IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 29, 2019, accepted September 5, 2019, date of publication September 23, 2019, date of current version October 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943207

LiDAR Data Integrity Verification for Autonomous Vehicle

RAGHU CHANGALVALA, (Member, IEEE), AND HAFIZ MALIK, (Senior Member, IEEE)
Electrical and Computer Engineering Department, University of Michigan-Dearborn, Dearborn, MI 48128, USA
Corresponding author: Raghu Changalvala (rchangal@umich.edu)

ABSTRACT Deterministic perception of the surrounding environment is both crucial and a challenging task for autonomous vehicles. A wide range of sensors, including LiDAR, RADAR, cameras, and so on, are used to build the perception layer of an autonomous vehicle. Many interfaces, such as OBD-II, Wi-Fi, Bluetooth, cellular networks, etc., have been introduced in autonomous vehicles to control various functionalities, including V2X communications, over-the-air updates, security, remote vehicle-health monitoring, and so on. These interfaces are introducing new attack surfaces that can be exploited via external as well as internal attacks. Attackers have successfully demonstrated how to exploit these attack surfaces by crafting attack vectors to launch both insider and external attacks. The sensor and sensor data are also vulnerable to both external and insider attacks. Developing safeguards against these attacks is a steppingstone toward the design and development of reliable autonomous vehicles. For instance, failure to detect and localize sensor data tampering can result in an erroneous perception of the environment and lead to wrong path-planning and control decisions. In this paper, we propose a novel semi-fragile data hiding-based technique for real-time sensor data integrity verification and tamper detection and localization. Specifically, the proposed data hiding-based method relies on 3-dimensional quantization index modulation (QIM)-based data hiding to insert a binary watermark into the LiDAR data at the sensing layer, which is used for integrity verification and tamper detection and localization at the decision-making unit, e.g., the advanced driver assistance system (ADAS). The performance of the proposed scheme is evaluated on a benchmarking LiDAR dataset. The impact of information hiding on the object-recognition algorithm is also evaluated. Experimental results indicate that the proposed method can successfully detect and localize data tampering attacks, such as fake object insertion (FOI) and target object deletion (TOD). Robustness to noise-addition attacks is also evaluated.

INDEX TERMS Autonomous vehicle, ADAS, LiDAR point cloud, data hiding, quantization index modulation (QIM).

I. INTRODUCTION

Fully autonomous vehicles are the future of our transportation, and the deterministic perception of an ego vehicle's environment is a crucial step in achieving autonomy. Many crucial driving decisions, such as acceleration, deceleration, braking, and evasive maneuvers, directly depend on the information from the perception layer. The perception layer builds an environment snapshot from the sensor data in a centralized data-fusion architecture. Multiple sensors like LiDAR, cameras, and RADAR connect to the decision-making microcontroller over various interfaces like ethernet, controller area network (CAN), and low-voltage differential signaling (LVDS). The decision block, referred to as the *sensor fusion core* understands the data and instructs the motion control system (actuators) to act.

Modern-day vehicles are cyber-physical systems (CPS) consisting of distributed networks with multiple sensors and electronic control units (ECUs) connected over multiple in-vehicle networks like controller area networks (CAN), ethernet, etc. These in-vehicle networks are interconnected over gateway modules transmitting data and control commands. In this distributed architecture, it is possible to inject code through available attack surfaces like the on-board diagnostics port (OBD-II) and CAN bus into the core ECU and bridge across multiple networks, thus exposing attack surfaces on different networks. Numerous attack vectors have been proposed in detail for the CAN and the OBD-II port over the past decade [1]. It was demonstrated that attackers can infiltrate any ECU and circumvent safety measures to

The associate editor coordinating the review of this manuscript and approving it for publication was Laxmisha Rai.

modify the outcome of safety-critical systems, disable brakes, perform steering control, or cause faulty cluster displays and even a complete engine shutdown [1]. Attackers can exploit these attack surfaces for sensor data manipulation. In-vehicle sensor data communication is unencrypted, and therefore an attacker just needs access to the in-vehicle network for sensor data manipulation, which can be realized through available attack surfaces. Though exposure of the external interface to unsolicited messages over a physical interface like the OBD-II port is a decreasing trend, growth in the integration of short- and long-range wireless interfaces is expanding the attack surfaces of connected vehicles. In a connected vehicle, attacks could be launched over wireless channels without physical access to the vehicle [2]. Shown in Fig. 1 is a high-level illustration of (a) data flow from various sensors to the sensor fusion core residing in a vehicle, also known as ADAS (advanced driver assistance system), and (b) available attack surfaces. For instance, a LiDAR sensor mounted on the vehicle sends a raw point cloud of the tracked environment over an ethernet/LVDS link to the sensor fusion ADAS core ECU placed in the vehicle. Inside the sensor fusion core, the object information from the raw sensor data is extracted and a list of tracked-object tracklets is computed and provided as an input to the perception estimator and other applications like the vehicle localizer.

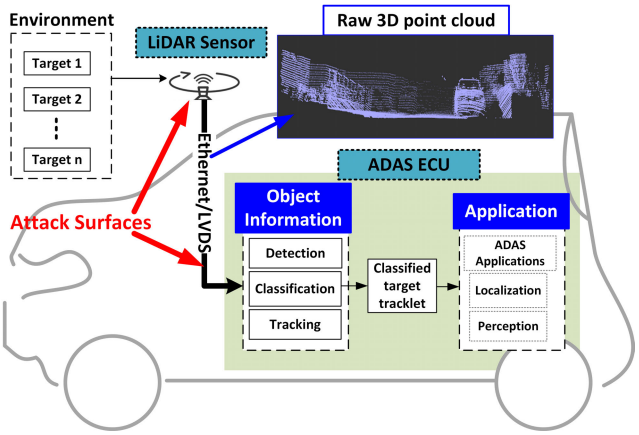
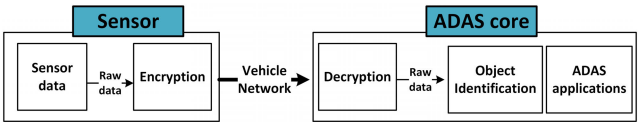


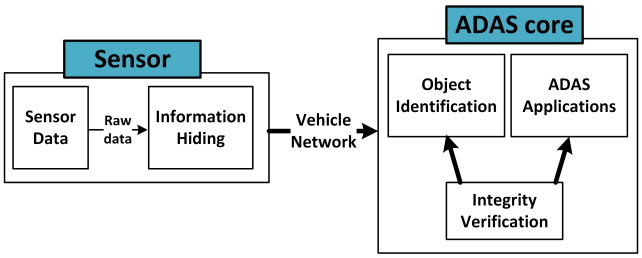
FIGURE 1. Autonomous vehicle general architecture: Sensor data flow path from origin to perception layer. Attack surfaces: Perception layer and sensor data transmission layer.

