AVMon: Securing Autonomous Vehicles by Learning Control Invariants and Residual Prediction

Ahmed Abdo
University of California, Riverside
aabdo003@ucr.edu

Sakib Md Bin Malek
University of California, Riverside; eBay Inc.
sbin003@ucr.edu

Xuanpeng Zhao
University of California, Riverside
xzhao094@ucr.edu

Nael Abu-Ghazaleh
University of California, Riverside
naelag@ucr.edu

***Abstract*—Autonomous systems are vulnerable to physical attacks that manipulate their sensors through spoofing or other adversarial inputs or interference. If the sensors' values are incorrect, an autonomous system can be directed to malfunction or even controlled to perform an adversary-chosen action, making this a critical threat to the success of these systems. To counter these attacks, a number of prior defenses were proposed that compare the collected sensor values to those predicted by a physics based model of the vehicle dynamics; these solutions can be limited by the accuracy of this prediction which can leave room for an attacker to operate without being detected. We propose AVMon, which contributes a new detector that substantially improves detection accuracy, using the following ideas: (1) Training and specialization of an estimation filter configuration to the vehicle and environment dynamics; (2) Efficiently overcoming errors due to non-linearities, and capturing some effects outside the physics model, using a residual machine learning estimator; and (3) A change detection algorithm for keeping track of the behavior of the sensors to enable more accurate filtering of transients. These ideas together enable both efficient and high accuracy estimation of the physical state of the vehicle, which substantially shrinks the attacker's opportunity to manipulate the sensor data without detection. We show that AVMon can detect a wide range of attacks, with low overhead compatible with real-time implementations. We demonstrate AVMon for both ground vehicles (using an RC Car testbed) and for aerial drones (using hardware in the loop simulator), as well as in simulations.**

I. Introduction

Autonomous Vehicles (AVs) [43] are capable of operating without the presence of a human controller (see Fig. 1). The National Highway Traffic Safety Administration (NHTSA) defines AVs as "those in which at least one aspect of safety-critical control function occurs without direct driver input" [3]. AVs include aerial, ground, and marine vehicles that are forecast to become an integral part of our life [48]. For example, the unmanned aerial vehicles market is already estimated at USD 19.3 billion in 2019 and projected to reach USD 45.8 billion by 2025 with applications ranging from agricultural management to aerial mapping and freight transportation [25].

We consider a threat model where an adversary compromises the sensors of a victim autonomous vehicle [11], [44]. Compromising or tampering with any of these sensors may destabilize an AV or even allow attackers to cause damage to the system and even injuries to people using it or in its vicinity. For instance, an attacker may use GPS spoofing [29], by leveraging a nearby radio transmitter to create malicious GPS signals, leading to wrong location or velocity estimates. Moreover, transducers are components that are responsible for converting physical signals into digital measurements. A transduction attack leverages the limitations of the physical processes of a transducer to cause measurement errors to the attacker's advantage. For example, sound waves can affect accelerometers and make them report incorrect values [57]. Manipulating the sensor readings can cause the vehicle controller to react in an erroneous way leading to safety and other concerns.

Conventional security approaches such as software security, memory protection, authentication, and cryptography fall short in safeguarding AVs from real-world physical threats. Physics-Based Attack Detection (PBAD) [21] offers a promising solution, modeling vehicle dynamics to predict future states and capturing relationships between user inputs, control actions, and system states. Anomalous sensor inputs are identified by deviations from the predicted model state. Recent defenses, like [11], [44], use Kalman Filter (KF) to predict future physical states and detect attacks by comparing them with sensor readings. The accuracy of the detector's estimate is crucial for detection performance. Models with inaccurate estimates require looser detection thresholds to avoid false positives, and therefore provide an attacker with greater opportunity to manipulate sensor data without being detected. We show that current PBADs have some limitations that impact their accuracy, enabling the development of attacks that remain below the PBAD's detection thresholds.

We propose AVMon a new framework for protection against sensor manipulation attacks. AVMon improves the detection performance relative to prior PBADs using three new ideas. First, it optimizes the parameters of the state estimator to tune the model to the vehicle dynamics and operating environment, leading to more accurate KF. The second op-

Symposium on Vehicles Security and Privacy (VehicleSec) 2024
26 February 2024, San Diego, CA, USA
ISBN 979-8-9894372-7-6
https://dx.doi.org/10.14722/vehiclesec.2024.23029
www.ndss-symposium.org

portunity arises due to the limitations of KF, even when using the Extended Kalman Filter (EKF) which accounts for non-linear dynamics. Due to the step-wise linear extrapolation, the EKF algorithm can experience inaccuracies in estimation under highly dynamic situations where the linear assumptions do not hold [7]. In parameter estimation, more advanced non-linear filters are used when high accuracy estimation is desired (e.g., Particle Filters [22], etc.). However, these filters are computationally expensive making it difficult to use them in a real-time setting. As a result, KF generates inaccurate predictions for trajectories with dynamic behavior and discontinuities (e.g., around turns and changes of direction). To reduce this error, AVMON incorporates a residual estimation machine learning model. The model compensates for errors resulting from the nonlinear dynamics of the system as well as external disturbances that are difficult to model. Residual learning model closes the gap between KF and advanced non-linear filters, and captures effects in the training data not captured by the physical model. Finally, we leverage a change detection model that analyzes the sensor data to discriminate true changes of state from transients, reducing false positives, and improving detection accuracy.

We study the performance of AVMON using a number of testbed and simulation studies. We use both a programmable Remote Controlled (RC) vehicle testbed and a hardware-in-the-loop Unmanned Aerial Vehicle (UAV) testbed to study how AVMON performs compared to recent defenses [44], [14]. In addition, we evaluate the system in different attack scenarios, and profile the code execution on the testbeds and show the performance overhead is low, making AVMON practical to deploy. Overall AVMON substantially improves detection performance relative to Savior [44] and PID-Piper [14]. It improves both accuracy and Time-To-Detection (TTD), which also limits the opportunity for the attackers to compromise the system.

In summary, the paper makes the following contributions.

- We propose AVMON which uses three new techniques to improve the performance of physics-based attack detection models. AVMON leverages a residual-based machine learning model to efficiently incorporate non-linear effects and unmodeled perturbations into the KF estimate. It also optimizes the KF parameters to tune them to the vehicle and environment and uses a change detection model to reduce false positives.
- We demonstrate attacks against AVs that can bypass previous defenses, enabling an attacker to impact AVs' operation substantially (including attacks where we crash a drone before the attack is detected). We show that AVMON with its higher detection accuracy, substantially reduces the threat surface available to attackers.
- We evaluate AVMON under different conditions using both simulation and testbeds and for both ground and aerial vehicles. We made the source code for the testbed implementation openly available at https://github.com/avmon/avmon_proj.git.

II. Sensor Modalities and Threat Model

This section provides some background on common sensor modalities in AVs and their use in constructing state-space

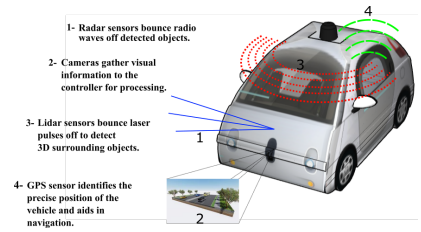


Fig. 1: General overview of the autonomous vehicle.

