p_R^2 + q_R^2)}{N_R - 1} &\quad (9) \end{aligned}$$

Now, let us suppose that the following expression is true:

$$\begin{aligned} E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U) = \\ \frac{(N_R - 1)(p_R^2 + q_R^2) + N_U(p_U p_R + q_U q_R)}{N_R + N_U - 1} \end{aligned} \tag{10}$$

Let us prove by recurrence that the equation holds true for $N_U + 1$ while N_R is fixed.

We can write the following:

$$\begin{aligned} E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U + 1) = \\ \sum_{k=0}^{N_R+N_U} \frac{k}{N_R + N_U}\zeta_k^{(N_R,N_U+1)} \end{aligned} \tag{11}$$

For $1 \leq j \leq N_R + N_U - 1$, we know that:

$$\begin{aligned} \zeta_j^{(N_R,N_U+1)} = (1 - q_U q_R - p_R p_U)\,\zeta_j^{(N_R,N_U)} \\ +(q_U q_R + p_R p_U)\zeta_{j-1}^{(N_R,N_U)} \end{aligned} \tag{12}$$

and,

$$\zeta_{N_R+N_U}^{(N_R,N_U+1)} = (q_U q_R + p_R p_U)\,\zeta_{N_R+N_U-1}^{(N_R,N_U)} \tag{13}$$

We shall explain the expression in Eq(12). $\zeta_j^{(N_R,N_U+1)}$ which is the probability of the reliable sensor in question agrees with exactly j sensors can be defined in a recursive manner as a function of $\zeta_j^{(N_R,N_U)}$ and $\zeta_{j-1}^{(N_R,N_U)}$. The associated agreement event takes place in two cases:

- if the reading of the sensor in question disagrees with the added unreliable sensor while agreeing with the rest j sensors. The probability of this event is $(1 - q_U q_R - p_R p_U)\zeta_j^{(N_R,N_U)}$
- if the reading of the sensor in question agrees in the same time with the added unreliable sensor and with the rest $j-1$ sensors, thus, in total, the sensor in question agrees with a total of j sensors. This event take place with probability equal to $(q_U q_R + p_R p_U)\zeta_{j-1}^{(N_R,N_U)}$.

$$\begin{aligned} &E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U + 1) = \\ &\sum_{j=1}^{N_R+N_U} \frac{j}{N_R+N_U}\Big((1 - q_U q_R - p_R p_U)\zeta_j^{(N_R,N_U)} \\ &\qquad + (q_U q_R + p_R p_U)\zeta_{j-1}^{(N_R,N_U)}\Big) \end{aligned} \tag{14}$$

$$
\begin{aligned}
(N_R + N_U)E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U + 1) = \\
(1 - q_R q_U - p_R p_U)(N_R + N_U)\zeta_{N_R+N_U-1}^{(N_R,N_U)} \\
+ \sum_{j=1}^{N_R+N_U-1} j\Big((1 - q_R q_U - p_R p_U)\zeta_{j-1}^{(N_R,N_U)} \\
+(q_R q_U + p_R p_U)\zeta_{N_R+N_U}^{(N_R,N_U)}\Big) \\
= (1 - q_R q_U - p_R p_U)\Big((N_R + N_U - 1)\zeta_{N_R+N_U-1}^{(N_R,N_U)} \\
+ \sum_{j=1}^{N_R+N_U-1} (j-1)\zeta_{j-1}^{(N_R,N_U)}\Big) \\
+(1 - q_R q_U - p_R p_U)\Big(\zeta_{N_R+N_U-1}^{(N_R,N_U)} \\
+ \sum_{j=1}^{N_R+N_U-1} \zeta_{j-1}^{(N_R,N_U)}\Big) \\
+(q_R q_U + p_R p_U)\sum_{j=1}^{N_R+N_U-1} j\zeta_j^{(N_R,N_U)}
\end{aligned}
\tag{15}
$$

Using a change of variable where $j-1$ is replaced by j, we obtain:

$$
\begin{aligned}
(N_R + N_U - 1)\zeta_{N_R+N_U-1}^{(N_R,N_U)} + \sum_{j=1}^{N_R+N_U-1} (j-1)\zeta_{j-1}^{(N_R,N_U)} = \\
\sum_{j=0}^{N_R+N_U-1} j\zeta_j^{(N_R,N_U)}
\end{aligned}
\tag{16}
$$

In addition, we observe too that:

$$
\zeta_{N_R+N_U-1}^{(N_R,N_U)} + \sum_{j=1}^{N_R+N_U-1} \zeta_{j-1}^{(N_R,N_U)} = \sum_{j=0}^{N_R+N_U-1} \zeta_j^{(N_R,N_U)}
\tag{17}
$$

Furthermore, we know that by the law of total probability that:

$$\sum_{j=0}^{N_R+N_U-1} \zeta_j^{(N_R,N_U)} = 1 \tag{18}$$

Therefore, using some simplification, we are able to obtain:

$$\begin{aligned}
&E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U + 1) = \\
&\frac{1 - q_R q_U - p_R p_U}{N_R + N_U} \sum_{j=1}^{N_R+N_U-1} j\zeta_j^{N_R,N_U} + \\
&+ \frac{q_R q_U + p_R p_U}{N_R + N_U} \cdot 1 + \\
&+ \frac{q_R q_U + p_R p_U}{N_R + N_U} \sum_{j=1}^{N_R+N_U-1} j\zeta_j^{(N_R,N_U)} \\
&= \sum_{j=1}^{N_R+N_U-1} \frac{j}{N_R + N_U} \zeta_j^{(N_R,N_U)} + \frac{q_R q_U + p_R p_U}{N_R + N_U}
\end{aligned} \tag{19}$$

We use the fact that we supposed that the following expression is true:

$$\begin{aligned}
&E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U) = \\
&\frac{(N_R - 1)(p_R^2 + q_R^2) + N_U(p_U p_R + q_U q_R)}{N_R + N_U - 1}
\end{aligned} \tag{20}$$

The above expression can be re-written as:

$$\begin{aligned}
\sum_{j=1}^{N_R+N_U-1} j\zeta_j^{(N_R,N_U)} &= (N_R - 1)\left(p_R^2 + q_R^2\right) \\
&+ N_U\left(p_U p_R + q_U q_R\right)
\end{aligned} \tag{21}$$

Therefore, Eq. (19) can be written as:

$$
\begin{aligned}
&E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U + 1) = \\
&\frac{N_R - 1}{N_R + N_U} \left(p_R^2 + q_R^2\right) + \frac{N_U}{N_R + N_U} \left(p_U p_R + q_U q_R\right) \\
&\qquad\qquad + \left(\frac{q_R q_U + p_R p_U}{N_R + N_U}\right) \qquad (22) \\
&= \frac{N_R - 1}{N_R + N_U} \left(p_R^2 + q_R^2\right) + \frac{N_U + 1}{N_R + N_U} \left(p_U p_R + q_U q_R\right) \qquad (23)
\end{aligned}
$$

Thus, we obtain

$$
\begin{aligned}
&E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U + 1) = \\
&\frac{(N_R - 1) \left(p_R^2 + q_R^2\right) + (N_U + 1) \left(p_U p_R + q_U q_R\right)}{N_R + N_U} \qquad (24)
\end{aligned}
$$

This ends the first step of the proof where we give the accurate expression of $E(\beta_0^i(t))$.

Now, we move to the second part of the proof.

