Malicious User Detection for Cooperative Mobility Tracking in Autonomous Driving

Wang Pi, Pengtao Yang, *Student Member, IEEE*, Dongliang Duan, Chen Chen, *Senior Member, IEEE*, Xiang Cheng, *Senior Member, IEEE*, Liuqing Yang, *Fellow, IEEE*, and Hang Li

***Abstract*—The mobility status of self and surrounding vehicles provides important information to various tasks in autonomous driving (AD) and intelligent transportation system (ITS). Accordingly, a precise, stable, and robust mobility tracking framework is essential. Compared with self-tracking that relies only on mobility observations from onboard sensors [e.g., global positioning system (GPS), inertial measurement unit (IMU), and camera], cooperative tracking markedly increases the precision and reliability of the mobility information by integrating observations from roadside units (RSUs) and nearby vehicles through vehicle-to-everything (V2X) communications in the Internet of Vehicles (IoV). Nevertheless, cooperative tracking can be quite vulnerable if there are malicious users sending bogus observations in the cooperative network. In this article, we present a malicious user detection framework, which includes two sequential detection algorithms and a secure mobility data exchange and fusion model to detect and remove bogus mobility information and integrate proposed detection algorithms with previous data fusion algorithms, which secures the cooperative mobility tracking in AD, ITS. Simulations validate the effectiveness and robustness of the proposed framework under different types of attacks.**

***Index Terms*—Autonomous driving (AD), cooperative mobility tracking, intelligent transportation system (ITS), Internet of Vehicles (IoV), malicious user detection, sequential detection.**

I. Introduction

AUTONOMOUS driving (AD) and intelligent transportation system (ITS) are expected to greatly improve the efficiency of the transportation system and reduce fatal accidents in the near future. Extensive research efforts regarding fundamental issues of AD and ITS have been carried out in the past decade, among which obtaining precise mobility information, such as the location, velocity, and acceleration of the self and surrounding vehicles is one of the most essential.

In practice, the prevailing global positioning system (GPS)-based tracking technique cannot provide the decimeter-level precision required by AD and ITS, especially under circumstances where the GPS signal is weak or even completely absent. Researchers proposed various methods to augment the precision of mobility tracking. Many of them fall into the category of single-vehicle multisensor independent tracking, which relies on multiple high-precision onboard sensors, such as GPS, inertial measurement unit (IMU), lighting detection and ranging (LIDAR), and simultaneous localization and mapping (SLAM), to achieve a precise independent mobility tracking. In [1], the state-of-the-art independent localization techniques were surveyed. As stated therein, though fusing data from onboard sensors could potentially achieve the required accuracy for AD and ITS, the cost of a single vehicle equipped with all these sensors may be too high. In addition, the performance may be compromised under extreme conditions. Thus, multivehicle–multisensor cooperative localization and tracking methods (see [2]–[9]) have been proposed to exploit the information shared by surrounding cooperative vehicles via vehicle-to-everything (V2X) communications (see [10]–[14]) in the Internet of Vehicles (IoV) to improve the localization and tracking performance. However, the security challenges of the cooperative tracking system are not considered in these works.

Similar to other Internet-of-Things (IoT) systems (see [15] and [16]), V2X-based cooperative mobility tracking systems also face security threats at three levels in practical implementation: 1) the perception level; 2) the transportation level; and 3) the application level. In terms of the threats at the perception and transportation levels, the current cooperative tracking framework can directly integrate physical layer protections, such as those summarized in [17] and [18], to prevent jamming or spoofing on sensors, and the communication authentication protocols presented in [19] and [20] to prevent unauthenticated malicious user from entering the cooperative network or conducting Denial-of-Service (DoS) and Sybil attacks.

Nevertheless, there is still a chance that malicious users obtain valid identities in an unexpected way, and then enter

Manuscript received September 9, 2019; revised January 22, 2020; accepted January 31, 2020. Date of publication February 13, 2020; date of current version June 12, 2020. This work was supported in part by the Ministry National Key Research and Development Project under Grant 2017YFE0121400, in part by the Guangdong Key Research and Development Project under Grant 2019B010153003, in part by the National Science Foundation under Grant CNS-1932413 and Grant CNS-1932139, and in part by the Open Research Fund from Shenzhen Research Institute of Big Data under Grant 2019ORF01006. *(Corresponding author: Xiang Cheng.)*

Wang Pi, Pengtao Yang, Chen Chen, and Xiang Cheng are with the State Key Laboratory of Advanced Optical Communication Systems and Networks, Department of Electronics, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: piwang@pku.edu.cn; ypt@pku.edu.cn; c.chen@pku.edu.cn; xiangcheng@pku.edu.cn).

Dongliang Duan is with the Department of Electrical and Computer Engineering, University of Wyoming, Laramie WY 82071 USA (e-mail: dduan@uwyo.edu).

Liuqing Yang is with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523 USA (e-mail: lqyang@engr.colostate.edu).

Hang Li is with the Data-Driven Intelligent Information System Laboratory, Shenzhen Research Institute of Big Data, Shenzhen 518172, China (e-mail: hangdavidli@163.com).

Digital Object Identifier 10.1109/JIOT.2020.2973661

the cooperative tracking network. They may choose not to conduct transportation level attack or jam others' sensors, in order to avoid being detected in perception and transportation level. Instead, they may send bogus mobility information to directly interfere the data fusion algorithm in cooperative tracking to compromise the tracking reliability and precision. Therefore, it is desirable to add an additional data-driven defense against bogus information sent by malicious users so that the cooperative tracking algorithm itself could remain reliable and robust against application-level bogus information attack.

Data-driven filtering and detection have been proposed as the application-level protection against bogus information in IoT, wireless sensor networks (WSN), and vehicular *ad hoc* networks (VANETs). Reputation management-based detection [21], [22] is proposed to manage the reputation score of each user and mark users whose scores are lower than the preset threshold as malicious. However, a reputation update in the framework is based on the abstract feedback values generated by involved users after the information exchange process. This means that the reputation management-based detection framework cannot be directly utilized for malicious user detection in cooperative tracking. Instead, they need to be built on some data processing algorithms that can extract crucial information from the dense high-dimensional raw mobility data and generate simple low-dimensional feedback.

Some data processing algorithms have been proposed in [23]–[26] to address a similar issue in WSN and VANETs, i.e., how to remove bogus information to secure the cooperative localization. Nevertheless, those algorithms are proposed for static localization problem, and hence only utilize single snapshot information to detect and identify malicious users. This means that for the dynamic cooperative tracking, they cannot exploit the temporal correlation of the mobility information sequence to improve the detection performance. In this regard, we propose two sequential malicious user detection algorithms to fully utilize the temporal correlation in mobility information sequence to improve the detection performance and provide cooperative mobility tracking with much better reliability and stronger robustness.

Our main contributions are summarized as follows.

1) We propose two sequential malicious user detection algorithms, namely, the dynamic model-based mean state detection (DMMSD) and mean residual error detection (MRED), to identify malicious users more precisely by exploiting the temporal correlation of mobility information sequence. To the best of our knowledge, this is the first time sequential malicious user detection algorithms are proposed in cooperative mobility tracking.
2) We propose a secure mobility data exchange and fusion model which can integrate the proposed malicious user detection algorithms with our previous cooperative tracking fusion algorithms [7].
3) We present an extended threat model which considers both temporal and spatial distribution of bogus information as attack parameters. As compared with existing models used in WSN and VANETs that only consider the spatial distribution, the extended model can characterize possible threats in dynamic cooperative tracking scenarios more comprehensively. Then, we evaluate the performance of our proposed algorithms and existing algorithms with different attack patterns.

The remainder of this article is organized as follows. Related work is reviewed in Section II. The system model, including the system state transfer model, the observation model, the secure data exchange and fusion model, and the extended threat model, is introduced in Section III. The detailed DMMSD and MRED algorithms are presented in Section IV. The performance of the proposed detection framework is evaluated in Section V. Finally, concluding remarks and future work are presented in Section VI.

II. Related Work

Most related work in the literature falls into three categories: 1) reputation management-based detection; 2) mobility data verification; and 3) secure localization.

A. Reputation Management-Based Detection

Reputation management-based malicious user detection is used in information/resources sharing in IoT or IoV (see [21] and [22]). In the reputation management framework, all users in the network are initially assigned with the same reputation score. Then, after each information exchange is completed, feedbacks are generated and sent to each other to update the reputation score of both vehicles. Sending bogus information will result in negative feedback and a reduction in the reputation score. Once the score of a vehicle is below a preset threshold, it is declared as malicious. Though reputation management-based detection is a good approach to detect malicious user by utilizing feedback to update reputation dynamically, it does not address how the feedback is generated from the raw information. For some systems, it might be straightforward, e.g., generating feedback based on the resolution of shared videos. However, it is not a trivial task in cooperative tracking considering the high dimensionality of the mobility information set when dozens of cooperative vehicles are involved. Furthermore, mobility information obtained by sensors of each cooperative vehicle is always noisy and not fully trustworthy. Therefore, there will not be a precise and stable reference to rely on during data processing, which makes the feedback generating even more challenging.

Therefore, for the malicious user detection in cooperative tracking, the reputation management-based detection serves more like a high-level framework which builds on the result of content-based processing algorithms (e.g., those summarized in Section II-C and the algorithms proposed in this article) and cannot be directly utilized to cope with the bogus mobility information in cooperative tracking.

B. Mobility Data Verification

Another category is the mobility data verification (see [27]–[30]) which seems to be similar to the problem considered in this article, while there are some major differences in the main objective and the trust assumption of them. The main task of mobility data verification is using its own

observations as the reference to verify whether other vehicles' self-claimed mobility information are honest. While the main objective of malicious user detection in cooperative tracking, in fact, is to identify and remove bogus information and corresponding users to maintain the reliability and precision of cooperative mobility tracking. In data verification, the observations made by the vehicle itself are usually assumed to be precise and it also assumes that there is always a few identified fully trustworthy users which can assist the verification process. However, in cooperative mobility tracking, these assumptions are generally quite impractical. Instead, the observations made by a vehicle itself are highly likely to be quite noisy or totally unavailable in some extreme environments. Furthermore, none of the cooperating vehicles can be treated as fully trustworthy all the time. Therefore, in terms of both the objective and problem setup, the existing algorithms proposed for mobility data verification are not applicable to the problem considered in this article.

C. Secure Localization

The third and the most relevant category is secure localization in WSN and VANETs, which considers how to detect and remove bogus information during the cooperative positioning process or reduce their effects on the reliability and precision of positioning.

