Algorithms and Analysis for Optimizing the Tracking Performance of Cyber Attacked Sensor-Equipped Connected Vehicle Networks

Zisheng Wang and Rick S. Blum, *Fellow, IEEE*

***Abstract*— Sensor-equipped connected vehicle networks (SECVNs) have the potential to enable substantially safer driving by improved object tracking, which is an important basic building block in SECVNs. Unfortunately, cyber-attacks on SECVNs pose a very serious threat which could lead to unacceptable outcomes, including fatalities. Recently there has been increasing focus on malicious attack detection and mitigation in SECVNs, and some of this work has considered attacks on sensor data to impact object tracking. Unfortunately, low complexity mitigation approaches which do not compromise performance are lacking. This paper describes an efficient machine-learning enhanced approach for tracking under cyber-attacks. By proper selection of some variances related to the sensor and prior probability density functions, under some assumptions the performance can be made as close as desired to a bound on the best possible performance. However, the complexity of this new approach is dramatically lower than the best existing published low complexity approach, which provides performance which is substantially inferior to that provided by the new approach. The new approach also provides much better scaling with the size of the SECVN. In particular, the complexity increases linearly in the number of sensors, while the best low complexity published approach has a complexity which grows quadratically in the number of sensors. The new approach is also applicable to other tracking applications.**

***Index Terms*— Cyber attacks, sensor-equipped connected vehicle networks, expectation-maximization algorithm.**

I. Introduction

IN RECENT years, there has been a surge in research and development efforts in advanced sensing, environmental perception, and intelligent driver assistance systems with the objective to make driving safer and reduce the number of on-road fatalities. To this end, sensor technology is being adopted by automotive manufacturers, creating sensor-equipped connected vehicle networks (SECVNs) which are distributed self-organized networks of many high-speed vehicles and infrastructure [1]. All vehicles in a SECVN employ sensors and onboard units (OBU), which integrate the vehicles' wireless communications, embedded systems, and global navigation satellite systems. These smart vehicles can then communicate with each other as well as with roadside units (RSUs), such as traffic lights or traffic signs, with the goal to improve the driving experience and make driving safer [2].

The constant monitoring of the position of a moving target in real time is termed tracking. Tracking of objects in the area monitored by the SECVN is an important fundamental building block for SECVNs which has received significant attention [3]–[5]. With the development of advanced sensing and wireless communications, the tracking performance can be enhanced through collaboration by sharing location-related information through vehicle-to-vehicle (V2V) communications. Tracking has been identified as a key issue for SECVNs since it informs all other required functions of the SECVN and intelligent transportation systems, such as navigation and collision avoidance [6].

Unfortunately, SECVNs are known for numerous security concerns [7] and are vulnerable to cyber-attacks. Since these attacks could lead to loss of human lives, cyber attacks on SECVNs are a very serious threat. A survey on the security and privacy concerns of SECVNs has been presented in [2]. An overview of some possible attacks on SECVNs has been presented in [8]. An analysis of the potential cyber-attacks specific to automated vehicles, with their special needs and vulnerabilities, is presented in [9].

Recently there has been increasing focus on malicious attack detection in SECVNs. A novel sybil attack detection scheme using a support vector machine (SVM) classifier is proposed in [10]. In [11], artificial neural networks are employed to detect denial of service (DoS) attacks. A distributed scheme to predict the behaviour of vehicles and prevent attacks in SECVNs is proposed in [12]. In Xue *et al.* [13], the authors consider detecting attacks that manipulate the stored vehicle location data on a vehicle. They identify manipulated data by comparison with known reliable data from another entity. None of the studies discussed in this paragraph are focused on approaches to detect and mitigate sensor data attacks

Manuscript received March 28, 2021; revised August 9, 2021 and September 13, 2021; accepted October 5, 2021. Date of publication October 25, 2021; date of current version November 9, 2021. This work was supported in part by the U.S. Army Research Laboratory, in part by the U.S. Army Research Office under Grant W911NF-17-1-0331, in part by the National Science Foundation under Grant ECCS-1744129, and in part by the Grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development through the Pennsylvania Infrastructure Technology Alliance (PITA). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mika Ylianttila. *(Corresponding author: Rick S. Blum.)*

The authors are with the Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18015 USA (e-mail: zs.wang.prc@gmail.com; rblum@lehigh.edu).

Digital Object Identifier 10.1109/TIFS.2021.3122070

on tracking in SECVNs and none provide low complexity algorithms that have been shown to provide near optimum after-attack performance as we do in this paper.

There are published research papers which study cyber-attacks on wireless sensor networks (WSNs) with stationary sensors which estimate the position of stationary objects [14]–[17]. The moving sensors and objects in SECVNs greatly change the problem, making the approaches in [14]–[17] unsuitable. In fact, we were unable to find any existing work focused on WSNs which was directly suitable for attack detection and mitigation of object tracking in SECVNs.

A number of publications have focused on studying cyber-security for state-space-modeled systems (SSMSs). The majority of this work focuses on general linear state-space equations which can be utilized for target tracking in SECVNs if the targets follow these linear models, a typical assumption. While work with very specific nonlinear state-space equations has appeared, it is not apparent that any of this work could be applied to target tracking for the types of SECVNs we consider. The most related linear SSMS work in terms of the goal of low complexity, near optimum, after-attack tracking appears to be [18]–[22]. In [21], a filter termed the observer is designed to detect if one or more of the sensors are under cyber-attacks for a linear SSMS described by linear equations and noise-free sensors. The observers proposed in [18], [20] consider the same linear SSMS except the sensor and dynamic model include Gaussian-distributed errors, which we call SSMS with Gaussian noise.

The work in [19] builds on that in [18], [20], ultimately demonstrating optimum performance for an observer based approach under the constraint of bounded noise, a case which does not strictly include Gaussian noise. To be clear, none of the approaches in [18]–[21] are shown to be optimum for SSMS with Gaussian noise. Further, the observer-based approaches generally require a large bank of observers running in parallel and generally search exhaustively over the outputs of all observers. The number of observers needed is equal to the number of all possible combinations of attacked and unattacked sensors, as expected from an exhaustive search. This leads to high complexity which grows rapidly with the system size, the number of sensors for example. Thus, none of the observer-based approaches can be considered low complexity. To address the unacceptable high complexity, [19] proposes a lower complexity approach. However, they show that this approach loses considerable performance, showing a large gap between its performance and that from their high complexity observer-based approach.

While the just discussed research [18]–[20] seems closest to our work, there has been other work that could be considered slightly related. For example, the authors in [22] discuss attacks on control signals and the authors in [23] discuss robust control approaches. However since we do not consider control, these works are not very related to our work. There is also some work [24]–[29] that tries to prevent or detect fake GPS signals. None of these works in [24]–[29] attempt to find a good track under attacked data and all of these methods seem complementary to our work. Therefore, one could use these approaches to try to detect fake GPS signals to augment the approaches taken in this paper. On the other hand, none of these approaches provide the analytical guarantees we provide (tracking performance as close to the best possible performance as desired by proper choice of the sensor characteristic) and we do not need these approaches to obtain our analytical guarantees, which are a major contribution of our work that other work does not provide.

In this paper, we describe a new approach, not based on the observer method, which we show provides much lower complexity when compared to [19]. On the other hand, we demonstrate numerically that our new method provides near optimum performance for sufficiently small (and practical) values of some variances related to the sensor and prior probability density functions (pdfs) by showing the performance is very close to a bound on the best possible performance. We also show analytically that the performance can be made as close to the bound as desired by choosing these variances sufficiently small under some assumptions. This improves the correct classification of both attacked and unattacked sensors which improves the tracking performance. In terms of scaling behavior, we specifically show that the complexity of our approach increases much more slowly than that of the best published low complexity approach we found [19], in terms of increases in the number of sensors or vehicles. Additionally, unlike some other algorithms, we do not need to have more than half the sensors be unattacked to correctly identify the attacked sensors due to our intelligent algorithm which uses all available information.

A. Motivational Example Using Realistic Urban Scenario Traffic Data

In order to overview the results presented later in a simple way, here we will present a motivational example using a realistic urban traffic scenario. The motivational example will show the utility of the work in this paper. We use the traffic simulator SUMO [30] which employs real maps of the city of Bethlehem, PA in the USA. The objective of this example is to track a Lehigh University commuter bus starting at the mountaintop campus of Lehigh University. The traffic includes 500 vehicles and 100 pedestrians with randomly chosen departure and arrival times.[1] Each vehicle is equipped with a GPS sensor and a RADAR sensor. In the motivational example, the sensors are assumed to obtain noisy and distorted measurements modeled by Gaussian noise-plus-distortion with signal-to-noise ratio of 20 dB (see numerical results for details). Later results generalize these assumptions. All data collected by the SECVN is used to generate an accurate track of the commuter bus.

Next, we discuss the cyber attacks considered in the motivational example. In the example, we consider GPS spoofing attacks which have been reported to be dangerous for SECVNs [24]–[29]. Later in the paper, we consider attacks on other sensor types besides GPS. For the cases considered in this motivational example and in our numerical results,

[1] In https://youtu.be/Z_5lC-kDB0o, we provide a video of the traffic.

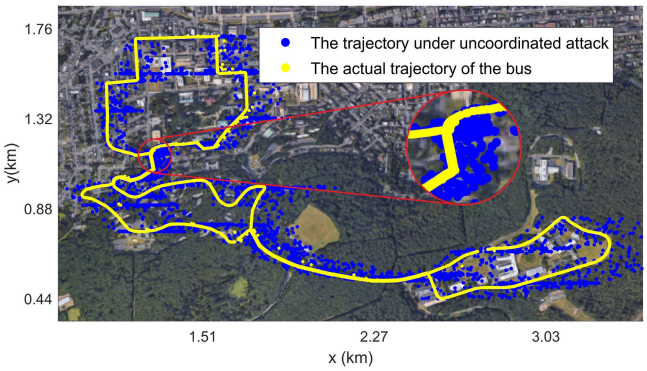


Fig. 1. The trajectory generated by the SECVN tracking algorithm under an uncoordinated GPS spoofing attack compared with the actual trajectory.

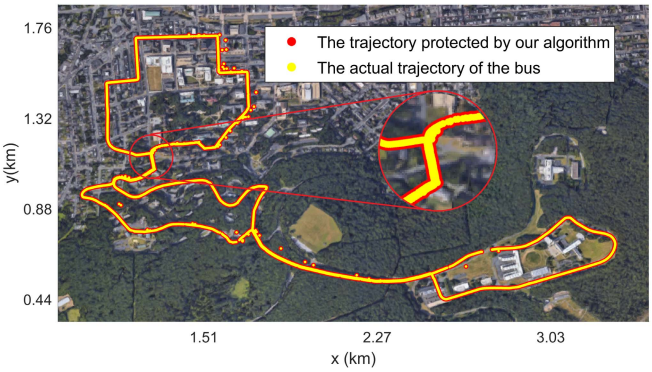


Fig. 2. The tracking generated trajectory when protected by the proposed algorithm compared with the actual trajectory.

our theory can handle all types of attacks on the considered sensors. The paper provides some minor assumptions required for more general sensor types beyond those considered in the motivational example and numerical results. As we focus on tracking in this paper, in the motivational example we assume that the trajectory the actual bus will take is fixed and not impacted by the attacks on the sensor data. We call this the actual trajectory. The attackers only alter the sensor measurements which only impact the tracking algorithm. We assume these sensor measurements are not used for control of the bus just to decouple tracking and control. The tracking algorithm does not know the actual trajectory of the bus. Two typical GPS spoofing attacks are considered:

- *Uncoordinated GPS spoofing attacks*: In these cases the attacker is assumed to not know the actual trajectory of the bus so the GPS sensors are assumed to be spoofed such that the measured position has an additional error (unrelated to the actual trajectory) to attempt to distort the tracked vehicle position. In the motivational example, we assume the errors at different time steps are uniformly distributed from -100 to 100 meters in both axes of each vehicle.
- *Coordinated GPS spoofing attacks*: In these cases the attacker is assumed to know the actual trajectory of the bus so the GPS sensors are spoofed such that the tracked trajectories after attack can be different from the actual trajectory in a carefully planned manner. In the motivational example, we spoof the GPS sensor of the commuter bus such that the attacked tracking trajectory of the commuter bus appears to detour off the desired path for a short distance.

GPS spoofing has been reported in practical scenarios. In [31], the authors are able to successfully spoof the GPS signal received by a cruise ship to steer it off course. In 2011 Iran was able to spoof the GPS in a CIA drone [32].

We will compare the after attack tracks generated by the SECVNs when applying either our algorithm or no defense. The actual track of the vehicle will also be presented for reference. Note that the uncoordinated and coordinated GPS spoofing attacks are simply motivational examples. As shown in the sequel, the proposed algorithm is able to handle cyber attacks with a general setting which may be launched on most existing types of sensors. Fig. 1 shows the trajectories generated by the SECVN. The yellow curve shows the actual trajectory while the blue curve shows the tracking trajectory under uncoordinated GPS spoofing attacks with no protection. Next, in Fig 2, we show the trajectory of the commuter bus when the SECVN is protected by the secure tracking algorithm that will be described in the sequel. In Fig. 2, the red curve shows the trajectory after protection by our algorithm. Based on Fig. 1 and Fig. 2, it is clear that cyber attacks without protection can dramatically impact the tracking and that the protected trajectory is much closer to the actual trajectory.

Next, in Fig. 3, we show the trajectories generated by the tracking algorithm employed by the SECVN under coordinated attacks with various levels of protection. The blue curve shows the trajectory under coordinated GPS spoofing attacks without any protection. As shown by the blue curve, the SECVN tracking trajectory deviates from the actual path (indicated by the yellow curve). However, our secure tracking algorithm (red curve) is able to generate a tracked trajectory that is nearly indistinguishable from the actual path.

In this paper, we will show (under reasonable assumptions) that the tracking performance of our algorithm can be made as close to the best possible performance as desired by proper choice of the sensor characteristic, e.g. the variance of sensor noise-plus-distortion. We next illustrate this with a motivational example in Fig. 4. Consider an uncoordinated GPS spoofing attack where the sensor noise-plus-distortion power is set to either 0.1, which is the high quality sensor case shown in orange, or where the sensor noise-plus-distortion power is set to 1 which is the low quality sensor case shown in blue. To obtain the blue and orange regions in Fig. 4, we run 10 Monte Carlo simulations and save the vehicle tracking trajectories with both high (orange) and low (blue) quality sensing. Then the blue and orange regions shade all points between the actual track and the farthest trajectory point either above or below the the actual track. The improvements from better sensing are clear from using the proposed protection approach and it is clear that performance can be made as close to the actual path as desired by improving the sensing, even if attacks are present in the sensor data with our approach.