We have:

$$
\begin{aligned}
&E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U) = \\
&p_R \frac{(N_R - 1)p_R + N_U p_U}{N_R + N_U - 1} + q_R \frac{(N_R - 1)q_R + N_U q_U}{N_R + N_U - 1} \qquad (25) \\
&= p_R \frac{(N_R - 1)p_R + N_U p_U}{N_R + N_U - 1} + q_R \left(1 - \frac{(N_R - 1)p_R + N_U p_U}{N_R + N_U - 1}\right) \qquad (26)
\end{aligned}
$$

We will now prove that $E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U) > 1/2$. In order to prove this inequality, let us consider the function $g(.)$ defined as the convex combination:

$$g(\rho) = p_R \cdot \rho + q_R \cdot (1 - \rho),$$

whence, it is easy to see that:

$$g(\frac{(N_R-1)p_R+N_Up_U}{N_R+N_U})=$$
$$E(\beta_0^i(t)\mid s_i\in S_R\,|S_R|=N_R,|S_U|=N_U). \tag{27}$$

Moreover, please note that the condition:

$$\frac{(N_R-1)p_R+N_Up_U}{N_R+N_U-1}>(N_R+N_u)/2 \tag{28}$$

gives that:

$$\frac{(N_R-1)p_R+N_Up_U}{N_R+N_U-1}>\frac{1}{2}+\frac{1}{2(N_R+N_U-1)} \tag{29}$$

Let us investigate the dynamics of $g(\rho)$ by studying its derivative function, $g'(\rho)$, which specifically, has the form $g'(\rho)=2p_R-1$. Since, by definition, $p_R>1/2$, we can confirm that $2p_R-1>0$ which is equivalent to stating that $g'(\rho)>0$. $g(\rho)$ is thus a *strictly increasing* function.

We further know that $g(1/2)=1/2p_R+1/2q_R=1/2$. Thus, by virtue of the strictly increasing property of the function $g(.)$:

$$\text{if}\rho>1/2\quad\Rightarrow\quad g(\rho)>g(1/2)=1/2. \tag{30}$$

Observe that, in particular, we can apply the inequality (30) for the particular case when $\rho=\frac{(N_R-1)p_R+N_Up_U}{N_R+N_U-1}$. Since we have previously demonstrated in Eq. (29) that $\frac{(N_R-1)p_R+N_Up_U}{N_R+N_U-1}>1/2$, if we replace ρ by $\frac{(N_R-1)p_R+N_Up_U}{N_R+N_U-1}$ in the inequality (30), we get:

$$E(\beta_0^i(t)\mid s_i\in S_R,,|S_R|=N_R,|S_U|=N_U)>1/2.$$

The last element of the proof is to observe that the following complementarity in our LA design:

$\beta_0^i(t)=1-\beta_1^i(t)$

Thus, $E(\beta_0^i(t)\mid s_i\in S_R,|S_R|=N_R,|S_U|=N_U)=1-E(\beta_1^i(t)\mid s_i\in$

$S_R, |S_R| = N_R, |S_U| = N_U)$ Thus,

$$E(\beta_0^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U) > E(\beta_1^i(t) \mid s_i \in S_R, |S_R| = N_R, |S_U| = N_U).$$

which concludes the proof. □

We shall now consider the converse case of omitting an unreliable sensor, and prove the analogous result.

Theorem 2. *Consider the scenario when* $(N_R - 1)p_R + N_U p_U > (N_R + N_U)/2$ *and when* $N_R + N_U \geq 2$. *Let* $s_i \in \mathcal{S}_R$. *Consider* $\beta_0^i(t)$ *the normalized ratio of the number of sensors agreeing with unreliable sensor* s_i *based on the responses of* N_R *reliable and* $N_U - 1$ *unreliable sensors. This number is proportional to the number of sensors agreeing with unreliable sensor* s_i.

$$\beta_0^i(t) = \frac{\sum_{\substack{k=1 \\ k \neq i}}^{N_R+N_U} I\{x_k(t) = x_i(t)\}}{N_R + N_U - 1} \tag{31}$$

Let $\beta_1^i(t)$ *as a normalized ratio of the number of sensors disagreeing with unreliable sensor* s_i.

$$\beta_0^i(t) = \frac{N_R + N_U - 1 - \sum_{\substack{k=1 \\ k \neq i}}^{N_R+N_U} I\{x_k(t) = x_i(t)\}}{N_R + N_U - 1} \tag{32}$$

$E(\beta_0^i(t) \mid s_i \in S_U, |S_U| = N_R, |S_U| = N_U) < E(\beta_1^i(t) \mid s_i \in S_U, |S_U| = N_R, |S_U| = N_U)$.

Proof:

The first part of the proof is similar to the first part of the proof in Theorem 1. Following the same lines of the proof as in the previous theorem and by invoking symmetry (exchanging p_R, N_R by p_U, N_U), we can prove that:

$$E(\beta_0^i(t) \mid s_i \in S_U, |S_R| = N_R, |S_U| = N_U) = \frac{(N_U - 1)(p_U^2 + q_U^2) + N_R(p_R p_U + q_R q_U)}{N_R + N_U - 1} \tag{33}$$

Let us rewrite the above expression as:

$$\begin{aligned} E(\beta_0^i(t) \mid s_i \in S_U, |S_R| = N_R, |S_U| = N_U) = \\ p_U \frac{(N_U - 1)p_U + N_R p_R}{N_U + N_R - 1} + q_U \frac{(N_U - 1)q_U + N_R q_R}{N_R + N_U - 1} \\ = p_U \frac{(N_U - 1)p_U + N_R p_R}{N_R + N_U - 1} + q_U (1 - \frac{(N_U - 1)p_U + N_R p_R}{N_R + N_U - 1}) \end{aligned} \tag{34}$$

We will now prove that $E(\beta_0^i(t) \mid |S_R| = N_R, |S_U| = N_U) < 1/2$. Following the same arguments of the proof as in Theorem 1 we obtain:

$$\begin{aligned} & N_R p_R + (N_U - 1)p_U > (N_R + N_U)/2 \\ & \Rightarrow \frac{N_R p_R + (N_U - 1)p_U}{N_R + N_U - 1} > \frac{N_R + N_U}{2(N_R + N_U - 1)} \\ & \Rightarrow \frac{(N_U - 1)p_U + N_R p_R}{N_R + N_U - 1} > \frac{1}{2} + \frac{1}{2(N_R + N_U - 1)}. \end{aligned}$$

Let us consider the function $h(.)$ defined by:

$$h(\rho) = p_U \cdot \rho + q_U \cdot (1 - \rho) \tag{35}$$

whence, it is easy to see that: $h(\frac{(N_U - 1)p_U + N_R p_R}{N_R + N_U - 1}) = E(\beta_0^i(t) \mid s_i \in S_U, |S_R| = N_R, |S_U| = N_U)$.

Let us investigate the dynamics of $h(\rho)$ by studying its derivative, $h'(\rho)$. Since $h'(\rho) = 2p_U - 1$, and $p_U < 1/2$, we see that $2p_U - 1 < 0$ which is equivalent to the conclusion that $h'(\rho) < 0$. Therefore $h(x)$ is a strictly *decreasing* function.

As a boundary condition, we see that $h(1/2) = 1/2 p_U + 1/2 q_U = 1/2$. Indeed, by virtue of the fact that the function $h(.)$ is strictly decreasing we obtain:

$$\text{If } \rho > 1/2 \quad \Rightarrow \quad h(\rho) < h(1/2) = 1/2. \tag{36}$$

In particular, we now apply the inequality (36) for the particular case when $\rho = \frac{(N_U-1)p_U+N_R p_R}{N_R+N_U-1}$. We know from [38] that $\frac{(N_U-1)p_U+N_R p_R}{N_R+N_U-1} > 1/2$.

Consequently, we obtain:

$$h(\tfrac{(N_U-1)p_U+N_R p_R}{N_R+N_U-1}) < 1/2$$

which is equivalent to:

$$E(\beta_0^i(t) \mid s_i \in S_U, |S_R| = N_R, |S_U| = N_U) < 1/2,$$

The last element of the proof is to observe that the following complementary in our LA design.