In survey papers [31], [32], secure localization algorithms are classified into two categories: 1) filtering algorithms and 2) detection algorithms. Filtering algorithms select a subset containing position observations that are believed to be benign to obtain the final estimate, such as gridding and voting in [23] and [33], or use some robust loss functions in the formulation of the location estimation to minimize the influence of bogus information, such as the least median squares (LMS) estimation proposed in [34] and the minimum mean absolute error (MMAE) estimation used in [35] and [36]. In contrast, the objective of various detection algorithms is to sort out all bogus information and exclude them from the cooperative positioning process. If properly designed, the detection algorithms usually would outperform the filtering algorithms since they attempt to retain as much benign information as possible and at the same time to exclude as much bogus data as possible. Therefore, in this article, we focus on the design of detection algorithms.

Early detection algorithms are mostly based on the minimum mean square error (MMSE) consistency check, which was first proposed in [23] as a part of the attack-resistant minimum mean square estimation (ARMMSE) algorithm. However, ARMMSE is sometimes regarded as a filtering algorithm since it only selects a subset of benign data. The cluster-based minimum mean square estimation (CMMSE) in [24] utilizes the consistency check and extends it to a true detection algorithm. Recently, hypothesis testing-based detection algorithms, such as the generalized-likelihood ratio test (GLRT) and malicious node detection algorithm (MNDC), were proposed in [25] and [26].

Most secure localization algorithms in WSN and VANETs except MNDC only deal with single-snapshot information at a specific time instant. This means that these algorithms cannot utilize the temporal correlation of the mobility information sequence entailed by the dynamic properties of vehicles to increase the detection accuracy. As for MNDC, though it directly averages data collected in a period and then conducts sequential hypothesis testing, its whole framework is based on the static node and fixed position assumption, which makes MNDC inapplicable in the dynamic cooperative mobility tracking scenario where all vehicles are moving in most of the time. In summary, most secure localization algorithms are designed for the static positioning problems and hence are not optimal solutions for dynamic cooperative tracking. Those being said, the classical secure localization algorithms will still be used as the baseline in this article and be compared with the proposed detection algorithms since there is no existing algorithm designed for malicious user detection in dynamic scenarios reported in the literature.

III. System Model

In general, the physical motion of a vehicle can be described by a first-order hidden Markov model [37]

$$\begin{aligned} \boldsymbol{s}[j] &= \boldsymbol{f}(\boldsymbol{s}[j-1], \boldsymbol{u}[j], \boldsymbol{w}[j]) \\ \boldsymbol{z}[j] &= \boldsymbol{g}(\boldsymbol{s}[j], \boldsymbol{v}[j]) \end{aligned} \tag{1}$$

where j is the discrete time instant index, $\boldsymbol{s}$ is the state of the vehicle which includes velocity and position, $\boldsymbol{u}$ is the command process or equivalently the driving input, and $\boldsymbol{w}$ is the state noise; $\boldsymbol{z}$ is the observations through sensing devices, such as GPS, IMU, LIDAR, etc., and $\boldsymbol{v}$ is the measurement noise; and $\boldsymbol{f}$ and $\boldsymbol{g}$ are the state transfer and measurement functions which can be obtained by the physical laws of the motion and the properties of the sensing devices, respectively.

We assume that all vehicles in cooperation are equipped with GPS, IMU, and an integrated sensing system which may include LIDAR, radar, camera, and so on. Each vehicle obtains its own position estimate from GPS, its own velocity estimate from IMU and wheel encoders, and the relative position and velocity with respect to other vehicles through the integrated sensing system. The detailed state transfer and observation model are presented as follows.

A. System State Transfer Model

For a vehicle V_i, we can describe its mobility in a system state transfer equation [37]

$$\boldsymbol{s}_i[j] = \boldsymbol{A}\boldsymbol{s}_i[j-1] + \boldsymbol{B}_{\boldsymbol{u}}\boldsymbol{u}_i[j] + \boldsymbol{w}_i[j] \tag{2}$$

where

$$\boldsymbol{s}_i = \begin{pmatrix} x_i \\ \dot{x}_i \\ y_i \\ \dot{y}_i \end{pmatrix}, \qquad \boldsymbol{u}_i = \begin{pmatrix} F_{i,x} \\ F_{i,y} \end{pmatrix}, \qquad \boldsymbol{w}_i = \begin{pmatrix} w_{x_i} \\ w_{\dot{x}_i} \\ w_{y_i} \\ w_{\dot{y}_i} \end{pmatrix} \tag{3}$$

$$\boldsymbol{A} = \begin{pmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{B}_{\boldsymbol{u}} = \begin{pmatrix} \frac{(\Delta t)^2}{2} & 0 \\ \Delta t & 0 \\ 0 & \frac{(\Delta t)^2}{2} \\ 0 & \Delta t \end{pmatrix} \tag{4}$$

x_i, y_i and $\dot{x}_i$, $\dot{y}_i$ are the Cartesian coordinates and velocities of V_i; $F_{i,x}$ and $F_{i,y}$ are the vehicle command process that provides acceleration, which can be provided by the IMU; $\boldsymbol{w}$ is the state noise which can be usually modeled as additive white Gaussian noise (AWGN); and Δt is the discrete time step.

B. Observation Model

The observation at an arbitrary vehicle V_s is composed of two parts: 1) the observation of its own mobility status, such as those provided by GPS and IMU, denoted as $\boldsymbol{z}_s$ and 2) the observation of the relative mobility status between another vehicle V_i and itself, such as those provided by the integrated sensing system, denoted as $\boldsymbol{z}_{i\to s}$. For $\boldsymbol{z}_s$, we have

$$\boldsymbol{z}_s[j] = \boldsymbol{H}_s \boldsymbol{s}_s[j] + \boldsymbol{v}_s[j] \tag{5}$$

where $\boldsymbol{H}_s$ is the measurement matrix and $\boldsymbol{v}_s$ is the measurement noise, both of which can be determined by the properties of the sensing devices of V_s. For $\boldsymbol{z}_{i\to s}$, we have

$$\boldsymbol{z}_{i\to s}[j] = \boldsymbol{H}_{i\to s} \boldsymbol{s}_{i\to s}[j] + \boldsymbol{v}_{i\to s}[j] \tag{6}$$

where $\boldsymbol{s}_{i\to s}[j] = \boldsymbol{s}_i[j] - \boldsymbol{s}_s[j]$ is the relative state between V_i and V_s. The detailed value of $\boldsymbol{H}_{i\to s}$ and the statistical property of $\boldsymbol{v}_{i\to s}$ depend on the sensing device involved and the way to extract the mobility-related information from the sensor data. Without loss of generality, in this article, we assume that in both cases, the sensing devices have direct unbiased measurement of the state, and the measurement noise is AWGN with known variance.

C. Secure Data Exchange and Fusion Model

The secure data exchange and fusion model proposed here is based on the cooperative mobility tracking algorithm developed in our previous work [7]. The structure of the model is shown in Fig. 1.

The vehicle observed by other vehicles is called the target vehicle and denoted as V_T. The cooperative vehicles are denoted as $V_1, V_2, \ldots, V_N$, where N is the number of cooperative vehicles. Instead of $\boldsymbol{z}_{T\to i}[j] + \boldsymbol{z}_i[j]$, mobility state observation of V_T from V_i at discrete-time instant t_j is written as $\boldsymbol{z}_{ij}$ for the rest of this article for simplicity. After receiving current observations from N vehicles, V_T passes observations to its detection module. Together with observations from all cooperative vehicles in previous $K-1$ time instants, they are stored in the detection module and form the observation matrix $\boldsymbol{M}_{\boldsymbol{z}}$

$$\boldsymbol{M}_{\boldsymbol{z}} = \begin{pmatrix} z_{11} & z_{12} & \cdots & z_{1K} \\ z_{21} & z_{22} & \cdots & z_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ z_{N1} & z_{N2} & \cdots & z_{NK} \end{pmatrix} \tag{7}$$

where K is the length of observation sequence. $\boldsymbol{M}_{\boldsymbol{z}}$ is then analyzed by the proposed sequential detection algorithm (DMMSD or MRED) in the detection module and generate the enabling signals for N independent Kalman filters on V_T (ellipses labeled as KF in Fig. 1). At the same time, N current observations are also sent to the corresponding Kalman filters to generate state estimates of V_T. As the outcome of the detection module, the enabling signals indicate which Kalman filters receive trustworthy state observation input and hence only those filters will generate outputs and pass them to the data fusion module to form the global mobility estimate of V_T at current time instant.

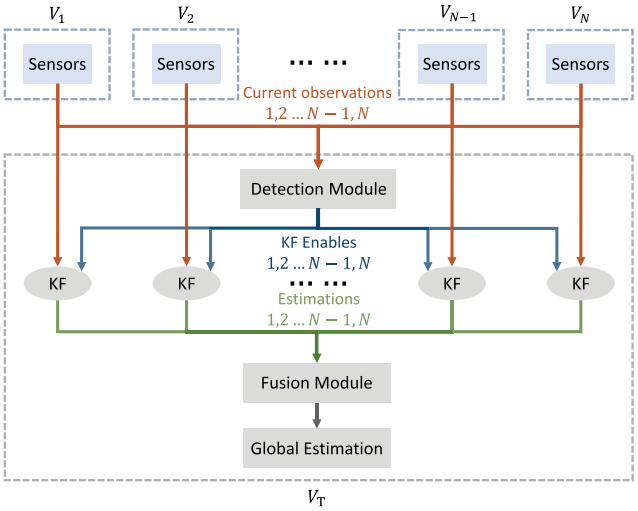


Fig. 1. Secure data exchange and fusion model.

D. Extended Threat Model

Existing threat models in the literature only include time-irrelevant attacks, which cannot fully characterize all attack patterns that may possibly be conducted in the case of dynamic cooperative tracking. Thus, to develop a comprehensive testing benchmark for evaluating the performance of the existing and proposed algorithms under different attack patterns, the extended threat model is presented here by considering both temporal and spatial distribution of mobility observations.

According to the spatial distribution of bogus observations, malicious attacks can be classified into two categories.

1) *Uncoordinated Attack:* There is no communication among malicious users, thus, the bogus observations from different malicious users are independent.
2) *Coordinated Attack:* Before sending their own bogus observations to the target vehicle, malicious users will first communicate with each other and agree on a center state (a mutual bogus state that deviates from the real state of V_T). Then, each malicious user generates its bogus observation around this center.

In uncoordinated attack, it is quite common that the deviations introduced by different bogus observations cancel out with others, which results in less degradation in the cooperative tracking performance. Therefore, the more threatening coordinated attack will be our focus and all attacks in the rest of this article are coordinated if not especially noted.