We compared our algorithm with what is said to be the lowest complexity approximation of an optimum performance approach in the literature [19], which we previously discussed in some detail, and found our approach is lower complexity.

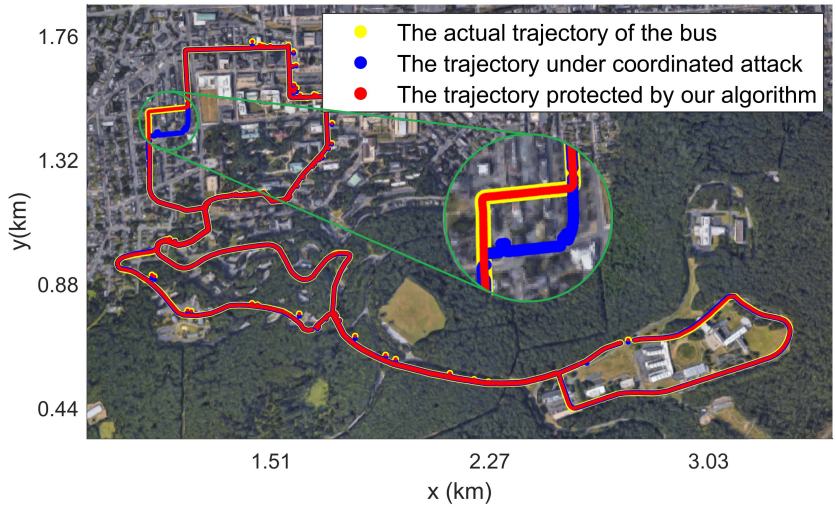


Fig. 3. The trajectories generated by the SECVN tracking algorithm under coordinated GPS spoofing attacks.

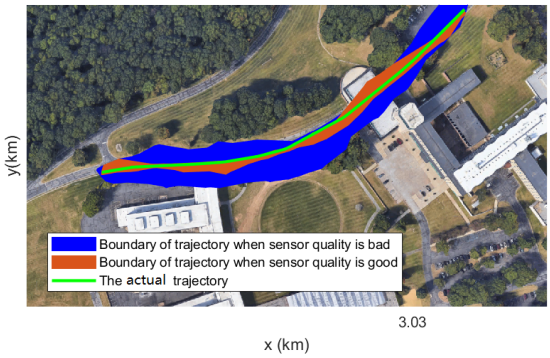


Fig. 4. Largest deviations from actual path after attack for with high (orange) or low (blue) quality sensing.

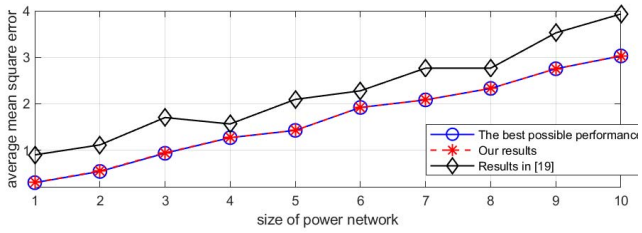


Fig. 5. The average mean squared error of our algorithm, the algorithm in [19], and the best possible approach for the power grid example from [19].

Consider the case with 10 vehicles and 10 pedestrians. Each vehicle has GPS and radar sensors for inter-vehicle relative distance and pedestrian-to-vehicle relative distance. In this case, at each time step, our algorithm requires overall 4.625×10^6 multiplication and 1.797×10^6 addition operations. On the other hand, one part of the approach in [19] requires 2.561×10^8 multiplication and 2.561×10^8 addition operations. Therefore, our algorithm is at least 79.7763 times faster. If we increase the number of vehicles and pedestrians to 20, we are at least 322.0020 times faster. As we will show in a later section, our algorithm will provide an even larger complexity advantage if there are more sensors or objects to track. Even with the lower complexity, our algorithm also provides better performance, for all cases tested, when compared to the results provided in [19]. We illustrate this for the power grid application considered in [19] (using their results), where performance is described in terms of the average mean squared error of the estimated power grid state variables. The average mean squared error of our algorithm, the algorithm in [19], and the best possible performance are shown in Fig. 5 for various sizes of the power system [19]. Note that our algorithm's performance is very close (indistinguishable in Fig. 5) to the best possible performance (which has been proven under some conditions as discussed later). This is not true for the approach in [19].

B. Notation and Organization

Any bold uppercase letter, like $\boldsymbol{A}$, denotes a matrix while any bold lowercase letter, like $\boldsymbol{a}$, denotes a vector. The plain lowercase and uppercase letters denote scalars. Let $\boldsymbol{A}\otimes\boldsymbol{B}$ denote the Kronecker product between $\boldsymbol{A}$ and $\boldsymbol{B}$. Let $\mathcal{N}(\boldsymbol{a},\boldsymbol{A})$ denote a multivariate Gaussian pdf with a covariance matrix $\boldsymbol{A}$ and a mean vector $\boldsymbol{a}$.

The remainder of this paper is organized as follows. Section II introduces general attacks on tracking in a general SECVN. Section III describes the notation and the attack model on which we focus. Example sensor measurement and tracked target motion models are also described. Section IV describes our unsupervised machine learning based methodology to track objects in the presence of attacks. Section V presents a bound on the optimum achievable tracking performance under attacks. Section VI presents numerical results and analysis. Section VII presents analytical results describing conditions under which our algorithm will provide performance close to the bound. Section VIII presents conclusions and discussions.

II. Overview of Tracking Under General Attacks

Object tracking is one of the most basic building blocks in sensor-equipped connected vehicle network applications, since outputs from the object tracking system are critical inputs for most of the other important systems in sensor-equipped connected vehicle network applications. In this

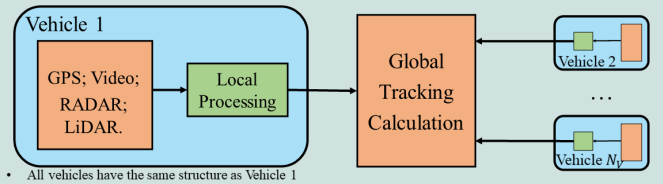


Fig. 6. Block diagram of unattacked tracking system in SECVNs.

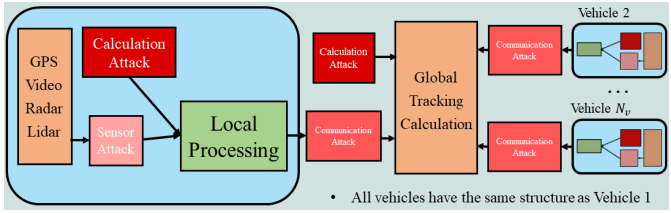


Fig. 7. Block diagram of attacked tracking system in SECVNs.

section, we provide an abstract high level description of a object tracking system in a sensor-equipped connected vehicle network. Then we describe attacks on object tracking and we outline an approach for modeling all possible attacks on object tracking.

A. Unattacked Tracking System in Sensor-Equipped Connected Vehicle Networks

Fig. 6 illustrates a system level block diagram illustrating the sensors, the processing subsystems and the data flows between them for object tracking in SECVNs. Data flows emerging from non-vehicles are omitted in Fig. 6. These include those from roadside infrastructures, but these would look similar to the addition to another vehicle. For simplicity, a single separate entity is shown where all the vehicle data is employed to perform a global tracking calculation. In practice, this calculation could be done at each vehicle. As shown, each vehicle is equipped with various sensors, such as GPS units, RADARs, LiDARs, and video cameras.

B. Attacked Tracking System in Sensor-Equipped Connected Vehicle Networks

Any attack that would degrade tracking accuracy would have to change some signals inside some blocks in Fig. 6. In general, all such changes can be reproduced by either changing the data coming into a processing subsystem, be it local or global processing, or by modifying a calculation in the processing subsystem. We define sensor attacks as those which change data prior to local processing in Fig. 6. These attacks also occur prior to any off vehicle communications. Their impact can always be exactly reproduced by changing the data produced by the sensor which justifies their name. Communication attacks change data entering a global processing subsystem. Their impact can always be exactly reproduced by changing the data communicated from one or more vehicles to the global tracking calculation which justifies their name. Calculation attacks modify calculations in a processing subsystem, be it a local processing subsystem or the global processing subsystem. Fig. 7 illustrates this efficient approach for modeling all possible attacks that impact object tracking. In the most general sense, all the signals in Fig. 6 can have a random aspect and so the attacks in Fig. 7 can generally modify the statistical descriptions of these signals.

C. Attack Model: Focus on Sensor Attacks Passing Bad Data Detection

Here we focus on sensor attacks since these attacks are very important. Examples of sensor attacks were given in the motivational example in the Introduction. Similar approaches can be developed for the other types of attacks discussed in Fig. 7 but we omit discussion here. We assume the tracking system will operate using standard statistical formulations which have been extremely successful over the years [4], [33] and which are employed in many existing systems which track objects. These statistical formulations employ a statistical model (which could be learned) for the observed sensor data. In such cases, the system can employ bad data detectors which check if the data fits the model and flag any data not fitting the model. Thus, our algorithms only need to consider attacks that can pass the bad data detector which justifies a focus on such attacks in this paper. The general form of attacks that we consider in all the numerical results in the paper, including the motivational example, subsume all possible attacks on the sensor data. Mathematical descriptions of the attacks, the sensor data and the objects being tracked are provided in the following section. We note that we can extend the ideas given here to cases with more general attacks and tracking approaches.

III. Notation, Attack Model, and Assumptions

Let $\mathcal{V} = \{1, 2, \ldots, N_v\}$ denote a set of vehicles located over a two-dimensional space who will cooperate to locate and track objects. Let $\mathcal{P} = \{1, 2, \ldots, N_p\}$ denote a set of objects which do not cooperate, including pedestrians, located in the same two-dimensional space as the cooperating vehicles. Let $N = N_v + N_p$ denote the total number of objects (vehicles, pedestrians, and other objects) in the SECVN. We employ discrete-time models with the time between samples fixed at T_s. At discrete time $k = 1, 2, \ldots$, each vehicle $i \in \mathcal{V}$ has state $\boldsymbol{x}_{i,k} = [\boldsymbol{p}_{i,k}^T \quad \boldsymbol{v}_{i,k}^T]^T$ in which $\boldsymbol{p}_{i,k} = [p_{x,i,k} \quad p_{y,i,k}]^T$ denotes the position vector and $\boldsymbol{v}_{i,k} = [v_{x,i,k} \quad v_{y,i,k}]^T$ denotes the velocity vector. At discrete time k, each non-cooperating object $j \in \mathcal{P}$ has state $\boldsymbol{y}_{j,k} = [\boldsymbol{p}_{\mathrm{NV},j,k}^T \quad \boldsymbol{v}_{\mathrm{NV},j,k}^T]^T$. Let $\boldsymbol{c}_{i,k}$ denote the acceleration of the i-th vehicle at time k. Let $\boldsymbol{c}_k = \left[\boldsymbol{c}_{1,k}^T \quad \ldots \quad \boldsymbol{c}_{N_v,k}^T\right]^T$. At discrete time $k = 1, 2, \ldots$, collect all the state information in

$$\boldsymbol{\theta}_k = \left[\boldsymbol{x}_{1,k}^T \quad \ldots \quad \boldsymbol{x}_{N_v,k}^T \quad \boldsymbol{y}_{1,k}^T \quad \ldots \quad \boldsymbol{y}_{N_p,k}^T\right]^T, \qquad (1)$$

a vector containing all positions and velocity vectors of cooperating vehicles and non-cooperating objects at time k. Let D denote the dimension of $\boldsymbol{\theta}_k$

As described previously, we focus on sensor attacks. Thus, we first describe the notation and models for the sensor observations. Each vehicle is equipped with various sensors, such as GPS units, RADARs, LiDARs, video cameras and

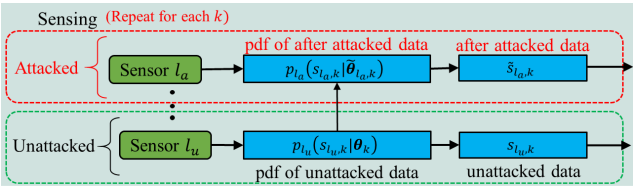


Fig. 8. Illustration of generation of attacked and unattacked sensor data.

other sensors, which produce data at each discrete time k that can be used to estimate the components of $\boldsymbol{\theta}_k$. Let $\boldsymbol{s}_k$ denote the vector of all the sensor data from all vehicles at time k. Each component of $\boldsymbol{s}_k$ is a scalar which comes from some sensor (may be one dimension of a larger position or velocity vector) and $\boldsymbol{s}_k$ has S components. Thus the scalar $s_{l,k}$ denotes the l-th component of $\boldsymbol{s}_k$, $l = 1, \ldots, S$ and for convenience we will call $s_{l,k}$ the data of the l-th sensor out of S total sensors. Let $p_l(s_{l,k}|\boldsymbol{\theta}_k)$ denote the pdf of the l-th sensor data $s_{l,k}$ conditioned on $\boldsymbol{\theta}_k$ prior to any attack, defined for $l = 1, \ldots, S$ and $k = 1, \ldots, \infty$.

A. Attack Model

As stated in the previous section, we focus on sensor attacks which are crafted to pass bad data detection. We consider all such attacks on the sensor data. This leads to the following definition.

Definition 1 (Attack Model): Let $\widetilde{s}_{l,k}$ denote the attacked sensor data. To pass bad data detection, the attacked data must come from $p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}_{l,k})$ where $\widetilde{\boldsymbol{\theta}}_{l,k}$ is called the attacked state.

Since the unattacked sensor data $s_{l,k}$ follows the statistical model (the pdf) $p_l(s_{l,k}|\boldsymbol{\theta}_{l,k})$, definition 1 states that we assume the attacks generate data $\widetilde{s}_{l,k}$ which follows the assumed model $p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}_{l,k})$ so the attacks cannot be detected by performing a check to see if this is true (bad data detection). Thus any such attacks are equivalent to changing $\boldsymbol{\theta}_k$ in the original unattacked model $p_l(s_{l,k}|\boldsymbol{\theta}_k)$ to

$$\widetilde{\boldsymbol{\theta}}_{l,k} = \boldsymbol{\theta}_k + \boldsymbol{a}_{l,k}, \quad \forall k, l, \tag{2}$$

for some vector $\boldsymbol{a}_{l,k}$. For the sensor models considered in any numerical results in this paper, which observe noisy and distorted versions of $\boldsymbol{\theta}_k$, any change in the sensor observations can be represented using (2) so we model all possible attacks.