If the input to the sensor fusion core is compromised, the resulting decisions down the understand-and-act pipeline would be erroneous and could result in significant damage. Though some redundancy could be built throughout the system by fusing different sensor information, the computational cost of path planning and other control algorithms to work around and ignore the tampered sensor data is much higher than detecting the tampering at the sensor level. It can be observed from Fig. 1(b) that given a vehicle architecture in which a sensor transmits raw data to the vehicle for data interpretation, an attacker can exploit vehicle attack surfaces to tamper with the raw sensor data by simple operations like fake object insertion (FOI) or target object deletion (TOD) to dupe the object information extractor and the perception estimation applications that are down the pipeline. By inserting tampered data containing fake objects or by deleting existing objects an attacker can influence the perception and localization algorithms to consider and act on the tampered data. This would result in the ADAS making wrong control decisions like decelerating or braking when it is not supposed to or driving right into a target object. These wrong control decisions can pose a serious safety threat to the occupants of an autonomous vehicle. *Integrity verification of sensor data before acting on it is crucial.*

Integrity verification of sensor data can be achieved using either data encryption or data hiding. Shown in Fig. 2 is the comparison between encryption-based and data hiding-based sensor data verification. It can be observed from Fig. 2(a) that encryption-based verification requires data decryption before ADAS processing, whereas data hiding-based integrity verification does not require any removal of embedded data (also known as a watermark) before ADAS processing. The additional step of decryption in the data encryption-based method adds delay, which becomes a bottleneck when dealing with high-frequency and high-bandwidth data like a LiDAR point cloud. To address this problem, we propose to use a digital watermarking-based method for integrity verification and tamper localization of sensor data. In a digital watermarking/data hiding-based method, the ADAS core can work on the embedded point cloud while the data verification engine does the integrity check in parallel, as shown in Fig. 2(b). The motivation behind selecting watermarking-based approaches is twofold: (1) these schemes introduce minimal latency/overhead, and (2) these schemes allow the system designer to achieve optimal watermark robustness and fidelity (embedding distortion) goals. Any digital watermarking technique can be used for integrity verification and tamper localization,



(a) Traditional cryptographic-based digital data integrity verification approach.



(b) Information hiding-based digital data integrity verification approach.

FIGURE 2. Comparison of traditional and data hiding-based digital data integrity verification frameworks.

including direct spread spectrum (DSS)-based watermarking, quantization index modulation (QIM), and so on.

To demonstrate the effectiveness of the proposed data hiding-based integrity verification for LiDAR data, a 3-dimensional quantization index modulation (3D-QIM) in the raw point-cloud (or direct) domain is proposed. Simplicity, low embedding/decoding complexity, quantifiable embedding distortion as a function of embedding parameter Δ, and detection performance as a function of channel distortion and embedding parameters are the salient features of QIM-based data hiding, which is the main motivation behind selecting it over the other information-hiding methods [3], [4]. The proposed method could be easily adapted into other 3D point-cloud data generators like RADAR, red-green-blue-depth (RGBD) cameras, etc.

A. PRINCIPAL CONTRIBUTION

Like any other autonomous vehicle sensor system, LiDAR is not only vulnerable to regular and side-channel attacks like sensor spoofing and saturation [5] but it is also vulnerable attacks on data transmission between LiDAR and ADAS. The major contributions of this paper are:

1) Investigated threat models and attack surfaces for sensors and proposed a framework for *sensor data integrity verification and tamper localization*. We focused on two transmission channel-level insider attacks on LiDAR sensor data and proposed a framework to identify them in real-time
2) Proposed a 3D data hiding based on 3D-QIM for the LiDAR raw point cloud.
3) Evaluated the performance of the proposed method on real-world LiDAR sensor data provided by KITTI vision benchmark suite [6], which is extensively used for training object-detection algorithms in autonomous vehicles.
4) Demonstrated that the proposed tamper-resistant approach does not cause the performance of the object-detection algorithms to deteriorate.

The rest of the paper is organized as follows: Section II presents an overview of the state of the art in LiDAR data forensics in autonomous vehicles. LiDAR data usage in autonomous vehicle applications is discussed in Section III. Section IV introduces the proposed data hiding method QIM and its adaptation to the LiDAR point cloud. Section V defines the attack vectors on LiDAR data considered in this research. Section VI details the countermeasure framework implemented to detect and localize the tampering. Section VII explains the experiments performed to evaluate the effectiveness of countermeasure framework and corresponding results followed by a conclusion in Section VIII

II. RELATED WORK

In this section, we provide a brief overview of the existing literature on LiDAR sensor attack models, sensor data integrity measures for automotive networks, and 3D QIM-based watermarking techniques for point-sampled data like a LiDAR point cloud.

LiDAR sensor spoofing by replay, relay attacks and sensor saturation/jamming attacks was extensively studied in [5], [7], [8]. Countermeasures such as wavelength redundancy, using a short pulse period, and probe redundancy was suggested to sustain these attacks. Though these techniques provide countermeasures against attacks on the LiDAR sensor using the same physical interface, also referred to as regular channel data injection attacks [9] they cannot counter insider attacks on the sensor data, that is, attacks originating from within the vehicle. Tampering attacks on LiDAR data that do not need any external hardware to launch are identified in [10]. These attack models focus on insider attacks, and the forensic algorithms proposed were based on the assumption that the forged area resolution and occlusion consistency of a given data frame are different from those of the original data. The equirectangular projection method proposed involves the 3D-to-2D projection of LiDAR data, increasing the runtime complexity when dealing with thousands of points in a LiDAR point cloud, and the proposed methods require multiple post-processing steps like median filtering and connected-component estimation to extract the forged regions. Also, the proposed methods for checking occlusion consistency are not useful for autonomous driving applications, since the motive of the attacker, in this case, would be more to confuse the algorithm interpreting the data than to hide data within a frame.