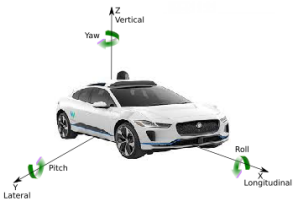


Fig. 2: Vehicle motion axes

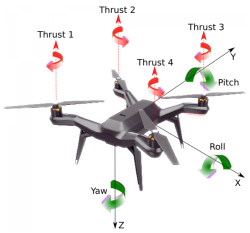


Fig. 3: Quadcopter motion axes and controls

estimation. We then present the threat model we assume in this paper. AVs [54] typically include a mix of sensors such as radar, camera, ultrasonic, Light Detection and Ranging (LiDaR), Inertial Measurement Units (IMUs), etc. Radar sensors monitor the position of neighboring vehicles. Video cameras are used to detect traffic lights, road signs and track other vehicles. Ultrasonic sensors detect curbs and other vehicles during parking maneuvers. LiDAR sensors detect road edges and identify lane markings. IMUs [15] measure a vehicle's angular rate and force by combining a 3-axis linear accelerometer and gyroscope to track a vehicle within six axes of motion. Specifically, IMU traces both linear (X, Y, and Z) and rotational components: (1) pitch, rotating a vehicle upwards or downwards; (2) roll: rotating, the vehicle sideways; and (3) yaw: rotating, the orientation of the vehicle. These six axes allow the full vehicle position and orientation to be tracked in real-time. These sensor streams are processed by software modules that use them to generate and adapt trajectory paths, which are effectuated by sending control signals to the vehicle's actuators to control acceleration, braking, and steering. To visualize these axes, the inertial frames of a ground vehicle and a quadcopter are shown in Figures 2 and 3 respectively.

Threat Model: We consider attacks where one or more of the sensors on an AV are interfered with by an attacker either directly or indirectly (e.g., using transduction attacks) [43]. Since these sensors are critical to the AV's estimation of its behavior and that of the environment, compromising sensors may lead to erroneous estimates of its operating state, leading to control actions from the AV that serve an attacker's goal. For example, the attacker may exploit the vehicle to cause property damage, block emergency traffic, or cause accidents and bodily injury. Prior work has shown that a range of common sensors are vulnerable to attacks including those that target IMUs [53], [58], RADAR sensors [34], LiDAR [50], ultrasonic sensors [34], camera sensors [34], [42], GPS signals [39], etc.

The required attacker access for successfully launching these attacks varies substantially across attacks and in ways that are specific to the target sensors and their implementations. For example, GPS signals [43] do not contain authentication information and are susceptible to spoofing attacks.

Conversely, LiDAR [9] is used for measuring distances to surrounding obstacles using infrared lasers, can provide 360° viewing angles, and generate 3-dimensional representations of the road environment. A LiDAR spoofing attack can be performed by replaying the LiDAR laser pulses from different positions to create fake points further than the location of the spoofer [42]. Our threat model is similar to previous research in physical attack detection, where we assume an adversary that can inject false signals in one or more of the sensors used by AVs at a time. Our defense can also be used to predict and monitor actuator commands that are potentially controllable by an attacker, although we do not demonstrate this in this paper.

We assume that the attacker cannot compromise or bypass our invariant-checking module, which can potentially be supported using hardware trusted execution environments [12], [31]. The primary strategy of these attacks is to inject a time series of biased attack values so that $y^a = y + bias$, where y^a is selected to harm the system. We also consider sophisticated attacks, where the attacker employs different strategies to evade detection; for example, an attacker may generate an adversarial example where y^a is selected to cause small incremental drift in the state of the AV. We do not consider the actions to be taken after an attack is detected; this is a difficult problem and context sensitive decision that can be handled in some scenarios by dropping to a safe operating mode or alerting the operator.

III. System invariants and attack detection

We next provide a brief review of the autonomous system's dynamics, including aerial and ground vehicles, which serve as the basis of the models used within AVMon. We then show how these models are used as part of a physics-based attack detection defense.

A. Physics-based System invariants

Aerial vehicles, such as quadcopters [8], control movement through four rotors. Motor pairs generate opposing torques for equilibrium, allowing constant heading during hovering. Yaw control adjusts motor pairs for a counter torque, and altitude is managed by equal thrust changes. Lateral movement is achieved by varying relative motor speeds. A quadcopter can move in six degrees of freedom; longitudinally (forward and backward), vertically (up and down), and laterally (right and left), by controlling the differential thrusts to the rotors. It can also move rotationally among each axis to produce roll, pitch, and yaw movements. The basic quadrotor parameters that depict Euler angles [26] including roll, pitch, yaw, and body coordinate frame, can be shown in Fig. 3. The model for a four-wheel vehicle is also well studied [4], [41]. This model has two degrees of freedom that are represented by the vehicle's lateral position and the yaw angle.

B. Physics-based attack detection

Monitoring the physics of cyber-physical systems [47] to capture sensor attacks is a growing area of research. Our contribution is to substantially improve these predictions for AVs by improving the extended Kalman Filter estimator by learning its optimal configuration, leveraging residual learning to compensate for nonlinear dynamics, and using context information to filter out transients and reduce false positives. Physics-based attack detection can be thought of as a security monitoring system that creates a time-series prediction model of sensor readings for the autonomous system and identifies anomalies as deviations between the predicted and actual sensor readings. Thus, such a framework consists of 1) Physical model prediction and 2) Anomaly detection. A physical vehicular system can predict the expected future measurements using a state-space representation that describes the physical system as a set of inputs, outputs, and state variables. In general, the control invariants model can be represented as follows [35]:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1}$$

where $x(t)$ is the state variables, $u(t)$ is the system inputs and $y(t)$ is the system outputs. Equations 1 determine the next state and output of the system based on the current state and control signals. Specifically, A, B, C, and D are matrices modeling the state and inputs of the system as follows: A represents the time-invariant dynamic state matrix; B, the time-invariant input matrix; C, the time-invariant measurement matrix; and D, the time-invariant feedforward matrix.

IV. AVMon design Overview

Fig. 4 shows AVMon design. It consists mainly of sensors pre-processing, prediction process, residual learning, and anomaly detection function. These components work together in a real-time/online manner to achieve substantially higher attack detection accuracy by improving the physics-based prediction and anomaly detection components. In our work, we use the Extended Kalman Filter (EKF) that was designed for nonlinear system estimation and filtration. AVMon starts by receiving the sensor data as input and uses it to predict the next state in the prediction model. Finally, the anomaly detection algorithm compares the predictions to the sensor values to detect attacks.

AVMon improves the prediction using two ideas: (1) it uses an optimization algorithm that is executed offline to configure the primary EKF module to operate more accurately with respect to the AV parameters; and (2) it uses residual learning to compensate for the nonlinear dynamics of the model that are not captured effectively by the EKF. Finally, in the anomaly detection process, a time series of residuals r_k, i.e., the difference between the received sensor measurement y_k and the predicted or expected measurement $\hat{y_k}$, is used to detect unusual deviations and raise an alarm if the sensor values are sufficiently different from the predicted values. We improve anomaly detection by using a change-aware model instead of just looking for deviations between predictions and sensor data to monitor the sensor data for self-consistency over time and reduce false positives. Fig. 5 shows the offline optimization or training components.