$\beta_0^i(t) = 1 - \beta_1^i(t)$

Thus, $E(\beta_0^i(t) \mid s_i \in S_U, |S_R| = N_R, |S_U| = N_U) = 1 - E(\beta_1^i(t) \mid s_i \in S_U, |S_R| = N_R, |S_U| = N_U)$

Thus,

$$E(\beta_0^i(t) \mid s_i \in S_U, |S_R| = N_R, |S_U| = N_U) < E(\beta_1^i(t) \mid s_i \in S_U, |S_R| = N_R, |S_U| = N_U).$$

proving the theorem. □

4.2. *Construction of the Learning Automata*

The results that we presented in the previous section form the basis of our LA-based solution. We explain this below, including the strategy by which the concept of ”proportionality” is invoked. We shall present first a baseline algorithm that we shall call S-LA [1]. Thereafter, we will also present two additional algorithms that use too the concept of S-Model environment.

[1]S reckons S-Model environment.

In the partitioning strategy, with each sensor s_i we associate a 2-action S-Model Learning automaton $\mathcal{A}^i$, $(\Sigma^i, \Pi^i, \Gamma^i, \Upsilon^i, \Omega^i)$, where Σ^i is the set of actions, Π^i is the set of action probabilities, Γ^i is the set of feedback inputs from the Environment, and Υ^i is the set of action probability updating rules.

1. *The set of actions of the automaton*: (Σ^i)
   The two actions of the automaton are α_k^i, for $k \in \{0, 1\}$, i,e, α_0^i and α_1^i
2. *The action probabilities*: (Π^i)
   $P_k^i(n)$ represent the probabilities of selecting the action α_k^i, for $k \in \{0, 1\}$, at step n. Initially, $P_k^i(0) = 0.5$, for $k = 0, 1$.
3. *The feedback inputs from the Environment to each automaton*: (Γ^i)

   Let the automaton select either the the action α_0^i or α_1^i. Then, the responses from the Environment is specified as fellows:

   - $\beta_0^i(t)$ response to action α_0^i
   - $\beta_1^i(t)$ response to action α_1^i

   Consider $\beta_0^i(t)$ as a normalized ratio of the number of sensors agreeing with sensor s_i.

$$\beta_0^i(t) = \frac{\sum_{\substack{k=1 \\ k \neq i}}^{N_R+N_U} I\{x_k(t) = x_i(t)\}}{N_R + N_U - 1} \tag{37}$$

   Let $\beta_1^i(t)$ as a normalized ratio of the number of sensors disagreeing with sensor s_i.

$$\beta_1^i(t) = \frac{N_R + N_U - 1 - \sum_{\substack{k=1 \\ k \neq i}}^{N_R+N_U} I\{x_k(t) = x_i(t)\}}{N_R + N_U - 1} \tag{38}$$

   A brief explanation about the feedback could be beneficial.

   (a) When the LA system chooses action α_0^i, in which case the the reward signal is proportional to the number of sensors agreeing with s_i.

(b) Alternatively, when the LA system chooses action α_1^i, in which case the reward signal is proportional to the number of sensors disagreeing with s_i.

4. *The action probability updating rules*: (Υ^i)
The way the action probability vector is updated leads to three possible different algorithms which we shall explain in Section 4.3.

4.3. Design of the update rules

We will present three different algorithms for updating the action probability vector.

4.3.1. Update rules for Algorithm 1: S-LA

First of all, since we are using the S-Model learning scheme [14, 18]. In the rest of the article, we shall call the following LA algorithm as S-LA. If α_k^i for $k \in \{0,1\}$ was chosen then, for $j \in \{0,1\}$. The LA update equations are given by:

$$P_j^i(t+1) \leftarrow P_j^i(t+1) + G\beta_k^i(t)(\delta_{jk} - P_j^i(t)) \tag{39}$$

where $0 < G \ll 1$ and:

$$\delta_{jk} = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{else} \end{cases} \tag{40}$$

The above equations can be re-written as follows. If α_k^i for $k \in \{0,1\}$ was chosen then,

$$\begin{aligned} P_k^i(t+1) &\leftarrow P_k^i(t) + G\beta_k^i(t) \times (1 - P_k^i(t)) \\ P_{1-k}^i(t+1) &\leftarrow 1 - P_k^i(t+1). \end{aligned}$$

The informed reader observes that, if the chosen action is reliable, it will see its probability increased by a quantity proportional to $\beta_0^i(t)$ which is the normalized ratio of the number of sensors agreeing with the sensor in question

[2]. In this case, the higher $\beta_0^i(t)$, the higher increase in the probability of the reliable action and vice-versa. By way of symmetry, similar explanation applies for the case where the chosen action is unreliable. In such case, the increase in the action probability is proportional to $\beta_1^i(t)$ which is the normalized ratio of the number of sensors disagreeing with the sensor in question.

4.3.2. Update rules for Algorithm 2: G_1 S-LA

We provide the design of the G_1 S-LA due to Simha et al. [27]. According to the G_1 S-LA algorithm [27], actions that have a higher average reward than the overall average reward of all actions (including itself) have their probability increased, whereas actions that have an average reward below the overall average reward have their probabilities decreased.

The details of the algorithm are given below:

Let $k \in \{0, 1\}$ be the chosen action at time instant t. The probability vector is updated as follows:

$$P_k^i(t+1) \leftarrow P_k^i(t) + G(\bar{\beta}_k^i(t) - \frac{\bar{\beta}_k^i(t)+\bar{\beta}_{1-k}^i(t)}{2})$$
$$P_{1-k}^i(t+1) \leftarrow 1 - P_k^i(t+1),$$

where $0 < G \ll 1$ is the update parameter $\bar{\beta}_k^i(t)$ is the time average of the reward.

$\bar{\beta}_k^i(t)$ represents the estimated average reward obtained for action k since the first step.

$$\bar{\beta}_k^i(t) = \frac{\sum_{l=1}^{t} J(l,k)\beta_k^i(l)}{\sum_{l=1}^{t} J(l,k)}$$

where $J(l,k) = 1$ if the action k action was deployed at the l^{th} time step.

Given that we only have two actions, we can write the difference between the average reward of action k and the the overall average reward:

[2]Please note that the increase in the action probability is $G\beta_0^i(t)(1 - P_0^i(t))$.

$$\bar{\beta}^i_k(t) - \frac{\bar{\beta}^i_k(t) + \bar{\beta}^i_{1-k}(t)}{2} = \frac{\bar{\beta}^i_k(t) - \bar{\beta}^i_{1-k}(t)}{2}$$

Therefore, the update algorithm reduces to:

$$P^i_m(t+1) \leftarrow P^i_k(t) + \tfrac{G}{2}(\bar{\beta}^i_k(t) - \bar{\beta}^i_{1-k}(t))$$
$$P^i_{1-k}(t+1) \leftarrow 1 - P^i_k(t+1)$$

4.3.3. Update rules for Algorithm 3: Pursuit S-LA

Inspired by the family of pursuit LA algorithm [23, 2, 39], we design a novel pursuit LA for S-Model that pursues the action that has the highest "average reward" among the two actions. Please note that the classical pursuit LA found in the literature [23, 2, 39] operate only with binary feedback while our scheme uses continuous feedback. Now, we shall provide the details of our Pursuit S-LA algorithm that is discretized. In fact, the concept of discretization takes place by updating the action probability using a fixed quantity G.

- Choose an action k according to the probability vector.
- Update $\bar{\beta}^i_k(t)$
- Let d the index of the action which has the maximum action average reward estimate.

$$P^i_d(t+1) \leftarrow Min(P^i_d(t) + G, 1)$$
$$P^i_{1-d}(t+1) \leftarrow 1 - P^i_d(t+1)$$

4.4. Optimality Results

At this juncture, we shall present the optimality results of the algorithms.

4.4.1. Optimality of Algorithm 1: S-LA

We give a theorem that documents the optimality of the S-LA algorithm.

