sensors

MDPI

Article

Sensor Compromise Detection in Multiple-Target Tracking Systems

Juan-Pablo Ramirez-Paredes [1]*, Emily A. Doucette [2], Jess W. Curtis [2] and Victor Ayala-Ramirez [1]

[1] Department of Electronics Engineering, University of Guanajuato, Salamanca, Gto. 36885, Mexico; ayalav@ugto.mx
[2] Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL 32542, USA; emily.doucette@us.af.mil (E.A.D.); jess.curtis@us.af.mil (J.W.C.)
* Correspondence: jpi.ramirez@ugto.mx; Tel.: +52-464-647-9940

Received: 30 December 2017; Accepted: 17 February 2018; Published: 21 February 2018

Abstract: Tracking multiple targets using a single estimator is a problem that is commonly approached within a trusted framework. There are many weaknesses that an adversary can exploit if it gains control over the sensors. Because the number of targets that the estimator has to track is not known with anticipation, an adversary could cause a loss of information or a degradation in the tracking precision. Other concerns include the introduction of false targets, which would result in a waste of computational and material resources, depending on the application. In this work, we study the problem of detecting compromised or faulty sensors in a multiple-target tracker, starting with the single-sensor case and then considering the multiple-sensor scenario. We propose an algorithm to detect a variety of attacks in the multiple-sensor case, via the application of finite set statistics (FISST), one-class classifiers and hypothesis testing using nonparametric techniques.

Keywords: sensor networks; estimation; cyberphysical systems

1. Introduction

Including information from multiple cooperative sensors in a target tracking scenario can increase performance over using a single sensor or uncooperative sensors. For example, a bearings-only sensor requires more than one instance to recover point estimates of the target location. One concern that arises from the application of such a multiple-sensor system is its security. The sensors that comprise the system may come from various manufacturers and may be susceptible to interference by an adversary, either electronically or at their physical location. Therefore, as the number of sensors in a network increases, as does the number of potential vulnerabilities. This concern has been the subject of many studies by the cyberphysical systems community [1–3].

While there are a wide variety of attacks that a sensor network can suffer, one potential vulnerability comes from an adversary compromising some sensor nodes. An attack that completely disables some sensors can be easily detected, and thus appropriate measures can be taken to diminish its impact. Alternatively, an attacker could inject false information or a bias into the system so that its core algorithms fail. Detecting such false information is a challenge, because the specific nature of that kind of disruption is not known in advance.

The computing community has long studied the intrusion detection problem in the context of network security. Most work has consisted of analyzing the data traffic in search of unusual patterns [4–6]. In a wider context, the terms *anomaly* and *outlier* detection describe the search of any patterns that do not fit the nominal operation of a system. Searching for anomalies can be used for different purposes, such as for finding system faults or detecting intrusions [7,8].

In the context of a target tracking system, an attacker would be interested in disrupting the operation of the whole system to prevent some or all targets from being detected or to enable a tracked target to become unobservable or evade pursuit. A multiple-sensor system would be less vulnerable to a full takeover as a result of redundancy. As an alternative, an attacker in control of several nodes could introduce false measurements that deviate the tracker from the true target, making it follow a decoy or causing any estimation algorithms to diverge. Depending on the model, a simple disagreement among sensors may suffice to flag a possible compromise [9].

When the underlying system dynamics follow a linear time-invariant (LTI) model, it has been shown that it is possible to deal with additive disturbance signals, successfully recovering the system state [10–12]. If the specific nature of the attack cannot be anticipated, the search for anomalies in a system can be treated as a one-class classifier [13]. In this case, the nominal operation of the system has to be known beforehand, and a model of the measurements that it produces is built for later comparison. One of the earliest approaches to outlier detection consists of building a probabilistic model, such that inliers are considered to originate from a known distribution [14]. This idea has been extended to multi-dimensional datasets [15]. A problem of multiple-sensor systems, which is closely related to our motivation, is fault detection and isolation (FDI). This issue is often approached by deploying per-sensor strategies, which are combined in a decision-making stage. Some recent works utilize this technique, combined with a priori knowledge of the system dynamics or for a class of systems [16–18].

In this work, we present an algorithm to detect an intrusion in a multiple-sensor system. The algorithm works for a class of attacks that introduce a malicious signal at some sensor nodes with the purpose of deviating a target state estimator from the truth. We have applied the principles of hypothesis testing and the one-class classifier, which collects information about the system during nominal operation. This knowledge has then been applied to Bayesian networks (BNs) to perform an online detection of anomalous behavior in any subset of the sensors.

2. Background

2.1. Problem Statement

We consider a multiple-target tracking system, in which the dynamics of each target are assumed identical and are described by a model of the form

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + w_k \tag{1}$$

The tracking system collects observations from one or more sources, each following the model

$$\mathbf{z}_k = h(\mathbf{x}_k) + \nu_k \tag{2}$$

In order to fully represent a multiple-target distribution, it is necessary to introduce the concept of a finite set. We consider some state space E_s for the targets. The set of all finite subsets of E_s is $\mathcal{F}(E_s)$. Then, the multiple-target state at some time instant k, assuming $M(k)$ targets are present, is given by

$$X_k = \{\mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,M(k)}\} \in \mathcal{F}(E_s) \tag{3}$$

Similarly, E_o is the space of all possible observations, and the $N(k)$ multiple-target measurements at time k are given by

$$Z_k = \{\mathbf{z}_{k,1}, \ldots, \mathbf{z}_{k,N(k)}\} \in \mathcal{F}(E_o) \tag{4}$$

In a manner analogous to the way in which a vector can be a realization of a random process, a finite set is a realization of a random finite set (RFS). The class of RFSs that is considered in the derivation of the probability hypothesis density (PHD) filter is a Poisson point process. The RFS that represents the multiple-target state at time k is

$$\Xi_k = S_k(X_{k-1}) \cup B_k(X_{k-1}) \cup \Gamma_k \tag{5}$$

where $S_k(X_{k-1})$ is the RFS of the targets X_{k-1} that survived to time k, $B_k(X_{k-1})$ is the RFS of targets born from X_{k-1}, and Γ_k is the RFS of targets spawned at time k.

As for the observations, their RFS at time k is

$$\Sigma_k = \Theta_k(X_k) \cup K_k(X_k) \tag{6}$$

where $\Theta_k(X_k)$ is the RFS of measurements derived from the multiple-target state X_k, and $K_k(X_k)$ is the RFS of observations not related to targets, also known as clutter.

2.2. Multiple-Target Estimation with the PHD Filter

The PHD filter is a solution to the multiple-target tracking problem based on finite set statistics (FISST) to propagate a moment of the full multiple-target probability density function. This solution reduces the computational complexity while still allowing the model to accommodate for target births and deaths and false positive detections, that is, sensor clutter. A complete derivation of the PHD filter is given in [19], while details about its sequential Monte Carlo (SMC) implementation can be found in [20]. We will mention some of the basic concepts behind the PHD filter in order to introduce this estimator. These concepts are used later in this text to explain our treatment of the sensor compromise detection problem.

While there exists a set of sequential Bayesian inference equations to propagate the multiple-target distribution, there is a very high computational cost associated to them, making any practical implementations unfeasible. The PHD filter does not propagate the full multiple-target distribution.

It has been demonstrated in [19] that the PHD is the first moment of an RFS; thus (omitting time indices),

$$\int_S D(\mathbf{x})d\mathbf{x} = \mathbb{E}\left[|\Xi \cap S|\right] \tag{7}$$

Similarly to sequential Bayesian inference, the PHD is a density that can be propagated using a prediction-update procedure. The PHD is defined in state space, instead of as a set-valued function. Thus, $\mathbf{x}$ and $\mathbf{w}$ in the following equations refer to points in the state space of the possible targets. The PHD prediction is given by

$$\begin{aligned} D_{k+1|k}(\mathbf{x}) &= b_{k+1|k}(\mathbf{x}) + \\ &\int \Big(p_S(\mathbf{w}) f_{k+1|k}(\mathbf{x}|\mathbf{w}) + b_{k+1|k}(\mathbf{x}|\mathbf{w}) \Big) D_{k|k}(\mathbf{w}) d\mathbf{w} \end{aligned} \tag{8}$$

with $b_{k+1|k}(\mathbf{x})$ as the target birth intensity function at location $\mathbf{x}$, $b_{k+1|k}(\mathbf{x}|\mathbf{w})$ as the intensity function of targets spawned from $\mathbf{w}$, $p_S(\mathbf{w})$ as the probability that a target at $\mathbf{w}$ still exists at $k+1$, and $f_{k+1|k}(\cdot|\cdot)$ as the transition probability density of individual targets.