In [11] Benjamin et al. propose integrity assurance mechanisms in hard real-time constrained automotive networks. Assuming the hacker has complete network access, they propose methods to check data integrity in automotive networks. Source authentication via challenge-response schemes and digital signatures based on symmetric and asymmetric keys were discussed. Symmetric-key-based message authentication codes (MACs) exchanged between a sensor and a verifier were determined to better suit the time- and resource-constrained automotive networks and hardware. Though these methods were designed to fit the constraints of a CAN network, they can be extended to any other transmission protocols. The MACs, which can be data-centric or signal-centric or decoupled, add additional computation and time delay at the receiver end to remove these MACs before processing the data. Forgery detection using digital watermarking techniques in structured 3D point clouds like meshes has been a well-researched area, whereas forensics on unstructured ones such as LiDAR 3D point clouds is less explored. In [12] Parag introduced a method to watermark unstructured point clouds for forensic applications. Watermarks are embedded into the cluster tree of 3D points built on clusters generated from a nearest-neighbor search. An extension of the 3D QIM technique is used to embed and extract the watermark on these clusters. This method achieved 100% robustness against uniform affine transforms on point-sampled data but failed in the presence of local or global noise. Application of this technique to the

LiDAR point cloud adds additional computation for cluster generation, and forgery localization cannot be achieved.

III. LiDAR POINT CLOUD: *APPLICATIONS*

The LiDAR sensor plays a key role in an autonomous-driving vehicle due to its ability to provide better perception in all light conditions in comparison to other sensors like digital cameras. Adverse weather conditions like fog and rain could reduce the accuracy of the data, but in moderate weather conditions, LiDAR is well suited for high-frequency applications such as building a perception layer for an autonomous vehicle. High-end LiDARs could generate detailed local maps of an ego vehicle working in all light conditions. These maps could be used for a variety of critical tasks such as behavior predictions of the surrounding vehicles and environment. This environmental behavior prediction, such as whether a vehicle ahead is making a turn or not, helps a self-driving vehicle in predictive path planning. Typically, LiDARs are used in medium-range {80 to 160 m} applications such as collision avoidance and pedestrian detection and also in long-range {160 to 300 m} applications like adaptive cruise control and critical object tracking. A smart LiDAR is equipped with integrated ECUs to perform preprocessing, object-recognition (detection and classification), and tracking functions and provides a list of tracked objects to a control system. On the other hand, a simple LiDAR provides a raw point cloud, and the object recognition and tracking are performed in the ADAS ECU, as shown in Fig. 1. The choice of the type of LiDAR depends on autonomous vehicle architecture and functional safety requirements. In this paper, we focus on autonomous vehicle systems built on LiDARs that provide a raw point cloud.

The LiDAR data returns have no shape attributes, as they represent the perceived environment. The density of the point cloud depends on the horizontal and vertical angular resolution of the LiDAR. For automotive applications in general, the point cloud is sparse, as the points are spread across the maximum range of the LiDAR, which could be up to 300 m. Each point in the point cloud is usually represented by its Cartesian coordinates and the intensity of reflection. In autonomous vehicle applications, most of the existing object detection and tracking models do not consider the intensity of reflection; hence, that value is neglected in this paper. In this work, the 3D point cloud is considered as a set of points $pc = \{p^1, p^2, p^3..p^n\}$, where each point is the combination of its x, y, z components $p^i = \{p^i_x, p^i_y, p^i_z\}$.

IV. QUANTIZATION INDEX MODULATION (QIM): *AN OVERVIEW*

Quantization index modulation (QIM)-based data hiding is a non-linear data hiding technique commonly used for digital watermarking and digital steganography applications [13]–[15]. It is an information hiding technique wherein a host signal is quantized based on the embedded message symbols. For binary QIM-based data hiding, a host signal $S = \{s_1, s_2, \cdots, s_N\}$ is quantized based on the message symbol to be embedded, $M = \{m_1, m_2, \cdots, m_N\}$, here, $m_i \in \{0, 1\}$, resulting in a processed signal that can be expressed as:

$$s_w(s_i, m_i) = q_{m_i}(s_i, \Delta), \ \ where\ i = 1, 2, \cdots, N \tag{1}$$

where, $q_{m_i}(\cdot)$ denotes a uniform quantizer. With quantization step-size Δ and a perturbation of $\Delta/2$, it can be expressed as:

$$q_{m_i}(s_i, \Delta) = round(\frac{s_i}{\Delta}) \cdot \Delta \pm m_i \cdot \Delta/2 \tag{2}$$

The QIM-based quantization operation uses a unique set of quantization/reconstruction grids. The dimensionality of the quantization/reconstruction grid depends on the size of the message symbols. Specifically, a ν-dimensional message's symbols would require a $\log_2(\nu)$-dimensional grid. For example, for a binary message (i.e., $\nu = 2$), a 1D quantization/reconstruction grid is required. Considering a 1-D quantization/reconstruction grid consists of two unique quantizers, (say $q_0(\cdot)$ and $q_1(\cdot)$) shown in Fig. 3, the ★ points represent quantization/reconstruction points for uniform quantization function $q_0(\cdot)$ with a step-size Δ for a message symbol $m_i = 0$, and ♦ points represent the quantization/reconstruction points for a similar uniform quantizer $q_1(\cdot)$ with a step-size of Δ for a message symbol $m_i = 1$. Here, uniform quantizer $q_1(\cdot)$ is obtained by shifting forward or backward uniform quantizer $q_0(\cdot)$ by $\pm\Delta/2$, that is, $q_1(\cdot) = q_0(\cdot) \pm \Delta/2$. If an i^{th} sample of the signal, s_i falls within the i^{th}quantization cell, it will be quantized to either $(i) \times$ ★ or $(i) \times$ ♦ based on the i_{th} embedded message symbol $m_i \in \{0, 1\}$. Essentially, a watermarked signal from a signal sample s_i with a binary hidden message m_i can be obtained from Eq. (1). This watermarked signal is sent over the transmission channel, and the received signal at the receiver input is used for watermark extraction. The received signal could get distorted due to the channel noise, intentional processing or malicious attacks. If $s'_w(s_i, m_d)$ represents a distorted received signal, the embedded message m_d from the received signal is estimated based on the difference between the received signal and the quantized value of the received signal for all the quantization functions used in the embedding step as shown in Eq. (3). Essentially, minimum distance-based framework is used for message decoding, that is, the reconstruction point closest to the received signal sample is used for message decoding, which can be expressed as:

$$m_d = \underbrace{argmin}_{i \in 0,1} |s'_w(s_i, m_d) - s_w(s_i, m_i)| \tag{3}$$

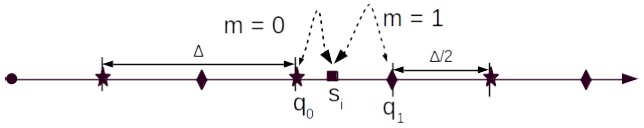


FIGURE 3. Illustration of 1-dimensional QIM-based data hiding.

