Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 54-20 (2021) 638–643

Secure Connected and Automated Vehicles against False Data Injection Attack using Cloud-based Data Fusion

Chunheng Zhao * **Gurcan Comert** ** **Pierluigi Pisu** ***

* *Clemson University International Center for Automotive Research, Greenville, SC 29607 USA (e-mail: chunhez@clemson.edu)*
** *Benedict College, Columbia, SC 29204 USA (e-mail: Gurcan.Comert@benedict.edu)*
*** *Clemson University International Center for Automotive Research, Greenville, SC 29607 USA (e-mail: pisup@clemson.edu)*

Abstract: It has been shown that interdependency in connected and automated vehicles (CAV) can be potentially beneficial in several aspects, however, it also poses a set of specific challenges in concern of safety and reliability due to the possibility of cyber-attacks. In this paper, we present a data fusion-based methodology to detect the false data injection (FDI) attack on CAVs, and generate a flow of trustworthy information for every CAV. The effectiveness of the proposed approach is validated using microscopic traffic simulation, which shows that our proposed methodology is able to detect and isolate the false data injection attacks on CAVs.

Keywords: Connected and automated vehicles, Data fusion, Particle filter, Attack detection.

1. INTRODUCTION

CAVs using technologies of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure communications (V2I) have been a great focus. A lot of research for CAVs on platooning, intersection control, and similar intelligent transportation applications have been conducted (Wang et al., 2019; Guo et al., 2019). Although the interdependence in V2X can be potentially beneficial in several aspects (e.g., traffic management, reduction of fuel consumption), it also poses a set of specific challenges in safety and reliability, due to the possibility of cyber-attacks aimed at influencing the behavior of vehicles like false data injection, packet dropping, and forced network congestion (Mo and Sinopoli, 2010; Chowdhury et al., 2020; Dash et al., 2021).

Many defenses have been proposed considering different attacks on multiple CAV applications. As CAV is a typical application of cyber-physical system (CPS), control-based solutions have to be addressed to secure CAVs. Cardenas et al. (2009) showed that several drawbacks are presented if considering only the cyber side of the CPS, for example, software patching and frequent updates are not well suited for control systems. Possible risks related to different types of cyber-attacks on vehicle platoons via Cooperative Adaptive Cruise Control (CACC) applications have been illustrated in (Biron et al., 2018; Rayamajhi et al., 2018). Petrillo et al. (2020) leveraged an adaptive synchronization-based control algorithm to solve the problem of cyber-secure tracking for a platoon undergoing different kinds of cyber-threats.

For the other attacking scenarios, Zeng et al. (2017) proposed an attack model in road navigation scenarios, and a complete framework to analyze and evaluate the spoofing attacks was developed. Lin et al. (2018) adapted modeling and analysis of data integrity attacks to investigate security issues of route guidance schemes. Several defenses to secure CAV navigation systems have also been exploited (Kong and Jun, 2017; Luo et al., 2019).

Although there are a batch of defenses proposed, they are specifically designed for one CAV scenario and require the details of the corresponding cooperative controller. The scope of this paper is then developing a more general and scalable technique that doesn't depend on cooperative controller's information and can be applied in different CAV scenarios. In this study, the main goal is to propose a methodology to assess the trustworthiness of information exchanged by CAVs, thus, to achieve higher resilience to false data injection attacks. The main contributions of this paper are summarized as follows:

(1) We show that particle filter is good for cooperative localization and can improve the results significantly.
(2) We adopt optimal threshold selection in diagnostics and design an attack detection algorithm that can detect and isolate false data injection attacks.
(3) The proposed approach can be scaled to a number of cooperation-based applications.

The rest of this paper is organized as follows. Section 2 provides the problem statement and assumptions. Section 3 discusses the proposed attack detection approach. Section 4 presents numerical experiments on microscopic traffic simulation. Lastly, conclusions are presented in Section 5.