The predicted PHD is then corrected using observations from the sensor in the update stage. First, we define the "pseudo-likelihood" term, given by

$$F_{k+1}(Z_{k+1}|\mathbf{x}) = 1 - p_D(\mathbf{x}) + \sum_{\mathbf{z} \in Z} \frac{p_D(\mathbf{x}) L_{\mathbf{z}}(\mathbf{x})}{\lambda c(\mathbf{z}) + \langle D_{k+1|k}, p_D L_{\mathbf{z}} \rangle} \tag{9}$$

where $c(\mathbf{z})$ is the clutter density function, the parameter λ is the average number of Poisson-distributed false alarms per sensor observation, and $P_D(\mathbf{x})$ is the detection probability. The function $L_{\mathbf{z}}(\mathbf{z}) = g(\mathbf{z}|\mathbf{x})$

is the measurement likelihood function. Finally, $\langle f, h \rangle = \int f(\mathbf{x})h(\mathbf{x})d\mathbf{x}$. Using $F_{k+1}(\cdot)$, the PHD update equation is

$$D_{k+1|k+1} = F_{k+1}(Z_{k+1}|\mathbf{x})D_{k+1|k} \tag{10}$$

2.3. Multiple-Sensor PHD Filter

The problem of multiple-target tracking can have an increased complexity if several sensors are used at the same time, as noted in [19]. There is a high computational cost associated with exact multiple-sensor processing. In particular, the order in which their observations are processed can influence the outcome of the estimator. A simplification for sensors with non-overlapping fields of view or coverage regions is described in [21], while [22] and [23] propose alternative methods to process information from multiple sensors. The simplest approximation is the "pseudo-sensor", which processes all of the observations as coming from a single source [24] but that has been shown to be inaccurate.

An approximate solution to the multiple-sensor processing problem that presents a reasonable compromise between accuracy and ease of implementation is the iterated-corrector [19]. We can define a modified Equation (9) to accommodate s different sensors as

$$\begin{aligned} F_{k+1}^{[i]}(Z_{k+1}^{[i]}|\mathbf{x}) =& 1 - p_D^{[i]}(\mathbf{x}) + \\ & \sum_{\mathbf{z} \in Z^{[i]}} \frac{p_D^{[i]}(\mathbf{x}) L_{\mathbf{z}}^{[i]}(\mathbf{x})}{\lambda^{[i]} c^{[i]}(\mathbf{z}) + \langle D_{k+1|k}^{[i-1]}, p_D^{[i]} L_{\mathbf{z}}^{[i]} \rangle} \end{aligned} \tag{11}$$

where $D_{k+1|k}^{[0]}(\mathbf{x}) = D_{k+1|k}(\mathbf{x})$, and

$$D_{k+1|k}^{[i]}(\mathbf{x}) = F_{k+1}^{[i]}(Z_{k+1}^{[i]}|\mathbf{x}) D_{k+1|k}^{[i-1]}(\mathbf{x}) \tag{12}$$

Finally, the multiple-sensor PHD update is given by

$$D_{k+1|k+1}(\mathbf{x}) = F_{k+1}^{[s]}(Z_{k+1}^{[s]}|\mathbf{x}) \tag{13}$$

2.4. Target Detection from the SMC–PHD Algorithm

The PHD is not a probability density, and moreover the PHD filter does not build target tracks on its own. In the case of the SMC implementation of the PHD filter, the PHD is encoded as a set of particles that represent different realizations of feasible targets over E_s. Each particle $x_k^{(i)}$ has an associated weight $w_k^{(i)}$. The number of particles is not normally constant over time; it changes with L_k, denoting how many particles are being used at time k.

In the SMC–PHD implementation, it is possible to obtain approximate target locations from the PHD with a clustering algorithm such as K-means [25] or K-medoids [26] acting on all $x_k^{(i)}$, $i = 1, \ldots, L_k$. These clustering algorithms require the number of cluster centers to be specified in advance, and it can be obtained from the SMC–PHD as

$$k = \left[\hat{N}_{k|k}\right] = \left[\sum_{j=1}^{L_{k-1}+J_k} w_k^{(j)}\right] \tag{14}$$

where $[\cdot]$ denotes rounding to the nearest integer.

The SMC–PHD algorithm has a resampling step at the end of each iteration, after which all of the particle weights are equal. Hence, the clustering algorithm operates only in the target state space with no consideration of $\{w_k^{(i)}\}_{i=1}^{L_k}$. We refer to the resulting finite set as $\hat{X}_k$.

3. Methods

3.1. Preliminaries

We assume that some region E_s is being surveyed by s sensors. We can pose the sensor compromise problem as an anomaly detection application. Referring back to the RFS formulation of the observation set in Equation (6), the observation set with added anomalies for sensor i can be described as

$$\tilde{\Sigma}_k^{[i]} = \tilde{\Theta}_k^{[i]}(X_k) \cup K_k^{[i]}(X_k) \cup \Delta_k^{[i]} \tag{15}$$

while in the nominal operation mode, the observation RFS $\Theta_k^{[i]}(X_k)$ follows a certain model that includes noise derived from a known distribution. In $\tilde{\Sigma}_k^{[i]}$, the observation $\tilde{\Theta}_k^{[i]}(X_k)$ has noise distributions that may not match the assumed model. The clutter model can also be modified by the attack. In addition, any attacks that consist of the introduction of decoys among the targets can be modeled as a new RFS, $\Delta_k^{[i]}$.

While it may not be possible to characterize each and every attack or sensor failure, the model used in Equation (15) allows us to enumerate the major categories into which an anomalous behavior may fall:

1. Modified observation of a given target: The sensor noise does not conform to the assumed distribution. This can be, for example, an added bias on the observations or an increased covariance in the noise distribution.
2. Modified probability of target detection: For instance, some targets can be omitted from the sensor observations.
3. Modified false alarm rate: This can include an increase in the clutter intensity, duplicate reports for a given target, and so on.

An attack or malfunction may combine any of the above.

Confronted with the impossibility of modeling every kind of attack on the sensor system, an alternative is to resort to the one-class classifier approach. A diagnostics system is built that is only trained to recognize the nominal operation of the sensor array. Any measurements that deviate significantly from the learned model are then classified as being caused by a breach in the system or a hardware failure. The overall architecture of such an anomaly detection system is shown in Figure 1.

Selecting which features to learn about a multiple-target tracker is not straightforward. Because we do not limit our analysis to a specific class of dynamical systems or observation models, we inspire our anomaly detection approach in the concept of hypothesis testing.

The multiple-target scenario includes the possibility of having target births, deaths and sensor clutter. Hence, it requires a way to quantify the deviation from each sensor with respect to the estimator as a whole. Before continuing with our exposition, we introduce some assumptions that we require to hold for the anomaly detection techniques studied.

Assumption 1. *The observation spaces for all sensors are such that* $E_o^{[i]} = E_o^{[j]}$ *for* $i, j = 1, \ldots, s$, *or there exist one-to-one transformations* T_i^j *such that* $E_o^{[j]} = T_i^j[E_o^{[i]}]$.

Assumption 2. E_o *is a Banach space, with metric* $d(x, y)$.

Assumption 3. *The observation noise for the sensor, with respect to each target, is a stationary process, with known mean. Moreover, the noise statistics for all observations from targets are equal and known.*

The first assumption allows for the direct comparison of observations stemming from different sensors. The second enables us to use a metric to compare sensor observations. Finally, the third simplifies the analysis of the relative error between observations from different sensors.