A. THE QIM-BASED DATA HIDING INTO LiDAR POINT CLOUD

Any given sample in a LiDAR dataset is the combination of the reflection intensity of a point in space and its corresponding 3D location coordinates. Since we are focusing on autonomous vehicle applications, the primary usage of LiDAR data would be in the areas of perception and localization. These applications require distance measurements to the detected objects in the LiDAR point cloud. In performing object detection on the raw LiDAR point cloud, the general norm is to reduce the redundancy and bring in fixed connectivity between the points and then feed this sensor data to a prediction model. Most of these prediction models are deep-learning-based, where the model extracts features based on the training data set. The existing prediction models cannot detect LiDAR point-cloud tampering. Cryptographic- or data hiding-based approaches can be developed to solve this problem. It can be observed from Fig. 2 that the data hiding-based solution outperforms the cryptographic-based solution as far as latency is concerned. The challenge for the data hiding-based integrity verification method for automotive and robotic applications is to ensure that message embedding distortion should not deteriorate the performance of prediction models used in the ADAS unit. The QIM-based data hiding provides the flexibility to select a desired embedding distortion level as a function of the quantization parameter, which is the main motivation behind selecting QIM over other available data-hiding methods. In the following, we outline QIM-based data hiding for a LiDAR point cloud.

The basic principle of quantizing the host signal using multiple quantizers, where each one of them could be treated as a set of reconstruction points, can be extended to a 3D point cloud such as LiDAR sensor data [16]. The point samples from the LiDAR sensor are randomly located by default and lack connectivity information. To give them shape and connectivity aspects, the point cloud is divided into fixed-size voxels. A voxel is a fixed-width cube in 3D space. Once the maximum range of the point samples from the sensor is determined, points are quantized with a specific step size Δ. After this quantization step, all the points that fall within a voxel are represented either by a fixed vertex of the corresponding voxel or by its centroid. This voxelization step also reduces the redundancy in reflections from the same target. After voxelization, all the points of the signal S, with position vectors that fall within a voxel k are represented by the vertex at the origin of the voxel $v_k = \{x_k, y_k, z_k\}$ that assumes a value given by a uniform scalar quantizer $Q(v_k, \Delta)$

$$Q(v_k, \Delta) = round(\frac{v_k}{\Delta}) \cdot \Delta \tag{4}$$

Extending the QIM based data-hiding method from Section IV to a 3D point cloud gives the ability to embed multiple bits in each point. Data is hidden in the spatial domain by modifying the three-dimensional position vector of each point; hence, we have three degrees of freedom in the selection of reconstruction points as shown in Fig. 4.

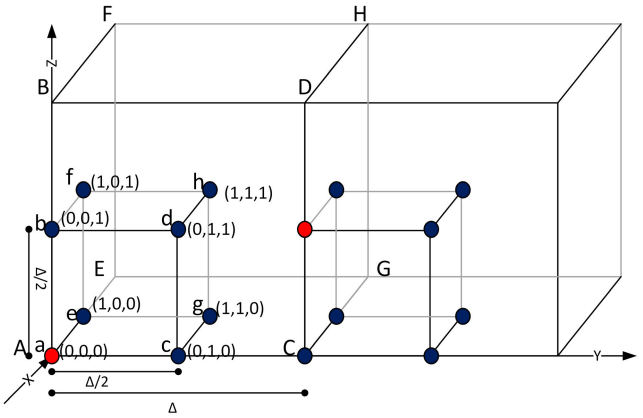


FIGURE 4. Illustration of 3D QIM-based data hiding, here axis representation is in LiDAR frame.

The non-intersecting nature of the reconstruction points results in host-signal interference rejection [17]. Based on the hidden message tuple, m_k, to be embedded, here $m_k = \{m_{xk}, m_{yk}, m_{zk}\}$ the vertex can be moved around a fixed inner cube of a given dither size. The resulting watermarked signal for 3D QIM can be represented as:

$$s_w(s_{v_k}, m_k) = q_{m_k}(s_{v_k}, \Delta) \tag{5}$$

where $q_{m_k}(\cdot)$, denotes 3D QIM quantizer which is expressed as:

$$q_{m_k}(v_k, \Delta) = round(\frac{v_k}{\Delta}) \cdot \Delta \pm \frac{\Delta}{2} \cdot \sqrt{\begin{bmatrix} m_{xk} \\ m_{yk} \\ m_{zk} \end{bmatrix}} \tag{6}$$

If we embedded three bits per host-signal sample to take advantage of the three-dimensional spread of points, with the embedding rate $R = 3$ *bits/sample*, the embedded message m_k would assume 2^R values. The range of m_k determines the count of the ensemble of quantizers hence the quantizer ensemble will have eight values $q_i \in \{q_1, q_2, q_3, \cdots, q_8\}$ in this case. Each one of these eight quantizers shifts the vertex point at A in Fig. 4 to one of the eight vertices $\{a, b, c, d, e, f, g, h\}$ within the inner cube. If, for example, all the points of a 3D point cloud are arranged in sequential order, Fig. 4 represents the first two voxels of the point cloud. If the point cloud is quantized with a step-size Δ, the points within these first two voxels are represented by vertices A and C. In the proposed 3D QIM method, the position of the vertex is moved within an inner hypercube of size $\Delta/2$ based on the embedded message, which is the sequence number of the voxel, i.e., 0 or 1. This shift in the vertex position is depicted by the red circle in Fig. 4. The proposed method of moving the vertex within an inner hypercube does not increase the vertex count in comparison to a normal quantization and hence does not introduce any additional transmission overhead.

V. ATTACK MODELING

Attacks on LiDAR sensors used in autonomous vehicle applications such as localization and perception can be broadly divided into two categories:

1) Regular-channel attacks at sensor level: Sensor saturation, spoofing.
2) Transmission-channel attacks at interface level: Point cloud tampering or deformation.

Regular-channel attacks such as sensor saturation (flooding the target with bright light) and relay and replay attacks (capturing and re-sending the target LiDAR pulse sequence) can be launched external to the vehicle but need precise knowledge of the target LiDAR pulse sequence, receiving angles, and listening time interval [5], [7]. These attacks could be nullified by introducing some pre-processing steps like random probing, correlation, and voting-based confidence estimators. The proposed data hiding-based method is unable to detect regular-channel attacks.