According to the temporal distribution of bogus observations, malicious attacks can be classified into three categories.

1) *Continuous Trajectory Attack:* The bogus state observations from one or a group of malicious users form a continuous and slow-varying malicious state trajectory

over a certain period. Its goal is to mislead the mobility state estimate of V_T to a predefined malicious state trajectory. Usually, a continuous trajectory attack lasts for a relatively long period.

2) *Continuous Random Attack:* No malicious state trajectory is designed, thus, the bogus observations from adjacent time instants have less correlation and may vary a lot in the state space. Its goal is to simply degrade the stability and accuracy of the mobility state estimates of V_T rather than leading it to a wrong trajectory. It is also assumed to last for a relatively long period.
3) *Sparse Random Attack:* The bogus observations only randomly occur at a few isolated time instants. For most of the time, observations from malicious users are consistent with those from benign users. While the effects of this type of attack would be less severe as compared with the previous two, the malicious users are more difficult to be detected and the effects could build up over time to degrade the tracking performance.

By considering both the spatial and the temporal distribution, three major types of attacks we focused on in this article are constructed and presented as follows.

1) *Coordinated Continuous Trajectory Attack:*

$$\boldsymbol{z}_{mi}[j] = \boldsymbol{s}_{(m\_\text{traj})}[j] + \boldsymbol{\delta}_{mi}[j], \quad (j \in T_m). \tag{8}$$

2) *Coordinated Continuous Random Attack:*

$$\boldsymbol{z}_{mi}[j] = \boldsymbol{s}_{(m\_\text{rand})}[j] + \boldsymbol{\delta}_{mi}[j], \quad (j \in T_m). \tag{9}$$

3) *Coordinated Sparse Random Attack:*

$$\boldsymbol{z}_{mi}[j] = \boldsymbol{s}_{(m\_\text{pulse})}[j] + \boldsymbol{\delta}_{mi}[j], \quad \left(j \in T_{pm}\right). \tag{10}$$

In (8)–(10), $i \in M$ and M is the set of the IDs of malicious users; $\boldsymbol{z}_{mi}[j]$ is the bogus state observation from the ith malicious user at discrete time instant $t_j = j\Delta t$; $T_m = [t_{\text{start}}, t_{\text{end}}]$ is the duration of the continuous trajectory and continuous random attack; AND $T_{pm} = \{t_{p1}, t_{p2}, \ldots, t_{pL}\}$ is the set of isolated time instants when sparse random attacks occur. $\boldsymbol{s}_{(m\_\text{traj})}$, $\boldsymbol{s}_{(m\_\text{rand})}$, and $\boldsymbol{s}_{(m\_\text{pulse})}$ are the center of bogus state that are agreed by all malicious users in coordinated attack. Small noises $\boldsymbol{\delta}_{mi}$ are added by each malicious user independently to make the spatial distribution of the malicious data at each time instant not abnormally dense.

IV. Proposed Algorithms

A. Dynamic Model-Based Mean State Detection

As discussed earlier in Section II, there is only one existing malicious user detection algorithm in the literature, namely, the MNDC, which works on a sequence of observations rather than single-snapshot information for detection. The simple averaging operation over the sequence of observations was introduced to reduce the influence of observation noise and increase the malicious user detection performance. However, this algorithm was originally developed only for the localization problem in WSN and based on the static node and fixed position assumption. When it comes to the mobility tracking problem, the tracked target vehicle V_T would be in different states at different time instants, which makes the simple direct averaging operation invalid in this scenario. However, by utilizing the dynamic model which captures the correlation between different states, the mobility observations sequence can still be "averaged" to reduce the influence of noise and improve the malicious user detection performance. Following this line of thought, we propose our first algorithm, namely, the DMMSD.

The DMMSD consists of three main steps: 1) prediction, which utilizes the dynamic model to convert all observations in the sequence into predictions of current mobility state of target vehicle; 2) averaging, the key operation to reduce the influence of observation noise and increase the detection accuracy; and 3) consistency check and clustering, the final operation to detect whether there are malicious users and identify all of them if so. The detailed procedures of each step are presented as follows.

1) Prediction With the Dynamic Model: The differences among the observations provided by one particular user at different time instants arise two major reasons.

1) The first reason is the effect of random observation noises, which also exists for the static WSN scenario.
2) The second reason is the motion of the target vehicle itself, which is exactly what invalidates the direct averaging of observations at different time instants.

To cope with the motion of the target vehicle and convert all the past observations into predictions of current mobility state of the target vehicle, we rely on the state transfer equation (2) or equivalently the dynamic model which completely depicts the theoretical trajectory of the target vehicle.

The core idea of this step is pretty straightforward. The acceleration of the target vehicle at each time instant is measured by its own onboard IMU very precisely and can be regarded as trustworthy. Therefore, for any single past observation provided by any particular cooperating vehicle, the target vehicle can predict its possible counterpart at current time instant with the dynamic models and the measured acceleration data over the period.

For instance, consider any particular cooperating vehicle V_i observing V_T. For a period of time $\{t_1, t_2, \ldots, t_K\}$ (t_K is the current time instant), we can use the single-observation data $\boldsymbol{z}_{i1}$ from V_i at time instant t_1 and the acceleration data $\{a_1, a_2, \ldots, a_{K-1}\}$ from V_T during this period to predict the possible observation V_i may send to V_T currently, and we denote this predication as $\hat{\boldsymbol{z}}_{i(1\to K)}$. Similarly, the corresponding predictions $\{\hat{\boldsymbol{z}}_{i(2\to K)}, \hat{\boldsymbol{z}}_{i(3\to K)}, \ldots, \hat{\boldsymbol{z}}_{i(K\to K)}\}$ of past observations $\{\boldsymbol{z}_{i2}, \boldsymbol{z}_{i3}, \ldots, \boldsymbol{z}_{iK}\}$ can be obtained. Note that $\hat{\boldsymbol{z}}_{i(K\to K)} = \boldsymbol{z}_{iK}$.

After the prediction process, state observations from V_i at different time instants are all converted into state predictions that correspond to the current time instant, which therefore effectively excludes the influence of the motion of the target vehicle in this period and makes the averaging strategy in the second step applicable.

2) Variance Reduction With the Averaging: At any particular time instant, the way to detect the malicious user is to check whether its observation is consistent with others. However, due to the observation noise, even single observation from the benign vehicle would have a large variance, which makes it

quite difficult to determine whether the inconsistency is due to the normal observation noise or the deliberate malicious deviation. It is especially so when the deviation injected by the malicious user is relatively small.

After averaging, the observation noise is significantly reduced, which in turn makes the determination of the cause of inconsistency much more accurate. The outcome of averaging is the mean of state predictions or equivalently the mean state $\bar{z}_{iK}$ of V_i, i.e.,

$$\bar{z}_{iK} = \frac{\hat{z}_{i(1\to K)} + \hat{z}_{i(2\to K)} + \cdots + \hat{z}_{i(K\to K)}}{K}. \tag{11}$$

Qualitatively, the variance of $\bar{z}_{iK}$ will be much smaller than the variance of any single prediction $\hat{z}_{i(j\to K)}$ or single observation z_{ij}. However, to determine the sequence length K that achieves the maximum variance reduction, one needs to obtain the quantitative relation between the variance of z_{ij}, the variance of $\bar{z}_{iK}$, and the sequence length K. The relation is obtained as the following theorem.

Theorem 1: The variance of the mean of state predictions given by a particular cooperating vehicle is approximately

$$D(\bar{z}_K) = \begin{pmatrix} D(\bar{x}_K) \\ D(\bar{v}_K) \end{pmatrix} \approx \frac{1}{K^2} \begin{pmatrix} K\sigma_x^2 + (\Delta t)^2 \sigma_v^2 \sum_{j=1}^{K-1} j^2 \\ K\sigma_v^2 + (\Delta t)^2 \sigma_a^2 \sum_{j=1}^{K-1} j^2 \end{pmatrix} \tag{12}$$

where $D(\bar{x}_K)$and$D(\bar{v}_K)$ are the variance of mean position and velocity predictions, respectively, σ_x^2andσ_v^2 are the variance of single position and velocity measurement of cooperating vehicle respectively, and σ_a^2 is the variance of single acceleration measurement.

Proof: See Appendix A. ∎

To get a clear idea of the amount of reduction in variance by averaging the predictions, we adopt a practical observation interval $\Delta t = 0.1$s and assume that $\sigma_x^2 = \sigma_v^2 = \sigma^2$, and in practice, $\sigma^2 \gg \sigma_a^2$. Accordingly

$$\frac{D(\bar{z}_K)}{\sigma^2} \approx \frac{1}{600K} \begin{pmatrix} 600 + (K-1)(2K-1) \\ 600 \end{pmatrix}. \tag{13}$$

Fig. 2 clearly shows the trend of $D(\bar{x}_K)/\sigma^2$ and $D(\bar{v}_K)/\sigma^2$ with varying K. As shown in the figure, $D(\bar{v}_K)/\sigma^2$ decreases in proportional to $1/K$. However, $D(\bar{x}_K)/\sigma^2$ has a minimum value due to accumulative noises brought by dynamic model-based prediction. Therefore, under the assumptions above, $K = 16$ is the optimal sequence length to minimize $D(\bar{x}_K)/\sigma^2$.

After dynamic model-based averaging, the observation matrix $\boldsymbol{M_Z}$ in (7) is converted into mean state vector $\bar{\boldsymbol{Z}}$ which includes the mean of state predictions from N cooperative vehicles

$$\bar{\boldsymbol{Z}} = \begin{pmatrix} \bar{z}_{1K} & \bar{z}_{2K} & \cdots & \bar{z}_{NK} \end{pmatrix}^T. \tag{14}$$

3) Malicious User Detection Using the Consistency Check and Clustering: By reducing the variance of the observations provided by cooperative vehicles, the dynamic model-based averaging could enlarge the difference between the bogus and normal observations, thus, make it easier to distinguish them. However, a criterion is yet set to determine whether there are bogus observations. Moreover, a tool is needed to identify all bogus observations and the corresponding malicious users generating them. For this task, we propose the following two-step procedure.

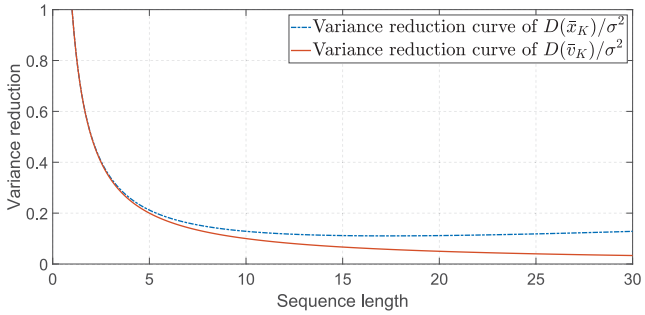


Fig. 2. Variance reduction amount with different sequence length.