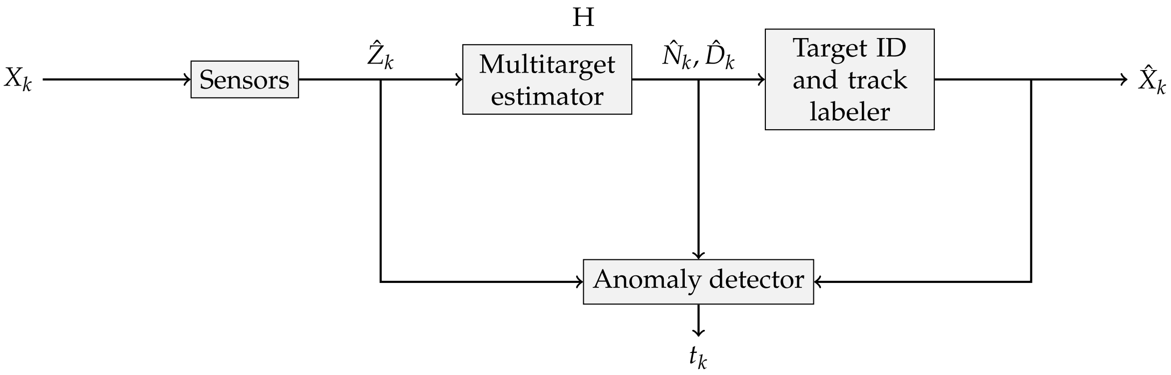


Figure 1. Anomaly detection scheme used in this work.

3.2. OSPA Metric

For every estimation algorithm, there is a metric that can quantify its performance. As an example, any single-target estimation algorithm can use the $\mathcal{L}_2$-norm to compute the difference between the estimated state and the true target state. This is no longer valid in the case of multiple targets, as the estimation error can consist not only of a distance in state space, but also of association errors between estimates and different target states. There can also be errors in the estimation of the cardinality of the target set. Metrics such as the Haussdorf distance have been proposed to capture these errors, and some developments have dealt with the limitations of this distance [27]. An alternative that solves most of these problems and remains a valid metric, that is, it satisfies the axioms of identity, symmetry and the triangle inequality, is optimal sub-pattern assignment (OSPA) [28]. To use this metric, one needs to specify a parameter called the cut-off $c > 0$, in addition to $1 \leq p < \infty$, which is the order of OSPA.

The OSPA metric is computed using the expression

$$\bar{d}_p^{(c)}(X, Y) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)}(\mathbf{x}_i, \mathbf{y}_{\pi(i)})^p + c^p(n-m) \right) \right)^{1/p} \tag{16}$$

where $d^{(c)}(\mathbf{x}, \mathbf{y}) = \min(c, d(x, y))$, and Π_k is the set of permutations on $\{1, \ldots, k\}$ for any $k \in \mathbb{N}$. The cardinalities of the sets being compared are $|X| = m$ and $|Y| = n$. Equation (16) is only valid if $m \leq n$; otherwise $\bar{d}_p^{(c)}(X, Y) := \bar{d}_p^{(c)}(Y, X)$.

In the rest of this work, we make use of a decomposition of the OSPA metric into two components: localization error and cardinality error. The localization component is given by

$$e_{p,\text{loc}}^{(c)}(X, Y) = \left(\min_{\pi \in \Pi_n} \frac{1}{n} \sum_{i=1}^{n} d^{(c)}(\mathbf{x}_i, \mathbf{y}_{\pi(i)})^p \right)^{1/p} \tag{17}$$

and the cardinality component is

$$e_{p,\text{card}}^{(c)}(X, Y) = \left(\frac{c^p(n-m)}{n} \right)^{1/p} \tag{18}$$

As before, these expressions only hold valid if $m \leq n$, and by definition, $e_{p,\text{loc}}^{(c)}(X, Y) := e_{p,\text{loc}}^{(c)}(Y, X)$ and $e_{p,\text{card}}^{(c)}(X, Y) := e_{p,\text{card}}^{(c)}(Y, X)$ if $m > n$.

3.3. Anomaly Detection: Single-Sensor Case

In the single-sensor case, the information available to the filter is limited to the estimated target states over time and the observations from a single source. Detecting an anomaly in the sensor behavior implies comparing its observations of the projections of the estimated states to E_o.

In the single-sensor, single-target case, one solution to identify whether an estimator is operating nominally is the use of residual analysis [29]. In a Kalman filter, the residual is given by $|y - C\hat{x}|$, where y is an observation and $\hat{x}$ is the estimated target state. The residual in this case follows a χ^2 distribution; thus a hypothesis test would reveal the existence of an anomaly.

The single-sensor, multiple-target case with general nonlinear dynamics and non-Gaussian noise requires a different treatment. Each pair of dynamical and observation models would require a separate derivation of the statistics of the residuals. A general approach needs a training phase in which a statistical model of the residuals is built, to be used as the basis for anomaly detection during a deployment phase.

In a single-sensor, multiple-target tracker, the ability of an attacker to introduce decoys is limited by $b_{k+1|k}(\mathbf{x})$. In a trusted sensors framework, the target birth intensity is typically chosen to minimize the response time of the PHD filter, such that new targets are tracked as soon as possible. However, this renders any decoy injection attacks quite effective, as they all become new objects in the tracker. Tuning $b_{k+1|k}(\mathbf{x})$ can delay the apparition of any decoys and enable the detection of an attack by imposing a limit on the expected number of new targets, but at the expense of delaying the detection of true targets. Another problem is that this countermeasure assumes that the attacker does not know about the modified $b_{k+1|k}(\mathbf{x})$, as otherwise, they would just introduce the decoys sequentially over time to avoid detection.

Proposition 1. *Consider a single-sensor, multiple-target tracker based on the PHD filter. If an attacker has full knowledge of $b_{k+1|k}(\mathbf{x})$ and the expected number of targets to track is always unknown a priori, the attacker can introduce decoys without being detected.*

Proof. We show that even for the simplest multiple-target tracking case, an attacker would be capable of spawning a new target in the system. We consider a basic PHD filter with $p_S(\mathbf{x}) = 1$, $b_{k+1|k}(\mathbf{x}|\mathbf{w}) = 0$ and a single target, such that $D_{k|k}(\mathbf{x}) = f(\mathbf{x})$ is some unimodal density. The sensor is assumed to produce no clutter. Then,

$$D_{k+1|k}(\mathbf{x}) = f_1(\mathbf{x}) + b_{k+1|k}(\mathbf{x})$$

after the prediction stage. An observation of the true target $\mathbf{z}_1$ is reported by the sensor. We assume that an attacker sends $\mathbf{z}_2$, a false report of the location of a second target. For illustration purposes, we consider $L_{z_1}(\mathbf{x})$ and $L_{z_2}(\mathbf{x})$ to have negligible overlap. From Equation (9), it follows that each observation will produce a term in the sum to compute $F_{k+1}(Z_{k+1}|\mathbf{x})$. The term due to the real observation is

$$\frac{L_{z_1}(\mathbf{x})f_1(\mathbf{x}) + L_{z_1}(\mathbf{x})b_{k+1|k}(\mathbf{x})}{\langle f_1, L_{z_1}\rangle + \langle b_{k+1|k}, L_{z_1}\rangle}$$

while the term corresponding to the decoy is

$$\frac{\overbrace{L_{z_2}(\mathbf{x})f_1(\mathbf{x})}^{\to 0} + L_{z_2}(\mathbf{x})b_{k+1|k}(\mathbf{x})}{\underbrace{\langle f_1, L_{z_2}\rangle}_{\to 0} + \langle b_{k+1|k}, L_{z_2}\rangle} \approx \frac{L_{z_2}(\mathbf{x})b_{k+1|k}(\mathbf{x})}{\langle b_{k+1|k}, L_{z_2}\rangle}$$

and as a result of this reduction, integrating the resulting PHD produces

$$\int D_{k+1|k+1}(\mathbf{w})d\mathbf{w} \approx N_{k+1|k+1} + 1$$

The introduction of the extra decoy can only be prevented if $L_{z_2}(\mathbf{x})b_{k+1|k}(\mathbf{x}) \approx 0 \; \forall \; \mathbf{x}$.