The subscript l on $\widetilde{\boldsymbol{\theta}}_{l,k}$ implies that a different attack can be launched at each sensor. Collect the attacked states at each sensor as $\widetilde{\boldsymbol{\Theta}}_k = \begin{bmatrix} \widetilde{\boldsymbol{\theta}}_{1,k}^T & \ldots & \widetilde{\boldsymbol{\theta}}_{S,k}^T \end{bmatrix}^T$. The difference between attacked and unattacked sensor data is summarized in Fig. 8.

B. Assumptions and Available Information

The following assumptions are made in the rest of this paper unless otherwise stated.

Assumption 1: The l-th sensor data at time k follows

$$s_{l,k} = g_l(\boldsymbol{\theta}_k) + n_{l,k}, \quad \forall l. \tag{3}$$

In (3), $g_l(\boldsymbol{\theta}_k)$ denotes the conditional mean of this sensor's observation conditioned on the state $\boldsymbol{\theta}_k$ and $n_{l,k}$ is random (not necessarily Gaussian) with a zero mean and variance σ_l^2, $l = 1, \ldots, S$. The variance σ_l^2 is not dependent on the state. Note that $g_l : \mathbb{R}^D \to \mathbb{R}$ is a known function (could be learned) describing how the state impacts the l^{th} sensor observation.

Assumption 2: Conditioned on $\boldsymbol{\theta}_k$, for $l = 1, \ldots, S$, the data generated by the l-th sensor is statistically independent of the data from the other sensors.

Assumption 3: The prior pdf $p(\boldsymbol{\theta}_k)$ is assumed known and unattacked. Let $\sigma_{S,l}^2$ denote the known variance of $g_l(\boldsymbol{\theta}_k)$ when $\boldsymbol{\theta}_k$ follows $p(\boldsymbol{\theta}_k)$. We call $\sigma_{S,l}^2$ the prior variance of the distortion-free, noiseless l^{th} sensor observation (see (3)).

Assumption 4: To be classified as an attack, we require attacked data to come from a pdf with a parameter $\widetilde{\boldsymbol{\theta}}_{l,k}$ (estimated in practice) satisfying

$$\left\| g_l(\widetilde{\boldsymbol{\theta}}_{l,k}) - E(g_l(\boldsymbol{\theta}_k)) \right\| \geq \epsilon_l, \quad \forall l, k. \tag{4}$$

where intuitively $\epsilon_l, \forall l$ is selected to ignore attacks which cause small damage (similar to expected statistical variations) and the expectation is taken with respect to the prior pdf $p(\boldsymbol{\theta}_k)$.

Numerical investigations suggest a good choice of ϵ_l is approximately equal to the square root of the sum of $\sigma_{S,l}^2$ and σ_l^2. Further discussion is given in Appendix C. Our algorithm assumes that the quantities listed in Table I are available. Generalizations are possible.

TABLE I
ASSUMED AVAILABLE QUANTITIES FOR OUR ALGORITHM

Notation	Description
$\widetilde{\boldsymbol{s}}_k, \forall k$	Sensor data observations.
$p(n_{l,k})$	Pdf of $n_{l,k}$.
$g_l(\boldsymbol{\theta}_k)$	See (3).
$p(\boldsymbol{\theta}_k)$	Pdf of unattacked state prior to sensor observations.
$\sigma_{S,l}^2$ and σ_l^2	See Assumption 3.

C. Discussion of Assumptions and Available Quantities

Let us first discuss Assumptions 1-4. Assumption 1 represents a large class of practical problems and allows significantly simpler analysis (see Section VII). Extensions are possible using similar ideas. In real systems, designers would work hard to approximately eliminate any of the dependencies disallowed by Assumption 2. While attacks on sensor data must be detected rapidly since this data will be used very shortly after it is generated, there is typically much more time available to check the prior pdf $p(\boldsymbol{\theta}_k)$. Thus, Assumption 3 seems to be a reasonable first order assumption which allows us to decouple this aspect. Protecting the prior pdf $p(\boldsymbol{\theta}_k)$ can be a topic of future work. In Assumption 4, (4) indicates that we only classify sensor data with $g_l(\widetilde{\boldsymbol{\theta}}_{l,k})$ which is sufficiently far from $E(g_l(\boldsymbol{\theta}_k))$ as attacked, thus much larger than the expected statistical changes due to the random state or noise. Ignoring attacks causing small or no change in the observed data as per Assumption 4 would seem to be required in practice. Next we discuss the quantities assumed to be available as mentioned in Table 1.

A pdf of the sensor data conditioned on $\boldsymbol{\theta}_k$, as described in Table I, is a pretty general way to provide a needed statistical description of the sensor data. This pdf could be, for example, derived from a mathematical model. As a simple

example of such a model, consider a case where each cooperating vehicle has a GPS unit providing an inexact estimate of its position vector, denoted by

$$\boldsymbol{\rho}_{i,k} = \boldsymbol{p}_{i,k} + \boldsymbol{n}_{\mathrm{GPS},i,k}, \quad i = 1, \ldots, N_v, \tag{5}$$

where $\boldsymbol{n}_{\mathrm{GPS},i,k}$ is an error vector with a possibly complicated distribution. Assume each cooperating vehicle has a sensor providing an inexact estimate of its velocity vector, denoted by

$$\boldsymbol{\xi}_{i,k} = \boldsymbol{v}_{i,k} + \boldsymbol{n}_{\mathrm{SPD},i,k}, \quad i = 1, \ldots, N_v, \tag{6}$$

where $\boldsymbol{n}_{\mathrm{SPD},i,k}$ is an error vector with a possibly complicated distribution. Similarly, each cooperating vehicle has an inexact estimate of each non-cooperating object velocity vector, denoted by

$$\boldsymbol{\eta}_{j,k} = \boldsymbol{p}_{\mathrm{NV},j,k} + \boldsymbol{n}_{\mathrm{SPDNV},j,k}, \quad j = 1, \ldots, N_p, \tag{7}$$

where $\boldsymbol{n}_{\mathrm{SPDNV},j,k}$ is an error vector with a possibly complicated distribution. In addition, in this example each cooperating vehicle obtains an estimate of the cooperating inter-vehicle and vehicle-to-non-cooperating-object distances, possibly from fusion of radar and video sensor data, which are modeled respectively as

$$\boldsymbol{d}_{i_1,i_2,k} = \begin{cases} \boldsymbol{p}_{i_1,k} - \boldsymbol{p}_{i_2,k} + \boldsymbol{n}_{\mathrm{VV},i_1,i_2,k}, & \|\boldsymbol{p}_{i_1,k} - \boldsymbol{p}_{i_2,k}\| < R_s \\ \boldsymbol{n}_{\mathrm{VV},i_1,i_2,k}, & \text{otherwise,} \end{cases} \tag{8}$$

and

$$\boldsymbol{g}_{i,j,k} = \begin{cases} \boldsymbol{p}_{i,k} - \boldsymbol{p}_{\mathrm{NV},j,k} + \boldsymbol{n}_{\mathrm{VN},i,j,k}, & \|\boldsymbol{p}_{i,k} - \boldsymbol{p}_{\mathrm{NV},j,k}\| < R_s \\ \boldsymbol{n}_{\mathrm{VN},i,j,k}, & \text{otherwise,} \end{cases} \tag{9}$$

where $i_1 = 1, \ldots, N_v, i_2 = 1, \ldots, N_v, i = 1, \ldots, N_v$, $j = 1, \ldots, N_p$, and R_s denotes the sensing range. In (8) and (9), $\boldsymbol{n}_{\mathrm{VV},i,j,k}$ and $\boldsymbol{n}_{\mathrm{VN},i,j,k}$ are error vectors with possibly complicated distributions. Then, the sensor measurements $\boldsymbol{s}_k$ can be collected into a vector as

$$\begin{aligned} \boldsymbol{s}_k = \Big[& \boldsymbol{\rho}_{1,k}^T \ \boldsymbol{\xi}_{1,k}^T \ \cdots \ \boldsymbol{\rho}_{N_v,k}^T \ \boldsymbol{\xi}_{N_v,k}^T \ \boldsymbol{d}_{1,1,k}^T \ \cdots \\ & \boldsymbol{d}_{N_v,N_v,k}^T \ \boldsymbol{g}_{1,1,k}^T \ \cdots \ \boldsymbol{g}_{N_v,N_p,k}^T \ \boldsymbol{\eta}_{1,k}^T \ \cdots \ \boldsymbol{\eta}_{N_p,k}^T \Big]^T. \end{aligned} \tag{10}$$

Thus $s_{1,k}$ is the first component of $\boldsymbol{\rho}_{1,k}^T$, $s_{2,k}$ is the second component of $\boldsymbol{\rho}_{1,k}^T$, and so on. The error vectors can be collected into a common vector

$$\begin{aligned} \boldsymbol{n}_k = \Big[& \boldsymbol{n}_{\mathrm{GPS},1,k}^T \ \cdots \ \boldsymbol{n}_{\mathrm{GPS},N_v,k}^T \ \boldsymbol{n}_{\mathrm{SPD},1,k}^T \ \cdots \ \boldsymbol{n}_{\mathrm{SPD},N_v,k}^T \\ & \boldsymbol{n}_{\mathrm{VV},1,1,k}^T \ \cdots \ \boldsymbol{n}_{\mathrm{VV},N_v,N_v,k}^T \ \boldsymbol{n}_{\mathrm{VN},1,1,k}^T \ \cdots \ \boldsymbol{n}_{\mathrm{VN},N_v,N_p,k}^T \\ & \boldsymbol{n}_{\mathrm{SPDNV},1,k} \ \cdots \ \boldsymbol{n}_{\mathrm{SPDNV},N_p,k} \Big]^T. \end{aligned} \tag{11}$$

Using $\boldsymbol{n}_k$, (5)-(10) and the definition of $\boldsymbol{\theta}_k$ in (1), the unattacked sensor measurements can be written as [34].

$$\begin{aligned} \boldsymbol{s}_k &= \begin{pmatrix} \boldsymbol{I}_{4N_v} & \boldsymbol{0}_{4N_v \times 4N_p} \\ \boldsymbol{1}_{N_p} \otimes \boldsymbol{D}_1 & -\boldsymbol{I}_{N_p} \otimes \boldsymbol{D}_2 \\ \boldsymbol{1}_{N_v} \otimes \boldsymbol{D}_1 - \boldsymbol{I}_{N_v} \otimes \boldsymbol{D}_2 & \boldsymbol{0}_{2N_v^2 \times 4N_v^2} \\ \boldsymbol{0}_{2N_p \times 4N_v} & \boldsymbol{D}_3 \end{pmatrix} \boldsymbol{\theta}_k + \boldsymbol{n}_k, \\ &= \boldsymbol{H}\boldsymbol{\theta}_k + \boldsymbol{n}_k, \end{aligned} \tag{12}$$

where $\boldsymbol{D}_1 = \boldsymbol{I}_{N_v} \otimes \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{0}_2 \end{bmatrix}$, $\boldsymbol{D}_2 = \boldsymbol{1}_{N_v} \otimes \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{0}_2 \end{bmatrix}$, $\boldsymbol{D}_3 = \boldsymbol{I}_{N_p} \otimes \begin{bmatrix} \boldsymbol{0}_2 & \boldsymbol{I}_2 \end{bmatrix}$ and $\otimes$ denotes the kronecker product. Let $\boldsymbol{h}_l^T$ denote the l-th row of $\boldsymbol{H}$. It is clear that (12) follows Assumption 1 by setting $g_l(\boldsymbol{\theta}_k) = \boldsymbol{h}_l^T \boldsymbol{\theta}_k, l = 1, \ldots, S$.

Obtaining $p(\boldsymbol{\theta}_k)$ from Table I:

There are many approaches to obtain the pdf of the state, prior to sensor observations, $p(\boldsymbol{\theta}_k)$ [35], which is called the prior pdf here since we track the state. The prior pdf can be determined based on physics or from data. We can also combine data and physics to produce a highly informative prior pdf. Here we describe one physics-based approach which uses an object motion model. One of the most popular object motion models is described in [4]. Let

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{I}_2 & T_s \boldsymbol{I}_2 \\ \boldsymbol{0}_2 & \boldsymbol{I}_2 \end{bmatrix} \otimes \boldsymbol{I}_N \quad \text{and} \quad \boldsymbol{B} = \begin{bmatrix} \frac{T_s^2}{2} \boldsymbol{I}_2 \\ T_s \boldsymbol{I}_2 \end{bmatrix} \otimes \boldsymbol{I}_N. \tag{13}$$

Based on (1) and assuming that the non-cooperating object accelerations (pedestrians) are zero for simplicity (easily generalized), the model is given by

$$\boldsymbol{\theta}_k = \boldsymbol{A}\boldsymbol{\theta}_{k-1} + \boldsymbol{B} \begin{bmatrix} \boldsymbol{c}_{k-1} \\ \boldsymbol{0}_{2N_p \times 1} \end{bmatrix} + \boldsymbol{r}_k, \tag{14}$$

where $\boldsymbol{r}_k$ is a random vector with a general distribution. This model can be generalized to include non-linear dynamics as

$$\boldsymbol{\theta}_k = \tilde{g}(\boldsymbol{\theta}_{k-1}, \boldsymbol{c}_{k-1}, \boldsymbol{r}_{i,k}), \tag{15}$$

when $\tilde{g}$ is a non-linear function. The prior pdf $p(\boldsymbol{\theta}_k)$ can be obtained from (14) or (15).

IV. Unsupervised Machine Learning Algorithm to Handle Attacks

In this section, we propose an algorithm to track objects accurately in the presence of sensor attacks under the assumptions given in the previous section. Fig. 9 illustrates the data flow and structure of the proposed algorithm. The algorithm attempts to identify which sensor data observed at time k is under attack and then uses this to predict the state at the next time (tracking) based on past predictions.

As shown in Fig. 9, the data $\widetilde{\boldsymbol{s}}_k$, possibly attacked, is sent to an attack analyzer and an object tracker. Let $\boldsymbol{z}$ denote a hidden discrete latent vector. When the l-th sensor is under a sensor attack, the l-th component of $\boldsymbol{z}$, denoted by z_l, is equal to one. Otherwise, $z_l = 0$. The attack analyzer employs a modified expectation-maximization (EM) algorithm to generate soft decisions $\pi_l \in [0, 1], l = 1, \ldots, S$ estimating the probability that the l-th sensor is under attack ($z_l = 1$) based on available information. Let $\boldsymbol{\pi} = [\pi_1, \ \ldots, \ \pi_S]^T$ denote a vector containing all soft decisions. The soft decisions are sent to the object tracker to produce an accurate track of all objects at time k even when the sensor data is attacked as per Definition 1 and Assumptions 1-4 in Section III.