1) *Consistency Check:* It determines whether there are bogus observations by analyzing the distribution of the mean states of all cooperative vehicles in state space.
2) *Clustering:* If the step above indicates the existence of bogus observations, a clustering algorithm is applied to classify those mean states into two clusters and identify the bogus mean states sent by malicious users accordingly.

In the first step, the MMSE consistency is considered, which was first proposed in [23]. Its core idea is concisely explained here, while the detailed derivations can be found in the original paper: since the benign state observations are the sum of the true target state and zero-mean Gaussian observation noise, MMSE of $\bar{\boldsymbol{Z}}$ should satisfy $P\{\text{MMSE} < \tau^2\} \to 1$ if all the users are benign and the normalized threshold τ^2 is selected properly. However, if there are bogus state observations, the MMSE would be very likely to exceed τ^2. Thus, the MMSE of $\bar{\boldsymbol{Z}}$ is computed to determine whether there are bogus mean states.

Content-based clustering has been shown to be an effective approach [24], [38] to classify vehicles into different groups based on the distribution of shared data in resource sharing optimization and malicious user detection scenario in IoV or WSN. Thus, it is adopted in DMMSD as the last step to classify the mean states and the corresponding users into a benign and malicious group. If the consistency check indicates the existence of bogus observations, clustering on $\bar{\boldsymbol{Z}}$ will be performed. Otherwise, all the observations will be regarded as from benign users. Considering the number of clusters is always two in the malicious user detection scenario, K-means clustering is a very effective and appropriate algorithm, thus, it is adopted in our implementation of DMMSD. However, any other clustering algorithm (e.g., density-based clustering, spectral clustering, and so on) is also fully compatible with this framework.

The cluster with more members will be regarded as the benign group and all the corresponding users will be marked as benign. All the users in the other cluster will be marked as malicious users. The result of consistency check and clustering are concluded as a boolean vector or equivalently a trust table, which describes each vehicle as benign or malicious.

Algorithm 1 DMMSD

Input: $\boldsymbol{M_Z}, \boldsymbol{A}, \boldsymbol{B_u}, \boldsymbol{a}$
Output: *trust_table*

```
for i = 1 → N do
    for j = 1 → K do
        for k = 1 → K − j do
            M_Z[i, j] = A M_Z[i, j] + B_u a[j + k]
        end for
    end for
end for
Z̄ = mean of all columns of M_Z
Res_MMSEcheck = MMSE_consistency_check(Z̄)
if Res_MMSEcheck indicates all vehicles are benign then
    all vehicles are marked as benign in trust_table
else
    Res_Cluster = Clustering(Z̄)
    the bigger cluster → benign
    the smaller cluster → malicious
    convert the clustering result to trust_table
end if
```

The pseudocode of the DMMSD algorithm is shown in Algorithm 1. After generating the trust table by executing the DMMSD algorithm, the subsequent data fusion process as shown in Fig. 1 will be performed. The target vehicle only uses the outcomes (mobility state estimates of V_T) of the Kalman filters that are enabled by the trust table as the input of our previously proposed fusion algorithms [7] to obtain the global estimate of the mobility state of itself.

B. Inherent Disadvantages of DMMSD

Though DMMSD reduces the variance of observations and substantially increases the accuracy of detection in continuous trajectory attack (8), it also has two inherent disadvantages.

1) Utilizing the dynamic model makes averaging operation and observation noise reduction applicable in dynamic scenarios, while it also brings in accumulative noises in the process of prediction, which limits the performance of DMMSD when bogus state observations do not deviate too much from the true state of the target vehicle.
2) Consistency check and clustering are only based on mean states $\bar{\boldsymbol{Z}}$ in DMMSD. However, as well known, both the mean and the variance are essential to depict the statistical property of a data sequence. Without including the variance in its development, DMMSD cannot detect abnormal fluctuations in the observation sequence, and thus, cannot resist the other two types of attacks in the extended threat model: 1) the continuous random attacks as described in (9) and 2) the sparse random attacks as described in (10).

The first disadvantage can be clearly seen in the $(K-1)(2K-1)$ term in (13), which limits the maximum variance reduction ratio. The detail of the reason why the variance information of the observation sequence cannot be integrated into DMMSD is presented in Appendix B.

Consequently, two inherent disadvantages of DMMSD make it insensitive to continuous trajectory attack with small deviation and cannot resist continuous and sparse random attack at all. To solve this problem, we propose our second detection algorithm, namely, the MRED.

C. Mean Residual Error Detection

Essentially, all sequential detection algorithms consist of different operations on the state observations matrix $\boldsymbol{M_Z}$. However, the order of different operations may lead to different performances. To have a vision of the connection and difference between MRED and DMMSD, the core steps of DMMSD are summarized and reviewed from the point of view of matrix operations.

1) Convert state observations into state predictions and average predictions of each cooperating vehicle (i.e., perform prediction operation on all columns of $\boldsymbol{M_Z}$, then average all columns) to obtain the mean states vector $\bar{\boldsymbol{Z}}$.
2) Conduct consistency check and clustering on $\bar{\boldsymbol{Z}}$ (i.e., analyze the relation of different rows in $\bar{\boldsymbol{Z}}$).

Simply speaking, the operation order in DMMSD is "column first, row second." Thus, it is natural to think about reversing the order to "row first, column second" and see the difference. The MRED algorithm is proposed based on this idea.

MRED consists of three steps: 1) computing the residual error, a better preprocessing approach to exclude the influence of the motion of target vehicle; 2) obtaining the squared residual error, the operation that makes MRED resistant to continuous and sparse random attack; and 3) consistency check and clustering. The detail of the three steps is introduced as follows.

1) Residual Error in MRED: The inevitable accumulative noise problem brought by dynamic model-based prediction in DMMSD is the result of the "column first" operation order. By switching to "row first," MRED not only excludes the influence of the motion of the target vehicle in this step but also naturally avoids the prediction process and accumulative noise.

The procedure of this step is described as follows. An arbitrary vehicle is selected as the reference and the residual errors between the observations of the reference and other vehicles are computed. The residual error between vehicles V_i and V_{Ref} at t_j is denoted as $\boldsymbol{r}_{ij}$

$$\boldsymbol{r}_{ij} = z_{ij} - z_{\text{Ref}j}. \tag{15}$$

The collection of all residual errors forms the residual error matrix, which is denoted as

$$\boldsymbol{M_r} = \begin{pmatrix} \boldsymbol{r}_{11} & \boldsymbol{r}_{12} & \cdots & \boldsymbol{r}_{1K} \\ \boldsymbol{r}_{21} & \boldsymbol{r}_{22} & \cdots & \boldsymbol{r}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{r}_{N1} & \boldsymbol{r}_{N2} & \cdots & \boldsymbol{r}_{NK} \end{pmatrix}. \tag{16}$$

This simple step could effectively exclude the influence of the motion since the movement of the target vehicle has the same effect on observations of all cooperating vehicles. Thus, such an effect is naturally canceled out after the subtraction

between the reference and other vehicles. Therefore, computing residual error matrix $\boldsymbol{M_r}$ already excludes the influence of the motion. With this property, we can average different columns of $\boldsymbol{M_r}$ directly to obtain the mean residual errors of all cooperative vehicles, which avoids bringing in the prediction process and any additional noises (hence overcomes the first inherent disadvantage of DMMSD). Moreover, this property also brings the variance reduction ratio to the theoretical lower bound and reduces the computation complexity by replacing the computation-heavy prediction process with simple residual error computing.

Similar to (14), we use N mean residual errors as the mean residual error vector, which will be used in the third step of MRED and denoted as

$$\bar{\boldsymbol{R}} = \begin{pmatrix} \bar{r}_1 & \bar{r}_2 & \cdots & \bar{r}_N \end{pmatrix}^T. \tag{17}$$

2) Squared Residual Error in MRED: The mean residual error vector $\bar{\boldsymbol{R}}$ plays the same role as the mean state vector $\bar{\boldsymbol{Z}}$ in DMMSD. Thus, to detect the continuous and sparse random attack, one still need another indicator which can characterize the fluctuation of the observation sequence. The squared residual error is exactly what is needed.

Briefly speaking, if any particular vehicle V_i is conducting a continuous or sparse random attack, the mean residual error $\bar{r}_i$ between this vehicle and the reference can be quite small due to the potential cancellation between positive and negative residual errors at different time instants. Nevertheless, the mean of squared residual error $\bar{r}_i^2$ will be abnormally large because of the nonnegative property of r_{ij}^2, which means $\bar{r}_i^2$ actually plays a similar role as the desired but unachievable variance indicator in DMMSD to detect continuous and sparse random attacks.

By elementwise squaring the residual error matrix $\boldsymbol{M_r}$, we can obtain the matrix of the squared residual error $\boldsymbol{M_r^2}$

$$\boldsymbol{M_r^2} = \begin{pmatrix} r_{11}^2 & r_{12}^2 & \cdots & r_{1K}^2 \\ r_{21}^2 & r_{22}^2 & \cdots & r_{2K}^2 \\ \vdots & \vdots & \ddots & \vdots \\ r_{N1}^2 & r_{N2}^2 & \cdots & r_{NK}^2 \end{pmatrix}. \tag{18}$$

Similarly, the mean of squared residual error vector $\bar{\boldsymbol{R}}^2$ is written as

$$\bar{\boldsymbol{R}}^2 = \begin{pmatrix} \bar{r}_1^2 & \bar{r}_2^2 & \cdots & \bar{r}_N^2 \end{pmatrix}^T. \tag{19}$$

3) Pairwise Consistency Check and Clustering: After obtaining $\bar{\boldsymbol{R}}$ and $\bar{\boldsymbol{R}}^2$, the final step of MRED is also consistency check and clustering.

The essence of consistency check is to determine whether there are mean predictions or mean residual errors deviating from the majority in the state space. In DMMSD, the MMSE of $\bar{\boldsymbol{Z}}$, a global indicator formed by all elements of $\bar{\boldsymbol{Z}}$, is computed to find out inconsistency. Nevertheless, since the difference between observations of other vehicles and the reference (residual error and squared residual error) are already obtained in the previous two steps of MRED, computing pairwise rather than a global indicator is a much more effective and appropriate way to perform the consistency check. Thus, a pairwise consistency check is adopted in MRED and the following theorem gives the criterion to determine whether there are malicious users.