Data Collection and Preprocessing: We collect the vehicle operation profile data, including input states and sensor values such as velocity, acceleration, latitude, and longitude. Simultaneously, we preprocess the data, converting GPS readings to flat-Earth coordinates (X, Y, and Z) and using Inertial Measurement Unit (IMU) data to calculate orientation parameters (roll, pitch, and yaw angles).

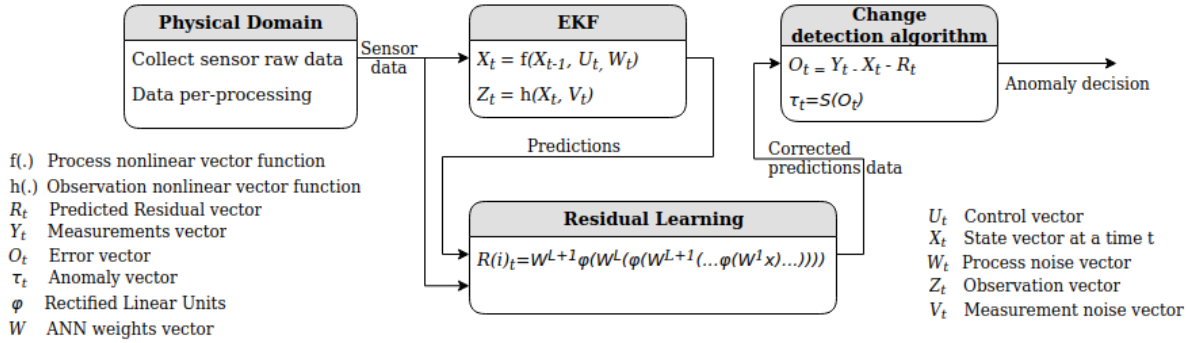


Fig. 4: AVMON system overview.

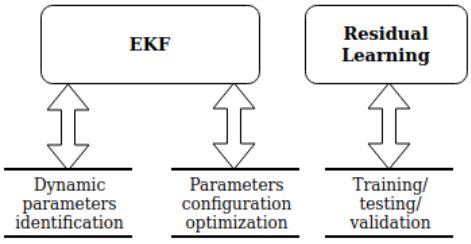


Fig. 5: AVMON offline learning and optimizing.

A. Model Parameter Optimization

AVMON predicts the AV states using an Extended Kalman Filter (EKF) [27]. EKF is a lightweight algorithm that does not require historical data, using only the previous state information to predict the next state. EKF is capable of modeling nonlinear estimation problems but uses piece-wise linearization between estimation steps. EKF uses Bayesian inference to provide an estimate of the joint probability distribution over the variables for each time frame.

We take sensory measurements and previously estimated outputs as the inputs to the EKF model to predict the following sensor states. The model has two procedures; (1) prediction and (2) correction. The first component takes the last sensors' values estimation and the current sensor readings to generate predicted sensor values for the next time step. However, these predicted values are refined to account for the nonlinear nature of the estimation process. Specifically, the covariance matrix of the estimation error (i.e., the error between the actual measurements and the predicted states) and the state transition matrix (encapsulating the equations for the vehicle dynamics) are used to obtain the predicted states.

The second component is the correction procedure: this component uses the previous sensors' values predictions, the observation matrix (i.e., a transformation matrix that transforms the AV system from the physical state space to measurement space), and the covariance of the measurement noise to compute the Kalman gain. The Kalman gain is defined as the ratio of the uncertainty in a predicted state to the uncertainty in the predicted state in addition to the uncertainty in measurement readings or message data. As a result, we get the sensor value predictions that are corrected using the measurement and covariance updated matrices. The outputs of this procedure will be used in the Residual Learning module and will feed the next iteration of this algorithm.

To define the dynamics and control algorithm to be used in the EKF for prediction, we use System Identification (SI) [52] to extract the AV control invariants and equations that describe how the vehicle behaves given the control objectives (e.g., a reference position) and the current states. We use a MATLAB built-in function [1] to derive such equations through regression over a set of collected traces of vehicle operation.

Since EKF can generate errors in predicting the dynamic behavior of many systems due to the poor tuning of some of its parameters, such as the covariance matrices Q and R respectively, these parameters have to be tuned to improve the model performance. Thus, we use a Genetic Algorithm (GA) to tune these parameters based on measurement data. Note that this is an offline procedure as shown in Fig. 5. GA [40] is a method for solving both constrained and unconstrained optimization problems that are inspired by the Human genetic process of passing genes from one generation to another. GA is a tool to aid in the system identification process as it is used to compute the optimal coefficients. We model the AV dynamics prediction competence by evaluating its accuracy based on its coefficients. The specific coefficients corresponding to the EKF model are represented as a typically binary array, which can be viewed as a chromosome carrying genetic information about the individual (candidate set of parameters for the EKF model in our case). Thus, GA starts from a widely dispersed initial population of coefficients setups for the EKF model design and converges to the best coefficients' estimation. Note that other non-linear optimization approaches could also be used.

B. Residual Learning

Since EKF approximates a nonlinear physical system (e.g., the quadcopter) using a piece-wise linear process, the prediction suffers large inaccuracies and filter instability in highly dynamic scenarios [60]. The prediction can be inaccurate in terms of the individual predicted values and the error can accumulate causing substantial divergence from the real-time state. When an AV vehicle changes its trajectory slowly, linear extrapolation between steps is effective and prediction errors are low. However, in more dynamic scenarios, such as sharp change of direction or start-stop behavior, the model will experience errors that can accumulate. For more effective detection, we need to minimize the generated error between the predicted and observed measurements.

One approach uses a more complex filter that models the nonlinear behavior between steps, but such complex filters rapidly become computationally prohibitive, especially for embedded controllers [6]. Moreover, they may not account

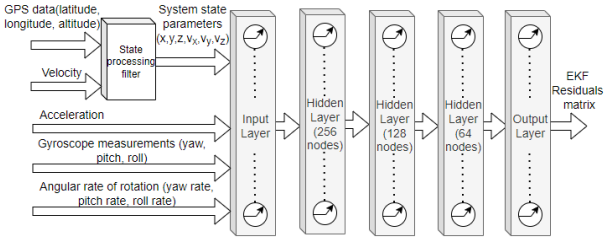


Fig. 6: General structure of AVMON residual learning neural network module.

for sensors' noise and other perturbations that arise in the real world. Thus, we approach this problem using machine learning to predict the residual sensor values (the expected deviation between the EKF prediction and the measurements). To learn the residual dynamics, we use a neural network model shown in Fig. 6. The model uses one input layer and three hidden layers, with 256, 128, and 64 neurons respectively, with rectified linear (ReLu) activation function. We used this architecture consistent with recommendations for the type of problem that recommend 3–5 hidden layers for this type and size of problem [19].

The inputs consist of the sensors readings such as the location and orientation of the AV. Having a large first hidden layer and following it up with smaller layers will lead to better performance as the first layer can learn a lot of lower-level features that can feed into a few higher order features in the subsequent layers. The output layer represents the prediction errors that can be used to validate the outputs coming from the EKF equations. Identifying the set of features that are most discriminative with respect to identifying anomalies from the large set of features (e.g., roll speed, pitch rate, yaw rate, altitude) can be challenging. We use a Sequential Forward Selection (SFS) algorithm, shown in Algorithm 1 [24], to reduce the feature space. SFS starts from an empty set and continues to add features based on their impact on the residual prediction model until a subset of the most salient features of size K is reached.