A. Attack Analysis and Object Tracking Algorithm

The first step in deriving any EM algorithm is the specification of a set of complete data and incomplete data for the problem. The pdfs for the complete and incomplete data are

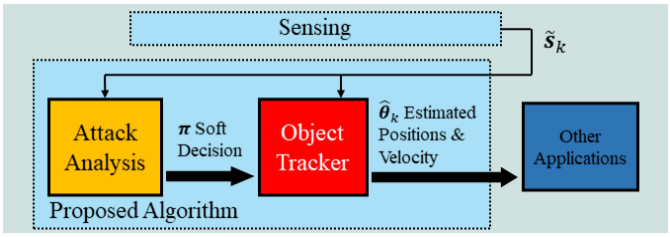


Fig. 9. Data flows of vehicle sensor networks and the proposed algorithm.

characterized by a common set of parameters. The complete data is not available, but if it was available it would simplify the estimation of the common parameters. Only the incomplete data is available. The EM algorithm addresses this situation and provides an iterative procedure for estimating the common parameters based on the incomplete data.

For our problem, the complete data includes the sensor data $\tilde{s}_k$, the state $\boldsymbol{\theta}_k$ and the latent vector z. The incomplete data includes just the sensor data $\tilde{s}_k$. The common parameters include the soft decision $\boldsymbol{\pi}$ and the attacked states $\widetilde{\boldsymbol{\Theta}}_k$. In the EM algorithm, we will update $\boldsymbol{\pi}$ and $\widetilde{\boldsymbol{\Theta}}_k$ so it is useful to denote the current value of these parameters, prior to update, by $\boldsymbol{\pi}'$ and $\widetilde{\boldsymbol{\Theta}}'_k$. The SAGE algorithm, one specific version of EM algorithm, is used. In SAGE, during each iteration $\boldsymbol{\pi}$ is updated with $\widetilde{\boldsymbol{\Theta}}_k$ held fixed, followed by the update of $\widetilde{\boldsymbol{\Theta}}_k$ with $\boldsymbol{\pi}$ held fixed. The sequence of estimates produced by the SAGE algorithm has non-decreasing likelihood for the incomplete data [36].

In the following we employ conditional probabilities and Bayes rule in deriving the expressions needed for the EM algorithm. Let $p(\tilde{s}_k|z, \boldsymbol{\theta}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k)$ denote the pdf of the sensor data conditioned on z and $\boldsymbol{\theta}_k$ when the parameters are set as the current estimates $\boldsymbol{\pi} = \boldsymbol{\pi}'$ and $\widetilde{\boldsymbol{\Theta}}_k = \widetilde{\boldsymbol{\Theta}}'_k$. Recall that z_l is an indicator variable describing if the l-th sensor is under attack. Using the attack model described in the previous section and Assumption 2, the pdf of the incomplete data $\tilde{s}_k$ conditioned on z and $\boldsymbol{\theta}_k$ when $\boldsymbol{\pi}$ and $\widetilde{\boldsymbol{\Theta}}_k$ are set to $\boldsymbol{\pi}'$ and $\widetilde{\boldsymbol{\Theta}}'_k$ becomes

$$p(\tilde{s}_k|z, \boldsymbol{\theta}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k) = \prod_{l=1}^{S} \left\{ \left[p_l(\tilde{s}_{l,k}|\tilde{\boldsymbol{\theta}}'_{l,k}) \right]^{\mathbb{1}_{z_l=1}} \left[p_l(\tilde{s}_{l,k}|\boldsymbol{\theta}_k) \right]^{\mathbb{1}_{z_l=0}} \right\}, \quad (16)$$

which does not depend on $\boldsymbol{\pi}'$. Using $p(A, B) = p(A|B)p(B)$ and (16), the probability description (pdf of continuous variables and probability mass function (pmf) of discrete variables) of the complete data $\tilde{s}_k$, z, and $\boldsymbol{\theta}_k$ when $\boldsymbol{\pi}$ and $\widetilde{\boldsymbol{\Theta}}_k$ are set to $\boldsymbol{\pi}'$ and $\widetilde{\boldsymbol{\Theta}}'_k$ is given by

$$\begin{aligned} p(\tilde{s}_k, z, \boldsymbol{\theta}_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k) &= p(\tilde{s}_k|z, \boldsymbol{\theta}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k)\, p(z, \boldsymbol{\theta}_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k) \\ &= p(\tilde{s}_k|z, \boldsymbol{\theta}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k)\, p(z|\boldsymbol{\theta}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k)\, p(\boldsymbol{\theta}_k) \\ &= p(\boldsymbol{\theta}_k) \prod_{l=1}^{S} \left[p_l(\tilde{s}_{l,k}|\tilde{\boldsymbol{\theta}}'_{l,k})\pi'_l \right]^{\mathbb{1}_{z_l=1}} \left[p_l(\tilde{s}_{l,k}|\boldsymbol{\theta}_k)(1-\pi'_l) \right]^{\mathbb{1}_{z_l=0}}, \end{aligned} \quad (17)$$

where we dropped conditioning variables that are unrelated to the probability being computed and used $p(z|\boldsymbol{\theta}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k) = \prod_{l=1}^{S} {\pi'_l}^{\mathbb{1}_{z_l=1}} (1-\pi'_l)^{\mathbb{1}_{z_l=0}}$ for the conditional pmf.

Let[2] $\upsilon_l = E\{\mathbb{1}_{z_l=1}|\tilde{s}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k\} = \int_{\boldsymbol{\theta}_k} p(z_l = 1, \boldsymbol{\theta}_k|\tilde{s}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k) d\boldsymbol{\theta}_k$. The E-step (expectation) of the EM algorithm performs an average of $\ln p(\tilde{s}_k, z, \boldsymbol{\theta}_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k)$ from (17) over the unavailable parts of the complete data $z, \boldsymbol{\theta}_k$ by conditioning on the incomplete data $\tilde{s}_k$ where $\boldsymbol{\pi}$ and $\widetilde{\boldsymbol{\Theta}}_k$ are set to $\boldsymbol{\pi}'$ and $\widetilde{\boldsymbol{\Theta}}'_k$, as in

$$\begin{aligned} &Q(\boldsymbol{\pi}, \widetilde{\boldsymbol{\Theta}}_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k) \\ &= E\left\{ \ln p(\tilde{s}_k, z, \boldsymbol{\theta}_k|\boldsymbol{\pi}, \widetilde{\boldsymbol{\Theta}}_k) \,\middle|\, \tilde{s}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k \right\} \\ &= \sum_{l=1}^{S} \Big\{ \upsilon_l \ln \pi_l + (1-\upsilon_l) \ln(1-\pi_l) + \upsilon_l \ln p_l(\tilde{s}_{l,k}|\tilde{\boldsymbol{\theta}}_{l,k}) \\ &\qquad + (1-\upsilon_l) E\big[\ln p_l(\tilde{s}_{l,k}|\boldsymbol{\theta}_k)|\tilde{s}_k, \boldsymbol{\pi}', \tilde{\boldsymbol{\theta}}'_{l,k} \big] \Big\} + B, \end{aligned} \quad (18)$$

where B is a term that is independent of $\boldsymbol{\pi}$ and $\widetilde{\boldsymbol{\Theta}}_k$. The EM algorithm updates the parameter estimates to new values $\boldsymbol{\pi}$ and $\widetilde{\boldsymbol{\Theta}}_k$ that maximize $Q(\boldsymbol{\pi}, \widetilde{\boldsymbol{\Theta}}_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k)$ in (18). By using the SAGE algorithm, $Q(\boldsymbol{\pi}, \widetilde{\boldsymbol{\Theta}}_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k)$ is first maximized with respect to $\boldsymbol{\pi}$ with $\widetilde{\boldsymbol{\Theta}}_k$ fixed at $\widetilde{\boldsymbol{\Theta}}'_k$. The new value of $\boldsymbol{\pi}$ should satisfy

$$\nabla_{\boldsymbol{\pi}} Q(\boldsymbol{\pi}, \widetilde{\boldsymbol{\Theta}}'_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k) = \mathbf{0}. \quad (19)$$

Substituting (18) into (19), it becomes

$$\upsilon_l \frac{1}{\pi_l} - (1-\upsilon_l)\frac{1}{1-\pi_l} = 0, \quad l = 1, \ldots, S, \quad (20)$$

which yields

$$\pi_l = \upsilon_l. \quad (21)$$

Then, $Q(\boldsymbol{\pi}, \widetilde{\boldsymbol{\Theta}}_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k)$ is maximized with respect to $\widetilde{\boldsymbol{\Theta}}_k$ with $\boldsymbol{\pi}$ fixed at $\boldsymbol{\pi}'$. Based on Assumption 4, the new estimate of $\widetilde{\boldsymbol{\Theta}}_k$ can be obtained by solving

$$\widetilde{\boldsymbol{\Theta}}_k = \arg\max Q(\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}'_k) \quad (22)$$

$$\text{s.t.:} \quad \left\| g_l(\tilde{\boldsymbol{\theta}}_{l,k}) - E\{g_l(\boldsymbol{\theta}_k)\} \right\| \geq \epsilon_l, \quad \forall l, k. \quad (23)$$

The objective function and the constraints are not always convex since we consider very general models. But for many widely accepted models [4] the objective function and the constraints are convex and the optimization can be solved by many algorithms such as the interior point method. In some nonconvex cases, the objective function may be locally convex and a good starting point might allow a similar approach.

While (23) identifies single step attacks, it is also prudent to identify long term attacks, which operate over a time window with well planned attacks which might involve small attacks at each time step which can add up over time. Detecting these attacks can be accomplished by summing the contributions $g_l(\tilde{\boldsymbol{\theta}}_{l,k}) - E\{g_l(\boldsymbol{\theta}_k)\}$ over a series of steps and then flagging any significant change in a given vector direction as an attack, in a way similar to (23), as per

$$\left\| \sum_{k_0=k-L}^{k} g_l(\tilde{\boldsymbol{\theta}}_{l,k_0}) - E\{g_l(\boldsymbol{\theta}_{k_0})\} \right\| \geq \epsilon_l, \quad \forall l \quad (24)$$

[2]See Appendix A for these calculations.

where L is the length of time window. Thus if desired we can flag sensors as satisfying either (23) or (24).

Let $\widetilde{\boldsymbol{s}}_{k,U}$ denote the vector of sensor observations corresponding to those sensors which the algorithm believes are unattacked. Thus, $\pi_l = 0$ for any index l for a sensor in this set. After being passed the soft decisions from the terminated EM run corresponding to discrete time k, the object tracker generates the next state prediction (track of objects) as

$$\widehat{\boldsymbol{\theta}}_k = E(\boldsymbol{\theta}_k|\widetilde{\boldsymbol{s}}_{k,U}), \tag{25}$$

which allows a recursive computation. In fact, under no attacks (25) simplifies to what would normally be considered the most popular standard tracking approach, the minimum mean square error estimate (MMSE). We call the algorithm proposed in this section (see Fig. 4) *Tracking with Soft Decision Attack Detection* (TSDAT).

V. Bound on Optimum After-Attack Tracking Performance

Object tracking performance can be described using a mean-square-error (MSE) matrix which categorizes the errors in estimated state vector (including the locations and velocities of all tracked objects). This paper employs the Bayesian Cramer Rao Bound (BCRB), an asymptotically achievable lower bound on the error correlation matrix. The BCRB using only the unattacked data provides a lower bound for the tracking performance. This follows directly since it is easy to show that the attacked data can never be used to improve the tracking performance for the given attack model. Let S_u denote the number of unattacked sensors. Let $\mathcal{U} = \{l_{u,1}, \ldots, l_{u,S_u}\}$ denote the set of unattacked sensors where $l_{u,m}$ denotes the index of the m-th unattacked sensor. Let $\boldsymbol{s}_{u,k} = \left[\widetilde{s}_{l_{u,1},k} \ \ldots \ \widetilde{s}_{l_{u,S_u},k}\right]$ denote the vector consisting of the unattacked sensor data. Using the notations previously defined in Table I and Assumption 2, the joint pdf of $\boldsymbol{s}_{u,k}$ and $\boldsymbol{\theta}_k$ is given by

$$p(\boldsymbol{s}_{u,k}, \boldsymbol{\theta}_k) = \prod_{m=1}^{S_u} \left\{ p_{l_{u,m}}(\widetilde{s}_{l_{u,m},k}|\boldsymbol{\theta}_k) \right\} p(\boldsymbol{\theta}_k). \tag{26}$$

To obtain the BCRB when the attacked sensors are *correctly* identified, we first formulate the Bayesian Fisher information matrix (BFIM) for tracking $\boldsymbol{\theta}_k$, given by [37]

$$\boldsymbol{J}_\theta \triangleq -E\left\{ \frac{\partial^2 \ln p(\boldsymbol{s}_{u,k}, \boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_k \partial \boldsymbol{\theta}_k^T} \right\} \tag{27}$$

where the expectation is taken with respect to $\boldsymbol{s}_{u,k}$ and $\boldsymbol{\theta}_k$. Then, the BCRB matrix is defined as the inverse of the BFIM $\boldsymbol{J}_\theta$, such that

$$\boldsymbol{C}_\theta = \boldsymbol{J}_\theta^{-1}, \tag{28}$$

where $\boldsymbol{J}_\theta$ is defined in (27). Tracking performance is characterized by the MSE matrix, given by

$$\boldsymbol{R} \triangleq E\left\{ \left(\widehat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k\right)\left(\widehat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k\right)^T \right\}, \tag{29}$$

which is bounded from below by the BCRB matrix as [37]

$$\boldsymbol{R} - \boldsymbol{C}_\theta \succeq \boldsymbol{0}, \tag{30}$$

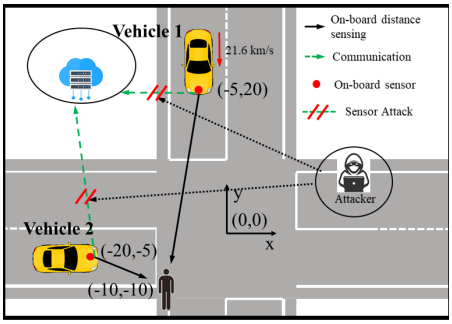


Fig. 10. An example SECVN under a sensor attack.

where the inequality is taken in the semidefinite positive sense. For brevity, we call the bound defined in (28) *the Best Achievable Tracking Performance (BATP).*

Upon calculating the BATP in a given case (sensor observation model and motion model), one often finds the BATP is insensitive to certain system parameters. This is especially interesting since our tracking approach is later shown to achieve after-attack performance close to this bound in the cases investigated. Thus, it is not surprising that the performance of our tracking approach is also insensitive to these same parameters in these same cases. This insensitivity, illustrated in our numerical results, is not very apparent prior to calculating the bound, which seems a valuable contribution of this bound.