2. PROBLEM STATEMENT

Consider a routing scenario with vehicles traveling on a three-lane highway with multiple routes and intersections,

10.1016/j.ifacol.2021.11.243

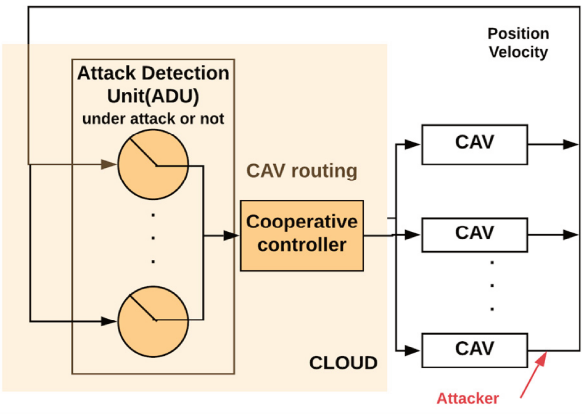


Fig. 1. Architecture of proposed cloud-based method

and the dynamic routing algorithms rely on accurate information from CAVs. CAVs publish their information as well as their surrounding information to the Cloud with a unique vehicle ID as shown in equation (1). It is assumed that only surrounding vehicles within a radius R from an ego CAV can be sensed. Thus, each CAV only publishes the information of itself and its neighboring CAVs.

Define S_i as the total information that a CAV i publishes to the cloud, X_i is the information about i itself and X_{ij} is the information about i and its neighboring vehicle j. All observation vectors can then be given in equation (1).

$$\begin{cases} S_i &= [X_i, X_{i1}, ..., X_{ij}]^T \\ X_i &= [x_i, y_i, \psi_i, v_i, a_i, \dot{\psi}_i]^T \\ X_{ij} &= [d_{ij}, v_{ij}]^T \end{cases} \tag{1}$$

where,

$i \in N_v$, set of ego CAVs
$j \in M_v$, set of neighboring vehicles
x_i, global longitudinal coordinate of vehicle i
y_i, global lateral coordinate of vehicle i
ψ_i, yaw angle of vehicle i
v_i, velocity of vehicle i
a_i, acceleration of vehicle i
$\dot{\psi}_i$, yaw rate of vehicle i
d_{ij}, relative distance between vehicle i and vehicle j
v_{ij}, relative velocity between vehicle i and vehicle j

The false data injection attack is assumed to be on the CAVs in the procedure of publishing information with Cloud as shown in Fig. 1. The published information can be injected with malicious data after the communication channels on the vehicles are compromised. Among all the published information, we consider position and velocity attack x_i, y_i, and v_i. Let I_k denote the original data at time k and $I \in \{x_i, y_i, v_i\}$. The equation of the sensor data I_k under FDI attack at time k can be described as

$$I_k^a = I_k + \Delta I_k \tag{2}$$

where ΔI_k is the malicious data injected by the attacker and I_k^a is the final attacked data.

3. METHODOLOGY

3.1 Cloud-based Solution

In the proposed method, Cloud will be responsible for gathering information from all the CAVs and generating a flow of trustworthy information for every CAV. The proposed structure requires the definition of a Cooperative Controller (CC) and an Attack Detection Unit (ADU) as shown in Fig. 1. The CC is a supervisory controller with enhanced performance relying on shared CAVs information to make decisions. In a navigation scenario, CC is a dynamic routing algorithm which uses real-time traffic data collected by CAVs for routes selection to reduce travel cost. However, when the shared information is malicious (i.e. false data injected), the ADU should detect it and prevent such data from being used by the Cooperative Controller. ADU should allow the shared information to be utilized again once the attack detection alert is cleared.

ADU will compare the published data of a single-vehicle with the estimated data from a cloud-based data collection and fusion system. Once a mismatch is identified, that vehicle should be elected as under attacks. ADU should be able to handle the task in real-time leveraging the powerful computational capability of cloud computing.

3.2 System Model

A simple steering and driving model that uses gyroscopes and accelerometers to find the vehicles' yaw rate and acceleration is considered to be the vehicle motion model. The current input of the system can be defined by a pose vector $u_t = [\dot{\psi} \quad a]$, where $\dot{\psi}$ and a are current yaw rate and acceleration respectively. The discrete-time state transition equation of the vehicle is shown in equation (3).

$$\begin{aligned} X_t &= f(u_t, X_{t-1}) \\ &= \begin{cases} x_t = x_{t-1} + v_t \cdot cos\psi \cdot \Delta t + \epsilon_{t1} \\ y_t = y_{t-1} + v_t \cdot sin\psi \cdot \Delta t + \epsilon_{t2} \\ v_t = v_{t-1} + a_t \cdot \Delta t + \epsilon_{t3} \\ \psi_t = \psi_{t-1} + \dot{\psi}_t \cdot \Delta t + \epsilon_{t4} \end{cases} \end{aligned} \tag{3}$$

where, $X_t = [x_t \quad y_t \quad v_t \quad \psi_t]^T$ is the state of the vehicle at time t, Δt is time step, and ϵ_{ti} $(i = 1, ... 4)$ is a set of random samples drawn from Gaussian distribution $\mathcal{N}(0, \sigma_a^2)$ representing system noise. Here, the standard deviations are $\sigma_{a1} = 0.4$, $\sigma_{a2} = 0.4$, $\sigma_{a3} = 0.01$, $\sigma_{a4} = 0.25$. These values are determined based on trial and error to make the model closer to reality.