For transmission-channel attacks, which can be launched from inside the vehicle, creating a fake scene could be as simple as copying or deleting a section of the point cloud at the desired location. These insider attacks can be launched with ease in real-time and can have a maximum impact on vehicle decision making if the ADAS core algorithms are designed on the assumption that the sensor data is credible. Most of the object-detection and classification algorithms in the data analysis pipeline are deep-learning-based and are run or inferenced in real-time. These deep-learning models do not differentiate between a fake object and a real object, which could result in erroneous object detections on tampered data. In this section, we describe the attack model for transmission-channel attacks. Transmission-channel attacks happen at the edge system when a hacker gets access to the network interfaces or the decision-making control unit. A hacker could modify the data or point cloud in real-time by some simple operations like copying the existing targets from the point cloud and pasting them in the direct path of the vehicle, which could prompt the vehicle to come to a sudden halt. If the point-cloud tampering is not detected before the inference engine runs on the raw data, it will put more burden on the decision-making logic (the ADAS unit) as it has to incorporate more checks and balances, thus increasing the processing time. Moreover, the ADAS outputs are also expected to be wrong. If we could detect and localize tampering in real-time, then that would ensure the integrity of the sensor data and therefore guarantee the expected ADAS performance.

A. ATTACK VECTORS

We have identified two main attack vectors for transmission-channel attacks, which require not only the detection of tampering but also the specific location of the tampering to neglect that area in the decision-making process.

1) **Fake Object Insertion** (FOI): A fake target is inserted in the direct path of the vehicle.
2) **Target Object Deletion** (TOD): An existing target in the path of the vehicle is removed.

VI. COUNTERMEASURES TO TRANSMISSION CHANNEL ATTACKS

To counter the above-mentioned attack scenarios on the transmission channel between the LiDAR sensor and the ADAS, we propose a QIM-based data-hiding method for tamper detection and localization. Shown in Fig. 5 is the block diagram of the proposed method. We divide the framework into an information-hiding processing block at the LiDAR sensor unit and a point-cloud verification and tamper-detection and localization processing block in the ADAS unit. The information-hiding processing framework that is implemented inside the sensor embeds a binary watermark in the raw point cloud, introducing a negligible distortion. After this step, the embedded point cloud is transmitted over the vehicle network. The watermarked point cloud can be directly worked on by the ADAS core to detect and track objects. The integrity-verification processing block runs in parallel to perform integrity checks on the point cloud data in real-time and inputs its decision to the ADAS unit. This verification-processing block localizes the tampered region once it determines that the point cloud is tampered. This approach secures the point cloud against any transmission-channel-intrusion insider attacks, which are hard to detect at the data inference stage.

A. IMPLEMENTATION DETAILS

In the QIM-based information-hiding processing framework, which runs in close vicinity to the physical sensor, the LiDAR point cloud is filtered to capture forward-looking points or the front camera view. This step can be skipped if the surround-view point cloud is required by the application. The resulting points are quantized based on the predefined step size Δ. After basic quantization, each voxel vertex is shifted to one of the eight positions at a minimum distance of $\Delta/2$ based on a binary message vector as discussed in section IV-A. For simplicity in this study, a repeating message sequence $\in \{0, 1, 2, 3, 4, 5, 6, 7\}$ was chosen. At each sample with index i the message value is given by:

$$m = i \ (mod \ (8)).$$

The computational complexity of the implemented algorithm is $O(3N)$ for N samples, as embedding each bit per sample is $O(1)$.

For the point-cloud verification framework, a blind watermark extraction mechanism is used, that is, the original point cloud is not used for the watermark extraction process. In the verification block, the embedded message is extracted by quantizing the received signal with the same step-size $\Delta/2$ and selecting the nearest reconstruction point. For simplicity, we have assumed that the verification block is aware of the repeating message-embedding pattern. If a dynamic message embedding is needed, the required pattern can be communicated to the verification block through any selected vehicle