As we have shown, even under the simplest conditions, it would not be possible to detect the insertion of a decoy by an attacker. As those conditions are relaxed, and because clutter, splitting and overlapping targets are included in the model, there are even more possible locations in which an attacker can successfully place a decoy without violating the sensor model. □

3.4. Anomaly Detection for Multiple-Sensor Systems

As discussed so far, the detection of anomalies in the single-sensor case, for a multiple-target tracking problem with nonlinear dynamics, has several difficulties that may not have a viable solution for all scenarios. If a redundant architecture with multiple sensors is introduced instead, new possibilities arise for compromise detection. For the multiple-sensor case, it is possible to use the observations from all sensors to measure the deviation of each from a consensus, under the following condition.

Assumption 4. *An attacker cannot control more than $\tilde{s} \leq \lceil s/2 - 1 \rceil$ sensors; thus a majority consensus set $\mathcal{Z}_k$ with all observations by all sensors at time k can be constructed. From Assumption 1, every element of $\mathcal{Z}_k$ belongs to a single E_o.*

As established in the previous section, the OSPA metric can indicate distances between finite sets with different cardinalities. Hence, we propose using quantities related to the OSPA metric to measure the distance between the observations of a sensor, $Z^{[i]}$, and the estimated target states $\hat{X}$. Because OSPA combines localization and cardinality errors in a single metric, we opt to use $e^{(c)}_{p,\text{loc}}(X,Y)$ and $e^{(c)}_{p,\text{card}}(X,Y)$ as a way to separate the two kinds of sensing errors in multiple-target scenarios. Our anomaly detection goal does not need the symmetry property of OSPA and can benefit from a distinction between localization and cardinality disparities among sensors.

We consider, for each sensor i, the set of its observations $Z^{[i]}_k$. Hence, we can construct the set of all observations with the exception of those from sensor i, which are $\mathcal{Z}_k \setminus Z^{[i]}_k$. In our exposition of the anomaly detection scheme, the p and c values of $e^{(c)}_{p,\text{loc}}(X,Y)$ and $e^{(c)}_{p,\text{card}}(X,Y)$ are assumed to remain constant; thus we do not refer to them explicitly unless necessary. A calculation of the localization and cardinality errors of the observations from each sensor i at time k, when compared to the consensus set, can be denoted by

$$\Lambda^{[i]}_k = e^{(c)}_{p,\text{loc}}(Z^{[i]}_k, \mathcal{Z}_k \setminus Z^{[i]}_k) \tag{19}$$

$$\Upsilon^{[i]}_k = e^{(c)}_{p,\text{card}}(Z^{[i]}_k, \mathcal{Z}_k \setminus Z^{[i]}_k) \tag{20}$$

Because the probability distributions of $\Lambda^{[i]}_k$ and $\Upsilon^{[i]}_k$ depend on many factors, such as the target dynamics, sensor models and number of targets, they can be constructed from experimental data during nominal system operation. A training phase with fully trusted sensors and estimator would be needed.

After the training data is gathered, sequences $\bar{\Lambda}^{[i]}_k = \Lambda^{[i]}_{k-l}, \ldots, \Lambda^{[i]}_k$ and $\bar{\Upsilon}^{[i]}_k = \Upsilon^{[i]}_{k-l}, \ldots, \Upsilon^{[i]}_k$ need to be analyzed for each sensor. An adequate hypothesis test is necessary to evaluate if the null hypothesis, $H_0 : \bar{\Lambda}^{[i]}_k, \bar{\Upsilon}^{[i]}_k \sim p(\Lambda^{[i]}_k, \Upsilon^{[i]}_k)$, can be rejected. Useful tools in this context are the Kolomogorov–Smirnov two-sample test, and the Wilcoxon rank-sum test [30,31]. For this work, we apply the latter.

There can be a coupling between the sensor outputs and the estimated state. It would be desirable to isolate a faulty sensor from others. Unfortunately, as shown by Equation (11), the iterated-corrector scheme for multiple-sensor information fusion processes the observations from each source in a sequential manner. Because we are proposing to compare each $Z^{[i]}$ to $\hat{X}$, and because $\hat{X}$ is extracted from $D_{k+1|k+1}(\mathbf{x})$, several sensors could be considered as anomalous, as H_0 is rejected for each of them.

The nature of the attack determines whether faulty sensors can be isolated or not, as evidenced later in our results section.

3.5. Algorithm for Multiple-Sensor Anomaly Detection

We summarize our anomaly detection methodology. We base the anomaly detection procedure on a BN. The rationale behind this choice is the observation that the two measurements Λ and Υ may have different values depending on the class of the attack. Most attacks, if they are of high enough intensity, will cause a mismatch in the number of objects detected by a sensor. Only a subset of attacks that introduce a small error signal, such as noise or bias, will provoke an increase in Λ without affecting Υ.

1. *Data collection*. Gather data from all sensors during nominal operation. This requires a trusted environment. This data is then used to compute $\Lambda_k^{[i]}$ and $\Upsilon_k^{[i]}$ for all k in the recorded measurements.
2. *Nonparametric statistical test*. During a target tracking task, keep a history of measurements for every sensor using a fixed horizon; that is, collect $\bar{\Lambda}_k^{[i]}, \bar{\Upsilon}_k^{[i]}$ for a lag l. At every time step k, perform a nonparametric statistical test between the history of measurements and the trusted data for each sensor, and save the p-value of the test.
3. *Bayesian inference for anomaly detection*. Denote the trustworthiness of each sensor by the indicator function:

$$t_k^{[i]} = \begin{cases} 0 & \text{Sensor } i \text{ cannot be trusted at time } k \\ 1 & \text{Sensor } i \text{ is fully reliable at time } k \end{cases} \tag{21}$$

There are two events associated with the result of a hypothesis test between a fixed number of observations of Λ and Υ:

$$c_k^{[i]} = \begin{cases} 0 & \text{Sensor } i \text{ reports an anomalous } \Upsilon \text{ at time } k \\ 1 & \text{Sensor } i \text{ has a nominal } \Upsilon_k \end{cases} \tag{22}$$

$$e_k^{[i]} = \begin{cases} 0 & \text{Sensor } i \text{ reports an anomalous } \Lambda \text{ at time } k \\ 1 & \text{Sensor } i \text{ has a nominal } \Lambda_k \end{cases} \tag{23}$$

The results of the nonparametric hypothesis test provide $p(c_k^{[i]}|t_k^{[i]} = 1)$ and $p(e_k^{[i]}|t_k^{[i]} = 1)$. Then the probability of $t^{[i]}$ can be updated over time in a Bayesian manner. The inspection of the effects of some attacks on Λ, Υ reveals that the dependence relations among $t_k^{[i]}$, $c_k^{[i]}$ and $e_k^{[i]}$ can be described by the BN depicted in Figure 2. Hence, the posterior probability for $t^{[i]}$ is given by

$$p\left(t_k^{[i]} \middle| e_{k-1}^{[i]}, c_{k-1}^{[i]}, t_{k-1}^{[i]}\right) = \sum_j f\left(t_k^{[i]} | t_{k-1}^{[i]} = j\right) p\left(t_{k-1}^{[i]} = j \middle| e_{1:k-1}^{[i]}, c_{1:k-1}^{[i]}\right) \tag{24}$$

$$p\left(t_k^{[i]} | e_k^{[i]}, c_k^{[i]}\right) = \frac{p\left(e_k^{[i]} | t_k^{[i]}\right) p\left(c_k^{[i]} | e_k^{[i]}, t_k^{[i]}\right) p\left(t_k^{[i]} \middle| e_{k-1}^{[i]}, c_{k-1}^{[i]}, t_{k-1}^{[i]}\right)}{\sum_j p\left(e_k^{[i]} | t_k^{[i]} = j\right) p\left(c_k^{[i]} | e_k^{[i]}, t_k^{[i]} = j\right) p\left(t_k^{[i]} = j \middle| e_{k-1}^{[i]}, c_{k-1}^{[i]}, t_{k-1}^{[i]}\right)} \tag{25}$$

The full anomaly detection system is described in Figure 3. The targets are (generally) visible to all sensors $S_1, \ldots, S_n$. The components of OSPA Λ and Υ are computed for every sensor, and banks of delays M_{i1} and M_{i2} produce $\bar{\Lambda}_k^{[i]}$ and $\bar{\Upsilon}_k^{[i]}$, respectively, which are then fed to the individual BNs. The output of each BN is the estimated posterior probability of sensor trustworthiness, $t_k^{[i]}$.