Theorem 3. *Consider the scenario when* $(N_R - 1)p_R + N_U p_U > (N_R + N_U)/2$ *and that* $N_R + N_U \geq 2$. *Given the S-LA scheme with a parameter G which is arbitrarily close to zero, the following is true:*

$$\text{If } s_i \in \mathcal{S}_R, \quad \text{then } \lim_{G \to 0} \lim_{n \to \infty} P_0^i(n) \to 1$$
$$\text{If } s_i \in \mathcal{S}_U, \quad \text{then } \lim_{G \to 0} \lim_{t \to \infty} P_1^i(n) \to 1.$$

Proof: To prove the theorem, we again treat the two cases separately.

Case 1: $s_i \in \mathcal{S}_R$. Based on the result of Theorem 1, we can see that the inequality $E(\beta_0^i(t)) > E(\beta_1^i(t))$ holds implying that for this case, action α_0^i is the optimal one. Therefore, using the results from [14, 18], $P_0^i(t) \to 1$ as $t \to \infty$ and $G \to 0$.

Case 2: $s_i \in \mathcal{S}_U$. In this case, based on the result of Theorem 2, we see that the following inequality holds: $E(\beta_1^i(t)) > E(\beta_0^i(t))$. This implies that action α_1^i is the optimal one, and for this action:

$P_1^i(t) \to 1$ as $t \to \infty$ and $G \to 0$.

The theorem is thus proven. □

4.4.2. Optimality of Algorithm 2: G_1 *S-LA*

The optimalty of the G_1 S-LA algorithm is a direct consequence of the work due to Simha et al. [27].

4.4.3. Optimality of Algorithm 3: Pursuit S-LA

Conjecture 1. *Consider the scenario when* $(N_R - 1)p_R + N_U p_U < (N_R + N_U)/2 - 1$ *and that* $N_R + N_U \geq 2$. *Given the Pursuit S-LA scheme with a parameter G which is arbitrarily close to zero, the following is true:*

$$\text{If } s_i \in \mathcal{S}_R, \quad \text{then } \lim_{G \to 0} \lim_{t \to \infty} P_0^i(t) \to 1$$
$$\text{If } s_i \in \mathcal{S}_U, \quad \text{then } \lim_{G \to 0} \lim_{t \to \infty} P_1^i(t) \to 1.$$

The proof of Conjecture 1 is beyond the aim of this article and we allude to the proofs reported in [39] as a potential possible way to justify it.

4.5. Communication Model

A possible message exchange model is depicted in Figure 1. In the first step, all sensors observe the ground truth $T(t)$. Each sensor s_i reports its own version of the ground truth called x_i. In step 2, we envisage an aggregation center that collects all observations from the pool of N sensors. This is a realistic assumption since sensor fusion is usually done in a centralized manner. Given sensor s_i, to which attached Learning automaton $\mathcal{A}^i$, there is a need to compute the individual feedback from the environment, which is in this case $\beta_0^i(t)$ or $\beta_1^i(t)$ depending on which action was chosen. A naive manner to compute the feedback involves contacting the rest of the sensors and comparing own reading against their individual readings as seen in Eq. (37) and Eq. (38). However, in order to query all the sensors, each sensor s_i needs to receive $N-1$ readings and to send its reading to the rest of sensors $N-1$. In other words, the number of exchanged messages is $N(N-1)$ for the whole pool of sensors, which is unfortunately quadratic. Such intensive message exchange is not desired in the context of sensor networks. We shall rather use a simple but rather subtle trick that involves $X(t) = \sum_{k=1}^{N} x_k(t)$, which is the sum of votes supporting the ground truth is 1. According to this simple trick, each sensor can compute its feedback based on $X(t)$. In fact, if $x_i = 1$, the number of agreeing sensor with s_i is simply the aggregate $X(t)-1$, therefore $\beta_0^i(t) = \frac{X(t)-1}{N_R+N_U-1}$ and $\beta_1^i(t) = 1 - \beta_0^i(t)$. We shall now show how we are able to derive the latter expression. If $x_i = 1$, then:

$$\beta_0^i(t) = \frac{\sum\limits_{\substack{k=1\\k\neq i}}^{N_R+N_U} I\{x_k(t) = x_i(t)\}}{N_R + N_U - 1}$$
$$= \frac{\sum\limits_{\substack{k=1\\k\neq i}}^{N_R+N_U} I\{x_k(t) = 1\}}{N_R + N_U - 1}$$
$$= \frac{\sum\limits_{k=1}^{N_R+N_U} I\{x_k(t) = 1\} - I\{x_i(t) = 1\}}{N_R + N_U - 1}$$
$$= \frac{X(t) - 1}{N_R + N_U - 1} \tag{41}$$

Whereas, if $x_i = 0$, the number of agreeing sensors is $N - X(t) - 1$. In this case where $x_i = 0$, by virtue of normalization, we obtain $\beta_0^i(t) = \frac{N-X(t)-1}{N_R+N_U-1}$. The value of $\beta_1^i(t)$ is deduced as $1 - \beta_0^i(t)$. Therefore, the message exchange is reduced to a single sent message per sensor containing its reading, and one received message per sensor containing the aggregate $X(t)$. The aggregate $X(t)$ can be broadcasted by the aggregation center using legacy broadcast protocols. In other words, $2N$ messages for the whole pool of sensors which is a reasonable number (linear message complexity) compared to the naive approach which requires $N(N-1)$ messages (quadratic message complexity). It is known that the LA update equations are simplistic and yield therefore negligible energy consumption. The part that consumes energy in sensor networks is mostly the communication part [13] which is significantly reduced thanks to introducing the idea of broadcasting the aggregate $X(t)$. Therefore, the non-naive approach seems efficient in terms of energy consumption due to the low number of exchanged messages and the lightweight computation complexity of LA.

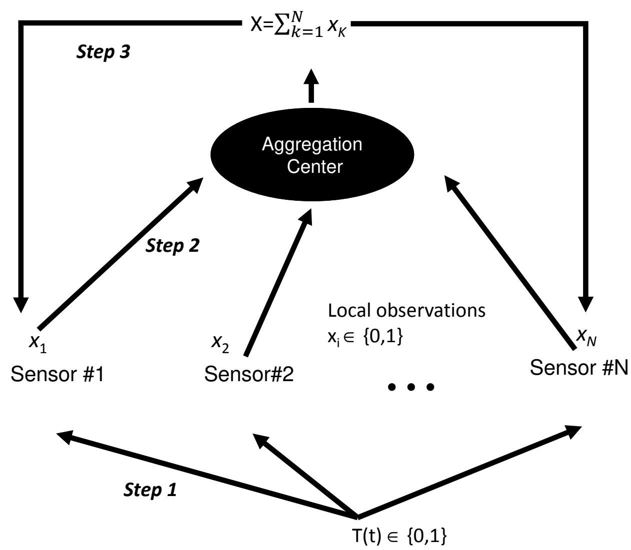

Figure 1: Message Exchange Model for Aggregation Center

5. Experimental results

The performance of the LA-based partitioning in terms of accuracy and convergence time, have been rigorously tested by simulation in a variety of parameter settings, and the results that we have obtained are truly conclusive. In the experiments, the settings were chosen so that the condition $N_R p_R + (N_U - 1) p_U > (N_R + N_U)/2$ was met, reflecting the phenomenon where "the truth prevails over lying".

We report some comparisons results of the three devised S-Model based LA algorithms: Pursuit S-LA, G_1 S-LA and the S-LA against the legacy L_{RI}. The LA is deemed to have converged if one of its action probabilities attained the value $1 - \epsilon$[3]. Formally:

- If $P_0^i(n) \geq 1 - \epsilon$, then the LA has converged to the action α_0^i;
- If $P_1^i(n) \geq 1 - \epsilon$, then the LA has converged to the action α_1^i.

We also initialized all the LA at time instant $t = 0$, to have the values: $P_0^i(t) = P_1^i(t) = 0.5$.

For the sake of clarity, we merely present below the algorithmic description of the S-LA. The algorithmic descriptions of the G_1 S-LA and the Pursuit S-LA are not included here for the sake of brevity and can be easily obtained based on the description in Section 4.3.