For the CAV application, the localization information of ego vehicle can not only be directly obtained from on-board GPS, but also from neighbors' GPS data, the relative distance, and speed between ego vehicle and its neighbors. Lidar and radar can provide measurements about relative distance and speed. A vehicle and its neighbors at time t are represented by i_t and $N_t^{(i)}$, respectively. Assuming that $j \in N_t^{(i)}$, its estimation about the location and velocity of i is expressed by

$$\begin{cases} x_t^{(ji)} = x_t^{(j)} + d_t^{(ji)} cos(\gamma_t^{(ji)}) \\ y_t^{(ji)} = y_t^{(j)} + d_t^{(ji)} sin(\gamma_t^{(ji)}) \\ v_t^{(ji)} = v_t^{(j)} + s_t^{(ji)} sin(\gamma_t^{(ji)}) \end{cases} \tag{4}$$

where, $x_t^{(ji)}$, $y_t^{(ji)}$, and $v_t^{(ji)}$ are the estimation of i's location and velocity in the coordinate frame of j. $d_t^{(ji)}$, $s_t^{(ji)}$, and $\gamma_t^{(ji)}$ are relative distance, relative velocity, and angle between two vehicles at time t using lidar and radar, respectively. $x_t^{(j)}$, $y_t^{(j)}$, and $v_t^{(j)}$ are the estimation of i's

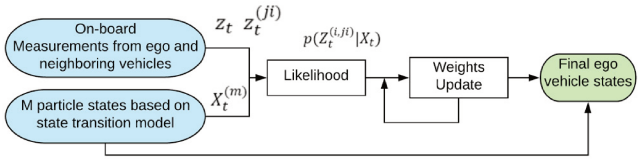


Fig. 2. Overall architecture of the proposed data fusion

neighbor j's location and velocity. False yaw angle attack is not considered as it is difficult to infer ego vehicle's yaw angle from neighboring vehicle's on-board sensors.

Full state observation is assumed which contains the measurement of the current position, speed, and yaw angle. Therefore, the onboard sensor readings are:

$$Z_t = [\tilde{x}_t \quad \tilde{y}_t \quad \tilde{v}_t \quad \tilde{\psi}_t]^T \tag{5}$$

The observation equations for this model are:

$$Z_t = g(X_t, n_t) = \begin{cases} \tilde{x}_t = x_t + n_t^{(1)} \\ \tilde{y}_t = y_t + n_t^{(2)} \\ \tilde{v}_t = v_t + n_t^{(3)} \\ \tilde{\psi}_t = \psi_t + n_t^{(4)} \end{cases} \tag{6}$$

where, n_t is a set of random samples drawn from $\mathcal{N}(0, \sigma_n^2)$ representing measurement noise. Note that for different states, measurement noises vary.

The routing algorithm for CAVs is adapted and simplified from the work by Tian et al. (2015). The criterion for the best route is the general travel cost. A route is a sequence of edges that describes a path through the network. For each of the edges in the network, the general cost of that edge i for period k is computed as a weighted sum of travel time T_i^k and travel distance d_i^k.

$$C_i^k = \alpha \cdot T_i^k + \beta \cdot d_i^k \tag{7}$$

where travel distance is determined by the geometry of the edges and travel time is computed depending on the traffic situation. The coefficients α and β can be defined by the user. During a simulation, travel times are measured for each edge in the network. All vehicles that leave the edge report the time they have spent on the edge. All travel times during one evaluation interval k are averaged and thus form the measured travel time for that edge. The general cost C_j^k for a route j is simply defined as the sum of the general costs C_i^k of all its edges i:

$$C_j^k = \sum_{i \in j} C_i^k \tag{8}$$

Then the route with minimum cost will be selected.