Discussion and Limitations: It is natural to ask why would a learning module be able to capture non-linear dynamics that are not captured by EKF. We believe that the key observation is that AVs have predictable behavior that may be learned. While EKF and other filters extrapolate based on the current state and the underlying model, they cannot predict likely control actions and operating contexts. The model also learns external disturbances such as impact of weather to the extent that it is exposed to them in the training data. A limitation of our approach is that the residual accuracy may drop if there is concept drift (e.g., completely novel operating conditions or regimes). We believe that the current generation of AVs is likely to be targeted towards a known subset of environments, with sufficient training data to reduce this risk. Moreover, there is a rich and growing body of work for learning and operation in the presence of concept drift [20]. We can leverage such techniques to incorporate continuous learning, or alternatively concept drift detection and model selection. It is also possible to increase the number of sensors (e.g., incorporating anemometers (wind sensors) and extending the model or the residual learning module to account for them. We hope to explore these directions in our future work.

C. Online Anomaly and attack detection

To mitigate false positives from transient states and sensor noise, we implement a change detection algorithm [55]. Instead of relying solely on the instantaneous error between prediction states and sensor observations, this algorithm examines the self-consistency of predicted data over time. In our anomaly detection method, pre-processed sensor readings $I(k)$ generate predicted sensor values $Y(k+1)$ using the EKF algorithm. We then update the predicted residual $e(K)$ using a neural network to compute the final residuals, expressed as:

$$r_i(k) = I(k) - Y(k) - e(K). \tag{2}$$

To address transient errors caused by PID control algorithm, we employ a statistical detection test based on sequentially discounting autoregression time series modeling (SDAR) [46]. SDAR discounts older data values, prioritizing recent ones, making it effective for online change point detection. This approach outperforms many detection algorithms, enabling early detection of malicious values and minimizing false positives from transient errors.

In AVMON, the anomaly detector algorithm, SDAR, keeps track of the historical changes of the residuals instead of a fixed time window to prevent an attacker from hiding their attack between time windows. In each iteration, new residuals data arrives each time frame with $(t = k+1, k+2, ...)$. Here, we define a parameter S that represents the anomaly score value and is calculated as:

$$S_k = \hat{\Theta}_i(S_{k-i} - \hat{\mu}) + \hat{\mu} \tag{3}$$

where variances-covariances vector $\hat{\mu}$ characterizes the variance and covariances among residuals data, and Θ represents the weights assigned to the past residuals at different time lags when calculating the current anomaly score. For example, if your anomaly detector is sensitive to recent changes and less influenced by distant past observations, you might have

Algorithm 1 Sequential Forward Selection Algorithm

Parameters:
$S \rightarrow$ a whole d-dimensional feature set as input
$e_k \rightarrow$ total error when using selected features.
Result:
$X_K \rightarrow \{x_j \mid j = 1, 2, ..., k; x_j \in S\},\ k = (0, 1, 2.., d)$
▷ a list of the critical features (R) used for our ANN model

Require:
$S \leftarrow \{s_1, s_2, ..., s_d\}$
$e \leftarrow \{\infty\}$
Initialization:
$X_0 \leftarrow 0$
$k \leftarrow 0$
while $k < K$ **do**
  $e^+ \leftarrow argmin_e(e, e(X_k + x^+)) < e,\ where\ x^+ \in S - X_k$
  $k \leftarrow k + 1$
  if $e^+ < e$ **then**
    $X_k \leftarrow X_k + x^+$
    $S \leftarrow S - x^+$
  end if
end while

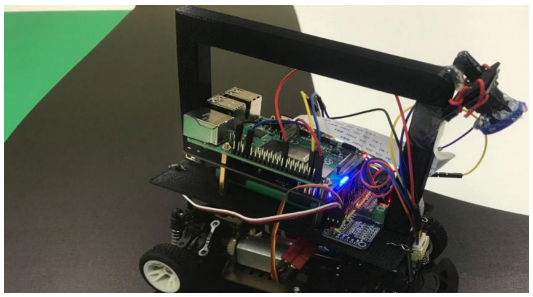
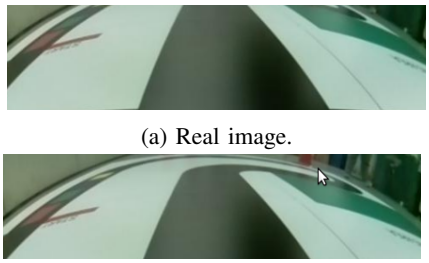


Fig. 7: Front wheel testbed vehicle.



(a) Real image.

(b) Injected image.

Fig. 8: A visual attack on a ground vehicle.

a diagonal matrix where the main diagonal contains higher weights for recent lags and lower weights for older lags. It is essential to note that the optimal values for these parameters often require experimentation and tuning based on a specific application and dataset. Finally, once S exceeds a predefined threshold, an anomaly alarm will be triggered.

V. Performance Evaluation

Implementation: We implemented our approach using CARLA simulator [16] which is an open urban driving simulator to support the development, training, and validation of autonomous urban driving systems. Then, our approach was applied to two different AVs (aerial and ground). In CARLA, the vehicle becomes autonomous through the subdivision of driving tasks into perception, planning, and continuous control. The perception stack utilizes semantic segmentation with the RefineNet [33] model, estimates lanes, road limits, and hazards. A state machine-based local planner manages driving states, and continuous control employs a PID controller for steering, throttle, and brake based on current position and waypoints. AVMon program implementing the system algorithm uses actors in the simulation and their interactions with sensors via designated APIs.

For the aerial AV, we utilized Dronecode's PX4 autopilot on dedicated hardware, Pixhawk 4 [2], powered by a 32-bit ARM Cortex M7 processor. The Pixhawk board includes sensors like accelerometer, gyroscope, magnetometer, and barometer. The simulation employs Gazebosim [30], and mission control inputs are managed through QGroundControl. Additionally, a Python module representing AVMon is implemented on a Raspberry Pi 3, acting as a flight controller for the PX4. Our ground vehicle, based on the AVWLtoys A242 model as shown in Fig. 7, is Raspberry Pi 4-powered with an HD camera and motor drive controller for autonomy. Motors are controlled through PWM signals from GPIO, using on-board battery power. The vehicle features infrared speed encoders, a TOF LiDAR (VL53L1X), and operates on the Robot Operating System (ROS). Our AVMon is implemented as a ROS node, receiving sensor measurements and showcasing adaptability to diverse autonomous vehicles despite differences in invariants, real-time requirements, and environments.

Experimental Setup: We evaluate AVMon on PX4 autopilot running on Pixhawk 4 and Gazebosim for the aerial vehicle and ROS Kinetic Kame[56] running on Raspberry Pi 4 for the ground vehicle. We also use the CARLA autonomous vehicle simulator that runs on Windows 10 64-bit.

Our experiments are based on maps designed in the CARLA simulator for training, testing, and validating. For real experiments, we performed missions to obtain real data sets containing information obtained from different trajectory scenarios, including simple (i.e., low curvature) and complex (i.e., increased curvature) ones with varying settings of velocity. We separate the dataset into separate training and testing data sets. To avoid overfitting, we use k-fold cross-validation and apply random drop-out to regularize the network during training. We also tried L2 regularization [38] as an alternative to random drop-out, which resulted in slightly lower test accuracy.