Algorithm S-LA

Initialization:
1. $P_k^i(0) = 0.5$, for $k \in \{0, 1\}$.
2. t:=1.

Method:
Loop

1. Select an action, α_k^i, for $k \in \{0, 1\}$, by randomly sampling using the action probability vector $[P_0^i(t), P_1^i(t)]$.
2. Sensor s_i observes $x_i(t)$

[3] The value of ϵ was set to be 0.01.

3. Compute number of agreeing sensors with s_i: $\sum_{\substack{k=1 \\ k \neq i}}^{N_R+N_U} I\{x_k(t) = x_i(t)\}$
4. Deduce β_k^i as per Eq. (37) or Eq. (38) according to whether $k = 0$ or $k = 1$ respectively.
5. Update action probability vector

$$P_k^i(t+1) \leftarrow P_k^i(t) + G\beta_k^i(t) \times (1 - P_k^i(t))$$
$$P_{1-k}^i(t+1) \leftarrow 1 - P_k^i(t+1).$$

6. /*If any $P_k^i(t+1) \geq 1 - \epsilon$, make $P_k^i(t+1)$ jump to 1 and break the loop*/
   If $\exists k \in \{0, 1\}$ such that $P_k^i(t+1) \geq 1 - \epsilon$
      $P_k^i(t+1) = 1$
      $Break$
   EndIf
   $t = t + 1$

End Algorithm S-LA

We computed the average convergence time in an ensemble of $1,000$ experiments for all the LA associated with the sensors in $\mathcal{S}_R$ and for those in $\mathcal{S}_U$ to converge, in addition, the average convergence time for all the LA to converge. In simple terms, in order to compute the average time for all sensors in $\mathcal{S}_R$, respectively $\mathcal{S}_U$, to converge, we record the maximum number of iterations it takes for all N_R in $\mathcal{S}_R$ to converge, respectively $\mathcal{S}_U$, in each of the experiments and we average out that number over all experiments.

The update parameter G was set to 0.05. We shall provide two representative scenarios:

- In the first scenario, we choose N_R to be twice N_U and thus, we call this case as biased case since S_R forms a clear majority. We provide experimental results cataloguing the convergence speed and accuracy for the cases where $(N_R, N_U) = (20, 10)$ and $(N_R, N_U) = (200, 100)$.

- In the second scenario, we choose to have N_R to be equal to N_U and thus, we call this case as balanced case. Namely, we provide experimental results cataloguing the convergence speed and accuracy for the cases where $(N_R, N_U) = (20, 20)$ and $(N_R, N_U) = (200, 200)$.

We also chose the value of (p_R, p_U) in a manner so that the condition $(N_R - 1)p_R + N_U p_U > (N_R + N_U)/2$ holds true in both scenarios.

We draw a set of interesting remarks:

- In Table 1, we report the average convergence time for $(N_R, N_U) = (20, 10)$ where we easily have a majority of reliable sensors. According to the Table, the Pursuit S-LA is the fastest to converge. It is almost 10 times faster than the Legacy L_{RI}.

  For example, in Table 1, we see that for $(p_R, p_U) = (0.8, 0.1)$ the Pursuit S-LA is more than 10 times faster than the L_{RI}. In fact, we report 32.358

convergence time while the L_{RI} converges within 355.311 time instants. Moreover, we see that for $(p_R, p_U) = (0.95, 0.2)$ we report 17.994 convergence time while the L_{RI} converges within 194.679 time instants.

As we increase the number of sensors in Table 3 by a factor of 10 compared to Table 1, we observe still that the pursuit S-LA still outperforms the L_{RI} but with a smaller factor, namely, 8.

For example, in Table 3, we report 34.904 convergence time while the L_{RI} converges within 283.546 time instants which gives a factor of 8. Moreover, we see that for $(p_R, p_U) = (0.95, 0.2)$ we report 29.566 convergence time while the L_{RI} converges within 260.618 time instants which makes it faster by a factor of almost 9.

- The G_1 S-LA and the L_{RI} have comparable results. The G_1 S-LA performance becomes worse than the S-LA as we increase by ten the number of sensors in Table 3.
- From the Table 3, the quickest convergence time takes place when p_R goes to 1 and p_U goes to 0 which in this case $(0.95, 0.1)$.
- The slowest convergence time takes when p_R and p_U goes both to 0.5 rendering the environment "difficult", i.e, difficult to differentiate between the identity of the sensors.
- We observe that the convergence time differs according to whether $s_i \in \mathcal{S}_R$ or $s_i \in \mathcal{S}_U$. We shall give a brief account of why we observe such difference. The reason for the latter difference in convergence time is the difference in :
  - $E(\beta_0^i(t))$ given by Eq. (11) for $s_i \in \mathcal{S}_R$.
  - $E(\beta_1^i(t))$ given by Eq. (33) for $s_i \in \mathcal{S}_U$.

  For example, from Table 3, we observe that whenever $(p_R, p_U) = (0.8, 0.1)$, $E(\beta_0^i(t)) = 0.535$ for $s_i \in \mathcal{S}_R$ while $E(\beta_1^i(t)) = 0.566$ for for $s_i \in \mathcal{S}_U$. We see that the environment is "easier" for $s_i \in \mathcal{S}_R$ than for $s_i \in \mathcal{S}_U$ and thus, theoretically, the convergence for $s_i \in \mathcal{S}_U$ is expected to be faster in this case. This confirmed too by the experimental results, where we see that, for example, for $(p_R, p_U) = (0.8, 0.1)$, the Pursuit S-LA records a convergence time of 31.147 and 14.027 for $s_i \in \mathcal{S}_R$ and $s_i \in \mathcal{S}_U$ respectively. In this particular case, the convergence for $s_i \in \mathcal{S}_U$ is approximately twice faster than for $s_i \in \mathcal{S}_R$.

  The opposite takes place when $(p_R, p_U) = (0.95, 0.2)$, i.e, the convergence for $s_i \in \mathcal{S}_R$ is faster than that for $s_i \in \mathcal{S}_U$. In this case, $E(\beta_0^i(t)) = 0.672$ (see Eq. (11)) for $s_i \in \mathcal{S}_R$ while $E(\beta_1^i(t)) = 0.630$ for for $s_i \in \mathcal{S}_U$. Therefore, the the convergence for $s_i \in \mathcal{S}_R$ is expected to be faster in this case. This is confirmed too in the Table, where we record that the Pursuit S-LA gives a convergence time of 11.00 and 16.573 for $s_i \in \mathcal{S}_R$ and $s_i \in \mathcal{S}_U$ respectively

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.8 , 0.1)	31.147	14.027	32.358	3656.21	774.21	3657.21	849.913	139.545	850.913	354.052	195.893	355.311
(0.8 , 0.2)	26.435	19.96	29.337	1795.274	756.325	1796.678	381.142	151.089	382.191	273.421	232.801	281.953
(0.85 , 0.1)	19.796	12.3	21.31	1483.86	587.716	1485.061	283.828	106.101	284.828	211.537	156.253	213.819
(0.85 , 0.2)	18.359	18.067	22.62	955.807	621.622	959.38	191.345	124.806	192.466	197.178	202.962	216.813
(0.9 , 0.1)	14.15	11.651	16.129	832.423	479.748	833.91	151.692	86.19	152.692	154.292	140.905	160.157
(0.9 , 0.2)	13.957	17.357	19.734	626.532	510.064	636.129	116.191	105.275	120.539	154.759	193.178	196.458
(0.95 , 0.1)	11.176	11.356	13.73	546.23	391.298	548.991	94.374	72.767	95.571	124.125	136.607	141.27
(0.95 , 0.2)	11.0	16.573	17.994	442.597	427.004	465.247	76.507	93.277	95.003	127.163	193.228	194.679