Algorithm 2 MRED

Input: $\boldsymbol{M_z}, L_l, L_u, L'_l, L'_u$
Output: *trust_table*

```
Ref= Identity of an arbitrary vehicle
Vec_Ref= state observations of reference
M_r = M_z − Vec_Ref
R̄ = mean of all columns of M_r
M_r^2 = (M_z − Vec_Ref)^2
R̄^2 = mean of all columns of M_r^2
if ∀i, L_l < R̄_i < L_u then
    all vehicles are marked as benign in trust_table_1
else
    Res_Cluster_1 = Clustering(R̄)
    the bigger cluster → benign
    the smaller cluster → malicious
    convert the clustering result to trust_table_1
end if
if ∀i, L'_l < R̄_i^2 < L'_u then
    all vehicles are marked as benign in trust_table_2
else
    Res_Cluster_2 = Clustering(R̄^2)
    the bigger cluster → benign
    the smaller cluster → malicious
    convert the clustering result to trust_table_2
end if
trust_table = trust_table_1 & trust_table_2
```

Theorem 2: If $\exists i \in \{1, 2, \ldots, N\}, \bar{r}_i \notin [L_l, L_u]$ or $\bar{r}_i^2 \notin [L'_l, L'_u]$, then V_i or V_{Ref} is a malicious user. Otherwise, all cooperative vehicles are benign. L_l, L'_l and L_u, L'_u are proper lower and upper bounds.

Proof: See Appendix C. ∎

After the pairwise consistency check, if the existence of malicious user is confirmed, clustering in the MRED algorithm is completed in the same way as in DMMSD; K-means clustering on $\bar{\boldsymbol{R}}$ and $\bar{\boldsymbol{R}}^2$ generates two local trust table. Considering mean residual error and the mean of the squared residual error are designed for detecting different types of attacks, so we use AND operation to integrate two local trust tables (boolean vectors) and form the global trust table, which ensures that MRED is immune to all types of attacks defined in Section III-D. The pseudocode of MRED is shown in Algorithm 2.

So far, the procedures of MRED and the principles behind them have been thoroughly interpreted. However, before presenting the evaluation results, we want to digress here to explain why the reputation management-based detection is not further adopted after MRED finishes, since in Sections I and II-A, we did mention that it can be utilized once feedback is generated by data processing algorithm. The major reason is that it is difficult for a reputation management-based detection framework to react to malicious behavior in a real-time manner. A qualitative illustration is presented as follows. For instance, malicious users may act with normal behavior over

a period of time to first obtain a sufficiently high reputation score before attacking. Then, for an extremely short duration (e.g., one or several discrete-time instants), they conduct a coordinated sparse random attack together. In this scenario, sparse attack will be detected by the proposed algorithms (e.g., MRED) and the negative feedback will be generated, while the reputation score of malicious users will not fall below the threshold immediately since their reputation score is sufficiently high to afford a few negative feedbacks (being detected by MRED) before dropping too much. Therefore, the bogus observations for those several time instants will bypass reputation-based detection and cause deviation in mobility estimates, which may expose vehicles to great danger considering that the AD, ITS, and IoV are highly dynamic scenarios. By contrast, the detection in MRED is naturally real time since the detection result is based on the trust table which is regenerated at every time instant. Therefore, considering the strict real-time detection requirement in a highly dynamic cooperative mobility tracking scenario, using the raw feedback (trust value in the trust table) generated by MRED to detect malicious users would be a more suitable approach.

V. Performance Evaluation

A. Simulation Setup and Parameter Settings

In this section, we evaluate our algorithms by comparing their performance with the detection and filtering algorithms that are originally designed for WSN. Although MNDC algorithm [26] has the best performance in existing detection algorithms designed for WSN, it is only designed for static scenario and inapplicable to cooperative mobility tracking scenario. Thus, we turned classical CMMSE from [24] into a sequential enhanced version, SeqMMSE, and compare it with DMMSD and MRED. Two classical filtering algorithms LMS estimation from [34] and MMAE estimation used in [35] and [36] are selected as representatives of filtering algorithms in our comparisons.

Since the proposed and existing algorithms all process position and velocity information in state observations independently, without loss of generality, in our following evaluation, we assume bogus observations only exist in the position information to simplify the simulation scenario for better figure readability. The true trajectory of the target vehicle is plotted as the blue line in Figs. 4(a), 5(a), and 6(a), which is obtained by a typical lane changing action. To further reduce the number of simulation parameters, we assume that bogus position information only deviates from true position information in the Y direction. Parameters of our simulation are listed in Table I and the performance of the proposed and existing algorithms are evaluated under three types of coordinated attack in the following sections.

B. Coordinated Continuous Trajectory Attack

In coordinated continuous trajectory attack simulation, the malicious trajectory $y_m[j]$ agreed by all malicious users is assumed to be obtained by adding a constant malicious deviation in Y direction ϵ_m to the true trajectory $y_t[j]$ of the target vehicle, i.e., $y_m[j] = y_t[j] + \epsilon_m$.

TABLE I
Simulation Parameters

Simulation Parameters	Value
Discrete time step	0.1 [s]
Duration of simulation	20 [s]
Number of total vehicles	30
Length of stored observation sequence	16
Number of malicious users	0-14 in Fig. 4b,5b,6b 8 in Fig. 3,4a,5a,6a
Variance of single benign observation	16 [m^2]
Variance of single bogus observation	12 [m^2]
Malicious deviation	8 [m] in Section V-B 20 [m] in Section V-C 60 [m] in Section V-D

1) Detection Rate Comparison: True-positive rate (TPR) and false-positive rate (FPR) are essential indicators to evaluate the performance of detection algorithms. Thus, we first compare the TPR and FPR of SeqMMSE, DMMSD, and MRED with varying malicious deviation ϵ_m and the variance of malicious data σ_m^2 in Fig. 3. As shown in the figures, DMMSD and MRED outperform SeqMMSE easily with higher TPR and lower FPR, so only DMMSD and MRED will be selected as representatives of the detection algorithms and be further compared with filtering algorithms in the following part.

2) Cooperative Tracking Performance Comparison: To compare the proposed algorithms with filtering algorithms LMS and MMAE, here we adopt the trajectory estimates and root mean square error (RMSE) as the performance indicators. The performance of different algorithms is shown in Fig. 4. It is clearly shown in Fig. 4(a) that the trajectory estimate of LMS has several sharp pulses. and the estimate of MMAE has an observable deviation in Y direction due to the bogus information. In contrast, estimates of DMMSD and MRED are much more stable and closer to the true trajectory of the target vehicle. RMSE curves with a varying number of malicious users in Fig. 4 provide us with more information. RMSEs of estimates of LMS, DMMSD, and MRED are quite stable as the number of malicious users increases, but RMSEs of DMMSD and MRED are significantly lower than LMS and just slightly above the RMSE curve of the "Ground Truth" (where no malicious exist). Though the RMSEs of DMMSD and MRED are slightly higher than MMAE when the ratio of malicious users is relatively low, this is actually a reasonable result caused by the detection property of DMMSD and MRED.

A brief explanation is given here: to make proposed algorithms more generalized and resistant to different types of attack, it is desirable to make the detection rate more balanced, i.e., make the TPR slightly smaller than 1 and FPR slightly larger than 0, instead of pushing one of them to the best. Therefore, when the ratio of malicious users is low, TPR smaller than 1 and FPR larger than 0 may cause few malicious users being regarded as benign and some benign vehicles being regarded as malicious. Consequently, RMSEs of the detection algorithms are slightly higher than MMAE. Though one is able to increase TPR or decrease FPR by fine-tuning parameters in the consistency check step, it often comes with the price of a significant increase in FPR or decrease TPR according to our test. To get a balanced and robust performance in all the

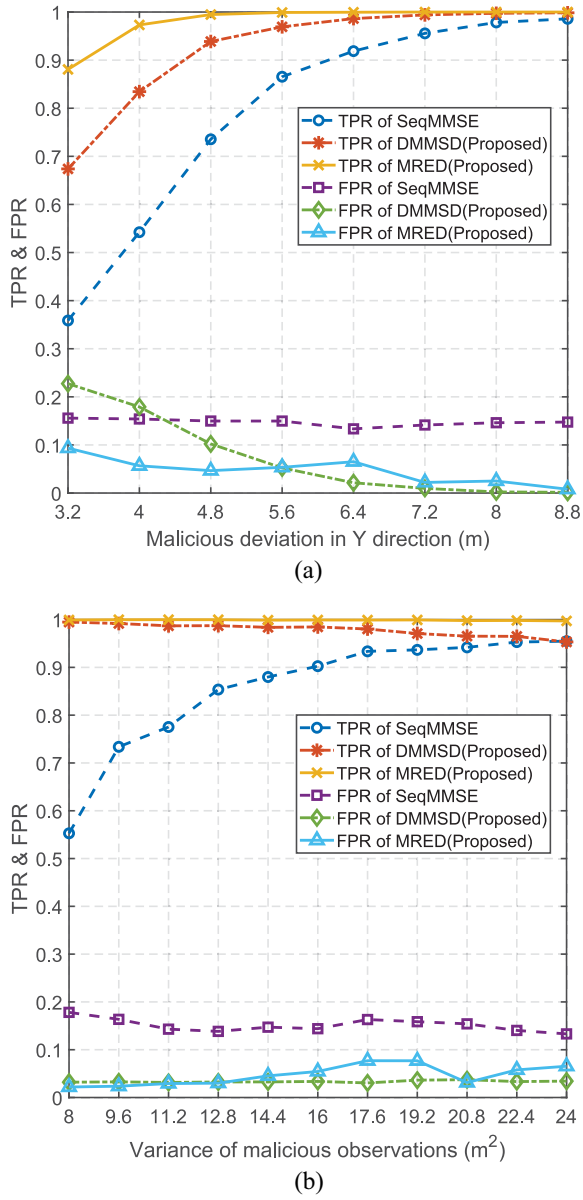


Fig. 3. TPR and FPR under continuous trajectory attack. (a) With varying deviation. (b) With varying variance.