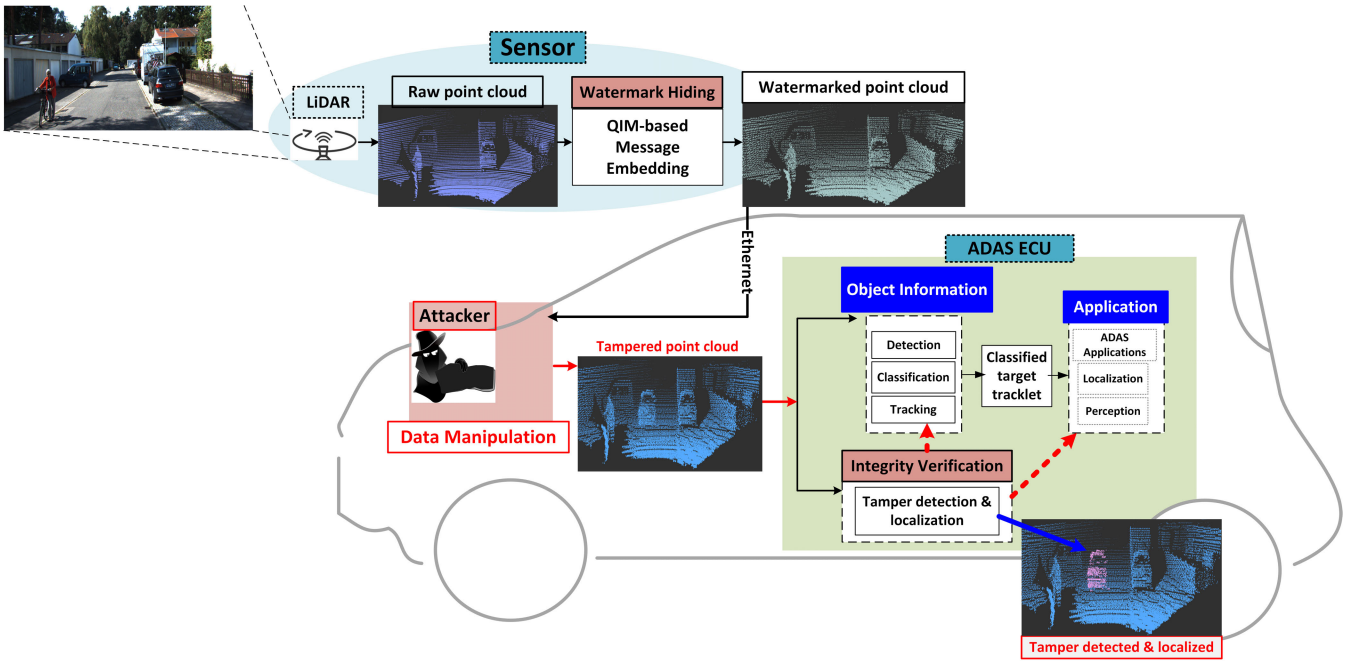


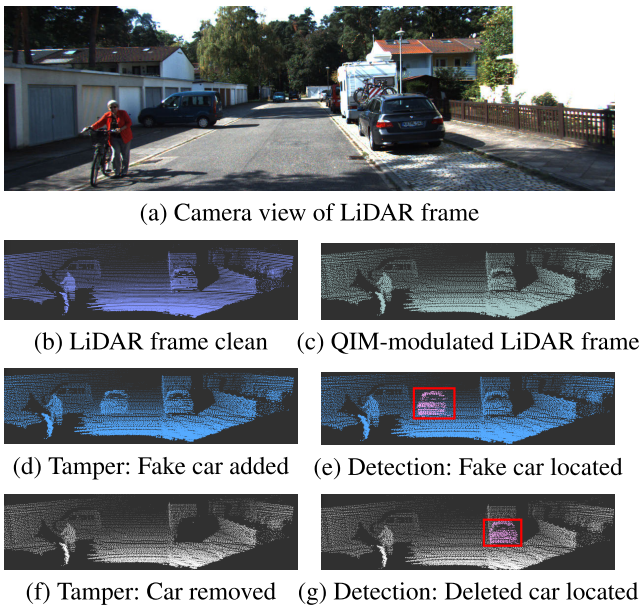
FIGURE 5. Block diagram of the proposed QIM-based tamper resistant framework.

interface, as shown in Fig. 5. Based on the correlation values and pattern matching between the embedded and extracted messages, the indices of the received signal where the embedded and extracted messages do not match are determined. From these indices, the corresponding LiDAR frame points are traced and localized as tampered. To measure the accuracy of tamper localization, the Hausdorff distance [12] is computed between the bounding boxes of the points detected as tampered against the ground truth bounding boxes.

B. PERFORMANCE EVALUATION

To evaluate the performance of the proposed framework, we used KITTI vision benchmark data collected using a 64-channel Velodyne HDL-64E LiDAR running at 10 Hz. The KITTI offers a labeled dataset with the location information of the objects in the data frame. The majority of deep-learning-based object detection and classification models for training and performance benchmarking rely on this dataset. We also chose this dataset to evaluate the performance of the proposed integrity verification and tamper-resistant method to keep the analysis as close as possible to real-world autonomous driving scenarios.

A fake object insertion (FOI) is simulated on the fly by copying a real target object's points from the LiDAR frame at a different location. Similarly, the other attack vector of target object deletion (TOD) is also simulated by removing points in the frame that represent a real labeled target object. These two techniques could be combined to move the targets to a different location, which could be considered as another attack vector. A sample representation of the LiDAR data from the KITTI dataset is shown in Fig. 6. It visualizes the fake object insertion and known target deletion along with



(a) Camera view of LiDAR frame

(b) LiDAR frame clean (c) QIM-modulated LiDAR frame

(d) Tamper: Fake car added (e) Detection: Fake car located

(f) Tamper: Car removed (g) Detection: Deleted car located

FIGURE 6. Attack models and tamper detection and localization results.

accurate detection and localization. To further understand the effects of channel noise, random Gaussian noise of varying variance is added globally to the LiDAR frame.

The minimum distance d_{min} between two reconstruction points measures the size of the noise vector that can be tolerated by the system [17]. For 1D QIM embedding this distance is $\Delta/2$ (as shown in Fig. 3). If we set a limit on message embedding distortion by choosing a fixed quantization step size Δ or, in other words, set a constraint that the composite signal in Eq. (5) should be closely equal to the s_{v_k} $\forall m_k$,

then the message detection accuracy would be high when channel noise is bounded by:

$$d_{min}^2 > 4 \cdot N \cdot \sigma_n^2 \tag{7}$$

where σ is the noise standard deviation and N is the number of dimensions. Our experimental results confirmed that the proposed algorithm adheres to this theoretical noise limit.

VII. EXPERIMENTAL RESULTS

Performance of the proposed framework is evaluated on KITTI's 3D object detection benchmark training dataset. For performance evaluation, LiDAR frames with cars in close proximity to the ego vehicle with no occlusions and less truncation are selected. Motivation behind this selection criteria is to aid the visual inspection of the simulated FOI and TOD attack vectors and to precisely evaluate the tamper localization accuracy in a controlled environment. To this end, first 1000 frames of KITTI's dataset are analyzed that resulted in 67 frames satisfying selection criteria. It is important to highlight that the proposed framework is applicable to all the LiDAR frames in the KITTI's dataset and LiDAR frame selection criteria is not the limitation of the proposed system. It is rather used to have more meaningful and fair performance analysis.

The performance of the proposed method is evaluated using four experiments, ranging from investigating the impact on the object detection performance of the ADAS unit of embedding induced distortion to embedding strength analysis and robustness in the presence of additive noise.

A. IMPACT ON ADAS PERFORMANCE OF EMBEDDING DISTORTION

The primary goal of this experiment is to investigate impact of embedding distortion on ADAS functionality. Specifically, this experiment studies the impact of QIM-based data hiding distortion on the performance of the object-detection and -recognition algorithms. The motivation behind using an object detection algorithm as a key performance indicator in this experiment is because it provides a direct error measurement in terms of distance between object(s) in the original point cloud and corresponding object(s) in the watermarked point cloud. In other LiDAR applications such as simultaneous localization and mapping (SLAM) and object tracking the message embedding distortion is estimated through indirect methods. In these methods, distortion is estimated as sensor bias and often gets compensated for or canceled based on the filters used in SLAM [18].