Table 1: Average convergence time for the case when $(N_R, N_U) = (20, 10)$

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.8 , 0.1)	0.994	0.999	0.9963	0.9296	0.999	0.953	1.0	1.0	1.0	1.0	1.0	1.0
(0.8 , 0.2)	0.997	0.998	0.9981	0.9962	0.999	0.997	1.0	1.0	1.0	1.0	1.0	1.0
(0.85 , 0.1)	0.99895	1.0	0.999	0.998	1.0	0.998	1.0	1.0	1.0	1.0	1.0	1.0
(0.85 , 0.2)	0.999	0.9986	0.999	0.999	1.0	0.999	1.0	1.0	1.0	1.0	1.0	1.0
(0.9 , 0.1)	0.9999	1.0	0.999	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
(0.9 , 0.2)	0.99995	0.998	0.9993	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
(0.95 , 0.1)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
(0.95 , 0.2)	1.0	0.998	0.999	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 2: Accuracy for the case when $(N_R, N_U) = (20, 10)$

- The G_1 S-LA yields the worst performance in terms of convergence speed. The reason is that the updates of the reward probability is proportional to the difference of the average reward of both actions k and $1-k$. Since the latter difference is small, the increase in the probability is small too and thus slow convergence. For example, for $(p_R, p_U) = (0.8, 0.1)$ in Table 1, the overall convergence time for G_1 S-LA is 849.913 while the Pursuit S-LA, S-LA and L_{RI} record respectively 32.358, 139.545 and 355.311. Thus, the G_1 S-LA is the slowest algorithm in terms of convergence speed in this case.

- In Table 2 and Table 4 we report the average convergence accuracy over 1000 experiments. We observe that the L_{RI} has slightly better accuracy than the Pursuit S-LA. In fact, the S-LA yields a performance in some cases around 0.99 while the L_{RI} has an optimal accuracy of 1. We see that the Pursuit S-LA is faster that the L_{RI} at the cost of a negligible loss of accuracy. We observe too that as we increase the number of sensors by 10, the convergence accuracy of Pursuit S-LA improves as reported in Table 4 compared to Table 2

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.8 , 0.1)	33.887	17.818	34.904	373.711	218.879	374.711	2645.976	1623.976	2649.362	282.272	188.878	283.546
(0.8 , 0.2)	33.132	29.647	36.053	262.499	232.529	264.894	1651.476	1433.194	1682.072	285.622	267.943	295.952
(0.85 , 0.1)	25.562	17.379	26.842	208.604	154.462	209.604	1387.807	1081.308	1398.216	234.724	187.858	237.031
(0.85 , 0.2)	25.356	28.811	31.551	164.042	186.742	188.615	1046.353	1052.894	1109.823	239.873	262.983	270.041
(0.9 , 0.1)	19.501	17.107	21.538	132.475	122.028	134.567	889.164	802.835	907.008	196.022	183.034	203.368
(0.9 , 0.2)	19.391	28.458	29.757	109.851	157.903	158.903	729.195	826.423	836.963	203.052	260.415	262.737
(0.95 , 0.1)	14.615	17.02	18.738	88.401	101.297	102.548	624.953	627.397	655.647	162.572	179.017	184.377
(0.95 , 0.2)	14.642	28.546	29.566	75.581	138.275	139.275	534.95	684.393	686.551	166.451	259.419	260.618

Table 3: Average convergence time for the case when $(N_R, N_U) = (200, 100)$

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.8 ,0.1)	0.999	1.0	0.999	1.0	1.0	1.0	0.996	0.999	0.997	1.0	1.0	1.0
(0.8 , 0.2)	0.999	0.999	0.999	1.0	1.0	1.0	0.999	0.999	0.999	1.0	1.0	1.0
(0.85 , 0.1)	0.9999	1.0	0.999	1.0	1.0	1.0	0.999	1.0	0.999	1.0	1.0	1.0
(0.85 , 0.2)	0.999	0.999	0.999	1.0	1.0	1.0	1.0	0.999	0.999	1.0	1.0	1.0
(0.9 , 0.1)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
(0.9 , 0.2)	1.0	0.9999	0.999	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
(0.95 , 0.1)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
(0.95 , 0.2)	1.0	0.9999	0.999	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 4: Accuracy for the case when $(N_R, N_U) = (200, 100)$

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.75 , 0.45)	36.506	123.439	124.441	306.317	2638.819	2639.819	1484.729	4849.221	4850.221	348.704	1647.921	1648.921
(0.75 , 0.4)	43.685	87.743	88.864	416.191	1257.994	1258.994	2075.922	3963.42	3977.232	418.396	985.353	986.47
(0.75 , 0.35)	60.071	79.534	82.03	667.384	1048.226	1049.575	3174.14	3949.2	4171.386	594.822	832.074	844.092
(0.75 , 0.3)	112.652	108.399	119.009	1799.474	1401.032	1832.618	5112.929	4670.952	5662.717	1283.817	1040.968	1354.298
(0.8 , 0.45)	24.648	116.997	118.005	193.827	2163.856	2164.856	1002.503	4370.92	4371.92	250.813	1556.19	1557.19
(0.8 , 0.4)	27.605	77.471	78.494	241.584	961.998	962.998	1262.017	3267.715	3268.736	277.377	804.248	805.286
(0.8 , 0.35)	30.906	59.951	61.188	331.121	712.443	713.443	1716.475	2902.92	2919.172	338.769	604.91	606.253
(0.8 , 0.3)	43.648	55.275	58.695	526.054	701.021	704.577	2730.808	3060.257	3302.393	482.809	575.425	594.835

Table 5: Average convergence time for the case when $(N_R, N_U) = (20, 20)$

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.75 , 0.45)	0.999	0.818	0.909	1.0	0.999	0.999	0.999	0.827	0.9134	1.0	0.966	0.983
(0.75 , 0.4)	0.999	0.919	0.959	1.0	1.0	1.0	0.995	0.916	0.955	1.0	0.996	0.998
(0.75 , 0.35)	0.993	0.956	0.975	1.0	1.0	1.0	0.968	0.928	0.948	0.999	0.999	0.999
(0.75 , 0.3)	0.942	0.965	0.954	1.0	1.0	1.0	0.814	0.873	0.843	0.987	0.995	0.991
(0.8 , 0.45)	0.999	0.837	0.918	1.0	0.999	0.999	0.999	0.870	0.935	1.0	0.976	0.988
(0.8 , 0.4)	0.999	0.934	0.967	1.0	1.0	1.0	0.999	0.959	0.979	1.0	0.999	0.999
(0.8 , 0.35)	0.999	0.967	0.983	1.0	1.0	1.0	0.998	0.976	0.987	1.0	0.999	0.999
(0.8 , 0.3)	0.996	0.980	0.988	1.0	1.0	1.0	0.985	0.971	0.978	1.0	0.999	0.999

Table 6: Accuracy for the case when $(N_R, N_U) = (20, 20)$

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.75 , 0.45)	44.231	399.538	400.538	325.59	3282.976	3283.976	1996.772	7805.564	7806.564	366.438	2147.455	2148.455
(0.75 , 0.4)	45.652	159.871	160.871	410.083	1407.051	1408.051	2799.829	6435.39	6437.561	369.265	965.676	966.676
(0.75 , 0.35)	48.497	95.622	96.67	579.474	1106.44	1107.44	4315.948	6452.977	6481.485	377.077	637.802	638.847
(0.75 , 0.3)	64.561	82.908	84.873	1113.158	1415.89	1416.965	7347.199	7955.676	8443.919	509.946	613.78	622.009
(0.8 , 0.45)	32.566	393.424	394.424	210.852	2915.685	2916.685	1295.549	6975.341	6976.341	299.348	2145.444	2146.444
(0.8 , 0.4)	33.335	155.621	156.621	253.462	1162.852	1163.852	1662.713	5176.411	5177.538	297.166	974.234	975.234
(0.8 , 0.35)	33.847	89.386	90.386	319.686	805.246	806.246	2278.318	4649.253	4650.253	296.939	612.6	613.6
(0.8 , 0.3)	35.855	63.434	64.494	455.011	756.269	757.269	3597.135	5172.978	5198.341	306.67	459.701	460.957