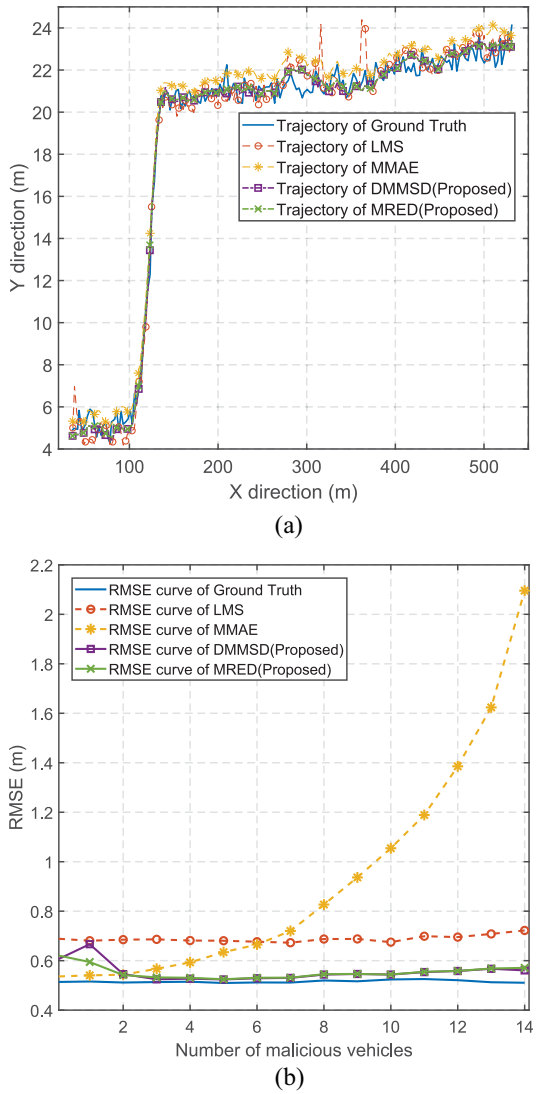


Fig. 4. Performance under coordinated trajectory attacks. (a) Trajectory estimates. (b) RMSEs.

malicious ratio, a little bit higher RMSE in the low malicious ratio case is completely acceptable.

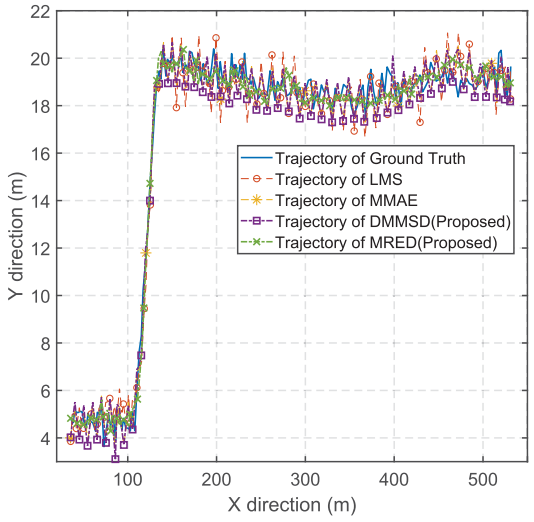
Quantitatively, RMSE of MRED is 16.2% higher than MMAE and 10.7% lower than LMS in the worst case, while 72.8% and 20.9% lower than MMAE and LMS in the best case, respectively. Since the detection algorithm has a fundamental influence on the reliability and security of cooperative mobility tracking in AD and ITS, it is apparently wiser to sacrifice a little bit precision for much stronger robustness.

C. Coordinated Continuous Random Attack

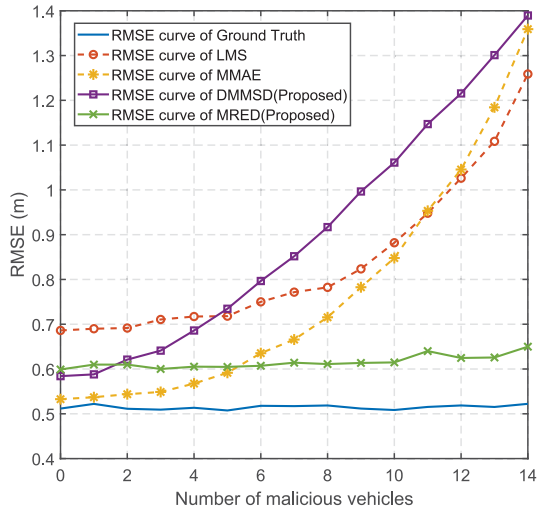
In coordinated continuous random attack simulations, bogus observations reported by malicious users are assumed to have a periodic positive and negative constant deviation ϵ_m from the true trajectory of the target vehicle in Y direction. Compared with deviation in continuous trajectory attack, ϵ_m is assumed to be larger in order to compensate for its periodic change and cause similar amount of deviation in the global estimate. From Fig. 5(a), one can see that DMMSD cannot resist this continuous random attack, which is consistent with the analysis in Section IV-B. Trajectory estimates and RMSE curves presented in Fig. 5(a) and (b) both indicate that MRED is still the most robust algorithm. RMSE of MRED is 13.6% higher than MMAE in the worst case but 52.2% lower in the best case. As for LMS and DMMSD, MRED easily outperforms them in any malicious ratio.

D. Coordinated Sparse Random Attack

Compared with the coordinated continuous random attack, bogus observations in coordinated sparse random attacks only occur at some isolated time instants. To compensate for the shorter duration, ϵ_m adopted in simulations is also set to be larger than the one in the continuous random attack. In Fig. 6(a), one can see that due to the very large deviation, the

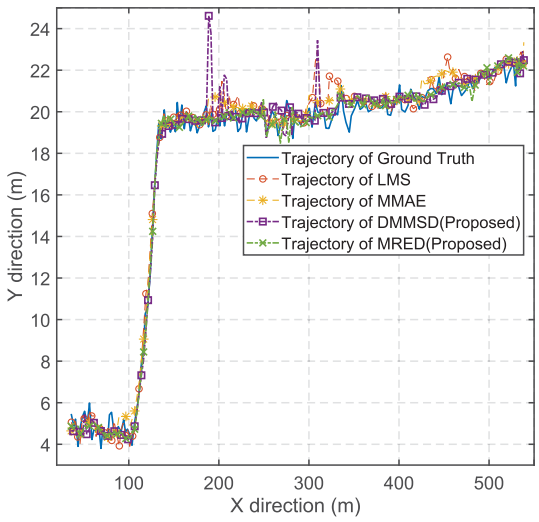


(a)

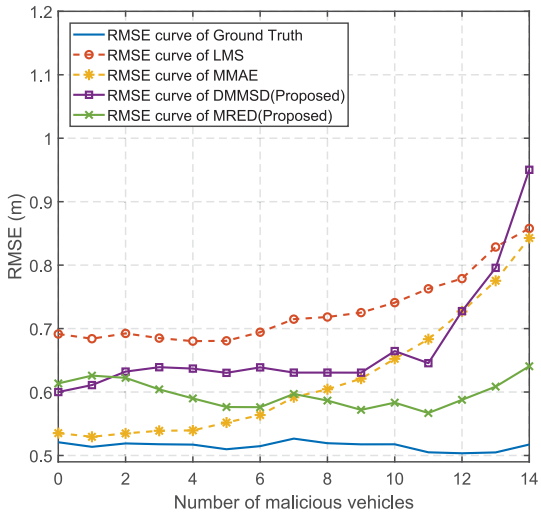


(b)

Fig. 5. Performance under coordinated continuous random attack. (a) Trajectory estimates. (b) RMSEs.



(a)



(b)

Fig. 6. Performance under coordinated sparse random attack. (a) Trajectory estimates. (b) RMSEs.

trajectory estimates of DMMSD and LMS are severely distorted at those attack moments. MMAE is better but still worse than the proposed MRED algorithm. RMSE curves in Fig. 6(b) have similar trend with those in Fig. 5(b). Quantitatively, the RMSE of MRED is 18.2% higher than MMAE in the worst case while 31.5% lower in the best case. Furthermore, if the attack frequency increases, the advantage of MRED would be more substantial.

E. Simulation Summary

In summary, the detection rate of MRED easily outperforms the existing SeqMMSE detection algorithm and the RMSE of MRED is about 0.6 m in almost any type of attack and any malicious user ratio. All the evaluation results strongly demonstrate the excellent robustness and precision of MRED as compared with existing classical detection and filtering algorithms.

VI. Conclusion

In this article, we proposed two sequential detection algorithms (namely, DMMSD and MRED) to exploit the temporal correlation of mobility observation sequence to improve the malicious user detection performance and provide cooperative mobility tracking with stronger robustness against bogus mobility information sent by malicious users. A secure mobility data exchange and fusion model was presented to integrate the proposed detection algorithms with existing cooperative tracking fusion algorithms and form a complete secure cooperative mobility tracking framework. Furthermore, to characterize all the possible threats and develop a more comprehensive testing benchmark for secure cooperative mobility tracking, we introduced an extended threat model. With this comprehensive testing benchmark, the performances of MRED and the existing filtering and detection algorithms are evaluated under different attack patterns. The advantage of the proposed MRED algorithm is clearly demonstrated with better detection accuracy and stronger robustness. Therefore, the MRED

algorithm is appropriate to detect bogus mobility information and the malicious users generating them, and secure the cooperative mobility tracking process. In the future, one may consider the utilization of the correlation among acceleration, velocity, and position of vehicles to conduct multilevel joint sequential detection.

APPENDIX A
PROOF OF THEOREM 1

The four dimensions of state and observation vector $\boldsymbol{s}$ and $\boldsymbol{z}$ are position and velocity in X direction and position and velocity in Y direction as we introduced in (3), (5), and (6). Without loss of generality, we assume the motions in X direction and Y direction are independent, so we only consider X direction in the following derivations.