Attack benchmarks: The attacks manipulate directly sensor data at the interface between the control code and sensor modules. This approach permits various malicious interferences, including publishing false sensor data to compromise inertial and GPS sensors, control signal spoofing affecting steering and the motor pulse width modulation (PWM) signals, and parameter corruption by modifying control parameters (e.g., PID control coefficients) at run time. As an example, an attack node replays a chosen image at a higher rate than the camera, compromising visual data and influencing steering decisions, as represented in Fig. 8.

Creating simple attacks by adding bias values stochastically to sensors can introduce significant errors – these attacks are straightforward to detect once the errors exceed the thresholds of the detectors. Consequently, we also consider an advanced adaptive attack where the attacker is aware of the internals of this detector and attempts to carry out attacks such that they fool the detector eventually (so called Frog Boiling attack [10], [13]). Specifically, the error at any point between observed values V_o and estimated output V_e has to be larger than a predefined threshold τ to raise an anomaly alarm. the attacker triggers stealthy attacks in a controlled manner where $|V_e - V_o|$ never exceeds τ because the error remains under the threshold τ at any time, the attack should remain undetected. We try the attacks across all of our scenarios and for both ground and aerial AVs to test the defenses across a range of operating conditions.

A. AVMon *characterization*

In the first set of experiments, the impact of the individual components of our solution is evaluated as they together track closed loop and straight line trajectories. The average and maximum errors of each component are shown in Table I. In a straight line track, using only the default EKF estimator after configuring its relevant parameters can cause an error up to almost 9 meters and an average of 1.77 meters, especially during the turn maneuver for the vehicle. After using GA optimization to configure the EKF, the prediction error becomes significantly lower. However, it experiences spikes of

	Cyclic track		Straight track	
Components	Avg	Max	Avg	Max
EKF	1.77	8.92	0.08	0.16
EKF+GA	1.06	6.46	0.042	0.08
EKF+GA+RL	0.16	0.49	0.03	0.05

TABLE I: Average and maximum errors (distances in meters) of main components in AVMON.

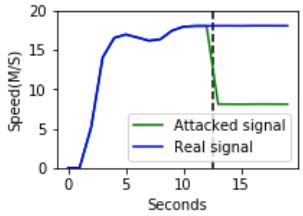


Fig. 9: An example of a velocity attack on an AV.

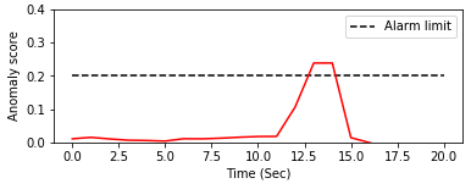


Fig. 10: Anomaly detection under the attack (in Fig. 9).

up to 6.5 meters during turns and an average of 1.06 meters. Finally, considering the Residual learning (RL) component further decreases the prediction error to be less than 0.5 meters at its maximum and 0.16 meters on average (a 10x reduction from just EKF) because it compensates for the non-linear effects that challenge the EKF estimator. For the straight track, the average error using only the default EKF estimator was not high (0.08 meters) with the RL component slightly improving prediction error (0.03 meters).

Next, we evaluate our anomaly detection performance. In Fig. 9, we trigger a malicious wheel speed sensor attack. As the attack started at time 12 secs, the signal was replaced with a malicious value. The ground vehicle tried to compensate to correct its orientation, causing large deviations from trajectory potentially leading to a crash. Fig. 10 shows the anomaly score over time using SDAR, which rapidly detects the attack crossing the detection threshold line. An anomaly score S quantifies the historical deviation based on SDAR. If $S > threshold$, then an alarm is raised.[1] Similarly, we launched GPS, IMU, and gyroscope attacks during quadrotor missions: AVMON caught all the attacks successfully within 0.2 secs (on average) after the attack launch.

B. Comparison to Savior

We performed a series of experiments comparing AVMON and SAVIOR [44]. The experiments use the CARLA simulator, as well as our ground, and aerial vehicle testbeds. Recall that SAVIOR uses an EKF for tracking vehicle state, and a cumulative sum (CUSUM) [37] algorithm for anomaly detection.

1) Prediction performance: In the first experiment, we use a quadrotor under our different scenarios, which differ in

[1]A video demonstrating the attack can be found at this link.

Curvature	Velocity	Mid-air stops	AVMon accuracy	SAVIOR accuracy
High	High	No	98.77%	89.19%
High	Low	No	100.0%	95.03%
High	Low	Yes	99.44%	88.72%
Low	High	No	100.0%	97.62%
Low	Low	No	100.0%	98.26%
Low	Low	Yes	100.0%	97.58%

TABLE II: Prediction accuracy for different route types.

Attack bias range	AVMon	SAVIOR
Less than 1m	Detected	Not detected
More than 1m	Detected	Not detected
More than 5m	Detected	Detected

TABLE III: Detectability using different bias values in the location readings for a ground vehicle.

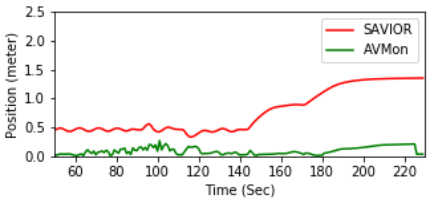


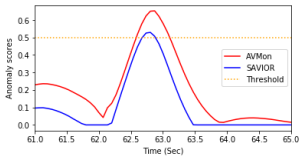
Fig. 11: Positional Error Ground AV.

trajectories' curvatures, velocity, presence of events such as multiple stopping in midair, and so on. Table. II shows that AVMON prediction quality is slightly higher than SAVIOR on simpler trajectories. However, it significantly improves the prediction quality in high dynamic routes.

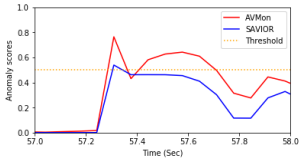
In the next attack, we use GPS Spoofing to inject a bias into the GPS sensor values. We measured if the anomaly detector in both schemes is capable of detecting the attacks for different bias ranges, as shown in Table. III. Due to its higher accuracy, AVMON can detect all the attacks in the bias ranges we tried, while Savior detected only attacks with more than 5m bias.

Time-To-Detect (TTD) is another essential metric for cyber-physical systems. Detection delays can provide an attacker with an opportunity to cause significant damage. Thus, we launched attacks where bias is injected into the AV sensor signals to compare TTD performance. We use different scenarios that vary in terms of trajectory curvatures, vehicle velocity, and presence/frequency of events where the vehicle stops and hovers in midair. We evaluate both schemes since they use different anomaly detectors to track the error throughout the quadcopter missions. AVMON was able to detect attacks faster than Savior, on average by 0.6 seconds with the anomaly detector and 0.3 seconds without across all scenarios. An accurate AV positioning, key for context awareness, is evaluated by our system AVMON vs. SAVIOR through position error. In both simulated ground vehicle and real quadrotor tests, AVMON outperforms SAVIOR, achieving significantly lower errors, reducing attacker maneuvering space within predicted paths. Fig. 11 shows an example of the positional error comparison using both scheme using a ground vehicle.

Finally, we experimented with different spoofed values for the internal parameters of the PID controller. Specifically, we



(a) Anomaly performance with high injected K_d value.



(b) Anomaly performance after spoofing steering.

Fig. 12: Detection comparison for two attacks.

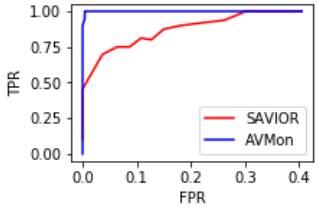


Fig. 13: ROC comparison implemented in a quadrotor.