Table 7: Average convergence time for the case when $(N_R, N_U) = (200, 200)$

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.75 , 0.45)	1.0	0.974	0.987	1.0	1.0	1.0	0.999	0.824	0.911	1.0	0.983	0.991
(0.75 , 0.4)	1.0	0.993	0.996	1.0	1.0	1.0	0.997	0.913	0.955	1.0	0.999	0.999
(0.75 , 0.35)	0.999	0.997	0.998	1.0	1.0	1.0	0.979	0.916	0.948	1.0	1.0	1.0
(0.75 , 0.3)	0.999	0.998	0.999	1.0	1.0	1.0	0.873	0.836	0.855	1.0	1.0	1.0
(0.8 , 0.45)	1.0	0.974	0.987	1.0	0.999	0.999	0.999	0.866	0.933	1.0	0.982	0.991
(0.8 , 0.4)	1.0	0.993	0.996	1.0	1.0	1.0	0.999	0.956	0.978	1.0	0.999	0.999
(0.8 , 0.35)	1.0	0.997	0.998	1.0	1.0	1.0	0.999	0.971	0.985	1.0	0.999	0.999
(0.8 , 0.3)	1.0	0.999	0.999	1.0	1.0	1.0	0.990	0.961	0.976	1.0	1.0	1.0

Table 8: Accuracy for the case when $(N_R, N_U) = (200, 200)$

6. Conclusion

The main stream of research on identifying unreliable sensors assumes that the sensor reliability can be assessed through comparison with the ground truth in a online manner or offline manner. In this paper, we tackle the counterpart case where the ground truth is unknown by invoking LA as a tool. The key idea behind our solution is the fact that comparing the readings of a sensor to the rest of the sensors gives an invaluable information about its reliability. Compared to the-state-of-the-art initial solution reported in [38], our solution is general in a sense that it does not impose any extra constraint on the parity of the number of sensors. Furthermore, we are able to devise a novel algorithm called Pursuit S-LA that is more than ten fold faster than the the-state-of-the-art solution while yielding high accuracy.

In the current work, we have only treated the case of binary sensor readings. We intend to investigate the case where the readings of the sensors admit continuous values in a future study. Furthermore, asymmetric error models have not been studied in this paper and they remain an interesting future research avenue.

ACKNOWLEDGMENT

This work was supported by the grants (No. TIN2013-40658-P and TIN2016-75850-R) from the FEDER funds.

Appendix A. Comparison Results with same settings as in [38]

In this experiment, the update parameter G was set to a larger value, 0.2, in order to allow comparison to the same settings as in [38][4]. The results confirm the superiority of the devised schemes compared to the L_{RI} scheme [38] . The analysis of the results shows a decline in the accuracy of all the learning schemes due to increasing G and in same time and an increase in convergence speed compared to the results reported in Section 5 where G was set to 0.05. In fact, specially for the Pursuit S-LA, the accuracy is reduced due to increasing the step size of the update. We adhere to the settings where $(N_R, N_U) = (20, 10)$ and $(N_R, N_U) = (400, 200)$. In Table A.9 and Table A.11, we report the comparisons results concerning the convergence time where $(N_R, N_U) = (20, 10)$ and $(N_R, N_U) = (400, 200)$ respectively. While, in Table A.10 and Table A.12, we report the comparisons results concerning the convergence time where $(N_R, N_U) = (20, 10)$ and $(N_R, N_U) = (400, 200)$ respectively.

[4] For the intrinsic parameter called θ of the L_{RI} scheme in [38] corresponds here to $1 - G$

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.8, 0.1)	11.083	5.303	12.19	136.82	55.402	137.821	336.201	215.767	344.506	62.21	46.94	63.31
(0.8, 0.2)	10.435	8.206	11.993	88.12	66.45	92.344	276.263	210.853	291.522	60.84	60.77	61.21
(0.85, 0.1)	7.886	4.244	9.105	66.526	39.034	67.695	259.004	167.895	264.79 3	47.88	38.912	49.12
(0.85, 0.2)	7.808	7.262	9.934	50.494	51.9	58.796	203.745	173.729	221.61	47.51	54.39	56.71
(0.9, 0.1)	5.599	3.956	7.12	38.97	30.453	41.067	178.064	130.171	184.528	37.908	35.465	39.313
(0.9, 0.2)	5.634	6.578	8.418	31.897	43.739	45.906	141.276	141.783	162.716	38.12	50.44	51.28
(0.95, 0.1)	3.652	3.746	5.784	24.349	25.496	28.714	126.641	105.719	134.218	31.38	34.43	36.12
(0.95, 0.2)	3.887	6.286	7.638	20.685	38.399	39.494	102.537	120.608	129.425	31.542	39.62	41.79

Table A.9: Average convergence time for the case when $(N_R, N_U) = (20, 10)$

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.8, 0.1)	0.836	0.88	0.85	1.0	1.0	1.0	0.798	0.929	0.842	0.993	0.719	0.902
(0.8, 0.2)	0.860	0.850	0.857	1.0	1.0	1.0	0.894	0.928	0.905	0.992	0.864	0.949
(0.85, 0.1)	0.878	0.885	0.88	1.0	1.0	1.0	0.920	0.974	0.938	0.989	0.941	0.973
(0.85, 0.2)	0.886	0.864	0.879	1.0	1.0	1.0	0.965	0.964	0.964	0.984	0.974	0.981
(0.9, 0.1)	0.895	0.891	0.893	1.0	1.0	1.0	0.978	0.990	0.982	0.999	0.735	0.911
(0.9, 0.2)	0.896	0.869	0.887	1.0	1.0	1.0	0.991	0.981	0.988	0.999	0.878	0.959
(0.95, 0.1)	0.899	0.894	0.897	1.0	1.0	1.0	0.994	0.995	0.995	0.998	0.949	0.982
(0.95, 0.2)	0.899	0.873	0.89	1.0	1.0	1.0	0.998	0.990	0.995	0.999	0.982	0.993

Table A.10: Accuracy for the case when $(N_R, N_U) = (20, 10)$

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.8, 0.1)	19.762	9.805	20.772	131.667	75.41	132.667	558.502	444.71	570.278	91.24	51.83	93.45
(0.8, 0.2)	20.55	18.884	22.251	103.467	95.59	108.118	455.169	412.62	479.512	91.546	83.36	92.78
(0.85, 0.1)	14.29	9.566	15.353	78.026	55.909	79.094	404.529	333.115	415.822	70.324	51.445	73.128
(0.85, 0.2)	15.132	17.258	18.758	65.415	81.536	83.418	320.651	330.58	357.284	70.73	82.06	83.17
(0.9, 0.1)	10.531	9.533	12.152	49.448	46.227	52.316	277.305	250.47	291.46	55.58	51.81	57.18
(0.9, 0.2)	10.966	16.574	17.634	43.174	71.453	72.464	216.547	266.903	272.23	55.89	81.93	82.672
(0.95, 0.1)	7.404	9.258	10.487	31.726	40.087	41.368	187.616	196.1	209.427	44.55	50.88	51.957
(0.95, 0.2)	7.587	16.582	17.585	28.341	65.247	66.249	151.522	224.596	226.486	31.542	39.62	41.59

Table A.11: Average convergence time for the case when $(N_R, N_U) = (400, 200)$

	Pursuit S-LA			G_1 S-LA			S-LA			L_{RI}		
(p_R, p_U)	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$	$s_i \in \mathcal{S}_R$	$s_i \in \mathcal{S}_U$	$s_i \in \mathcal{S}$
(0.8, 0.1)	0.998	0.999	0.999	1.0	1.0	1.0	0.8265	0.895	0.849	0.997	0.730	0.908
(0.8, 0.2)	0.999	0.999	0.999	1.0	1.0	1.0	0.9105	0.912	0.911	0.997	0.886	0.960
(0.85, 0.1)	0.999	0.999	0.999	1.0	1.0	1.0	0.937	0.959	0.944	0.997	0.960	0.985
(0.85, 0.2)	0.999	0.998	0.999	1.0	1.0	1.0	0.972	0.955	0.966	0.997	0.988	0.994
(0.9, 0.1)	0.999	0.999	0.999	1.0	1.0	1.0	0.983	0.984	0.984	0.999	0.731	0.910
(0.9, 0.2)	0.999	0.998	0.999	1.0	1.0	1.0	0.993	0.976	0.988	0.999	0.885	0.961
(0.95, 0.1)	0.999	0.999	0.999	1.0	1.0	1.0	0.997	0.993	0.996	0.999	0.960	0.986
(0.95, 0.2)	1.0	0.997	0.999	1.0	1.0	1.0	0.999	0.987	0.995	0.999	0.988	0.995

Table A.12: Accuracy for the case when $(N_R, N_U) = (400, 200)$

References

[1] M. Agache and B. J. Oommen. Generalized pursuit learning schemes: New families of continuous and discretized learning automata. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 32(6):738–749, December 2002.