VI. Numerical Results and Analysis

Numerical results are presented in this section to illustrate the tracking performance of the TSDAT proposed in Section IV. This includes comparisons to the bound proposed in Section V. In order to statistically characterize the performance of the TSDAT, we perform Monte Carlo simulations with different scenario settings.

Consider the SECVN with 2 cooperating vehicles and 1 pedestrian as shown in Fig. 10. The two vehicles are initially located at $(-5, 20)m$ and $(-20, -5)m$ and have a common velocity $(10, 0)m/s$. One stationary pedestrian, the only non-cooperating object, is located at $(-10, -10)m$. The sensor measurements are sampled every $T_s = 0.1s$. The sensor measurements and motion follow the models defined in (12) and (14), respectively. We assume that all objects for the considered SECVN are within the sensing range defined in (8) and (9) for simplicity. Based on the definition of $\boldsymbol{s}_k$ in (12) and our considered vehicle sensor network with $N_v = 2$ and $N_p = 1$, the dimension of $\boldsymbol{s}_k$ is $S = 2N_v(2 + N_v + N_p) = 20$. To illustrate how duplicate sensors with independent errors improve performance, we assume each vehicle has q duplicate versions of all the sensors employed in (12). This yields a sensor measurement vector $\boldsymbol{s}_k$ with a dimension of $20q$.

The error vectors $\boldsymbol{r}_k$ and $\boldsymbol{n}_k$ in (12) and (14) follow

$$\boldsymbol{r}_k \sim \mathcal{N}(0, \sigma_d^2 \boldsymbol{I}) \quad \text{and} \quad \boldsymbol{n}_k \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}). \tag{31}$$

Each vector $\boldsymbol{n}_k$ corresponding to each set of duplicate sensors is independent of the others. Unless otherwise stated,

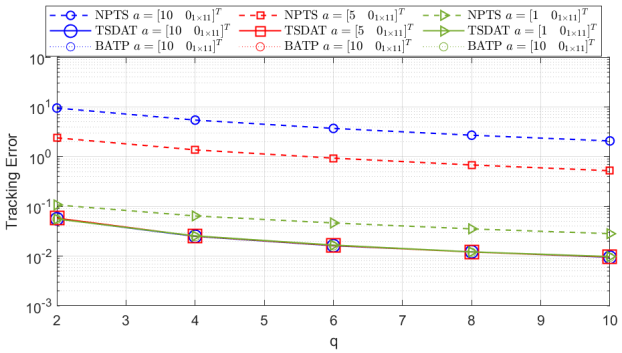


Fig. 11. Tracking error versus q of TSDAT and NPTS for 3 different values of $\boldsymbol{a}$.

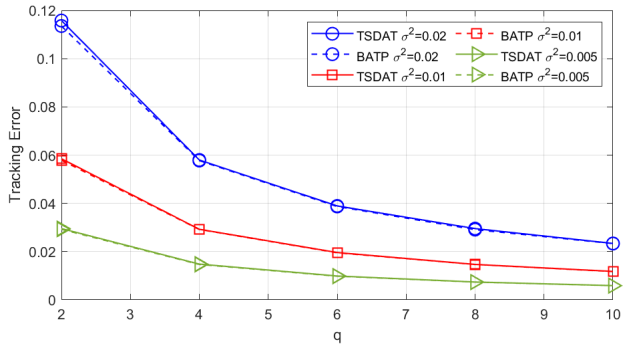


Fig. 12. Tracking error versus q of the TSDAT compared against the BATP for 3 different values of σ^2.

TABLE II
THE PR DEFINED IN (32)

$\boldsymbol{a}_{l,k}$	$q=2$	$q=4$	$q=6$	$q=8$	$q=10$
$[10\ \ \mathbf{0}_{1\times11}]^T$	1.13%	0.95%	0.55%	0.24%	0.20%
$[5\ \ \mathbf{0}_{1\times11}]^T$	4.56%	2.93%	1.95%	1.50%	1.28%
$[1\ \ \mathbf{0}_{1\times11}]^T$	7.36%	4.34%	3.07%	2.06%	1.48%

we set[3] $\sigma_d^2=1$ and $\sigma^2=0.01$. As previously stated after (12), $g_l(\boldsymbol{\theta}_k)=\boldsymbol{h}_l^T\boldsymbol{\theta}_k$. Then, $\sigma_{S,l}^2=\boldsymbol{h}_l^T\boldsymbol{h}_l\sigma_d^2$. We employ Monte-Carlo simulations using 10^4 runs. We define the tracking error as the trace of the MSE matrix defined in (29).

To illustrate the protection TSDAT can provide, we first compare the performance of TSDAT with that of a system which uses the attacked data as if it was not attacked which we call *a non-protected tracking system* (NPTS). Consider an attack which sets $\boldsymbol{a}_{l,k}=\boldsymbol{a}$ in (2) for all l representing the GPS sensors on vehicle 1 with either $\boldsymbol{a}=[10\ \ \mathbf{0}_{1\times11}]^T$ or $\boldsymbol{a}=[5\ \ \mathbf{0}_{1\times11}]^T$ or $\boldsymbol{a}=[1\ \ \mathbf{0}_{1\times11}]^T$ which implies all GPS sensors on vehicle 1 are attacked in the same way. We set $\boldsymbol{a}_{l,k}=\mathbf{0}$ for all other sensors so they are not attacked. Fig. 11 illustrates the BATP and NPTS comparison under these attacks as a function of q the number of duplicated sensors. Fig. 11 shows that for all cases shown, TSDAT dramatically decreases the tracking error when compared to that for NPTS. As shown in Fig. 11, the performance of TSDAT is very close to BATP. To further quantify how close the performance is, define the percentage error as

$$\text{PR}=|\text{Tr}(\boldsymbol{R})-\text{Tr}(\boldsymbol{C}_\theta)|\div\text{Tr}(\boldsymbol{C}_\theta)\times100\%, \tag{32}$$

Table II shows the percentage error is smaller than 6.31% for all cases considered. Fig. 11 and Table II consider cases where all the GPS sensors on vehicle 1 are attacked similarly. If some GPS sensors on vehicle 1 are not attacked, better performance is obtained as expected. On the other hand, the inter-vehicle distances measured by the non-GPS sensors in (8) also provide information concerning the position of vehicle 1.

For this same attack, Fig 12 shows the tracking error of both TSDAT and BATP as a function of q when the sensor noise power σ^2 from (31) is set to each of the values $\{0.005, 0.01, 0.02\}$. For all values of σ^2 considered, the tracking error of TSDAT is close to the BATP defined in (28). For the same attack, Fig. 13 shows the tracking error of TSDAT compared to the BATP for a larger range of q when $\sigma^2=0.01$. Fig. 13 shows increasing q will dramatically decrease the tracking error for the given attack. It indicates that we can achieve any desired tracking performance by simply adding more sensors in this case, even though some of the added sensors are also attacked.

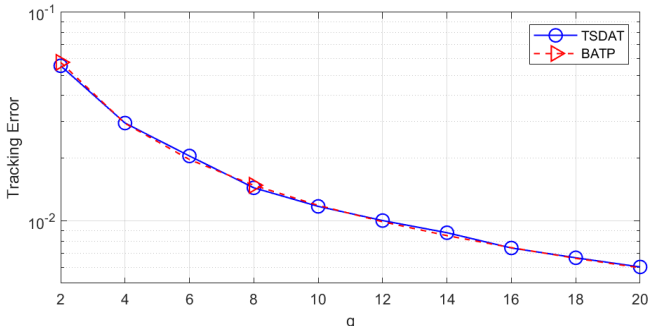


Fig. 13. Tracking error versus q of the TSDAT compared against the BATP.

Fig. 14 illustrates the tracking error of both TSDAT and BATP as a function of q when half of the duplicate vehicle 1 GPS sensors (see (5)), half of the vehicle 1 velocity sensors (see (6)), and half of the vehicle 1 inter-vehicle distance sensors (see (8)) are attacked. For the attacked GPS sensors and the attacked inter-vehicle distance sensors, the attack parameter $\boldsymbol{a}_{l,k}$ in (2) is set to $[10\ \ \mathbf{0}_{1\times11}]^T$. For the attacked velocity sensors, the attack parameter $\boldsymbol{a}_{l,k}$ is set to $[\mathbf{0}_{1\times3}\ \ 10\ \ \mathbf{0}_{1\times8}]^T$. While these attacks are on a larger number of sensors, Fig. 14 shows that the tracking error is still reasonably close to the best possible performance predicted by the BATP bound. For all cases studied, the percentage error is smaller than 3.10%.

We have shown that the TSDAT is close to the BATP for many considered cases. Next, we show that the performance seems almost invariant to the previous position of the pedestrian at time $k-1$ in the process of estimating the state at time k. Consider an attack which sets $\boldsymbol{a}_{l,k}=[10\ \ \mathbf{0}_{1\times11}]^T$ in (2) for all l representing the GPS sensors on vehicle 1. We set $q=2$. At time k, the pedestrian position at time $k-1$ is set to (d,d)m where $d\in\{-60,-40,-20,20,40,60\}$. Table III shows the tracking errors of the TSDAT versus d, showing that the tracking error of the TSDAT seems almost

[3]We choose $\sigma_d^2=1$ as in [4] and consider $\sigma^2=0.01$ to represent accurate sensors that are typically applied in practice. Similar results are obtained in other cases.

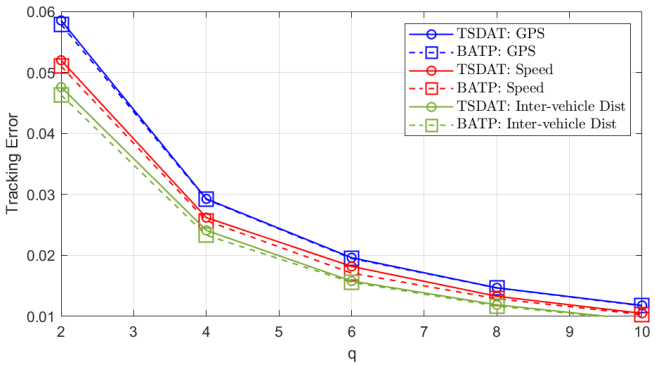


Fig. 14. Tracking error versus q of the TSDAT compared against the BATP when different components are attacked.

TABLE III

TRACKING ERRORS VERSUS d OF TSDAT WHEN THE INITIAL POSITION OF THE PEDESTRIAN IS SET TO (d, d)

d	-60	-40	-20	20	40	60
TSDAT	0.0583	0.0582	0.0583	0.0582	0.0582	0.0581
BATP	0.0576	0.0576	0.0576	0.0576	0.0576	0.0576
PR	1.15%	1.01%	1.23%	1.08%	0.99%	0.89%

invariant to d. The approximate invariance of the results in Table III with d seem to be explained by the invariance of the BATP with respect to d. The BATP for this problem is given by

$$\boldsymbol{C}_\theta = \left[\frac{1}{\sigma^2}\sum_{m=1}^{S_u} \boldsymbol{h}_{l_{u,m}}^T \boldsymbol{h}_{l_{u,m}} + \frac{1}{\sigma_d^2}\boldsymbol{A}^T\boldsymbol{A}\right]^{-1}, \tag{33}$$

where $l_{u,m}$ was defined as the index of the unattacked sensors, $\boldsymbol{A}$ was defined in (13) and $\boldsymbol{h}_{l_{u,m}}$ denotes the $l_{u,m}$-th row vector of $\boldsymbol{H}$ defined in (12). Note that this same approximate invariance should be observed for all quantities for which the BATP does not depend on whenever the TSDAT gives performance close to the BATP. We have tested many practical cases with different number of vehicles and pedestrians under varying attacks. In all those cases, the performance of the TSDAT is very close to the BATP. The cases we show here appear to be representative. On the other hand, we can't test every possible case numerically, so we provide insight in Section VII which provides conditions under which we can guarantee good performance.

Next, we show that the performance of TSDAT improves as either the prior variance σ_d^2 or the sensor variance σ^2 is decreased. We set $q = 2$. Consider an attack which sets $\boldsymbol{a}_{l,k} = [5 \ \ \boldsymbol{0}_{1\times 11}]^T$ in (2) for all l representing the GPS sensors on vehicle 1. Table IV shows the tracking errors of TSDAT and BATP versus the prior variance σ_d^2 while the sensor variance is fixed at $\sigma^2 = 0.01$. Table V shows the tracking errors of TSDAT and BATP versus the sensor variance σ^2 while the prior variance is fixed at $\sigma_d^2 = 1$. Table IV and V show if either σ_d^2 and σ^2 is decreased in this case, the tracking error of TSDAT decreases and the performance is closer to the BATP.

TABLE IV

TRACKING ERRORS OF TSDAT AND BATP VERSUS σ_d^2 WHILE $\sigma^2 = 0.01$

σ_d^2	1	0.316	0.1	0.031	0.01
TSDAT	0.0604	0.0595	0.0565	0.0495	0.0373
BATP	0.0576	0.0569	0.0545	0.0487	0.0370
PR	4.90%	4.73%	3.79%	1.75%	1.01%

TABLE V

TRACKING ERRORS OF TSDAT AND BATP VERSUS σ^2 WHILE $\sigma_d^2 = 1$

σ^2	0.01	0.009	0.008	0.007	0.006	0.001
TSDAT	0.0604	0.0541	0.0467	0.0406	0.0348	0.0058
BATP	0.0576	0.0520	0.0462	0.0404	0.0347	0.0058
PR	4.90%	4.10%	1.01%	0.59%	0.37%	0.16%

In Section VII, we analytically demonstrate a generalization of these findings under some conditions.