3.3 Particle Filter-based Data Fusion

Cooperative localization is introduced here in order to fuse information from multiple sources (i.e., neighboring CAVs) and obtain accurate localization information for an ego CAV, which will be utilized in the routing algorithm. The proposed data fusion scheme incorporates a particle filter with cloud communication. Compared with the Kalman filter, the particle filter is able to deal with non-Gaussian noises, which provides the scalability and flexibility to be applied to different scenarios. In this proposed method, the core idea is to use neighboring vehicles as additional measurements in the observation equation, which could be regarded as a multi-sensor architecture as shown in Fig. 2.

More specifically, each neighboring vehicle is able to provide an estimate of the ego vehicle's states using equation (9) which is based on equation (4).

$$Z_t^{(ji)} = \begin{cases} x_t^{(ji)} = x_t^{(j)} + d_t^{(ji)} cos(\gamma_t^{(ji)}) + a_{t1} \\ y_t^{(ji)} = y_t^{(j)} + d_t^{(ji)} sin(\gamma_t^{(ji)}) + a_{t2} \\ v_t^{(ji)} = v_t^{(j)} + s_t^{(ji)} sin(\gamma_t^{(ji)}) + a_{t3} \end{cases} \tag{9}$$

where $a_{ti} (i = 1, \ldots, 3)$ is a set of random samples drawn from $\mathcal{N}(0, \sigma_h^2)$ representing measurement noise. Therefore, each neighboring vehicle could be regarded as an additional "sensor" besides the ego vehicle's own onboard GPS. And the observation model is not based on measurement from one sensor but from multiple sensors. For time t, a set of measurements Z_t is provided by $j+1$ sensors:

$$Z_t = \{z_t^i\} \cup \{z_t^{(1i)}, ..., z_t^{(ji)}\} \tag{10}$$

where z_t^i represents the measurement from ego vehicle's sensor which is computed based on equation (6), $z_t^{(ji)}$ represents the measurements from neighboring vehicles' sensor which is based on equation (9) and j denotes the number of neighboring vehicles. Therefore, in total, there are $j+1$ sets of measurements. As the measurement sets of different sensors are independent, then the observation likelihood $p(Z_t|X_t)$ is computed as

$$\begin{aligned} p(Z_t|X_t) &= p(\{z_t^i\} \cup \{z_t^{(1i)}, \ldots, z_t^{(ji)}\}|X_t) \\ &= p(z_t^i|X_t) \prod_{j=1}^{N} p(z_t^{(ji)}|X_t) \end{aligned} \tag{11}$$

As we can see from equation (11), there are two parts in the total observation likelihood: $p(z_t^i|X_t)$ denotes the observation likelihood based on the vehicle's onboard sensor measurements and $p(z_t^{(ji)}|X_t)$ denotes the observation likelihood based on neighboring vehicles' measurements. The likelihoods can be computed using

$$\begin{cases} p(Z_t^{(i,ji)}|X_t) = \dfrac{exp\left(-\frac{1}{2} \cdot (e_t^{(i,ji)})^2\right)}{\sqrt{(2\pi)^N \prod_{n=1}^{N} \sigma_n(n)}} \\ e_t^{(i,ji)} = \sqrt{\sum_{n=1}^{N} \left(\dfrac{Z_t^{(i,ji)}(n) - X_t(n)}{\sigma_n(n)} \right)^2} \end{cases} \tag{12}$$

where $p(Z_t^{(i,ji)}|X_t)$ denotes either $p(z_t^i|X_t)$ or $p(z_t^{(ji)}|X_t)$; $e^{(i,ji)}$ denotes either e^i or $e^{(ji)}$, which is the corresponding normalized error between actual measurement (i.e., z_t^i or $z_t^{(ji)}$) and estimation X_t; N represents the number of states in observation equation or state transition equation. For X_t and Z_t, N is 4 as shown in equation (3) and equation (6); for Z_t^{ji}, N is 3 as shown in equation (9).

Then the weights for each particle m can be updated using equation (13) and the final output can be computed using equation (14). Note that in the routing scenario, every vehicle can be the ego vehicle. Thus, there is one particle filter running for each vehicle that aims to fuse its onboard sensor information with its neighboring vehicle information in the radius R.

$$\begin{cases} \omega_{t-1}^{(m)} &= \dfrac{\tilde{\omega}_{t-1}^{(m)}}{\sum_{m=1}^{M} \tilde{\omega}_{t-1}^{(m)}} \\ \tilde{\omega}_t^{(m)} &= \omega_{t-1}^{(m)} \cdot p(Z_t|X_t^{(m)}) \end{cases} \tag{13}$$

$$E[X_t] \simeq \sum_{m=1}^{M} \omega_t^{(m) \cdot X_t^{(m)}} \tag{14}$$

3.4 Attack Detection Scheme

The core idea of attack detection is the discrimination between normal data and attacked data, and we discriminate them by comparing the data content in this case. Normal data is modeled using the proposed data fusion method in Section 3.3. In order to identify if false data injected or not, the results of the data fusion algorithm for one ego vehicle are going to be compared with the information sent from neighboring vehicles as shown in Algorithm 1.