[2] M. Agache and B. J. Oommen. Generalized pursuit learning schemes: new families of continuous and discretized learning automata. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 32(6):738–749, 2002.

[3] A. F. Atlassis, N. H. Loukas, and A. V. Vasilakos. The use of learning algorithms in ATM networks call admission control problem: A methodology. *Computer Networks*, 34:341–353, 2000.

[4] A. F. Atlassis and A. V. Vasilakos. The use of reinforcement learning algorithms in traffic control of high speed networks. *Advances in Computational Intelligence and Learning*, pages 353–369, 2002.

[5] M. Barzohar and D. B. Cooper. Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:707–722, 1996.

[6] P. J. Boland. Majority systems and the condorcet jury theorem. *The Statistician*, pages 181–189, 1989.

[7] R. L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5:51–72, 1986.

[8] J. Frolik, M. Abdelrahman, and P. Kandasamy. A confidence-based approach to the self-validation, fusion and reconstruction of quasi-redundant sensor data. *IEEE Transactions on Instrumentation and Measurement*, 50(6):1761–1769, 2001.

[9] O.-C. Granmo and B. J. Oommen. Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata. *IEEE Transactions on Computers*, 59(4):545–560, 2010.

[10] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Information Fusion*, 35:68–80, 2017.

[11] H. Jiang, R. Wang, J. Gao, Z. Gao, and X. Gao. Evidence fusion-based framework for condition evaluation of complex electromechanical system in process industry. *Knowledge-Based Systems*, 124:176–187, 2017.

[12] H. Jin, X. Chen, J. Yang, H. Zhang, L. Wang, and L. Wu. Multi-model adaptive soft sensor modeling method using local learning and online support vector regression for nonlinear time-variant batch processes. *Chemical Engineering Science*, 131:282–303, 2015.

[13] F. Luo, C. Jiang, H. Zhang, X. Wang, L. Zhang, and Y. Ren. Node energy consumption analysis in wireless sensor networks. In *2014 IEEE 80th Vehicular Technology Conference (VTC Fall)*, pages 1–5. IEEE, 2014.

[14] L. Mason. An optimal learning algorithm for s-model environments. *IEEE Transactions on Automatic Control*, 18(5):493–496, 1973.

[15] S. Misra, P. V. Krishna, K. Kalaiselvan, V. Saritha, and M. S. Obaidat. Learning automata-based qos framework for cloud iaas. *IEEE Transactions on Network and Service Management*, 11(1):15–24, 2014.

[16] S. Misra and B. J. Oommen. GPSPA: A new adaptive algorithm for maintaining shortest path routing trees in stochastic networks. *International Journal of Communication Systems*, 17:963–984, 2004.

[17] K. S. Narendra and A. M. Annaswamy. *Stable Adaptive Systems*. Prentice-Hall, New Jersey, 1989.

[18] K. S. Narendra and M. A. L. Thathachar. *Learning Automata: An Introduction*. Prentice-Hall, New Jersey, 1989.

[19] P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis. Learning automata-based polling protocols for wireless lans. *IEEE Transactions on Communications*, 51(3):453–463, 2003.

[20] P. Nicopolitidis, G. I. Papadimitriou, A. S. Pomportsis, P. Sarigiannidis, and M. S. Obaidat. Adaptive wireless networks using learning automata. *IEEE Wireless Communications*, 18(2):75–81, 2011.

[21] M. S. Obaidat, G. I. Papadimitriou, A. S. Pomportsis, and H. S. Laskaridis. Learning automata-based bus arbitration for shared-edium ATM switches. *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, 32:815–820, 2002.

[22] B. J. Oommen and M. Agache. Continuous and discretized pursuit learning schemes: Various algorithms and their comparison. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 31:277–287, 2001.

[23] B. J. Oommen and J. K. Lanctôt. Discretized pursuit learning automata. *IEEE Transactions on systems, man, and cybernetics*, 20(4):931–938, 1990.

[24] B. J. Oommen and T. D. Roberts. Continuous learning automata solutions to the capacity assignment problem. *IEEE Transactions on Computers*, C-49:608–620, 2000.

[25] R. Polikar. Ensemble learning. In *Ensemble machine learning*, pages 1–34. Springer, 2012.

[26] A. S. Poznyak and K. Najim. *Learning Automata and Stochastic Optimization*. Springer-Verlag, Berlin, 1997.

[27] R. Simha and J. F. Kurose. Relative reward strength algorithms for learning automata. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(2):388–398, 1989.

[28] S. Sun, H. Lin, J. Ma, and X. Li. Multi-sensor distributed fusion estimation with applications in networked systems: A review paper. *Information Fusion*, 38:122 – 134, 2017.

[29] M. A. L. Thathachar and B. J. Oommen. Discretized reward-inaction learning automata. *Journal of Cybernetics and Information Science*, pages 24–29, Spring 1979.

[30] M. A. L. Thathachar and P. S. Sastry. *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic, Boston, 2003.

[31] T. Tian, S. Sun, and N. Li. Multi-sensor information fusion estimators for stochastic uncertain systems with correlated noises. *Information Fusion*, 27:126–137, 2016.

[32] M. L. Tsetlin. *Automaton Theory and the Modeling of Biological Systems*. Academic Press, New York, 1973.

[33] S. M. Vahidipour, M. R. Meybodi, and M. Esnaashari. Learning automata-based adaptive petri net and its application to priority assignment in queuing systems with unknown parameters. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(10):1373–1384, 2015.

[34] A. V. Vasilakos, M. P. Saltouros, A. F. Atlassis, and W. Pedrycz. Optimizing QoS routing in hierarchical ATM networks using computational intelligence techniques. *IEEE Transactions on Systems, Man and Cybernetics: Part C*, 33:297–312, 2003.

[35] X. Wang, J. Zhu, Y. Song, and L. Lei. Combination of unreliable evidence sources in intuitionistic fuzzy mcdm framework. *Knowledge-Based Systems*, 97:24–39, 2016.

[36] A. Yazidi, O.-C. Granmo, and B. J. Oommen. Service selection in stochastic environments: a learning-automaton based solution. *Applied Intelligence*, 36(3):617–637, 2012.

[37] A. Yazidi, O.-C. Granmo, B. J. Oommen, M. Gerdes, and F. Reichert. A user-centric approach for personalized service provisioning in pervasive environments. *Wireless Personal Communications*, 61(3):543–566, 2011.

[38] A. Yazidi, B. J. Oommen, and M. Goodwin. On solving the problem of identifying unreliable sensors without a knowledge of the ground truth: The case of stochastic environments. *IEEE Transactions on Cybernetics*, 47(7):1604–1617, July 2017.

[39] X. Zhang, B. J. Oommen, O.-C. Granmo, and L. Jiao. A formal proof of the -optimality of discretized pursuit algorithms. *Applied Intelligence*, 44(2):282–294, 2016.