Next we show the scalability of the proposed methodology. For the linear model defined in (12), the bottleneck of the proposed methodology is evaluating a multivariate Gaussian pdf and computing the expectation of $Q(\boldsymbol{\pi}, \widetilde{\boldsymbol{\Theta}}_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k')$ defined in (18). Recall D denotes the dimension of $\boldsymbol{\theta}_k$. Based[4] on [38], [39], the expectation in $Q(\boldsymbol{\pi}, \widetilde{\boldsymbol{\Theta}}_k|\boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k')$ has time complexity (number of operations) $O(D^{2.376})$. The time complexity for evaluating a multivariate Gaussian pdf is $O(SD)$. In the worst case, the time complexity of the TSDAT is $O[D^{2.376} + SD]$. Hence, the time complexity of the proposed methodology increases linearly as the number of sensors increases. To validate these ideas, we plot actual running time on a PC with a 2.7GHz CPU and a 64GB RAM. Fig. 15 shows that the average running time increases linearly with respect to S. Here we increase S by increasing the number of duplicate sensors. Next, we consider the time complexity in terms of number of the vehicles N_v. Recall that the state contains the 2-dimensional velocities and positions of all vehicles and pedestrians. So,

$$D = 4(N_v + N_p). \tag{34}$$

Using the per vehicle sensor choices as per (12), the number of sensors S is given by

$$S = 2N_v(2 + N_v + N_p). \tag{35}$$

Using (34) and (35) in the previous time complexity formula yields $O[(N_v + N_p)^{2.376} + 8N_v(2 + N_v + N_p)(N_v + N_p)] \in O(N_v^3)$. Using the same computer as in Fig. 15, Fig. 16 shows a cubic increase in the average running time which agrees with the predicted time complexity.

We compare the proposed methodology to the robust control approach in [19] which is shown to be the most efficient among the previously published classical robust control approaches in [20], [21], [40]. According to Algorithm 3 in [19], the matrix calculations (a $DS \times DS$ matrix multiples a $DS \times 1$ vector) in updating the state are the most time consuming steps. At each time k, the time complexity for matrix

[4] [38], [39] employ the same basic computations as we do in different ways.

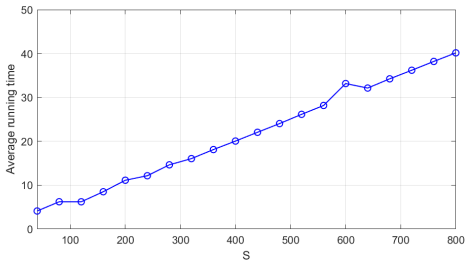


Fig. 15. Average running time versus S which is increased by employing multiple duplicates.

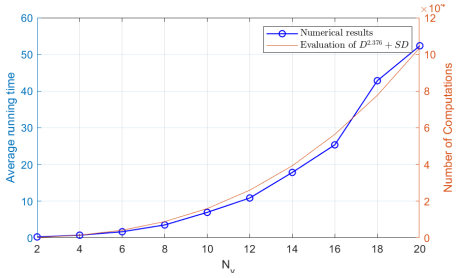


Fig. 16. Average running time versus N_v.

calculations is lower bounded by $O(D^2S^2)$. Hence, the time complexity of the algorithm in [19] is at least $O(D^2S^2)$. Using (34) and (35), we find that the time complexity is lower bounded by $O(N_v^6)$. Recall that the time complexity of the TSDAT is $O[SD]$ in terms of S and $O(N_v^3)$ in terms of N_v. Therefore TSDAT is much more efficient than the robust control approach in [19]. It is worth reminding the reader that TSDAT was shown in the Introduction to provide uniformly better performance than the approach in [19] even while providing these significant complexity reductions that grow with system size.

A. Non-Gaussian Noise Example

Next, we consider cases were a practical heavy-tailed non-Gaussian sensor noise model is employed [41]. Consider the case where $q = 1$ and the first component of the noise vector $\boldsymbol{n}_k$ in (12), which is the noise added to the x-component GPS sensor measurement for vehicle 1, is described by an M-term mixture of Gaussian pdf

$$p(\boldsymbol{n}_{k,1}) = \sum_{i=1}^{M} \left[\lambda_i (2\pi\sigma_i^2)^{-0.5} \exp\left(-0.5\sigma_i^{-2}\boldsymbol{n}_{k,1}^2\right)\right], \quad (36)$$

where $\sum_{i=1}^{M} \lambda_i = 1$. The other components of $\boldsymbol{n}_k$ still follow the previous Gaussian model summarized by $\mathcal{N}(0, \sigma^2)$. All components of $\boldsymbol{n}_k$ are statistically independent. Consider the cases where $M = 2$, $\sigma_1^2 = 0.01$, $\sigma_2^2 = 1$, $\sigma^2 = 0.01$, and $\lambda_1 = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Fig 17 illustrates the heavy-tailed behavior of the pdfs of $\boldsymbol{n}_{k,1}$ for $\lambda_1 = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ as compared to the previous model $\mathcal{N}(0, 0.01)$. For the non-Gaussian cases, the object tracking in (25) is approximated using a particle filter [42], which roughly approximates the posterior density $p(\boldsymbol{\theta}_k|\widetilde{\boldsymbol{s}}_k)$ required in (25) by a discrete-approximation using discrete objects called particles. As the number of particles increases, the track

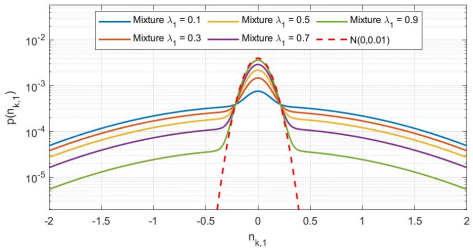


Fig. 17. Pdf of $\boldsymbol{n}_{k,1}$ for $\lambda_1 = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ with $\sigma_1^2 = 0.01$ and $\sigma_2^2 = 1$ compared against $\mathcal{N}(0, 0.01)$.

TABLE VI
TRACKING ERRORS VERSUS λ_1 OF TSDAT COMPARED AGAINST THE BATP

λ_1	0.01	0.1	0.3	0.5	0.7	0.9
TSDAT	0.0987	0.0975	0.0953	0.0967	0.0967	0.0939
BATP	0.0949	0.0944	0.0933	0.0957	0.0943	0.0934
PR	4.01%	3.25%	2.18%	1.08%	2.57%	0.58%

of the state $\hat{\boldsymbol{\theta}}_k$ computed by the particle filter approaches the optimal one defined in (25). Table VI shows the tracking errors of TSDAT and BATP, versus λ_1 when $\boldsymbol{a}_{l,k} = \begin{bmatrix}10 & \boldsymbol{0}_{1\times 11}\end{bmatrix}^T$ for all index l representing GPS sensors. All other sensor data is unattacked. The tracking errors of TSDAT and BATP are very close. It is worth mentioning that particle filtering is also useful for nonlinear motion models.

VII. DEMONSTRATION OF SUITABLE ATTACK DETECTION FOR SUITABLE SENSORS AND PRIOR PDFS

In this section, we will show that we can achieve any desired high level of performance from our algorithm, in terms of correctly detecting attacked and unattacked data, provided we have sufficiently small σ_l^2 and $\sigma_{S,l}^2$ defined in Assumption 1 and 3. Better performance in terms of correctly detecting attacked and unattacked data will directly translate into better tracking performance (closer to the bound) since we optimally process the unattacked data and the attacked data is easily shown to be unneeded. The next paragraph initiates the discussion by interpreting how our algorithm determines the vector $\boldsymbol{\pi}$ which describes the attacked and unattacked data.

From [43], each step of an EM algorithm will improve the estimates to increase the incomplete likelihood. Thus, EM is attempting to maximize the incomplete likelihood. In our case, based on the sensor observations $\widetilde{\boldsymbol{s}}_k$, our algorithm attempts to choose $\boldsymbol{\pi}$ and $\widetilde{\boldsymbol{\Theta}}_k$ to maximize the incomplete likelihood

$$p(\widetilde{\boldsymbol{s}}_k|\boldsymbol{\pi}, \widetilde{\boldsymbol{\Theta}}_k) = \prod_{l=1}^{S}\{\pi_l p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}_{l,k}) + (1-\pi_l)p_l(\widetilde{s}_{l,k})\}, \quad (37)$$

where $\widetilde{\boldsymbol{\theta}}_{l,k}$ is constrained such that $\|g(\widetilde{\boldsymbol{\theta}}_{k,l}) - E(g(\boldsymbol{\theta}_k))\| \geq \epsilon_l$ (see (23)). To maximize (37), it is clear one should select $\pi_l = 0$, indicating the data at sensor l is unattacked with certainty, only if

$$p(\widetilde{s}_{l,k}) > p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}_{l,k}), \quad (38)$$

otherwise one should select $\pi_l = 1$. Let $\mathcal{C}$ and $\overline{\mathcal{C}}$ denote the range of sensor data identified as unattacked and attacked

based on (38), respectively. Let $\widetilde{\boldsymbol{\theta}}^*_{l,k}$ denote the true value of the after-attack state and recall that an attacked sensor follows $p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}^*_{l,k})$. To judge our algorithm's performance in terms of correctly detecting attacked and unattacked data, the following two error probabilities are useful. Since we check each sensor separately, define

$$P_1 = \int_{\widetilde{s}_{l,k}\in\mathcal{C}} p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}^*_{l,k})\, d\,\widetilde{s}_{l,k}, \tag{39}$$

as the probability of identifying an attacked sensor as unattacked. Define

$$P_2 = \int_{\boldsymbol{\theta}_k}\int_{\widetilde{s}_{l,k}\notin\mathcal{C}} p_l(\widetilde{s}_{l,k}|\boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k)d\widetilde{s}_{l,k}d\boldsymbol{\theta}_k, \tag{40}$$

as the probability of identifying an unattacked sensor as attacked. We analyze these two probabilities separately next.

A. P_1 Analysis

To facilitate a useful expression for an upper bound on P_1 (the probability of identifying an attacked sensor as unattacked), the following assumption is useful.

Assumption 5: Let $f(\widetilde{s}_{l,k})$ denotes a unimodal and symmetric pdf with zero mean and unit variance. The pdfs $p(\widetilde{s}_{l,k}|\boldsymbol{\theta}_k)$ and $p(\widetilde{s}_{l,k})$ follow $\frac{1}{\sigma_l} f\left(\frac{\widetilde{s}_{l,k}-g(\boldsymbol{\theta}_k)}{\sigma_l}\right)$ and $\frac{1}{\sqrt{(\sigma_l^2+\sigma_{S,l}^2)}} f\left(\frac{\widetilde{s}_{l,k}-E(g(\boldsymbol{\theta}_k))}{\sqrt{(\sigma_l^2+\sigma_{S,l}^2)}}\right)$ respectively.

Theorem 1: Given Assumption 5, the probability of the TSDAT algorithm identifying an attacked sensor as unattacked P_1 is upper bounded as per

$$P_1 < \frac{\sigma_l^2}{2[-\epsilon_l + E[g(\boldsymbol{\theta}_k)] - g(\widetilde{\boldsymbol{\theta}}^*_{l,k})]^2}. \tag{41}$$

Proof: Based on (37), the TSDAT will choose $\widetilde{\boldsymbol{\theta}}_{l,k}$ to maximize the likelihood $p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}_{l,k})$ constrained by $\|g(\widetilde{\boldsymbol{\theta}}_{k,l}) - E(g(\boldsymbol{\theta}_k))\| \geq \epsilon_l$ for each sensor. If $\widetilde{s}_{l,k} \in [-\infty, E(g(\boldsymbol{\theta}_k)) - \epsilon_l) \cup (E(g(\boldsymbol{\theta}_k)) + \epsilon_l, \infty]$, the $\widehat{\widetilde{\boldsymbol{\theta}}}_{l,k}$ satisfying $g_l(\widehat{\widetilde{\boldsymbol{\theta}}}_{l,k}) = \widetilde{s}_{l,k}$ obeys the constraint and achieves the maximum likelihood since $p(\widetilde{s}_{l,k})$ has a peak at $g_l(\boldsymbol{\theta}_k)$ as per Assumption 5. Since $1/\sigma_l f(0) > p(\widetilde{s}_{l,k})$, all such $\widetilde{s}_{l,k}$ will be identified as attacked. Thus, the set of unattacked data $\mathcal{C}$ from (38) must be contained in $[-\epsilon_l + E(g(\boldsymbol{\theta}_k)), \epsilon_l + E(g(\boldsymbol{\theta}_k))]$. This allows us to bound P_1 as per

$$\begin{aligned} P_1 &< \Pr\{\widetilde{s}_{l,k} \in [-\epsilon_l + E(g(\boldsymbol{\theta}_k)), \epsilon_l + E(g(\boldsymbol{\theta}_k))]\} \\ &< \Pr\{\widetilde{s}_{l,k} > -\epsilon_l + E(g(\boldsymbol{\theta}_k))\} \\ &= \Pr\{\widetilde{s}_{l,k} - g(\widetilde{\boldsymbol{\theta}}^*_{l,k}) > -\epsilon_l + E(g(\boldsymbol{\theta}_k)) - g(\widetilde{\boldsymbol{\theta}}^*_{l,k})\} \\ &= \frac{1}{2}\Pr\{|\widetilde{s}_{l,k} - g(\widetilde{\boldsymbol{\theta}}^*_{l,k})| > -\epsilon_l + E(g(\boldsymbol{\theta}_k)) - g(\widetilde{\boldsymbol{\theta}}^*_{l,k})\}, \end{aligned} \tag{42}$$

where $\widetilde{s}_{l,k}$ follows $p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}^*_{l,k})$. For any random variable X with mean μ_X and variance σ_X^2, Chebyshev's inequality states that

$$\Pr(|X - \mu_X| > k\sigma_X) \leq 1/k^2, \tag{43}$$

for any given scalar k. Using (43) with $X = \widetilde{s}_{l,k}$, $\mu_X = g(\widetilde{\boldsymbol{\theta}}^*_{l,k})$, $\sigma_X = \sigma_l$, and, $k = (-\epsilon_l + E(g(\boldsymbol{\theta}_k)) - g(\widetilde{\boldsymbol{\theta}}^*_{l,k}))/\sigma_l$, (42) implies that P_1 is upper bounded as per (41). □

Remark 1: The major assumption in Assumption 5 is that the sensor pdf has a single peak which is highly desirable in practice and people will use sensors which exactly have this property or that closely approximate this property. Assumption 5 also helps keep the analysis tractable.

B. P_2 Analysis

To facilitate a useful expression for an upper bound on P_2 (the probability that an unattacked sensor is identified as attacked), some assumptions are useful. In practice, high quality sensors are usually employed where $\sigma_l^2 \ll \sigma_{S,l}^2$ (see Assumption 3). We exploit this property in the following assumption.

Assumption 6: The values of $\sigma_{S,l}^2$ and σ_l^2 satisfy

$$10^2 \leq \sigma_{S,l}^2/\sigma_l^2 \leq 10^5. \tag{44}$$

We also formalize our previous suggestion (see discussion near Assumption 3) on the choice of ϵ_l in the following assumption.