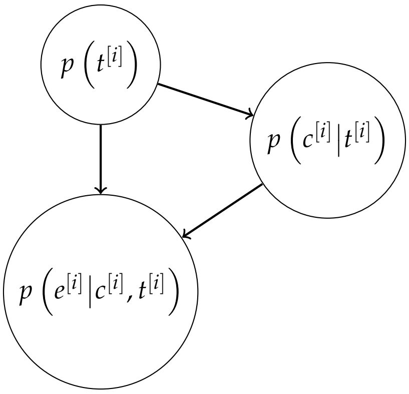

$p\left(t^{[i]}\right)$
$p\left(c^{[i]}|t^{[i]}\right)$
$p\left(e^{[i]}|c^{[i]},t^{[i]}\right)$

Figure 2. Bayesian network architecture for anomaly detection, for a single sensor.

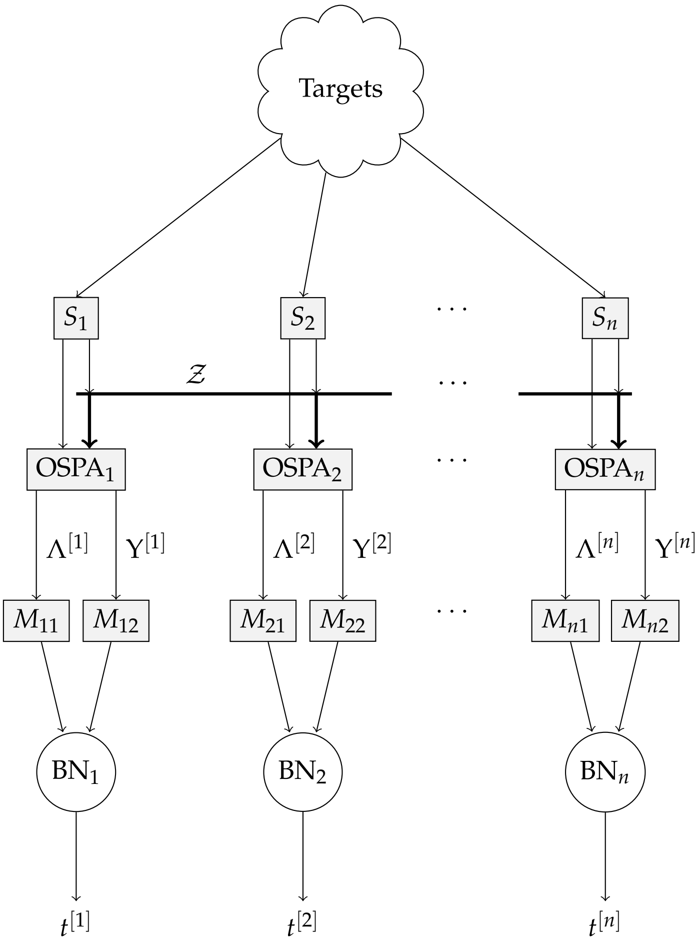


Figure 3. Complete anomaly detection scheme using the Bayesian network approach.

4. Results

In order to test the performance of the sensor intrusion methodology developed in this work, we provide results of simulations. To offer a comparison, we have also used a one-class classifier technique from the machine learning domain: support vector machines (SVMs). The main difference between the two approaches is that ours uses specific transition probabilities for each sensor, while the SVM algorithm is trained on pure data and does not include information about the likelihood of the reliability of each sensor. As in the case of our Bayesian approach, the SVM algorithm uses data from the Λ and Υ measurements, as different attacks would affect each of these in a different way. The SVM is trained using a time window of length l for Λ and Υ, just as for the Bayesian approach. Because it is a one-class classifier on a fairly high dimensional space, we have used a radial basis function kernel transform [32,33].

We introduce the motion model for the simulations before describing the full environment.

4.1. Coordinated Turn Model

The coordinated turn (CT) model is popular for aircraft tracking applications, as it assumes a constant forward velocity and has a state-space representation that uses Cartesian coordinates [34]. A useful property of this model is that it has an exact discrete-time conversion [35,36].

The continuous-time version of the CT model has a state vector

$$\mathbf{x} = [x\ y\ \dot{x}\ \dot{y}]^T$$

with a motion model given by

$$\dot{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\omega & 0 \\ 0 & 0 & 0 & \omega \end{bmatrix} \mathbf{x} = A(\omega)\mathbf{x} \tag{26}$$

where ω is the turning rate. Its discrete-time equivalent, with the turning rate added as an extra state, is

$$\mathbf{x}(t+T) = \begin{bmatrix} x + \frac{\dot{x}}{\omega}\sin(\omega T) + \frac{\dot{y}}{\omega}[\cos(\omega T) - 1] \\ y + \frac{\dot{x}}{\omega}[1 - \cos(\omega T)] + \frac{\dot{y}}{\omega}\sin(\omega T) \\ \dot{x}\cos(\omega T) - \dot{y}\sin(\omega T) \\ \dot{x}\sin(\omega T) + \dot{y}\cos(\omega T) \\ \omega \end{bmatrix} + \begin{bmatrix} T^2/2 & 0 & 0 \\ 0 & T^2/2 & 0 \\ T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_\omega \end{bmatrix} \tag{27}$$

where w_x, w_y and w_ω are zero-mean random variables to model velocity and turning rate uncertainties, and T is the sampling period.

4.2. Simulated Attacks

In the simulations, we introduced seven attacks. It is important to emphasize that these did not cover the whole spectrum of attacks that could occur, but they did test variations of the basic attacks described in Section 3.1. The simulated attacks were performed on a subset of the sensors, and were as follows:

1. Zero-mean noise introduced.
2. Mobile decoy added.
3. Static decoy inserted.
4. Bias added to the observation of one target.
5. Multiple decoys added.
6. A target surpressed.
7. Bias added to all observations.

For each type of attack, we ran 30 simulations, each with a length of 200 time steps, using the SMC implementation of the PHD filter. The sampling period of the simulations was 1 s. All attacks occurred in the interval $30 \leq k \leq 170$, with randomly selected start and end times, and also with random duration. For both intrusion detection approaches, we trained the classifiers using a 400 time-step simulation in which the sensors were assumed to be trustworthy. The simulated target tracks, along with the positions of the eight sensors, are shown in Figure 4.

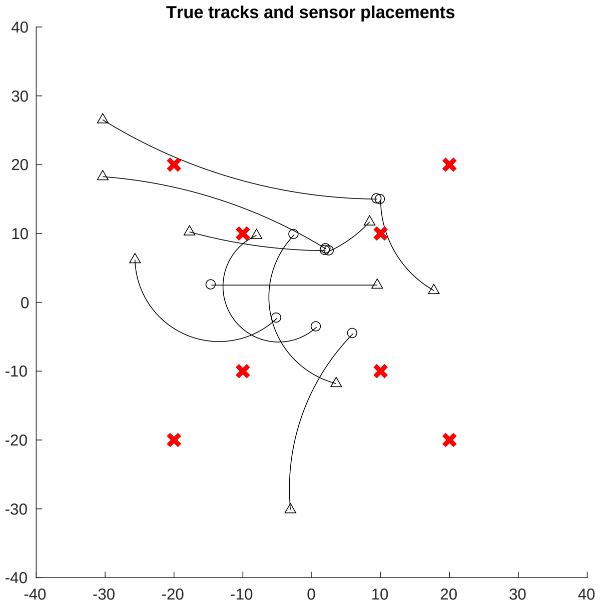


Figure 4. Ground truth of the target tracks. The ($\circ$) markers indicate the start of the target tracks, and their ends are the ($\triangle$) markers. The ($\times$) markers indicate the locations of the eight sensors.

5. Discussion

The numerical results report the precision and recall for each classifier, which we have combined by presenting the *F*-score in Table 1. We present two figures per scenario: local and global classification scores. Local indicates that we measured the *F*-score per sensor, so that the overall attack was detected and the responsible sensors were flagged as being compromised. The global *F*-score measured whether an attack was detected by the system, independently from which sensor may have caused it. Hence, the global *F*-score was always expected to be higher.