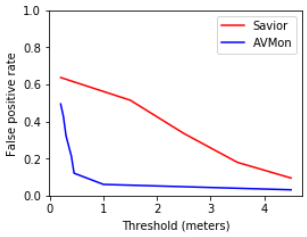


Fig. 14: FPR vs thresholds' values.

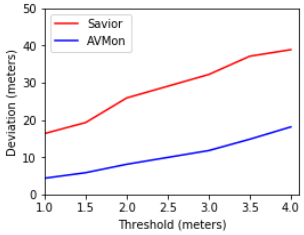


Fig. 15: Deviation vs thresholds' values.

spoof the derivative coefficient K_d and integral coefficient K_i. These derived values can efficiently cause the AV to get out of its reference trajectory. K_d reduces the overshoot caused by the proportional component of the controller, which drives the control output in proportion to the error. K_i fixes the systematic bias caused by the steering angle over time, which could eventually drive the vehicle out of the track. We also spoofed the steering angle output from the PID controller. In Fig. 12, we see a small advantage in TTD and anomaly score maximum threshold for AVMON.

2) End-to-End Attack Detection: AVMON is assessed against Savior for end-to-end performance. Threshold selection for each detector influences the trade-off between sensitivity to attacks and false positive probability. Adjusting thresholds through empirical experiments, including complex trajectories, allows for Receiver Oriented Characteristics (ROC) curve plotting, aiding in the identification of suitable thresholds balancing sensitivity and specificity. From Fig. 13, we observe that: (1) with the same threshold, AVMON outperforms and has lower FP rates, (2) by selecting smaller threshold values, we can detect attacks faster due to the smaller FP values, (3) AVMON can detect the attack with a probability close to 100% while having an FP rate (below 1%) using the same thresholds values for detecting the anomaly values as SAVIOR uses. To achieve the same sensitivity, SAVIOR would have over 30% FP rate.

3) Stealthy/adaptive attacks: We test the systems against stealthy attacks (described earlier under attack benchmarks). In these attacks, we inject false sensor data just under the detection threshold to avoid detection while causing maximum

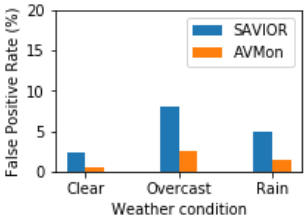


Fig. 16: False positives with weather condition (AV model in CARLA).

damage. Fig. 14 shows the impact of these stealthy attacks: how much deviation can a stealthy attack cause without being detected for different rates of false alarms. The resulting deviations are shown in Fig. 15. AVMON experiences lower deviations from the stealthy attacks at the same threshold value. Moreover, due to its higher accuracy AVMON can operate with a lower threshold with low false positives to even more efficiently avoid deviations from stealthy attacks compared to Savior. Of course, this only prolongs the duration needed to cause the required deviation, but this could allow sufficient time for a proximity detector, or an infrequent more expensive estimate (perhaps from the roadside infrastructure) that can help detect slow/accumulating deviations.

4) Sensitivity to external disturbances and noise: In real-world deployments, factors such as weather, road conditions, or system degradation affect the state of the vehicle and the sensors but are not captured in our EKF model, which tracks only vehicle dynamics. However, these disturbances could exhibit patterns when the residual learning component could learn present in the training data. To study the effect of these disturbances, we measured the prediction errors under different wind speeds and environmental conditions (e.g., clear, overcast, and rain). The baseline EKF model (SAVIOR) predictions are impacted appreciably. However, the residual learning component can compensate for these external disturbances and therefore experiences substantially lower false positive rates, as shown in Fig. 16.

C. Comparison to PID-Piper

We also compare AVMON to PID-Piper [14], a recently proposed framework that incorporates a machine learning predictor. PID-Piper uses a Feed-Forward Controller (FFC), as opposed to the Feed-Back Controller (FBC), to predict the future output of the PID controller rather than using feedback to minimize perceived error. The design relies on a Long Short-Term Memory (LSTM) [23] to learn the temporal correlation between inputs over time. This allows the system to predict future states of the vehicle, for example, predicting transition states from steady state to landing in drones.

We use the publicly available implementation of PID-Piper and evaluate it on the quadrotor using the PX4 Solo system for our different benchmark scenarios (recall that these include various trajectories' curvatures, velocity, as well as other events such as stopping and starting events). From Table. IV, we can see that AVMON achieves slightly better prediction quality (measured as a percentage of predictions within a preset detection threshold) on low curvature trajectories, which are simpler. However, it significantly improves the prediction quality in trajectories with high dynamics (marked as high

Trajectory	AVMon	PID-Piper
High curvature	97.7%	85.0%
Low curvature	99.0%	96.38%

TABLE IV: Prediction accuracy for different routes.

	Quadrotor	Ground vehicle
No residual learning	0.0075 sec	0.0013 sec
Full AVMON	0.012 sec	0.0041 sec
CPU overhead	10%	4.1%
Memory overhead	3%	8%

TABLE V: AVMon Overheads.

curvature); we conjecture that because of its open loop control (no feedback), PID-Piper can accumulate errors, which is a known drawback of FFC controllers.

D. Performance Overhead

To ensure real-time decisions, AVmon's components are lightweight (225.5 KB model) and seamlessly integrate with the vehicle's ROS controller. We measured the execution time of the AVMON module (e.g., conducting its $Path\ planning$ algorithm) for 5 minutes using two different tracks. The average execution time metric was calculated as an indication for evaluating the runtime performance. The overhead represents the average execution time with respect to the real-time constraints or timestamp (i.e., each 0.1 sec). So for the ground vehicle, Table V shows the performance of AVMon (second row) and AVMon without residual learning (first row), which is similar to the overhead of Savior. While the residual learning increases the overhead, the overhead remains low (about 4% for the ground vehicle). The execution time for the aerial vehicle was, on average, 0.012 seconds with 10% overhead. For our implementation in the aerial vehicle, we used the latest stable version of a popular drone operating system, PX4 v1.11.0. The size of the AVMON module is 8.5 KB. The module uses the Dronekit library to provide access to telemetry data and to send an interface with the controller through action commands such as take off and landing through accessor functions. Finally, AVMON incurs over 3X lower overhead than PID-Piper, on average across all the tested scenarios.

E. Potential Mitigations

Mitigating attacks once detected is a difficult problem deserving of its own investigation; it is critical to detect attacks accurately and in a timely way to effectively trigger mitigations. A potential approach is to replace the sensor data with the data generated from the predictor as PID-piper does [14]; however, this approach could be dangerous. Another approach is to identify more carefully the anomalous set of sensors and use predicted values for them, or alternatively estimate the state without them if possible; having some redundancy in the sensors can also help this approach. When such actions are not supported or possible, the AV should drop into a safe operating mode and/or alert human controllers.

F. Possible Limitations

Using AVMON in AVs has limitations due to the dynamic and complex nature of real-world driving environments. Operating without detailed maps, especially in unmapped construction zones, poses challenges for Kalman Filters (KF) that rely on accurate initial state estimates and prior knowledge. Additionally, modeling the complex and sometimes irrational behavior of human drivers, pedestrians, and cyclists is difficult, introducing inaccuracies in state estimation. Extreme weather conditions can degrade sensor performance, requiring assistance for KF to handle abrupt sensor quality changes. Real-time processing is crucial for AVMON, and while advanced nonlinear filters like Particle Filters and Unscented Kalman Filters offer superior accuracy, their computational expense makes them unsuitable for high-frequency prediction, involving tuning parameters that can impact performance.