Algorithm 1 Decision Logic

for each $j \in M_v$(Number of neighboring vehicles) **do**
  if $E_t^j \geq E_{threshold}$ **then**
    j is publishing false information;
  end if
end for

In this logic, E_t^j shows the residue between estimated results after data fusion and information sent from neighboring vehicle j at time t. E_t^j is a vector which is defined in equation (15). As false yaw angle attack is not considered, there are only three states for E_t^j and $E_{threshold}$. E_t^j exceeds $E_{threshold}$ if at least one element in the vector E_t^j exceeds the corresponding value in matrix $E_{threshold}$.

$$E_t^j = \begin{bmatrix} e_x \\ e_y \\ e_v \end{bmatrix} \tag{15}$$

where, e_x, e_y, and e_v are the errors for x position, y position, and velocity, respectively.

False data is identified to be published from that vehicle to the cloud when the residue of that vehicle is larger than a threshold ($E_{threshold}$). However, due to the existence of Gaussian noises in the measurement data as shown in equation (6), we need to carefully determine the threshold so that we know to what extent the biased data can be regarded as under attack, and to what extent the biased data can be regarded as normal data with noises. Therefore, the optimal threshold selection method is used to determine the threshold by minimizing false alarms P_F and misdetection P_M as shown in equation (16).

$$\min_h (P_F + P_M) \tag{16}$$

where P_F and P_M are defined in equation (17), and h is the threshold.

$$P_F = \int_h^{+\infty} p_0(x)dx \quad P_M = \int_{-\infty}^{h} p_1(x)dx \tag{17}$$

Here, binary hypothesis testing is the basics of this optimal threshold selection. Probability density function (*pdf*) p_0 represents the residuals distribution at a normal condition H_0 (i.e., no attacks) while p_1 represents the residuals distribution when the system is under attack H_1. Then P_F refers to the probability that hypothesis H_1 is chosen when H_0 is true (*i.e.*, probability of a false alarm), and P_M refers to the probability that hypothesis H_0 is chosen when H_1 is true (*i.e.*, probability of a misdetection).

As one neighboring vehicle j could also be the neighbors of other ego vehicles, which means there are multiple fusion systems using information from vehicle j, the majority rule is used here for consensus decision making as shown in Algorithm 2. If one neighboring vehicle j is identified as publishing false information by a data fusion system, then all the fusion systems using information from j will report the detection result of j. If more than half of the ego vehicles in the circle of interest report j is publishing false information, then j is considered as under attack.

Algorithm 2 Detection Scheme

$f = 0$;
for each $i \in I$ (Number of vehicles can sense j) **do**
  if $E_t^j(i) \geq E_{threshold}$ **then**
    j is publishing false information according to i;
    $f = f + 1$;
  end if
end for
if $f \geq \frac{1}{2}I$ **then**
  j is under false data injection attack;
end if

If the vehicle j is considered as under attack for more than T seconds, the vehicle j is removed from the fusion system and stopped from being used in the CC. At this stage, the Cloud is still receiving information from vehicle j and evaluating its trustworthiness. If more than half of the fusion systems report j is not publishing false information for T seconds, j will be put back in the data fusion systems.

Both the proposed data fusion algorithm and attack detection scheme are independent of the implementation details of routing algorithms, which gives the possibility of adopting the proposed approach to other CAV applications.

The number of CAVs required to implement the proposed algorithm depends on the radius of the circle of interest, the CAV penetration rate, and the number of attackers (i.e., we assume multiple attackers in this case). We assume there is a dynamic relationship between them instead of a simple static minimal value. This problem is not in the scope of this paper and left for the future work.

4. SIMULATION AND RESULTS

4.1 Simulation Setup

The Cloud setup is based on Microsoft Azure and MATLAB. A Linux virtual machine (VM) is created in Microsoft Azure with MATLAB installed. On the local machine, VISSIM is used to simulate the routing traffic scenario and VISSIM-MATLAB co-simulation is used to stream real-time traffic and vehicle data. To upload data to VM, a UDP communication between two MATLAB sessions is established. In this scenario, the local machine acts as an onboard embedded computer on vehicles, and the VM acts as a kind of traffic management center.