The attack identifier number from the leftmost column coincides with the seven disruption modalities described before. The numbers in bold font highlight the highest *F*-score in that table row. The column titled "*F*-Score Mode" indicates whether the intrusion detection was computed on a sensor-by-sensor basis ("Per sensor") or globally ("Global"). In most cases, the "Global" score is significantly higher than the "Per sensor" score. These attacks that were mostly related to the introduction of decoys were particularly well detected. A notable case is attack 6, for which some sensors stopped reporting a subset of the targets. The "Per sensor" *F*-score is low, as the sensors under attack were identified as nominal, and those in normal operation were flagged as compromised. However, the "Global" score is on par with those obtained for other attacks. In this instance, the SVM classifier outperformed the BN on the "Per sensor" *F*-score but not on the "Global" score.

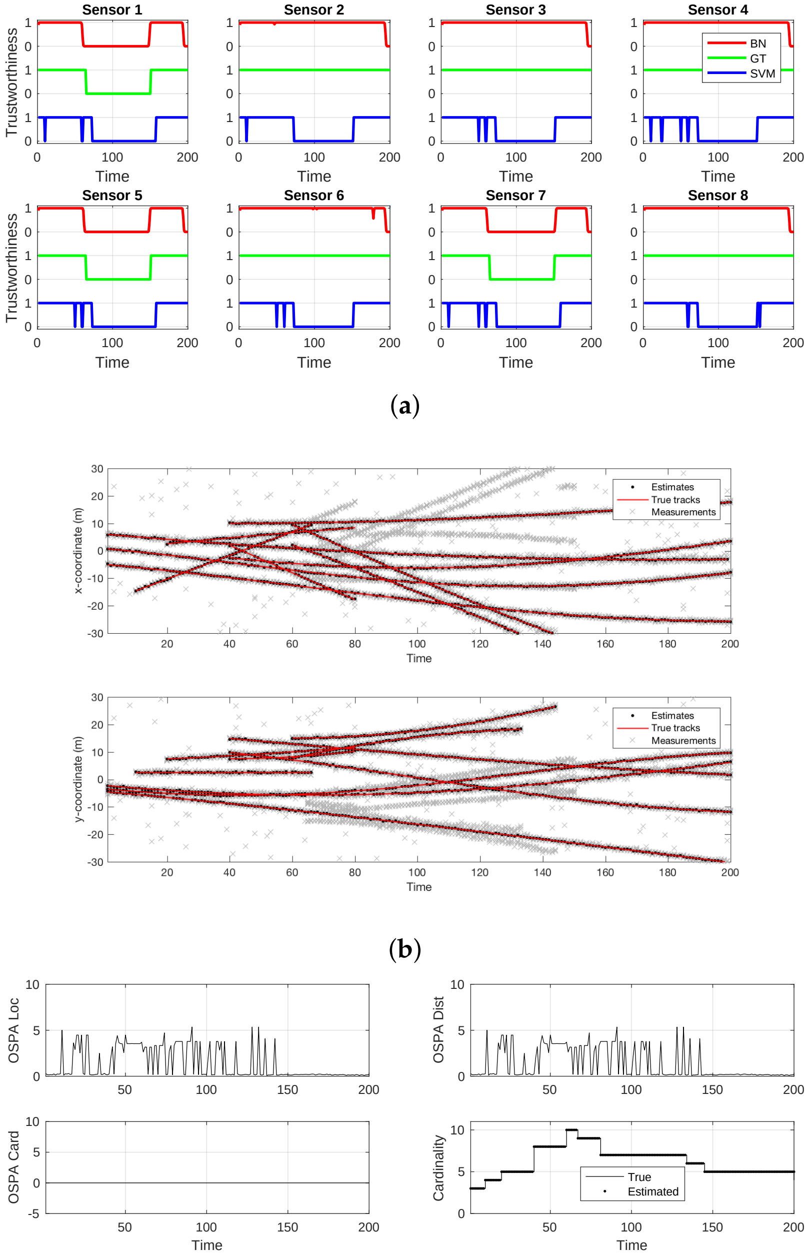
Table 1. *F*-scores for different attack classes, on three out of eight sensors.

Attack	*F*-Score Mode	BN Mean	BN Std	BN Best	BN Worst	SVM Mean	SVM Std	SVM Best	SVM Worst
1	Per sensor	0.4447	0.0711	**0.5104**	0.2555	0.3042	0.0720	0.4294	0.1582
1	Global	0.8534	0.0829	**0.9618**	0.6667	0.6190	0.1089	0.7826	0.3214
2	Per sensor	0.8547	0.0501	**0.9174**	0.7494	0.5160	0.0550	0.6008	0.3817
2	Global	0.8919	0.0618	**0.9576**	0.6986	0.8024	0.0944	0.9243	0.5472
3	Per sensor	0.8414	0.0625	**0.9329**	0.6667	0.6434	0.0948	0.7742	0.4393
3	Global	0.8741	0.0656	**0.9600**	0.7126	0.7931	0.0914	0.9200	0.5882
4	Per sensor	0.4843	0.0232	**0.5180**	0.4398	0.4552	0.0489	0.5125	0.3557
4	Global	0.8628	0.0597	**0.9531**	0.7423	0.8162	0.0998	0.9434	0.6133
5	Per sensor	0.8485	0.0547	**0.9300**	0.7550	0.4759	0.0568	0.5684	0.3623
5	Global	0.9051	0.0424	**0.9627**	0.8148	0.8077	0.0871	0.9339	0.6500
6	Per sensor	0.0355	0.0281	0.0724	0.0026	0.2126	0.0636	**0.3232**	0.0778
6	Global	0.8968	0.0449	**0.9565**	0.8000	0.8146	0.0764	0.9381	0.6389
7	Per sensor	0.4812	0.0233	**0.5203**	0.4279	0.4333	0.0383	0.4901	0.3587
7	Global	0.8758	0.0586	**0.9573**	0.7209	0.7842	0.0818	0.9239	0.6250

In order to illustrate the effectiveness of both the BN and the SVM approaches under some attacks, we show in detail the results of three simulated attacks. Each attack had a different nature and degree of success. The iterated-corrector SMC–PHD filter had not been altered in any way to change its resilience when presented with the attacks. In these examples, the attack start time was 60 s, with an end time of 150 s.

First, we show the result of adding several decoys to a minority of sensors, three out of eight. Figure 5 displays plots that convey insight about the effects of the attack. The first plot is Figure 5a, and it compares the ground truth for the attack (green line) to the detection offered by the BN (red line) and to that of the SVM classifier (blue line). In this case, both approaches successfully detected the presence of an attack from a global perspective. However, the SVM classifier flagged every sensor as being under attack. In contrast, the BN correctly identified which sensors were producing the false decoys. Figure 5b shows the tracks detected by the SMC–PHD filter, compared to the true tracks. The gray crosses represent the sensor observations. While there was an abundance of false target reports, along with clutter, this attack was not sufficient to make the filter fail or report the decoys as true targets. The effects of the attack, in terms of the cardinality and localization errors, can be verified in Figure 5c. As this attack was not successful in introducing decoys, the cardinality error was zero, while the localization error remained small.

The second case, in Figure 6, exemplifies a successful attack that made the SMC–PHD filter arrive to a wrong cardinality estimate during the times at which the sensors were compromised. Again, three out of the eight total sensors were being attacked. This attack introduced an elevated level of white noise, more than twice what was expected. As Figure 6a depicts, the BN approach incurred several mistakes while identifying individual sensors under attack. The same happened to the SVM classifier. In both cases, even if the compromised sensors were not correctly isolated, the presence of an attack in the overall system was successfully reported. The effects of the attack on the estimated target positions are shown in Figure 6b. Both the localization and the cardinality components of the OSPA error show adverse effects from the attack in Figure 6c. When the estimated cardinality is compared to the ground truth, it can be verified that the noise attack resulted in missed target reports by the filter.