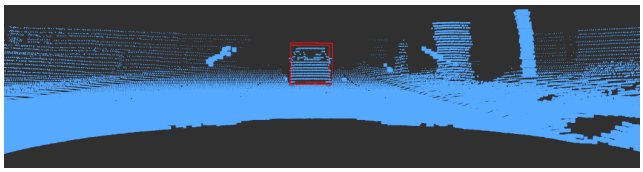
To verify the effect of embedding induced distortion on object detection, as a first step, we ran inference on a selected KITTI dataset frame processed using 3D QIM with nine different step sizes using a pre-trained 3D FCN [19] deep-learning model. We used an existing implementation of the 3D FCN which was pre-trained on raw KITTI data frames [20] for this experiment. From the KITTI training data set, we selected a frame in which a target vehicle is within a 50 m range with zero heading angle. The selected frame is processed using 3D-QIM with nine different step-sizes $\Delta \in \{1, 4, 6, 8, 10, 30, 40, 50\, cm\}$. The resulting frames are run individually through the 3D FCN deep-learning model inference engine, and the resulting bounding box prediction is compared with the ground-truth bounding box. The deviation in terms of the Hausdorff distance between the ground truth and the predicted bounding boxes is compared. The results depicted in Table 1, show that the inferencing of the deep-learning model resulted in good accuracy for a step-size of 30 cm and below. It can be observed from Fig. 7 that the shape of the raw point cloud shown in Fig. 7 (a) with a distinguishable car in the red bounding box (ground truth) deteriorates as we increase the step size Δ. It can be observed that the green bounding box corresponding to the model prediction starts moving away from the red box corresponding to the ground truth as the step-size increases beyond 30 cm. As we move farther, with the 64-channel LiDAR data, the number of points representing a target becomes much smaller and falls into single digits. For those labels, we observed a variation in prediction from the raw frame to the QIM-modulated frame even at a smaller $\Delta = 5$ cm. Since the probability of false alarms is high for objects with low reflections points, they are generally filtered by the decision-making process. The range at which this filtering occurs depends on the LiDAR resolution.

TABLE 1. QIM-induced distortion at different step-sizes.

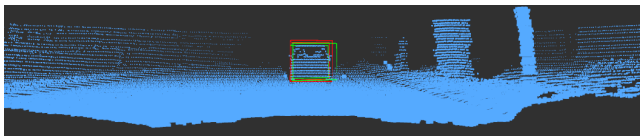
Δ/Step size (cm)	Bounding box shape distortion (m)
1	0.44
4	0.78
6	0.73
8	0.78
10	0.73
30	0.90
35	17.36
40	22.70
50	28.08

To further understand the effect of message embedding induced distortion on LiDAR object detection accuracy, we tested watermarked LiDAR frames on another 3D object detection model called VoxelNet [21]. VoxelNet is an end to end deep learning network that stacks the voxelization, convolution, and region proposal network (RPN) operations to detect and localize objects from the raw 3D LiDAR point cloud and its performance is claimed to be better than 3D FCN [21]. The VoxelNet implementation determines object detection precision based on the 70% overlap of the predicted 3D and 2D (bird's eye view) bounding boxes with their corresponding ground truth. In this experiment, we used an existing implementation of the VoxelNet that is trained on KITTI benchmark data to detect cars [22]. We chose a validation set of 25 frames that fall under the easy detection category defined by KITTI and ran inference on them using a pre-trained model checkpoint. A base-line average precision

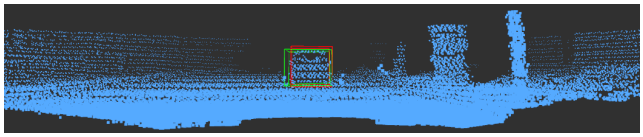
(a) Camera view of LiDAR point cloud



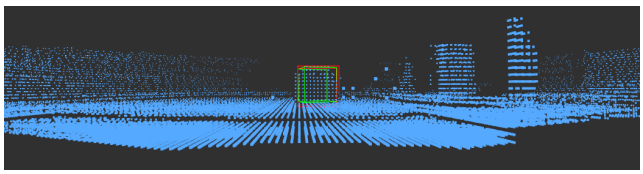
(b) LiDAR: Raw point cloud



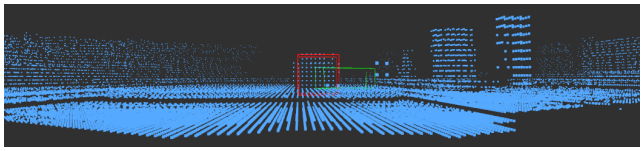
(c) QIM-modulated point cloud with $\Delta = 5\ cm$



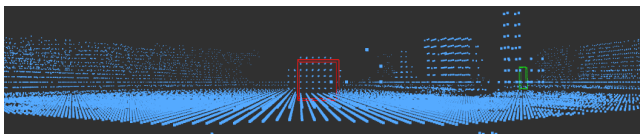
(d) QIM-modulated point cloud with $\Delta = 8\ cm$



(e) QIM-modulated point cloud with $\Delta = 30\ cm$



(f) QIM-modulated point cloud with $\Delta = 35\ cm$



(g) QIM-modulated point cloud with $\Delta = 50\ cm$

FIGURE 7. Bounding box estimation of a ground truth label at different QIM-embedding step sizes.

score of VoxelNet is established by running inference on the selected validation set multiple times. Same set of 25 LiDAR frames watermarked using 3D-QIM with different step-sizes $\Delta \in \{1, 4, 6, 8, 10, 20, 30, 40, 50\ cm\}$ are then generated. VoxelNet inference is executed on each individual watermarked dataset to get the average precision score as shown in Table 2. The first row ($\Delta = 0$) of Table 2 shows the average precision of the model for raw data frames. For the raw data

TABLE 2. VoxelNet average precision scores (in %) for car detection, at different step-sizes.

Δ/Step size (cm)	Bird's Eye View	3D Detection
0	96.92	77.38
1	96.59	83.05
4	96.96	73.12
6	96.21	72.90
8	97.61	75.18
10	89.40	65.30
20	83.50	57.39
30	73.42	34.65
35	54.77	23.51
40	42.01	13.40
50	12.20	2.38

frames the average precision of 2D and 3D detections was 96.92% and 77.38% respectively.

It can be observed from Table 2 that for watermarked frames there is no significant deterioration in the average precision of the model for up-to a step size of 8 *cm*. Within this range, the bird's eye view detection scores had a mean of 96.85% with a 0.51% standard deviation. The 3D detection scores averaged at 76.32% with a standard deviation of 4.17%. The additional spread in 3D prediction scores in comparison to the 2D scores could be attributed to the fact that the 3D detection is a more challenging task as it requires more accurate localization of shapes in 3D space [23] and hence model needs to be trained on a larger data-set to be able to generalize well. It can also be observed from Table 2 that for both methods, the average precision score of the VoxelNet model decreases significantly as the step size goes above 8 *cm*. Modifying the voxel dimensions of the model and training the model on the modulated point cloud could improve the model performance in general. Nevertheless, since our proposed data-hiding technique provides flexibility to select the desired embedding distortion level as a function of the step-size, for any given application an optimum step size could be selected based on the empirical evaluation of the model and application needs.