Three links are created with the same origin and destination. The links are single direction and single lane. Lane changing behaviors are not considered in this paper. The simulation step is 0.2 second (s) and the routing algorithm is based on the one illustrated in section 3.2. Without attack, vehicles are able to select a suitable route with minimum travel cost. Route selection changes as vehicles move along the network based on travel distance and travel time. Travel time is related to traffic volume density and velocity, which is based on the position and speed of a group of vehicles. As the velocity of CAVs on road will

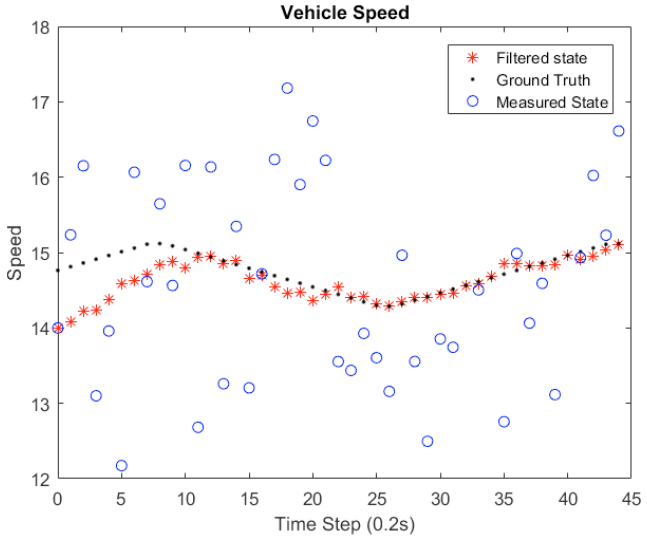


Fig. 3. Data fusion results for an ego vehicle with ID 17

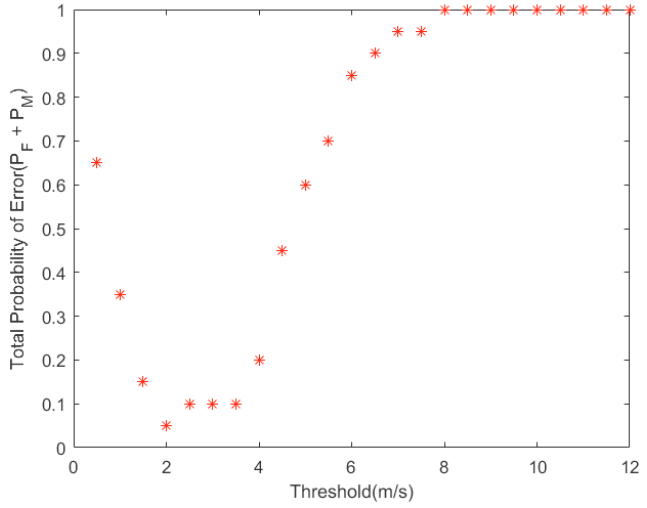


Fig. 4. Threshold selection for velocity attacks

affect the travel cost, false velocity attacks are performed in this scenario. Also, attacks on multiple vehicles are conducted as single malicious velocity data on the road is not able to affect the total travel time and travel cost a lot. More specifically, false velocity attacks are applied on 4 vehicles on the road where there are around 8 vehicles in total. The attacks are injected from 4 s to 8 s in the simulation and all the attacked vehicles have a malicious velocity which is 6 m/s lower than the original normal velocity. Therefore, fake congestion is created for that route. The CAV penetration rate is 90% for all runs.

4.2 Simulation Results

The estimated velocity for one ego vehicle are shown in Fig. 3. Each data point indicates the velocity at a single time step (with an interval of 0.2 s). As shown in Fig. 3, a good estimation of vehicle states can be obtained by the particle filter-based data fusion algorithm using the noisy sensor data and the speed error is reduced. The optimal threshold selection result is shown in Fig. 4. We ran the simulation multiple times with different thresholds for