(**a**)

(**b**)

(**c**)

Figure 5. Example of an unsuccessful attack. Several decoys were added to three sensors, out of eight. The cardinality of the detected target set is still correct, even in the presence of the false observations. The Bayesian scheme was able to identify the compromised sensors, while the SVM classifier incurred many false positives. (**a**) Sensor compromise detection by Bayesian network (BN) and support vector machine (SVM), compared to ground truth; (**b**) Tracks detected by the sequential Monte Carlo—probability hypothesis density (SMC—PHD) filter, with every sensor observation displayed; (**c**) Optimal sub-pattern assignment (OSPA) metric, along with its location and cardinality error components. True and estimated cardinalities are also displayed.

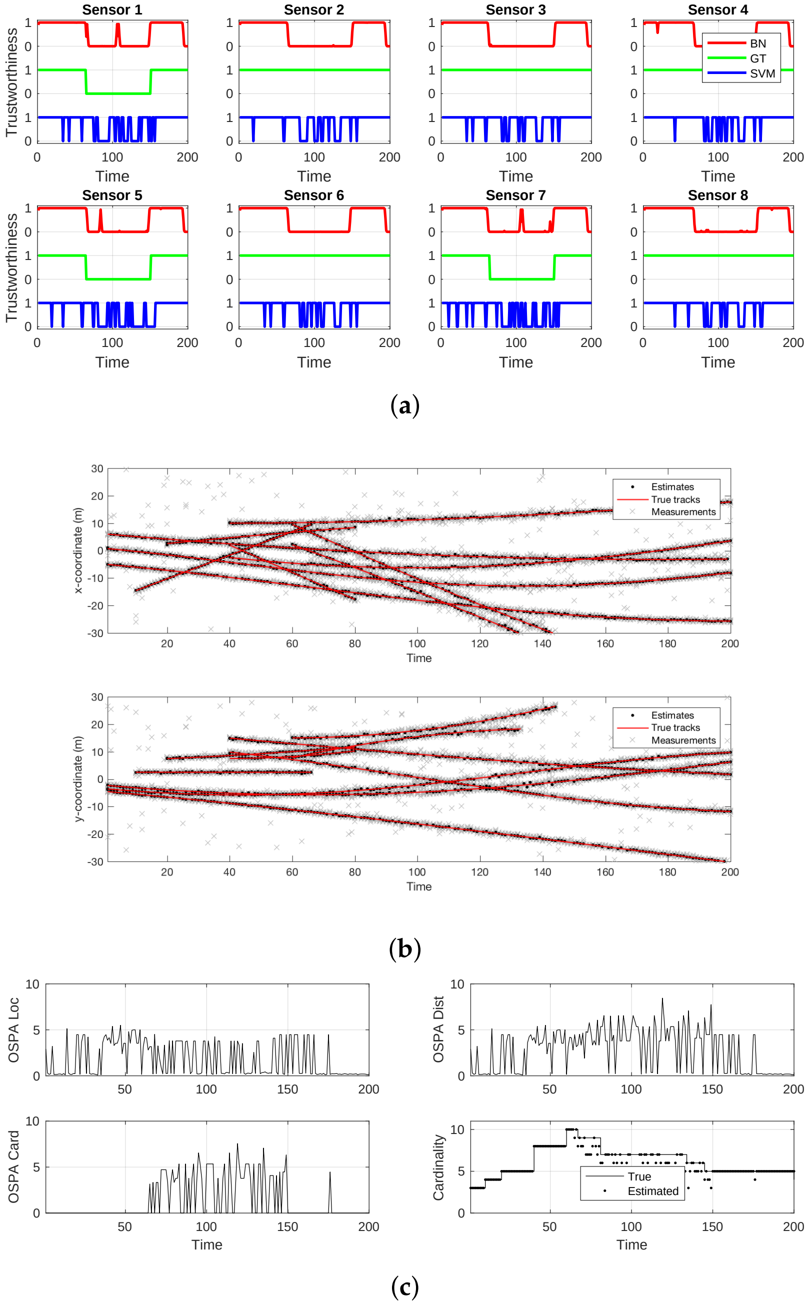
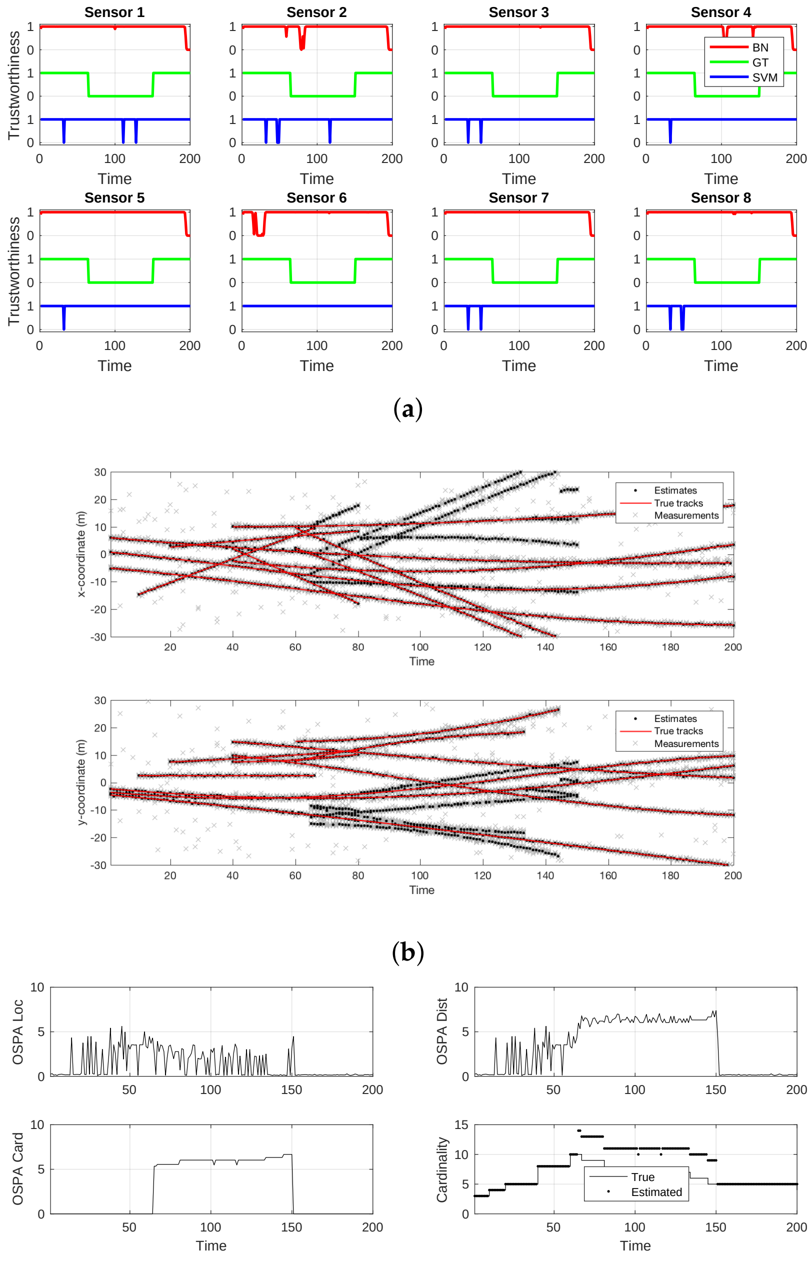


Figure 6. A successful attack. An increased level of white noise was injected to three sensors, out of eight. The cardinality of the detected target set is incorrect for the time of the attack. Both the Bayesian scheme and the SVM classifier flagged many false positives. (**a**) Sensor compromise detection by Bayesian network (BN) and support vector machine (SVM), compared to ground truth; (**b**) Tracks detected by the sequential Monte Carlo—probability hypothesis density (SMC—PHD) filter, with every sensor observation displayed; (**c**) Optimal sub-pattern assignment (OSPA) metric, along with its location and cardinality error components. True and estimated cardinalities are also displayed.

The final example scenario, which depicts the effects of an attack on all sensors, is shown in Figure 7. Here, a decoy introduction attack was performed on every sensor, which could not be detected under the schemes discussed in this work. Figure 7a shows that, while the ground truth contained attacks for all sensors, no detection technique reported any compromise. The new, false tracks that resulted from the attack can be seen in Figure 7b. The cardinality error was constantly high during the time of the attacks, as seen in Figure 7c. The comparison between the true and estimated cardinalities shows the increased number of targets during the decoy introduction attack.