First, the variance of single prediction $\hat{z}_{(j\to K)}$ from any particular cooperating vehicle is derived. For simplicity, we use scalars x, v, a as the position, velocity observation from cooperating vehicle, and acceleration observation from IMU of V_T. According to the state transfer function (2), we have

$$x_{j+1} = x_j + v_j\Delta t + \frac{1}{2}a_j(\Delta t)^2 \tag{20}$$

$$v_{j+1} = v_j + a_j\Delta t. \tag{21}$$

Then, we can get position and velocity prediction at t_K from the observation at t_1

$$\hat{x}_{1\to K} = x_1 + \Delta t\sum_{j=1}^{K-1} v_j + \frac{1}{2}(\Delta t)^2\sum_{j=1}^{K-1} a_j \tag{22}$$

$$\hat{v}_{1\to K} = v_1 + \Delta t\sum_{j=1}^{K-1} a_j. \tag{23}$$

Considering that we only have the velocity at t_1, we need to write $\hat{x}_{1\to K}$ as

$$\hat{x}_{1\to K} = x_1 + \Delta t\sum_{j=1}^{K-1} v_1 + (\Delta t)^2\sum_{j=1}^{K-1}\sum_{k=1}^{j-1} a_k + \frac{1}{2}(\Delta t)^2\sum_{j=1}^{K-1} a_j. \tag{24}$$

All noises of x, v, and a in (23) and (24) are assumed to be AWGN with known variance as mentioned in Section III-B. Use σ_x^2, σ_v^2, and σ_a^2 to represent their variances. Thus, the variances of $\hat{x}_{1\to K}$ and $\hat{v}_{1\to K}$ can be derived as

$$D(\hat{x}_{1\to K}) = D(x_1) + (\Delta t)^2 D\left(\sum_{j=1}^{K-1} v_1\right) + \frac{1}{4}(\Delta t)^4 D\left(\sum_{j=1}^{K-1} a_j\right) + (\Delta t)^4 D\left(\sum_{j=1}^{K-1}\sum_{k=1}^{j-1} a_k\right) \tag{25a}$$

$$D\big(\hat{v}_{1\to K}\big) = D(v_1) + (\Delta t)^2 D\left(\sum_{j=1}^{K-1} a_j\right). \tag{25b}$$

With further simplifications

$$D\big(\hat{x}_{1\to K}\big) = \sigma_x^2 + (K-1)^2(\Delta t)^2\sigma_v^2 + \frac{1}{4}(\Delta t)^4(K-1)\sigma_a^2 + (\Delta t)^4\sigma_a^2\Big[1^2 + 2^2 + \cdots + (K-2)^2\Big] \tag{26a}$$

$$D\big(\hat{v}_{1\to K}\big) = \sigma_v^2 + (K-1)^2(\Delta t)^2\sigma_a^2. \tag{26b}$$

A practical observation interval $\Delta t = 0.1$ s is used in our assumption. The variance of the observations of IMU is quite small and the length of sequence K will not be larger than 30. Therefore, the last two terms in (26a) are high-order small amount and can be neglected. Then, the approximation of $D(\hat{x}_{1\to K})$ has the same form as $D(\hat{v}_{1\to K})$, so we can use state vector to integrate them and simplify the expression

$$D\big(\hat{\boldsymbol{z}}_{1\to K}\big) = \begin{pmatrix} D\big(\hat{x}_{1\to K}\big) \\ D\big(\hat{v}_{1\to K}\big)\end{pmatrix} \approx \begin{pmatrix} \sigma_x^2 + (K-1)^2(\Delta t)^2\sigma_v^2 \\ \sigma_v^2 + (K-1)^2(\Delta t)^2\sigma_a^2\end{pmatrix}. \tag{27}$$

The form of variance of other prediction $D(\hat{\boldsymbol{z}}_{j\to K})$ is similar

$$D\big(\hat{\boldsymbol{z}}_{j\to K}\big) \approx \begin{pmatrix} \sigma_x^2 + (K-j)^2(\Delta t)^2\sigma_v^2 \\ \sigma_v^2 + (K-j)^2(\Delta t)^2\sigma_a^2\end{pmatrix}. \tag{28}$$

Eventually, the variance of the mean of state predictions $D(\bar{\boldsymbol{z}}_K)$ can be obtained

$$D(\bar{\boldsymbol{z}}_K) = \frac{D\big(\hat{\boldsymbol{z}}_{1\to K}\big) + D\big(\hat{\boldsymbol{z}}_{2\to K}\big) + \cdots + D\big(\hat{\boldsymbol{z}}_{K\to K}\big)}{K^2} \approx \frac{1}{K^2}\begin{pmatrix} K\sigma_x^2 + (\Delta t)^2\sigma_v^2\left(\sum_{j=1}^{K-1} j^2\right) \\ K\sigma_v^2 + (\Delta t)^2\sigma_a^2\left(\sum_{j=1}^{K-1} j^2\right)\end{pmatrix}.$$

APPENDIX B
DISCUSSION ON THE SECOND DISADVANTAGE OF DMMSD

It is natural to think of using variance estimate of all state predictions of each vehicle as an indicator to check the fluctuation of original observation sequence and determine the user is benign or malicious. For instance, for any particular cooperating vehicle V_i observing V_T, the variance estimate $\boldsymbol{S}_i^2$ of state predictions can be obtained as

$$\boldsymbol{S}_i^2 = \sum_{j=1}^{K}\frac{\big[\hat{\boldsymbol{z}}_{i(j\to K)} - \bar{\boldsymbol{z}}_{iK}\big]^2}{K-1}. \tag{29}$$

If state observations are from a benign user, the variance estimate of state predictions is very likely to fall in a reasonable range $[L_l, L_u]$, where the lower and upper bounds L_l, L_u are chosen properly according to the variance of a single observation. In contrast, the variance estimate of state predictions from a malicious user is very likely to be out of this range if it is conducting continuous or sparse random attack since the fluctuations of those observations would be abnormally large. So far, it seems that variance estimate can indeed be included as a proper indicator in DMMSD. However, the further analysis below shows that it is challenging to precisely determine L_l and L_u in DMMSD.

All dimensions of $\hat{\boldsymbol{z}}_{i(j\to K)}$ and $\bar{\boldsymbol{z}}_{iK}$ follow the Gaussian distribution, thus, $\hat{\boldsymbol{z}}_{i(j\to K)} - \bar{\boldsymbol{z}}_{iK}$ also follow the Gaussian distribution. It is well known that if independent random variables $X_i (i = 1, 2, \ldots, K)$ all follow the normalized Gaussian distribution, the sum of the square of X_i will follow the χ-square distribution. However, in our case, the variances of $\hat{\boldsymbol{z}}_{i(j\to K)}$ are different since different amount of noises are accumulated in the process of prediction, which is clearly shown in (28).

Therefore, each dimension of S^2 only follows generalized χ-square distribution, which does not have a closed-form probability density function. It means that we need to numerically recompute L_l, L_u according to our desired confidence level in every detection, which greatly increases the computation load of the algorithm. Worse still, the square operation in (29) will also enlarge accumulated noises and make it difficult to find out the proper lower and upper bound L_l, L_u, which further degrades the detection accuracy of the DMMSD algorithm in the continuous and sparse random attacks. Therefore, it is quite difficult to include a precise variance-based consistency check to detect the continuous and sparse random attacks in DMMSD.

APPENDIX C
PROOF OF THEOREM 2

The effectiveness and completeness of the pairwise consistency check can be first proved in continuous trajectory attack in all three cases as follows.

1) If reference and vehicle V_i are both benign, the residual error $\boldsymbol{r}_{ij}$ at t_j would follow the Gaussian distribution $\mathcal{N}(0, \sigma_i^2 + \sigma_{\text{Ref}}^2)$, and then the corresponding mean residual error $\bar{\boldsymbol{r}}_i$ follows $\mathcal{N}(0, [(\sigma_i^2 + \sigma_{\text{Ref}}^2)/K])$. Therefore, $P\{\bar{\boldsymbol{r}}_i \in [L_l, L_u]\} \to 1$ can be satisfied by choosing proper bounds L_l, L_u according to the distribution.
2) For the selected reference and a cooperative user V_i, if one of them is malicious and the other is benign, $P\{\bar{\boldsymbol{r}}_i \in [L_l, L_u]\}$ will be significantly decreased since the bogus observations usually have a large deviation from the true state of the target vehicle as compared with the normal unbiased noisy observations from a benign vehicle. Thus, the mean residual error will remain abnormally large after the averaging as compared with the case where both vehicles are benign.
3) If both of them are malicious, $\bar{\boldsymbol{r}}_i$ is very likely to fall in $[L_l, L_u]$ since they are conducting coordinated attack where the difference between their observation is small as we assumed. Nevertheless, this situation will not affect the detection since the consistency check step only needs to determine whether there are malicious users rather than identify all of them. Thus, as long as there are benign vehicles, the pair of one malicious user and one benign vehicle will always occur and then be detected by the pairwise consistency check.

The detection of continuous and sparse random attacks is almost the same except the indicator is switched to squared residual errors. $\boldsymbol{r}_{ij}$ at t_j follows the Gaussian distribution $\mathcal{N}(0, \sigma_i^2 + \sigma_{\text{Ref}}^2)$, then $[(K\bar{\boldsymbol{r}}_i^2)/(\sigma_i^2 + \sigma_{\text{Ref}}^2)]$ follows the χ-square distribution $\chi(K)$. Thus, we can also find proper bounds L_l', L_u' to guarantee that if both the reference and V_i are benign, $P\{\bar{\boldsymbol{r}}_i^2 \in [L_l', L_u']\} \to 1$. The existence of malicious users that are conducting continuous or sparse random attacks will be detected if $\bar{\boldsymbol{r}}_i^2$ falls out of the range.

REFERENCES

[1] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 829–846, Apr. 2018.

[2] S. Fujii *et al.*, "Cooperative vehicle positioning via V2V communications and onboard sensors," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, 2011, pp. 1–5.

[3] M. Rohani, D. Gingras, V. Vigneron, and D. Gruyer, "A new decentralized Bayesian approach for cooperative vehicle localization based on fusion of GPS and inter-vehicle distance measurements," in *Proc. Int. Conf. Connected Veh. Expo. (ICCVE)*, 2013, pp. 473–479.

[4] L. Altoaimy and I. Mahgoub, "Fuzzy logic based localization for vehicular ad hoc networks," in *Proc. IEEE Symp. Comput. Intell. Veh. Transp. Syst. (CIVTS)*, 2014, pp. 121–128.

[5] C.-H. Chen, C.-A. Lee, and C.-C. Lo, "Vehicle localization and velocity estimation based on mobile phone sensing," *IEEE Access*, vol. 4, pp. 803–817, 2016.

[6] L. Conde, R. Chelim, and U. Nunes, "Collaborative vehicle self-localization using multi-GNSS receivers and V2V/V2I communications," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, 2015, pp. 2525–2532.

[7] P. Yang, D. Duan, C. Chen, X. Cheng, and L. Yang, "Optimal multi-sensor multi-vehicle (MSMV) localization and mobility tracking," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, 2018, pp. 1223–1227.

[8] X. Cheng, D. Duan, L. Yang, and N. Zheng, "Cooperative intelligence for autonomous driving," *ZTE Commun.*, vol. 17, no. 2, pp. 44–50, 2019.

[9] Y. Li, D. Duan, C. Chen, X. Cheng, and L. Yang, "Occupancy grid map formation and fusion in cooperative autonomous vehicle sensing," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*. Chengdu, China, Dec. 2018, pp. 204–209.

[10] X. Cheng, R. Zhang, and L. Yang, *5G-Enabled Vehicular Communications and Networking*. Cham, Switzerland: Springer, 2018.