VI. RELATED WORK

A number of prior studies have considered how to protect vehicles against software attacks that originate from a malicious or compromised component. For example, several studies [59], [36], [28] demonstrate detection approaches to protect against attacks on electronic control units (ECU) within vehicles. If an attacker successfully compromises an ECU, they control the functionality it governs, leading to a potentially catastrophic compromise of the vehicle. These detection schemes look for pre-defined attack signatures and achieve a low false positive rate. However, they cannot capture novel attacks or variations of known attacks and require maintaining an up-to-date attack signature database.

In order to limit an attacker with access to the internal Controller Area Network (CAN) bus, the standard bus that interconnects components within a vehicle, Siddiqui et al. [51] propose hardware-based mutual authentication and encryption. Seshadri et al.[49] introduce the notion of Indisputable Code Execution (ICE), which supports the secure execution of functionality on a network node from a trusted component based on measuring the integrity of the system from the firmware and up. Redundancy-based techniques [17], [18] duplicate important system components and cross-check their states and outputs at run-time for detecting attacks and anomalies. The redundancy can include software and/or hardware modules. These approaches target a different threat model and can work orthogonally to protect an AV.

Recently, Choi et al. [11] proposed LTC, a detection framework for sensor tampering attacks. They use standard model templates that are fit on profiling data from just a few test missions and use it to detect physical or sensor attacks. However, the algorithm uses a linear prediction filter which can cause large errors, especially in dynamic scenarios. Quinonez et al. [44] improve on this model by using a second-degree nonlinear Kalman Filter as the basis of the model. Although the model is nonlinear, the extrapolation between computation steps in the model is linear; this is the issue we address using the residual learning component of AVMON. Li et al. [32] proposed using a dual extended Kalman filter (DEKF) using linear parameter-varying system proposed to achieve faster convergence. However, the authors did not compare their scheme with any machine learning-based work to compare accuracy.

A number of recent proposals integrate a Machine Learning component with a physical dynamics model, similar to

our approach. The real-time adaptive sensor attack detection ramework [5] can dynamically adapt the detection delay and false alarm rate to meet the detection deadline and improve usability according to different system statuses. It is based on a deep learning model that is offline extracted from sensor data through leveraging convolutional neural network (CNN) and recurrent neural network (RNN). Thus, the overhead could be larger. PID-Piper [14] uses a Machine learning based Feed-Forward Controller (FFC) to monitor the deviation and predict the potential disturbances (due to attacks) and directly rectifies the autonomous robotic vehicles' trajectory based on the prediction. If the deviation exceeds a pre-defined threshold, PID-Piper signals an attack. However, this technique has a longer run time behavior since it is based on Long Short Term Memory (LSTM) architecture. As we showed earlier, our model outperforms PID-piper while also being significantly less computationally expensive. KalmanNet [45], also proposes combining KF with a machine learning model. Specifically, KalmanNet replaces the Kalman gain (KG) computation with a Recurrent Neural Network (RNN). However, the complex RNN, while improving accuracy, is computationally expensive, making the solution unsuitable for real-time use.

ACKNOWLEDGEMENTS

This material is partially supported by the National Science Foundation (NSF) grants CCF-2212426, CNS-2053383, and CNS-1955650. It is also partially supported by U.S.. Army Research Office/Department of Defense award number W911NF2020267.

VII. CONCLUDING REMARKS

We proposed a new framework for protecting autonomous vehicles from attacks that manipulate sensor data, based on monitoring the control invariants of the AV. We identified several opportunities to improve on recent state-of-the-art defenses that rely on using predictions of the model's state. We proposed a machine learning residual estimation module to compensate for non-linear effects, as well as other optimizations to improve accuracy and reduce false positives. Our solution substantially improves the prediction quality in highly dynamic trajectories, and in the presence of unmodeled effects such as weather, with fewer false positives. We evaluated the scheme in both ground vehicles and quadrotors, using both simulation and hardware testbeds, demonstrating the effectiveness and practicality of the solution. We believe our defense represents an important step forward in improving this classes of defenses.

REFERENCES

[1] "Matlab system identification toolbox," accessed 2021 from https://www.mathworks.com/products/sysid.html.

[2] "Pixhawk 4," 2020, accessed 2021 from http://https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk4.html/.

[3] "Automated Vehicles for Safety," 2021, accessed 2017 from https://www.nhtsa.gov/technology-innovation/automated-vehicles.

[4] M. Abdullah, J. Jamil, and A. E. Mohan, "Vehicle dynamics modeling simulation," 02 2020.

[5] F. Akowuah and F. Kong, "Real-time adaptive sensor attack detection in autonomous cyber-physical systems," in *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2021, pp. 237–250.

[6] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[7] F. Bakhshande and D. Söffker, "Adaptive step size control of extended/unscented kalman filter using event handling concept," *Frontiers in Mechanical Engineering*, vol. 5, p. 74, 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/fmech.2019.00074

[8] R. W. Beard, "Master thesis: Quadrotor dynamics and control," brigham young university, Tech. Rep., 2008.

[9] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial sensor attack on lidar-based perception in autonomous driving," *the Computing Research Repository (CoRR)*, vol. abs/1907.06826, 2019. [Online]. Available: http://arxiv.org/abs/1907.06826

[10] E. Chan-Tin, D. Feldman, N. Hopper, and Y. Kim, "The frog-boiling attack: Limitations of anomaly detection for secure network coordinate systems," vol. 19, 09 2009, pp. 448–458.

[11] H. Choi, W.-C. Lee, Y. Aafer, F. Fei, Z. Tu, X. Zhang, D. Xu, and X. Deng, "Detecting attacks against robotic vehicles: A control invariant approach," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 801–816. [Online]. Available: https://doi.org/10.1145/3243734.3243752

[12] V. Costan and S. Devadas, "Intel sgx explained." *International Association for Cryptologic Research*, vol. 2016, no. 86, pp. 1–118, 2016.

[13] P. Dash, M. Karimibiuki, and K. Pattabiraman, "Out of control: Stealthy attacks against robotic vehicles protected by control-based techniques," in *Proceedings of the 35th Annual Computer Security Applications Conference*, ser. ACSAC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 660–672. [Online]. Available: https://doi.org/10.1145/3359789.3359847

[14] P. Dash, G. Li, Z. Chen, M. Karimibiuki, and K. Pattabiraman, "Pid-piper: Recovering robotic vehicles from physical attacks," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2021, pp. 26–38.

[15] D. Dempsey , "Learn the hidden secret of autonomous car navigation guidance: Imus," 2019, accessed 2021 from https://www.fierceelectronics.com/components/learn-hidden-secret-autonomous-car-navigation-guidance-imus.

[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[17] F. Fei, Z. Tu, R. Yu, T. Kim, X. Zhang, D. Xu, and X. Deng, "Cross-layer retrofitting of uavs against cyber-physical attacks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 550–557.