(**a**)

(**b**)

(**c**)

Figure 7. An attack on all sensors. No compromise detection method was able to identify the intrusion. (**a**) Sensor compromise detection by Bayesian network (BN) and support vector machine (SVM), compared to ground truth; (**b**) Tracks detected by the sequential Monte Carlo—probability hypothesis density (SMC—PHD) filter, with every sensor observation displayed; (**c**) Optimal sub-pattern assignment (OSPA) metric, along with its location and cardinality error components. True and estimated cardinalities are also displayed.

6. Conclusions

Given the possibility of an intruder taking over a subset of sensors in a target detection and tracking system, we present an algorithm for an effective intrusion detection scheme. We use the principle of a one-class classifier to detect anomalies in the measurements collected by the sensors, without assuming a particular attack form is employed by the intruder. The main tool that enables us to detect the anomalies is the OSPA metric, along with its localization and cardinality error components. While we show that in the single-sensor case, the task of detecting an intrusion can vary from hard to impossible, depending on the type of attack, using multiple sensors opens up the option of forming a consensus set to have some reference behavior with which to compare the performance of individual elements.

As evidenced by our simulation results, the use of a BN per sensor, along with hypothesis testing for the localization and cardinality errors of each sensor with respect to the consensus set, is a powerful tool that identifies the anomalies in a wide variety of situations. Some border cases cannot be handled by this approach if an accurate identification of the culprit sensors is required, but even in these scenarios, an overall intrusion alarm can be raised.

Acknowledgments: This work was supported by the OSD sponsored Autonomy Research Pilot Initiative project entitled "A Privileged Sensing Framework". Publication costs have been kindly covered by the Engineering Division of the Irapuato-Salamanca Campus, University of Guanajuato.

Author Contributions: J.-P.R. and J.W.C. conceived the concept and formulated the solution strategy; E.A.D. reviewed the manuscript and provided the experiment design; V.A. reviewed and helped with the manuscript and provided the necessary equipment to perform the experiments; J.-P.R. wrote the article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BN	Bayesian network
LTI	Linear time-invariant
OSPA	Optimal sub-pattern assignment
PHD	Probability hypothesis density
SVM	Support vector machine

References

1. Cardenas, A.A.; Amin, S.; Sastry, S. Secure control: Towards survivable cyber-physical systems. In Proceedings of the 28th International Conference on Distributed Computing Systems Workshops, Beijing, China, 17–20 June 2008; pp. 495–500.
2. Kwon, C.; Liu, W.; Hwang, I. Security analysis for Cyber-Physical Systems against stealthy deception attacks. In Proceedings of the 2013 American Control Conference, Washington, DC, USA, 17–19 June 2013; pp. 3344–3349.
3. Karim, M.E.; Phoha, V.V. Cyber-Physical Systems Security. In *Applied Cyber-Physical Systems*; Suh, S.C., Tanik, U.J., Carbone, J.N., Eroglu, A., Eds.; Springer: New York, NY, USA, 2014; pp. 75–83.
4. Lunt, T.F. A survey of intrusion detection techniques. *Comput. Secur.* **1993**, *12*, 405–418.
5. Biermann, E.; Cloete, E.; Venter, L. A comparison of Intrusion Detection systems. *Comput. Secur.* **2001**, *20*, 676–683.
6. Zhang, Y.; Lee, W.; Huang, Y.A. Intrusion detection techniques for mobile wireless networks. *Wirel. Netw.* **2003**, *9*, 545–556.
7. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 15.
8. Zhang, Y.; Meratnia, N.; Havinga, P. Outlier detection techniques for wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* **2010**, *12*, 159–170.

9. Zhang, Q.; Yu, T.; Ning, P. A framework for identifying compromised nodes in sensor networks. In Proceedings of the Securecomm and Workshops, Baltimore, MD, USA, 28 August–1 September 2006; pp. 1–10.
10. Fawzi, H.; Tabuada, P.; Diggavi, S. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Trans. Autom. Control* **2014**, *59*, 1454–1467.
11. Chong, M.S.; Wakaiki, M.; Hespanha, J.P. Observability of linear systems under adversarial attacks. In Proceedings of the American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015; pp. 2439–2444.
12. Bai, C.Z.; Gupta, V.; Pasqualetti, F. On Kalman Filtering with Compromised Sensors: Attack Stealthiness and Performance Bounds. *IEEE Trans. Autom. Control* **2017**, *62*, 6641–6648.
13. Pimentel, M.A.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A review of novelty detection. *Signal Process.* **2014**, *99*, 215–249.
14. Grubbs, F.E. Procedures for detecting outlying observations in samples. *Technometrics* **1969**, *11*, 1–21.
15. Aggarwal, C.C.; Yu, P.S. Outlier detection with uncertain data. In Proceedings of the 8th SIAM International Conference on Data Mining, Atlanta, GA, USA, 24–26 April 2008; pp. 483–493.
16. Reppa, V.; Polycarpou, M.M.; Panayiotou, C.G. Adaptive approximation for multiple sensor fault detection and isolation of nonlinear uncertain systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 137–153.
17. Reppa, V.; Polycarpou, M.M.; Panayiotou, C.G. Distributed sensor fault diagnosis for a network of interconnected cyberphysical systems. *IEEE Trans. Control Netw. Syst.* **2015**, *2*, 11–23.
18. Keliris, C.; Polycarpou, M.M.; Parisini, T. Distributed fault diagnosis for process and sensor faults in a class of interconnected input–output nonlinear discrete-time systems. *Int. J. Control* **2015**, *88*, 1472–1489.
19. Mahler, R.P.S. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1152–1178.
20. Vo, B.N.; Singh, S.; Doucet, A. Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 1224–1245.
21. Erdinc, O.; Willett, P.; Bar-Shalom, Y. Probability hypothesis density filter for multitarget multisensor tracking. In Proceedings of the 7th International Conference on Information Fusion, Philadelphia, PA, USA, 25–28 July 2005.
22. Delande, E.; Duflos, E.; Vanheeghe, P.; Heurguier, D. Multi-sensor PHD: Construction and implementation by space partitioning. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 3632–3635.
23. Xu, J.; Huang, F.; Huang, Z. The multi-sensor PHD filter: Analytic implementation via Gaussian mixture and effective binary partition. In Proceedings of the 16th International Conference on Information Fusion, Istanbul, Turkey, 9–12 July 2013; pp. 945–952.
24. Mahler, R. Objective functions for bayesian control-theoretic sensor management, 1: multitarget first-moment approximation. In Proceedings of the 2003 IEEE Aerospace Conference Proceedings, Big Sky, MT, USA, 8–15 March 2003; pp. 1905–1923.
25. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666.
26. Park, H.S.; Jun, C.H. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* **2009**, *36*, 3336–3341.
27. Hoffman, J.R.; Mahler, R.P. Multitarget miss distance via optimal assignment. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2004**, *34*, 327–336.
28. Schuhmacher, D.; Vo, B.T.; Vo, B.N. A Consistent Metric for Performance Evaluation of Multi-Object Filters. *IEEE Trans. Signal Process.* **2008**, *56*, 3447–3457.
29. Mehra, R.; Peschon, J. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica* **1971**, *7*, 637–640.
30. Lehmann, E.L.; D'abrera, H. *Nonparametrics: Statistical Methods Based on Ranks*; Holden-Day, Inc.: San Francisco, CA, USA, 1975.
31. Gibbons, J.D.; Chakraborti, S. *Nonparametric Statistical Inference*; CRC Press: Boca Raton, FL, USA, 2010.
32. Burges, C.J. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **1998**, *2*, 121–167.
33. Mahadevan, S.; Shah, S.L. Fault detection and diagnosis in process data using one-class support vector machines. *J. Process Control* **2009**, *19*, 1627–1639.

34. Gustafsson, F.; Isaksson, A.J. Best choice of coordinate system for tracking coordinated turns. In Proceedings of the 35th IEEE Conference on Decision and Control, Kobe, Japan, 13 December 1996; Vol. 3, pp. 3145–3150.
35. Nabaa, N.; Bishop, R.H. Validation and comparison of coordinated turn aircraft maneuver models. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 250–259.
36. Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1333–1364.