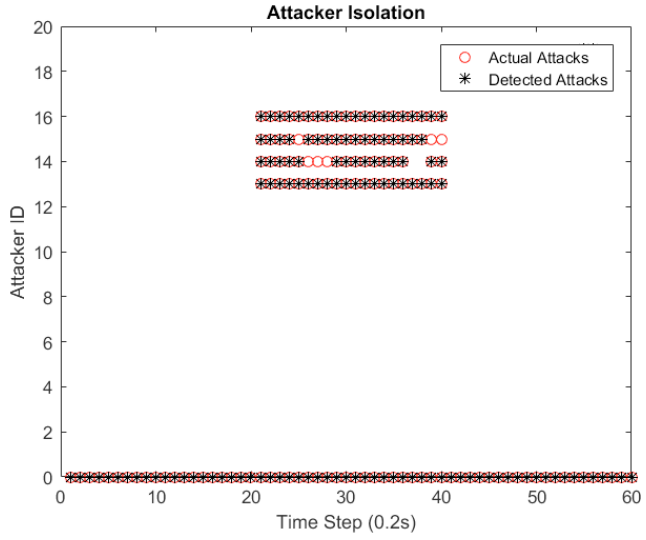


Fig. 5. Attackers isolation results

ADU, and the relationship between the total probability of error P_E and threshold h is shown in Fig. 4. P_E first decreases with the increase of h, and then P_E begins to increase after h reaches 4 m/s. Therefore, the threshold can be set to 2 m/s for false velocity attacks. Then the decision-making scheme using the selected h is shown to be able to identify the attackers which are vehicle identity (ID) 13, 14, 15, and 16 as shown in Fig. 5. False velocity attacks are injected from time step 20 to 40.

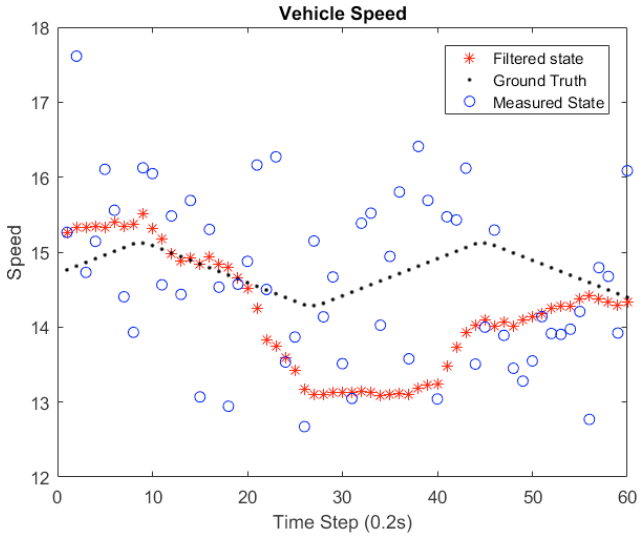
Fig. 6(a) shows the change of the fused speed for one ego vehicle due to the false data received from its neighbors. After adding the ADU using $h = 2$ selected from Fig. 4, the attack can be mitigated as shown in Fig. 6(b). The attacked vehicles will be removed from the fusion algorithm once they are detected, thus they won't affect the fusion results anymore. It is shown that those false data can affect the performance of the particle filter but the effect can be reduced with the proposed simple mitigation. We could see that this mitigation approach can generate a flow of trustworthy information for CAVs, which can ensure the safety and correct behavior of CAVs. A preliminary comparison is also conducted between our proposed algorithm and works in the literature as shown in Table 1. As shown in the table, our proposed method can achieve state-of-the-art performance with respect to the existing methods.

Table 1. Comparison with respect to the RMSE (Root Mean Squared Error)

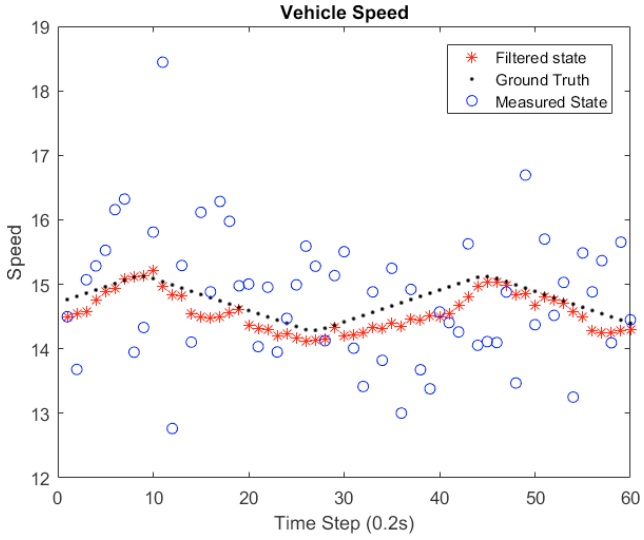
	Proposed Approach	***Kong and Jun (2017)***
Methods	Particle Filter	Kalman Filter
RMSE	0.315	≈ 0.5