Assumption 7: In the sequel, we set $\epsilon_l = \sqrt{\sigma^2 + \sigma_{S,l}^2} + c$ where $c > 0$ is an constant.

Finally, one last Assumption is employed which states that $p_l(\widetilde{s}_{l,k}|\boldsymbol{\theta}_k)$ and $p(\widetilde{s}_{l,k})$ come from a popular class.

Assumption 8: The pdfs $p(\widetilde{s}_{l,k})$ and $p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}^*_{l,k})$ follow a generalized Gaussian distribution (GGD) [44] with the same parameter β as per

$$p(\widetilde{s}_{l,k}) = \frac{\beta\Lambda_\beta}{2\Gamma(1/\beta)\sqrt{\sigma_l^2+\sigma_{S,l}^2}}\left[-\left(\frac{|\widetilde{s}_{l,k}-E(g(\boldsymbol{\theta}_k))|\Lambda_\beta}{\sqrt{\sigma_l^2+\sigma_{S,l}^2}}\right)^\beta\right], \tag{45}$$

and

$$p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}^*_{l,k}) = \frac{\beta\Lambda_\beta}{2\Gamma(1/\beta)\sigma_l}\exp\left[-\left(\frac{|\widetilde{s}_{l,k}-g(\widetilde{\boldsymbol{\theta}}^*_{l,k})|\Lambda_\beta}{\sigma_l}\right)^\beta\right], \tag{46}$$

where $\beta > 0$ denotes the GGD shape parameter, $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}\,dt$ denotes Gamma function, and $\Lambda_\beta = \Gamma^{0.5}(3/\beta)/\Gamma^{0.5}(1/\beta)$.

Similar to the proof of Theorem 1, we obtain an upper bound on P_2 by finding a set containing $\overline{\mathcal{C}}$ in the following lemma.

Lemma 1: Given Assumption 8 and assuming $\sigma_{S,l}^2/\sigma_l^2$ is upper bounded as per,

$$\sigma_{S,l}^2/\sigma_l^2 \leq \exp\left[-2\beta^{-1} + 2^{-\beta+1}\left(\epsilon_l\sigma_l^{-1}\right)^\beta \Lambda_\beta^{\beta/2}\right] - 1, \tag{47}$$

the set $\overline{\mathcal{C}}$ of sensor data deemed to be attacked (defined after (38)) is a subset of $(-\infty, -\epsilon_l/2 + E(g(\boldsymbol{\theta}_k))] \cup [\epsilon_l/2 + E(g(\boldsymbol{\theta}_k)), \infty)$.

Proof: The proof is shown in Appendix B. □

Based on Lemma 1 and Asssumptions 6-8, we describe an upper bound on P_2 as per the following theorem.

Theorem 2: Given Asssumptions 6-8, the probability of identifying an unattacked sensor as attacked P_2 is upper bounded as per

$$P_2 < 1 - \frac{1}{\Gamma(1/\beta)}\gamma_{1/\beta}\left[\left(\frac{\Lambda_\beta}{2} + \frac{c\Lambda_\beta}{2\sqrt{\sigma_l^2+\sigma_{S,l}^2}}\right)^\beta\right]. \tag{48}$$

Proof: Recall that unattacked sensor data $\widetilde{s}_{l,k}$ follows

$$p(\widetilde{s}_{l,k}) = \int_{\boldsymbol{\theta}_k} p(\widetilde{s}_{l,k}|\boldsymbol{\theta}_k) p(\boldsymbol{\theta}_k)\, d\,\boldsymbol{\theta}_k. \tag{49}$$

Using (49) in (40),

$$P_2 = \int_{\widetilde{s}_{l,k} \notin \mathcal{C}} p(\widetilde{s}_{l,k})\, d\,\overline{s}_{l,k}. \tag{50}$$

Based on $\epsilon_l = \sqrt{\sigma_l^2 + \sigma_{S,l}^2} + c$ in Assumption 7 and $10^2 \leq \sigma_{S,l}^2/\sigma_l^2 \leq 10^5$ in Assumption 6, we have $\epsilon_l/\sigma_l > 10$. Substituting $\epsilon_l/\sigma_l \geq 10$ into (47), the upper bound in (47) is greater than 10^5 for all β, which covers the range in Assumption 6. Hence, we can apply Lemma 1 to upper bound P_2 as per

$$\begin{aligned} P_2 &< \int_{-\infty}^{E(g(\boldsymbol{\theta}_k))-\epsilon_l/2} p(\widetilde{s}_{l,k}) d\widetilde{s}_{l,k} + \int_{E(g(\boldsymbol{\theta}_k))+\epsilon_l/2}^{\infty} p(\widetilde{s}_{l,k}) d\widetilde{s}_{l,k} \\ &= 2F_\beta(E(g(\boldsymbol{\theta}_k)) - \epsilon_l/2) \quad \text{(by symmetry)}. \end{aligned} \tag{51}$$

where

$$F_\beta(x) = \tfrac{1}{2} + \tfrac{\text{sign}[x - E(g(\boldsymbol{\theta}_k))]}{2\Gamma(1/\beta)} \gamma_{\frac{1}{\beta}} \left[\tfrac{|x - E(g(\boldsymbol{\theta}_k))|^\beta}{(\sigma_l^2 + \sigma_{S,l}^2)^{\beta/2}} \Lambda_\beta^\beta \right], \tag{52}$$

denotes the cumulative distribution function (cdf) [44] corresponding to the pdf in $p(\widetilde{s}_{l,k})$ in (45). In (52), $\gamma_y[x] = \int_0^x t^{y-1} e^{-t} dt$ denotes the lower incomplete gamma function and sign$[x]$ is the sign of x. Using the definition of $F_\beta(x)$ in (52) with $x = E(g(\boldsymbol{\theta}_k)) - \epsilon_l/2$ in (51), we obtain (48). □

C. Discussion of Assumptions 6-8

- *Discussion of Assumption 6:* Typically $\sigma_l^2 \ll \sigma_{S,l}^2$, which is why sensors are so useful. This can be roughly interpreted as the informativeness of the sensor data is much greater than the information prior to sensing. Motivated by typical settings, the range defined in Assumption 6 exploits these ideas.
- *Discussion of Assumption 7*: Based on our numerical investigations, the suggested value of ϵ_l appears to work well. Additionally, ϵ_l can be selected to optimize some suitable criterion involving both P_1 and P_2 (possibly a linear combination). More detailed discussion on the selection of ϵ_l is provided in Appendix C, along with some analysis.
- *Discussion of Assumption 8*: The GGD class includes a large number of pdfs of interest, including the Gaussian pdf and many continuous pdfs in the exponential family.

D. Discussion of Theorems

- *Summary of Theorem 1:* It is clear that (41) in Theorem 1 indicates that the upper bound of P_1 can be made sufficiently small for sufficiently small σ_l^2.
- *Summary of Theorem 2:* Since the lower incomplete gamma function $\gamma_{1/\beta}(x) = \int_0^x t^{1/\beta - 1} e^{-t} dt$ is strictly increasing towards $1/\Gamma(1/\beta)$, the upper bound on P_2 in (48) is monotonically decreasing towards zero as the sum $\sigma_l^2 + \sigma_{S,l}^2$ decreases to zero. Thus, with sufficiently small σ_l^2 and $\sigma_{S,l}^2$, both P_1 and P_2 can be made sufficiently small and the tracking performance can be made sufficiently close to the BATP.

Based on the just described analysis, we can achieve any desired sufficiently high level of performance from our algorithm, in terms of correctly detecting attacked and unattacked data, provided σ_l^2 and $\sigma_{S,l}^2$ are sufficiently small.

VIII. Conclusion

In this paper, we describe an efficient machine-learning enhanced approach for tracking under cyber-attacks called TSDAT. For appropriate sensor and prior pdfs, the tracking performance is shown to be close to the BATP. The time complexity of our approach is dramatically lower than the best existing published low complexity approach. In particular, the complexity of our approach increases linearly in the number of sensors, while the best low complexity published approach has a complexity which grows quadratically in the number of sensors. We can achieve any desired high level of performance from our algorithm, in terms of correctly detecting attacked and unattacked data, and ultimately tracking performance, provided we have sufficiently small σ_l^2 and $\sigma_{S,l}^2$ (defined in Assumptions 1 and 3).

Appendix A
Computation of υ_l

Recall that $\upsilon_l = E\{\mathbb{1}_{z_l=1}|\widetilde{\boldsymbol{s}}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k'\} = \int_{\boldsymbol{\theta}_k} p(z_l = 1, \boldsymbol{\theta}_k|\widetilde{\boldsymbol{s}}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k') d\boldsymbol{\theta}_k$. Based on $p(A, B) = p(A|B)p(B)$, we have

$$\upsilon_l = \int_{\boldsymbol{\theta}_k} p(z_l = 1|\boldsymbol{\theta}_k, \widetilde{\boldsymbol{s}}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k') p(\boldsymbol{\theta}_k) d\boldsymbol{\theta}_k \tag{53}$$

$$= E\{p(z_l = 1|\boldsymbol{\theta}_k, \widetilde{\boldsymbol{s}}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k')\}. \tag{54}$$

Based on Bayes theorem and the law of total probability, $p(z_l = 1|\boldsymbol{\theta}_k, \widetilde{\boldsymbol{s}}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k')$ in (54) is simplified as

$$p(z_l = 1|\boldsymbol{\theta}_k, \widetilde{\boldsymbol{s}}_k, \boldsymbol{\pi}', \widetilde{\boldsymbol{\Theta}}_k') \tag{55}$$

$$= \frac{p(\widetilde{s}_{l,k}|z_l=1, \pi_l', \widetilde{\boldsymbol{\theta}}_{l,k}') p(z_l=1|\boldsymbol{\theta}_k, \pi_l', \widetilde{\boldsymbol{\theta}}_{l,k}')}{\sum_{i=0}^{1} p(\widetilde{s}_{l,k}|z_l=i, \pi_l', \widetilde{\boldsymbol{\theta}}_{l,k}') p(z_l=i|\boldsymbol{\theta}_k, \pi_l', \widetilde{\boldsymbol{\theta}}_{l,k}')}$$

$$= \frac{\pi_l' p_l\left(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}_{l,k}'\right)}{\pi_l' p_l\left(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}_{l,k}'\right) + (1-\pi_l') p_l\left(\widetilde{s}_{l,k}|\boldsymbol{\theta}_k\right)}. \tag{56}$$

We can use efficient approaches, such as the importance sampling [45], to compute (54).

Appendix B
The Proof of Lemma 1

Recall that $\overline{\mathcal{C}}$ denotes the range of sensor data identified as attacked based on $p(\widetilde{s}_{l,k}) > p(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}_{l,k})$ in (38). The point where the two functions in (38) are equal is called the crossing point. For convenience, consider the case that the mean of $p(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}_{l,k}^*)$, which is $g(\widetilde{\boldsymbol{\theta}}_{l,k}^*)$, is more negative than the mean of $p(\widetilde{s}_{l,k})$, which is $E(g(\boldsymbol{\theta}_k))$. A similar result can be obtained with $g(\widetilde{\boldsymbol{\theta}}_{l,k}^*)$ which is more positive than $E(g(\boldsymbol{\theta}_k))$.

Consider the case where $g(\widetilde{\boldsymbol{\theta}}_{l,k}^*) = E(g(\boldsymbol{\theta}_k)) - \epsilon_l$, which is right on the edge of being identified as attacked from per (23). If $p(\widetilde{s}_{l,k})$ and $p(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}_{l,k}^*)$ have the same pdf (after any non-zero mean is extracted), the two functions cross

exactly midway between the means of the two functions $g(\widetilde{\boldsymbol{\theta}}^*_{l,k})$ and $E(g(\boldsymbol{\theta}_k))$, which is at $\widetilde{s}_{l,k} = E(g(\boldsymbol{\theta}_k)) - \epsilon_l/2$. Since $p(\widetilde{s}_{l,k})$ decreases monotonically as $\widetilde{s}_{l,k}$ becomes more negative than the mean $E(g(\boldsymbol{\theta}_k))$, if

$$p(\widetilde{s}_{l,k}) \geq p(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}^*_{l,k})\big|_{\widetilde{s}_{l,k}=E(g(\boldsymbol{\theta}_k))-\epsilon_l/2}, \tag{57}$$

the crossing point is more negative than the mid point.

For convenience, let

$$y = 0.5\,\epsilon_l\,\Lambda_\beta(\sigma_l^2 + \sigma_{S,l}^2)^{-0.5}. \tag{58}$$

Using (45), (46), (58), $\widetilde{s}_{l,k} = E(g(\boldsymbol{\theta}_k)) - \epsilon_l/2$ and $g(\widetilde{\boldsymbol{\theta}}^*_{l,k}) = E(g(\boldsymbol{\theta}_k)) - \epsilon_l$, the inequality (57) becomes

$$y\exp(-y^\beta) \geq 0.5\epsilon_l\,\sigma_l^{-1}\Lambda_\beta \exp\left[-\left(0.5\epsilon_l\,\sigma_l^{-1}\Lambda_\beta\right)^\beta\right]. \tag{59}$$

To denote the term on the left hand side of (59), let

$$f(y) = y\ \exp(-y^\beta). \tag{60}$$

Take the first and the second derivative of $f(y)$ with respect to y to obtain $f'(y) = \exp(-y^\beta)(1 - \beta y^\beta)$, and $f''(y) = \exp(-y^\beta)y^{\beta-1}(y^\beta - 1 - \beta)$. Solving $f'(y) = 0$ shows the extremum of $f(y)$ occurs at $y^* = \beta^{-1/\beta}$. Since $(y^\beta - 1 - \beta) < 0$ and $\exp(-y^\beta)y^{\beta-1} > 0$ when $y < y^*$, $f''(y) < 0$ so that $f(y)$ is concave when $y < y^*$. By concavity, in $[0, y^*]$, the function $f(y)$ must be above a line connecting $(0, f(0))$ and $(y^*, f(y^*))$ as per

$$f(y) \geq \exp(-1/\beta)y. \tag{61}$$

Thus (61) indicates that any y satisfying

$$\exp(-1/\beta)y \geq \frac{\epsilon_l}{2}\,\sigma_l^{-1}\Lambda_\beta \exp\left[-\left(0.5\epsilon_l\,\sigma_l^{-1}\Lambda_\beta\right)^\beta\right], \tag{62}$$

satisfies (59). Applying the definition of y in (58) in (62), we obtain (47). Therefore, for any $\sigma_{S,l}^2/\sigma_l^2$ satisfying (47), the crossing point is more negative than the mid point when the mean of $p_l(\widetilde{s}_{l,k}|\widetilde{\boldsymbol{\theta}}^*_{l,k})$, which is $g(\widetilde{\boldsymbol{\theta}}^*_{l,k})$, is more negative that the mean of $p(\widetilde{s}_{l,k})$, which is $E(g(\boldsymbol{\theta}_k))$. Therefore, $\overline{\mathcal{C}}$, the range of data identified as attacked, is a subset of $(-\infty, -\epsilon_l/2 + E(g(\boldsymbol{\theta}_k))] \cup [\epsilon_l/2 + E(g(\boldsymbol{\theta}_k)), \infty)$. This completes the proof.