[11] X. Cheng, R. Zhang, and L. Yang, "Wireless toward the era of intelligent vehicles," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 188–202, Feb. 2019.

[12] X. Cheng, C. Chen, W. Zhang, and Y. Yang, "5G-enabled cooperative intelligent vehicular (5GenIV) framework: When Benz meets Marconi," *IEEE Intell. Syst.*, vol. 32, no. 3, pp. 53–59, May/Jun. 2017.

[13] B. Hu, L. Fang, X. Cheng, and L. Yang, "Vehicle-to-vehicle distributed storage in vehicular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*. Kansas City, MO, USA, May 2018, pp. 1–6.

[14] B. Hu, L. Fang, X. Cheng, and L. Yang, "In-vehicle caching (IV-Cache) via dynamic distributed storage relay (D^2SR) in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 843–855, Jan. 2019.

[15] R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, and M. Viroli, "Modelling and simulation of opportunistic IoT services with aggregate computing," *Future Gener. Comput. Syst.*, vol. 91, pp. 252–262, 2019.

[16] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, Aug. 2018.

[17] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," in *Proc. Black Hat Europe*, vol. 11, 2015, p. 2015.

[18] J. Cui, L. S. Liew, G. Sabaliauskaite, and F. Zhou, "A review on safety failures, security attacks, and available countermeasures for autonomous vehicles," *Ad Hoc Netw.*, vol. 90, Jul. 2019, Art. no. 101823.

[19] M. Muhammad and G. A. Safdar, "Survey on existing authentication issues for cellular-assisted V2X communication," *Veh. Commun.*, vol. 12, pp. 50–65, Apr. 2018.

[20] A. Alnasser, H. Sun, and J. Jiang, "Cyber security challenges and solutions for V2X communications: A survey," *Comput. Netw.*, vol. 151, pp. 52–67, Mar. 2019.

[21] G. M. L. Sarné, "A reputation agent model for reliable vehicle-to-vehicle information," in *Proc. 19th Workshop Objects Agents (WOA)*, 2018, pp. 33–38.

[22] P. De Meo, F. Messina, M. N. Postorino, D. Rosaci, and G. M. L. Sarné, "A reputation framework to share resources into IoT-based environments," in *Proc. IEEE 14th Int. Conf. Netw. Sensing Control (ICNSC)*, 2017, pp. 513–518.

[23] D. Liu, P. Ning, and W. K. Du, "Attack-resistant location estimation in sensor networks," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, 2005, p. 13.

[24] C. Wang, A. Liu, and P. Ning, "Cluster-based minimum mean square estimation for secure and resilient localization in wireless sensor networks," in *Proc. Int. Conf. Wireless Alg. Syst. Appl. (WASA)*, 2007, pp. 29–37.

[25] S. A. AlRoomi, I. Ahmad, and T. Dimitriou, "Secure localization using hypothesis testing in wireless networks," *Ad Hoc Netw.*, vol. 74, pp. 47–56, May 2018.

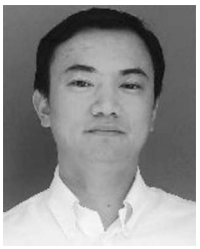
[26] X. Liu, S. Su, F. Han, Y. Liu, and Z. Pan, "A range-based secure localization algorithm for wireless sensor networks," *IEEE Sensors J.*, vol. 19, no. 2, pp. 785–796, Jan. 2019.
[27] A. Jaeger, N. Bißmeyer, H. Stübing, and S. A. Huss, "A novel framework for efficient mobility data verification in vehicular ad-hoc networks," *Int. J. Intell. Transp. Syst. Res.*, vol. 10, no. 1, pp. 11–21, 2012.
[28] M. Schäfer, V. Lenders, and J. Schmitt, "Secure track verification," in *Proc. IEEE Symp. Security Privacy*, 2015, pp. 199–213.
[29] L. Altoaimy and I. Mahgoub, "Mobility data verification for vehicle localization in vehicular ad hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2016, pp. 1–6.
[30] M. Sun, M. Li, and R. Gerdes, "A data trust framework for vanets enabling false data detection and secure vehicle tracking," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2017, pp. 1–9.
[31] H. Wang, Y. Wen, Y. Lu, D. Zhao, and C. Ji, "Secure localization algorithms in wireless sensor networks: A review," in *Proc. Adv. Comput. Commun. Comput. Sci.*, 2019, pp. 543–553.
[32] Y. Zeng, J. Cao, J. Hong, S. Zhang, and L. Xie, "Secure localization and location verification in wireless sensor networks: A survey," *J. Supercomput.*, vol. 64, no. 3, pp. 685–701, 2013.
[33] G. Yan, X. Chen, and S. Olariu, "Providing vanet position integrity through filtering," in *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.*, 2009, pp. 1–6.
[34] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, 2005, p. 12.
[35] D. Wang, J. Wan, M. Wang, and Q. Zhang, "An MEF-based localization algorithm against outliers in wireless sensor networks," *Sensors*, vol. 16, no. 7, p. 1041, 2016.
[36] M. Shanthi and D. K. Anvekar, "Secure localization for underwater wireless sensor networks based on probabilistic approach," in *Proc. 2nd Int. Conf. Adv. Electron. Comput. Commun. (ICAECC)*, 2018, pp. 1–6.
[37] L. Mihaylova, D. Angelova, S. Honary, D. R. Bull, C. N. Canagarajah, and B. Ristic, "Mobility tracking in cellular networks using particle filtering," *IEEE Trans. Wireless Commun.*, vol. 6, no. 10, pp. 3589–3599, Oct. 2007.
[38] K. Lin, F. Xia, and G. Fortino, "Data-driven clustering for multimedia communication in Internet of vehicles," *Future Gener. Comput. Syst.*, vol. 94, pp. 610–619, May 2019.

Wang Pi received the B.S. degree from Peking University, Beijing, China, in 2019. He is currently pursuing the M.S. degree in computer science and engineering with the University of California at San Diego, San Diego, CA, USA.

His research interests include statistical signal processing techniques for wireless communications with a primary focus on cooperative tracking, data sequence analysis, and malicious user detection in the Internet of Vehicles.

Pengtao Yang (Student Member, IEEE) received the B.S. degree from Nankai University, Tianjin, China, in 2017. He is currently pursuing the M.S. degree in electrical and information engineering with Peking University, Beijing, China.

His research interests include data analytics and statistical signal processing of wireless communications with a primary focus on signal processing algorithm in the vehicular network, including cooperative localization, trajectory prediction, and signal error detection.



Dongliang Duan received the B.S. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2006, the M.S. degree in electrical engineering from the University of Florida, Gainesville, FL, USA, in 2009, and the Ph.D. degree in electrical engineering from Colorado State University, Fort Collins, CO, USA, in 2012.

He joined the Department of Electrical and Computer Engineering, University of Wyoming, Laramie, WY, USA, where he is currently an Associate Professor. His research interests include signal processing techniques for wireless communications and power systems, estimation and detection theory, and energy resource management in wireless communication systems. He is particularly interested in statistical signal processing and wireless communications and networking for power grid monitoring and control.

Chen Chen (Senior Member, IEEE) received the Ph.D. degree from Peking University, Beijing, China, in 2009.

He is currently an Associate Professor with Peking University. Since 2010, he has been the Principal Investigator of over 20 funded research projects. He has authored or coauthored over 100 journal and conference papers and four books. His current research interests include signal processing, and wireless communications and networking.

Dr. Chen was a recipient of two Outstanding Paper Awards from the Chinese Government of Beijing in 2013 and 2018, respectively, and the Best Paper Awards of IEEE ICNC'17, ICCS'18, and Globecom'18. He is currently an Associate Editor of *IET Communications*. He has served as the symposium co-chair, the session chair, and a member of the Technical Program Committee for several international conferences.

Xiang Cheng (Senior Member, IEEE) received the Ph.D. degree from Heriot-Watt University, Edinburgh, U.K., and the University of Edinburgh, Edinburgh, U.K., in 2009.

He is currently a Professor with Peking University. His general research interests are in areas of channel modeling, wireless communications, and data analytics, subject on which he has published more than 200 journal and conference papers, five books, and holds six patents.

Prof. Cheng was a recipient of the IEEE Asia–Pacific Outstanding Young Researcher Award in 2015; and the co-recipient of the 2016 IEEE JSAC Best Paper Award: Leonard G. Abraham Prize, the NSFC Outstanding Young Investigator Award, and the First-Rank and Second-Rank Award in Natural Science, Ministry of Education, in China. He has also received the Best Paper Awards at IEEE ITST'12, ICCC'13, ITSC'14, ICC'16, ICNC'17, GLOBECOM'18, ICCS'18, and ICC'19 and the Postgraduate Research Thesis Prize from the University of Edinburgh. He has served as the symposium leading chair, the co-chair, and a member of the Technical Program Committee for several international conferences. He is currently an Associate Editor of the the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE WIRELESS COMMUNICATIONS LETTERS, and the *Journal of Communications and Information Networks*, and is an IEEE Distinguished Lecturer.

Liuqing Yang (Fellow, IEEE) received the Ph.D. degree from the University of Minnesota, Minneapolis, MN, USA, in 2004.

Her main research interests include communications and signal processing.

Dr. Yang received the Office of Naval Research Young Investigator Program Award in 2007; the National Science Foundation Career Award in 2009; the IEEE GLOBECOM Outstanding Service Award in 2010; the George T. Abell Outstanding Mid-Career Faculty Award; the Art Corey Outstanding International Contributions Award from CSU in 2012 and 2016, respectively; and the Best Paper Awards at IEEE ICUWB'06, ICCC'13, ITSC'14, GLOBECOM'14, ICC'16, WCSP'16, GLOBECOM'18, ICCS'18, and ICC'19. She has been actively serving in the technical community, including the organization of many IEEE international conferences and on the editorial boards of a number of journals, including the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and the IEEE TRANSACTIONS ON SIGNAL PROCESSING. She is currently serving as the Editor-in-Chief for *IET Communications*.

Hang Li received the B.E. and M.S. degrees from Beihang University, Beijing, China, in 2008 and 2011, respectively, and the Ph.D. degree from Texas A&M University, College Station, TX, USA, in 2016.

He was a Postdoctoral Research Associate with Texas A&M University from September 2016 to August 2017 and the University of California–Davis, Davis, CA, USA, from September 2018 to March 2018. He was a Visiting Research Scholar with the Shenzhen Research Institute of Big Data, Shenzhen, China, from April 2018 to June 2019, where he has been a Research Scientist since June 2019. His current research interests include wireless networks, Internet of Things, stochastic optimization, and applications of machine learning.