[18] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—a survey and some new results," *Automatica*, vol. 26, no. 3, p. 459–474, May 1990. [Online]. Available: https://doi.org/10.1016/0005-1098(90)90018-D

[19] A. Gad, "Beginners ask "how many hidden layers/neurons to use in artificial neural networks?"," Jun 2018. [Online]. Available: https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e

[20] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[21] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, and R. Candell, "A survey of physics-based attack detection in cyber-physical systems," *ACM Computing Surveys*, vol. 51, no. 4, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3203245

[22] N. J. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," 1993. [Online]. Available: https://api.semanticscholar.org/CorpusID:12644877

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[24] S. Huang, W. Wong, Y. Feng, Q. A. Chen, Z. Mao, and H. Liu, "Impact

evaluation of falsified data attacks on connected vehicle based traffic signal control," 10 2020.

[25] D. Jenkins and B. Vasigh, "The economic impact of unmanned aircraft systems integration in the United States," 2013, accessed 2017 from https://www.ros.org/news/2016/05/ros-kinetic-kame-released.html.

[26] J. Junkins and M. Shuster, "Geometry of the euler angles," *Journal of The Astronautical Sciences - J ASTRONAUT SCI*, vol. 41, pp. 531–543, 10 1993.

[27] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960. [Online]. Available: https://doi.org/10.1115/1.3662552

[28] S. Kaur and M. Singh, "Automatic attack signature generation systems: A review," *IEEE Security Privacy*, vol. 11, no. 6, pp. 54–61, 2013.

[29] A. Kerns, D. Shepard, J. Bhatti, and T. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of Field Robotics*, vol. 31, 07 2014.

[30] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004, pp. 2149–2154.

[31] D. Lee, D. Kohlbrenner, S. Shinde, K. Asanović, and D. Song, "Keystone: An open framework for architecting trusted execution environments," in *Proceedings of the Fifteenth European Conference on Computer Systems*, 2020, pp. 1–16.

[32] C. Li, Y. Liu, L. Sun, Y. Liu, M. Tomizuka, and W. Zhan, "Dual extended kalman filter based state and parameter estimator for model-based control in autonomous vehicles," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 327–333.

[33] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," *CoRR*, vol. abs/1611.06612, 2016. [Online]. Available: http://arxiv.org/abs/1611.06612

[34] J. Liu, C. Yan, and W. Xu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicles," DEF CON, 2016, https://doi.org/10.5446/36252 $Lastaccessed : 30Jan2021$.

[35] Ljung, "The control handbook," *CRC Press*, 1996.

[36] T. Morris and W. Gao, "On cyber attacks and signature based intrusion detection for modbus based industrial control systems," *Journal of Digital Forensics, Security and Law*, vol. 9, pp. 37–56, 01 2014.

[37] C. Murguia and J. Ruths, "Cusum and chi-squared attack detection of compromised sensors," in *2016 IEEE Conference on Control Applications (CCA)*, 2016, pp. 474–480.

[38] A. Y. Ng, "Feature selection, l1 vs. l2 regularization, and rotational invariance," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 78. [Online]. Available: https://doi.org/10.1145/1015330.1015435

[39] J. Noh, Y. Kwon, Y. Son, H. Shin, D. Kim, J. Choi, and Y. Kim, "Tractor beam: Safe-hijacking of consumer drones with adaptive gps spoofing," *ACM Transactions on Privacy and Security*, vol. 22, no. 2, 2019. [Online]. Available: https://doi.org/10.1145/3309735

[40] Y. Oshman and I. Shaviv, "Optimal tuning of a kalman filter using genetic algorithms," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2000-4558

[41] R. Pepy, A. Lambert, and H. Mounier, "Path planning using a dynamic vehicle model," in *2nd International Conference on Information Communication Technologies*, vol. 1, 2006, pp. 781–786.

[42] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," in *Black Hat Europe*, Nov 2015. [Online]. Available: https://www.blackhat.com/docs/eu-15/materials/eu-15-Petit-Self-Driving-And-Connected-Cars-Fooling-Sensors-And-Tracking-Drivers-wp1.pdf

[43] M. Pham and K. Xiong, "A survey on security attacks and defense techniques for connected and autonomous vehicles," *Computers and Security*, vol. 109, no. C, oct 2021. [Online]. Available: https://doi.org/10.1016/j.cose.2021.102269

[44] R. Quinonez, J. Giraldo, L. Salazar, E. Bauman, A. Cardenas, and Z. Lin, "SAVIOR: Securing autonomous vehicles with robust physical invariants," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 895–912. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/quinonez

[45] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, "KalmanNet: Neural network aided kalman filtering for partially known dynamics," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.

[46] A. Saaid, D. Nur, and R. King, "Change points detection of vector autoregressive model using sdvar algorithm," in *Proceedings of the Fifth Annual ASEARC conference*, 01 2012.

[47] F. Sabatino, "Quadrotor control: modeling, nonlinearcontrol design, and simulation," Master's thesis, KTH, Automatic Control, 2015.

[48] G. Seetharaman, A. Lakhotia, and E. Blasch, "Unmanned vehicles come of age: The darpa grand challenge," *IEEE Computer*, vol. 39, pp. 26 – 29, 01 2007.

[49] A. Seshadri, M. Luk, A. Perrig, L. van Doorn, and P. Khosla, "Scuba: Secure code update by attestation in sensor networks," vol. 2006, 01 2006, pp. 85–94.

[50] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications," in *Cryptographic Hardware and Embedded Systems – CHES 2017*, W. Fischer and N. Homma, Eds. Cham: Springer International Publishing, 2017, pp. 445–467.

[51] A. S. Siddiqui, Y. Gui, J. Plusquellic, and F. Saqib, "Secure communication over canbus," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017, pp. 1264–1267.

[52] C. Simpkins, "System identification: Theory for the user, 2nd edition (ljung, l.; 1999) [on the shelf]," *Robotics Automation Magazine, IEEE*, vol. 19, pp. 95–96, 06 2012.

[53] Y. Son, H. Shin, D. Kim, Y. Park, J. Noh, K. Choi, J. Choi, and Y. Kim, "Rocking drones with intentional sound noise on gyroscopic sensors," in *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, Aug. 2015, pp. 881–896. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/son

[54] I. Synopsys, "What is an Autonomous Car?" 2021, accessed 2021 from https://www.synopsys.com/automotive/what-is-autonomous-car.html.

[55] W. Taylor, "Change-point analysis: A powerful new tool for detecting changes," *Deerfield, IL: Baxter Healthcare Corporation*, 03 2000.

[56] T.Foote, "ROS Kinetic Kame Released," 2016, accessed 2017 from https://www.ros.org/news/2016/05/ros-kinetic-kame-released.html.

[57] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks," in *2017 IEEE European Symposium on Security and Privacy (EuroS P)*, 2017, pp. 3–18.

[58] Y. Tu, Z. Lin, I. Lee, and X. Hei, "Injected and delivered: Fabricating implicit control over actuation systems by spoofing inertial sensors," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018, pp. 1545–1562. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/tu

[59] Z. Tyree, R. A. Bridges, F. L. Combs, and M. R. Moore, "Exploiting the shape of CAN data for in-vehicle intrusion detection," *CoRR*, vol. abs/1808.10840, 2018. [Online]. Available: http://arxiv.org/abs/1808.10840

[60] A. Umamageswari, J. Ignatious, and R. Vinodha, "A comparitive study of kalman filter, extended kalman filter and unscented kalman filter for harmonic analysis of the non-stationary signals," vol. 3, 2012.