APPENDIX C
DISCUSSION ON THE SELECTION OF ϵ_l

The selection of ϵ_l impacts the performance from our algorithm in terms of correctly detecting attacked and unattacked data. If ϵ_l decreases towards 0, the range of data identified as unattacked shrinks. For fixed sensor and prior pdfs, this increases P_2, the probability of identifying an unattacked sensor as attacked, but decreases P_1, the probability of identifying an attacked sensor as unattacked. Similarly, as ϵ_l increases, P_2 decreases while P_1 increases. Therefore, there is a trade off in selecting ϵ_l to achieve the best performance, which could be expressed as some function of both P_1 and P_2. For simplicity, let ϵ_l^* denote the optimal ϵ_l that minimizes $P_1 + P_2$.

As previously stated after Assumption 4 (and shown analytically in Section VII), a value of ϵ_l close to $(\sigma_l^2 + \sigma_{S,l}^2)^{1/2}$ often provides good performance. To show this can be a good selection, we compute P_1 and P_2 numerically for the generalized Gaussian distribution in Assumption 8 for a scalar sensor. We set the after attack parameter $\widetilde{\theta}_{l,k}$ to $E(\theta_k) + \alpha$ where $\alpha \in (-\infty, -\sigma_{S,l}] \cup [\sigma_{S,l}, \infty)$. The probability P_1 decreases as the absolute value of α increases. So, we only consider $\alpha = \sigma_{S,l}$. We have similar results for $\alpha = -\sigma_{S,l}$. Fig. 18 shows P_1, P_2, and $P_1 + P_2$ versus ϵ_l when $\sigma_{S,l} = 1$, $\sigma = 0.1$, and $\beta = [1, 2, 3]$. Fig. 18 shows that when $\beta = 1, 2$, and 3, $P_1 + P_2$ is minimized for $\epsilon_l = 1.062, 1.048$, and 1.032, respectively, which is very close to the suggested value $\epsilon_l = 1.005$. Limited further numerical investigations provide similar conclusions.

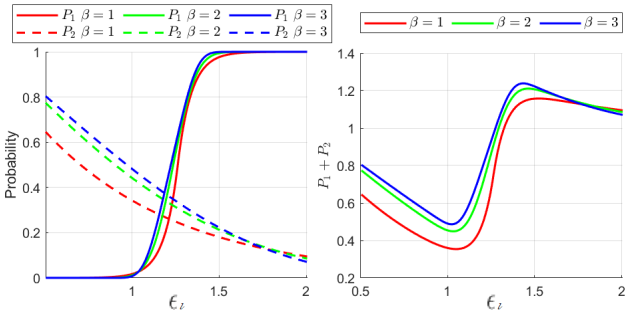


Fig. 18. Error probabilities versus ϵ_l when $\sigma_d = 1$ and $\sigma = 0.1$.

TABLE VII
THE RATIO BETWEEN THE OPTIMAL ϵ_l AND THE SUGGESTED VALUE FOR VARIOUS VALUES OF σ AND $\sigma_{S,l}$ FOR $\beta = 2$

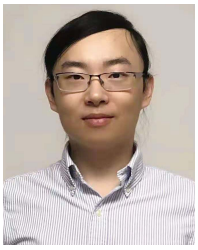
$\sigma_{S,l}$	$\sigma = 0.1$	$\sigma = 0.05$	$\sigma = 0.01$	$\sigma = 0.005$	$\sigma = 0.001$
1	1.1883	1.1087	1.0274	1.0140	1.0035
2	1.1080	1.0602	1.0147	1.0080	1.0020
10	1.0272	1.0146	1.0033	1.0017	1.0003

Next we show that the suggested value $(\sigma_{S,l}^2 + \sigma^2)^{1/2}$ approaches the optimal one as σ^2 decreases. Define $r = \epsilon_l^*/(\sigma_{S,l}^2 + \sigma^2)^{1/2}$ as the ratio between the optimal ϵ_l and the suggested value. Table VII shows the ratio between the optimal ϵ_l and the suggested value for various values of σ, $\sigma_{S,l}$ and $\beta = 2$. Table VII shows that as σ decreases, the ratio r approaches 1 (for a given $\sigma_{S,l}$).

REFERENCES

[1] A. J. Maidamwar and R. Sadakale, "Comprehensive study for localization techniques in MANET and VANET," in *Proc. Int. Conf. Adv. Commun. Comput. Technol. (ICACCT)*, Feb. 2018, pp. 349–352.
[2] F. Qu, Z. Wu, F.-Y. Wang, and W. Cho, "A security and privacy review of VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 2985–2996, Dec. 2015.
[3] G. Soatti, M. Nicoli, N. Garcia, B. Denis, R. Raulefs, and H. Wymeersch, "Implicit cooperative positioning in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3964–3980, Dec. 2018.
[4] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.
[5] H. S. Ramos, A. Boukerche, R. W. Pazzi, A. C. Frery, and A. A. F. Loureiro, "Cooperative target tracking in vehicular sensor networks," *IEEE Wireless Commun.*, vol. 19, no. 5, pp. 66–73, Oct. 2012.
[6] A. Boukerche, H. A. B. FOliveira, E. F. Nakamura, and A. A. F. Loureiro, "Vehicular ad hoc networks: A new challenge for localization-based systems," *Comput. Commun.*, vol. 31, no. 12, pp. 2838–2849, 2008.
[7] J. J. Blum, A. Eskandarian, and L. J. Hoffman, "Challenges of intervehicle ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 347–351, Dec. 2004.
[8] M. S. Al-kahtani, "Survey on security attacks in vehicular ad hoc networks (VANETs)," in *Proc. 6th Int. Conf. Signal Process. Commun. Syst.*, Dec. 2012, pp. 1–9.

[9] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 546–556, Apr. 2015.
[10] Y. Yao, B. Xiao, G. Yang, Y. Hu, L. Wang, and X. Zhou, "Power control identification: A novel Sybil attack detection scheme in VANETs using RSSI," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2588–2602, Nov. 2019.
[11] K. M. A. Alheeti, A. Gruebler, and K. D. McDonald-Maier, "An intrusion detection system against malicious attacks on the communication network of driverless cars," in *Proc. 12th Annu. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2015, pp. 916–921.
[12] T. Bouali, H. Sedjelmaci, and S.-M. Senouci, "A distributed prevention scheme from malicious nodes in VANETs' routing protocols," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2016, pp. 1–6.
[13] X. Xue, N. Lin, J. Ding, and Y. Ji, "A trusted neighbor table based location verification for vanet routing," in *Proc. IET ICWMNN*, Sep. 2010, pp. 1–5.
[14] J. Zhang, X. Wang, R. S. Blum, and L. M. Kaplan, "Attack detection in sensor network target localization systems with quantized data," *IEEE Trans. Signal Process.*, vol. 66, no. 8, pp. 2070–2085, Apr. 2018.
[15] J. Zhang, R. S. Blum, X. Lu, and D. Conus, "Asymptotically optimum distributed estimation in the presence of attacks," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1086–1101, Mar. 2015.
[16] B. Alnajjab, J. Zhang, and R. S. Blum, "Attacks on sensor network parameter estimation with quantization: Performance and asymptotically optimum processing," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6659–6672, Dec. 2015.
[17] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: A survey," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 2, pp. 52–73, 2nd Quart., 2009.
[18] A.-Y. Lu and G.-H. Yang, "Secure state estimation for cyber-physical systems under sparse sensor attacks via a switched Luenberger observer," *Inf. Sci.*, vol. 417, pp. 454–464, Nov. 2017.
[19] Y. Shoukry *et al.*, "SMT-based observer design for cyber-physical systems under sensor attacks," *ACM Trans. Cyber-Phys. Syst.*, vol. 2, no. 1, pp. 1–27, Feb. 2018.
[20] S. Mishra, Y. Shoukry, N. Karamchandani, S. Diggavi, and P. Tabuada, "Secure state estimation: Optimal guarantees against sensor attacks in the presence of noise," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 2929–2933.
[21] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013.
[22] X.-J. Li and X.-Y. Shen, "A data-driven attack detection approach for DC servo motor systems based on mixed optimization strategy," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 5806–5813, Sep. 2020.
[23] A.-Y. Lu and G.-H. Yang, "Input-to-state stabilizing control for cyber-physical systems with multiple transmission channels under denial of service," *IEEE Trans. Autom. Control*, vol. 63, no. 6, pp. 1813–1820, Jun. 2017.
[24] M. A. Hayes and M. A. Capretz, "Contextual anomaly detection framework for big sensor data," *J. Big Data*, vol. 2, no. 1, pp. 1–22, Dec. 2015.
[25] M. L. Psiaki and T. E. Humphreys, "Protecting GPS from spoofers is critical to the future of navigation," *IEEE Spectr.*, vol. 10, pp. 1–6, Jul. 2016.
[26] R. Quinonez, J. Giraldo, L. Salazar, E. Bauman, A. Cardenas, and Z. Lin, "SAVIOR: Securing autonomous vehicles with robust physical invariants," in *Proc. 29th USENIX Secur. Symp. (USENIX Security)*, 2020, pp. 895–912.
[27] C. Fu, Q. Zeng, and X. Du, "Hawatcher: Semantics-aware anomaly detection for appified smart homes," in *Proc. 30th USENIX Secur. Symp. (USENIX Security)*, 2021, pp. 1–18.
[28] S. Dasgupta, M. Rahman, M. Islam, and M. Chowdhury, "Prediction-based GNSS spoofing attack detection for autonomous vehicles," 2020, *arXiv:2010.11722*. [Online]. Available: https://arxiv.org/abs/2010.11722
[29] P. Kapoor, A. Vora, and K.-D. Kang, "Detecting and mitigating spoofing attack against an automotive radar," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–6.
[30] P. A. Lopez *et al.*, "Microscopic traffic simulation using sumo," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.
[31] J. Bhatti and T. E. Humphreys, "Hostile control of ships via false GPS signals: Demonstration and detection," *J. Inst. Navigat.*, vol. 64, no. 1, pp. 51–66, 2017.
[32] R. M. (2015). *Wondering How to Hack a Military Drone? It's All on Google*. [Online]. Available: https://www.ibtimes.co.uk/wondering-how-hackmilitary-drone-its-all-google-1500326
[33] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22–35, Mar. 2017.
[34] Z. Wang and R. S. Blum, "Cybersecurity of inference in vehicular ad-hoc networks: Invited presentation," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2020, pp. 1–6.
[35] B. P. Carlin and T. A. Louis, *Bayesian Methods for Data Analysis*. Boca Raton, FL, USA: CRC Press, 2008.
[36] J. A. Fessler and A. O. Hero, "Space-alternating generalized expectation-maximization algorithm," *IEEE Trans. Signal Process.*, vol. 42, no. 10, pp. 2664–2677, Oct. 1994.
[37] H. L. Van Trees and K. L. Bell, "Bayesian bounds for parameter estimation and nonlinear filtering/tracking," *AMC*, vol. 10, p. 12, Apr. 2007.
[38] L. S.-Y. Wu, J. S. Pai, and J. R. M. Hosking, "An algorithm for estimating parameters of state-space models," *Statist. Probab. Lett.*, vol. 28, no. 2, pp. 99–106, Jun. 1996.
[39] C. Montella, "The Kalman filter and related algorithms: A literature review," May 2011, pp. 1–17. [Online]. Available: https://www.researchgate.net/publication/236897001_The_Kalman_Filter_and_Related_Algorithms_A_Literature_Review
[40] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1454–1467, Jun. 2014.
[41] D. Middleton and A. Spaulding, "Elements of weak signal detection in non-Gaussian noise," *Adv. Stat. Signal Process.*, vol. 2, pp. 137–215, Jun. 1993.
[42] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[43] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
[44] S. A. Kassam, *Signal Detection in Non-Gaussian Noise*. Berlin, Germany: Springer, 2012.
[45] C. Robert and G. Casella, *Monte Carlo Statistical Methods*. Berlin, Germany: Springer, 2013.



Zisheng Wang received the B.S. and M.S. degrees in electronic information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2015 and 2018, respectively, and the Ph.D. degree in electrical engineering from Lehigh University, Bethlehem, PA, USA, in 2021. His research interests include signal detection and estimation theory, cybersecurity, wireless sensor networks, vehicular networks, and design of radio detecting systems.

Rick S. Blum (Fellow, IEEE) received the B.S. degree in electrical engineering from Pennsylvania State University in 1984 and the M.S. and Ph.D. degrees in electrical engineering from the University of Pennsylvania in 1987 and 1991, respectively. He graduated from GE's Advanced Course in engineering.

From 1984 to 1991, he was a member of the Technical Staff, General Electric Aerospace, Valley Forge, PA, USA. Since 1991, he has been with the Electrical and Computer Engineering Department, Lehigh University, Bethlehem, PA, USA, where he is currently a Professor and holds the Robert W. Wieseman Chaired Research Professorship in electrical engineering. He holds several patents. His research interests include signal processing for security, smart grid, communications, sensor networking, and radar and sensor processing.

Dr. Blum was a member of the SAM Technical Committee (TC) of the IEEE Signal Processing Society. He was a member of the Signal Processing for Communications TC of the IEEE Signal Processing Society. He is a member of the Communications Theory TC of the IEEE Communication Society. He was on the Awards Committee of the IEEE Communication Society. He is a member of Eta Kappa Nu and Sigma Xi. He was an IEEE Third Millennium Medal Winner. He was awarded the ONR Young Investigator Award in 1997 and the NSF Research Initiation Award in 1992. His IEEE Fellow Citation "for scientific contributions to detection, data fusion and signal processing with multiple sensors" acknowledges contributions to the field of sense. He was on the Editorial Board for the *Journal of Advances in Information Fusion* of the International Society of Information Fusion. He was an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING and IEEE COMMUNICATIONS LETTERS. He has edited special issues for IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. In his second term, he is an IEEE Signal Processing Society Distinguished Lecturer.