5. CONCLUSION

In this paper, a data fusion-based attack detection method is proposed to mitigate false data injection attacks in CAV scenarios. Particle filters and cloud communication are integrated in order to fuse the location and speed information published from multiple vehicles, then the results of data fusion are evaluated by the ADU to detect



(a) Without attack mitigation



(b) With attack mitigation

Fig. 6. Comparison of using attack mitigation and no attack mitigation

attackers. The decision scheme leverages the knowledge of diagnostics and consensus decision-making. The detection capability of the proposed approaches on CAVs has been verified in the simulation. The results show that the proposed data fusion algorithm can improve the localization and speed estimation of vehicles, and the ADU is able to detect the vehicles sending false information. Modeling the communication channel delay in the CAV network and the behavior prediction of unconnected vehicles could be the future work. The investigation of other types of attacks (i.e., false acceleration, false yaw angle) using the data fusion method could also be a future research direction.

ACKNOWLEDGEMENTS

This study is partially supported by the Center for Connected Multimodal Mobility (C^2M^2) (U.S. DOT Tier 1 University Transportation Center) headquartered at Clemson University, Clemson, South Carolina. G. Comert was partially supported by U.S. Department of Homeland Security Summer Research Team Program Follow-On grant and NSF Grant No. 1719501.

REFERENCES

Biron, Z.A., Dey, S., and Pisu, P. (2018). Real-time detection and estimation of denial of service attack in connected vehicle systems. *IEEE Transactions on Intelligent Transportation Systems*, 19(12), 3893–3902.

Cardenas, A., Amin, S., Sinopoli, B., Giani, A., Perrig, A., Sastry, S., et al. (2009). Challenges for securing cyber physical systems. In *Workshop on future directions in cyber-physical systems security*, volume 5. Citeseer.

Chowdhury, M., Islam, M., and Khan, Z. (2020). Security of connected and automated vehicles. *arXiv preprint arXiv:2012.13464*.

Dash, P., Karimibiuki, M., and Pattabiraman, K. (2021). Stealthy attacks against robotic vehicles protected by control-based intrusion detection techniques. *Digital Threats: Research and Practice*, 2(1), 1–25.

Guo, Q., Li, L., and Ban, X.J. (2019). Urban traffic signal control with connected and automated vehicles: A survey. *Transportation research part C: emerging technologies*, 101, 313–334.

Kong, S.H. and Jun, S.Y. (2017). Cooperative positioning technique with decentralized malicious vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 19(3), 826–838.

Lin, J., Yu, W., Zhang, N., Yang, X., and Ge, L. (2018). Data integrity attacks against dynamic route guidance in transportation-based cyber-physical systems: Modeling, analysis, and defense. *IEEE Transactions on Vehicular Technology*, 67(9), 8738–8753.

Luo, Q., Cao, Y., Liu, J., and Benslimane, A. (2019). Localization and navigation in autonomous driving: Threats and countermeasures. *IEEE Wireless Communications*, 26(4), 38–45.

Mo, Y. and Sinopoli, B. (2010). False data injection attacks in control systems. In *Preprints of the 1st workshop on Secure Control Systems*, 1–6.

Petrillo, A., Pescape, A., and Santini, S. (2020). A secure adaptive control for cooperative driving of autonomous connected vehicles in the presence of heterogeneous communication delays and cyberattacks. *IEEE transactions on cybernetics*.

Rayamajhi, A., Biron, Z.A., Merco, R., Pisu, P., Westall, J.M., and Martin, J. (2018). The impact of dedicated short range communication on cooperative adaptive cruise control. In *2018 IEEE International Conference on Communications (ICC)*, 1–7. IEEE.

Tian, D., Yuan, Y., Qi, H., Lu, Y., Wang, Y., Xia, H., and He, A. (2015). A dynamic travel time estimation model based on connected vehicles. *Mathematical Problems in Engineering*, 2015.

Wang, Z., Bian, Y., Shladover, S.E., Wu, G., Li, S.E., and Barth, M.J. (2019). A survey on cooperative longitudinal motion control of multiple connected and automated vehicles. *IEEE Intelligent Transportation Systems Magazine*, 12(1), 4–24.

Zeng, K.C., Shu, Y., Liu, S., Dou, Y., and Yang, Y. (2017). A practical gps location spoofing attack in road navigation scenario. In *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, 85–90.