B. EMBEDDING DISTORTION ANALYSIS

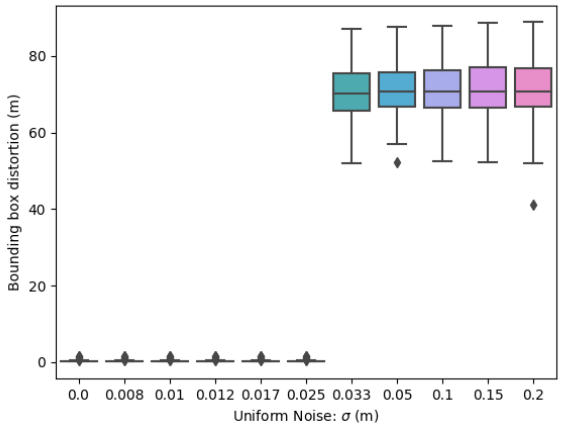
This experiment is designed to investigate the impact of single- vs multiple-bit message embedding on tamper localization accuracy. Specifically, for a given step size $\Delta = 10\ cm$, we compared bounding box prediction results of the following three QIM message embedding methods under various added noise levels:

1) 1*D* QIM with one-bit embedding along x axis
2) 2*D* QIM with two-bit embedding along x,y axes
3) 3*D* QIM with three-bit embedding along x,y,z axes

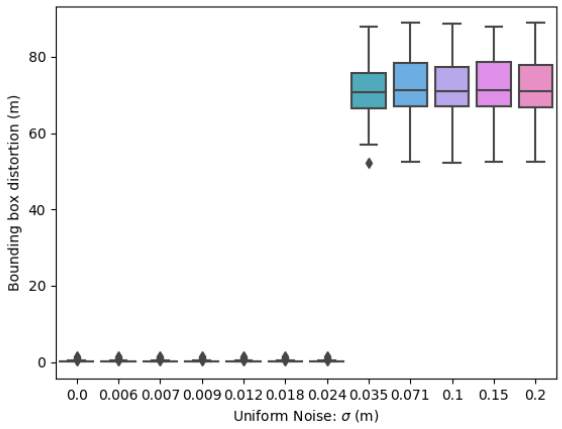
Shown in Fig. 9 is the performance of different bit-embedding methods in localizing the tampered area in a point cloud forged with FOI and global uniform noise.

It is observed that the tampered point localization accuracy decreased as the additional uniform noise levels increased.

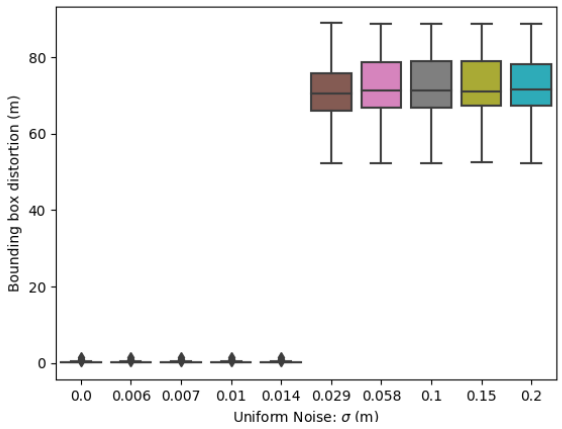
For a given step size $\Delta = 10\ cm$, the tamper localization accuracy of $1D$ QIM is good up to an additional uniform noise of $\sigma = 2.5\ cm$, $2D$ QIM is good up to $\sigma = 2.4\ cm$, and $3D$ QIM is good up to $\sigma = 1.4\ cm$. The noise tolerance values for a given Δ are within the limits of the σ values for $N = 1, 2, and\ 3$ as per Eq. (7). Fig. 8 shows the performance of different bit-embedding QIM methods in the presence of additional Gaussian noise. It is observed that there is no significant difference in accuracy between multi- and single-bit embedding.



(a) One-bit embedding



(b) Two-bit embedding



(c) Three-bit embedding

FIGURE 8. Bounding box distortion analysis for different bit-embedding schemes under uniform additive noise attack.

One of the goals of the data hiding is to detect tampering under high lossy conditions such as compression. The data-hiding method should detect any global intentional attacks on the integrity of data such as sensor saturation by external noise addition or affine transforms in multiple dimensions, along with local attacks like FOI and TOD. Though single-bit embedding offers higher robustness to noise levels, it will not detect targeted attacks in multiple dimensions. Hence, we propose using $3D$ QIM for autonomous vehicle applications.

C. ROBUSTNESS ANALYSIS

After choosing a range of step-sizes that result in acceptable embedding-induced distortion to analyze the tamper detection and localization accuracy of the proposed 3D QIM method, three attributes are considered:

1) Bit error rate (BER) of embedding
2) Tamper localization distortion
3) Tamper detection false-alarm rate f_{ar}

In different experiments, these three attributes were measured under various variance levels of both uniform and Gaussian noise addition, along with analysis of the TOD and FOI attack vectors and results.

1) BIT ERROR RATE

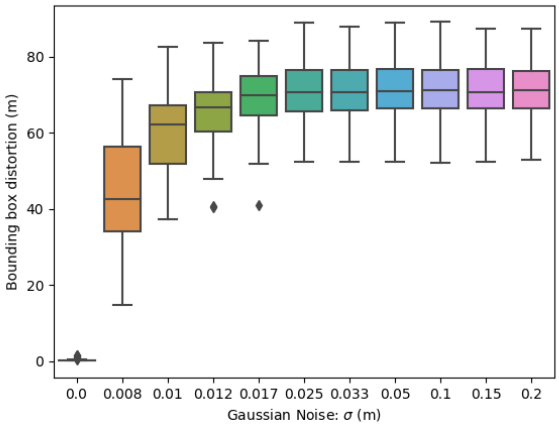
This experiment is designed to measure the performance of the proposed method in terms of bit error rate (BER) in the presence of additive channel-noise. The output of the message-decoding step in our method is a decoded message bit stream from the received signal $\hat{M} = \{\hat{m}_1, \hat{m}_2, \cdots, \hat{m}_N\}$.

In this experiment, the bit error rate is measured for each LiDAR frame after the decoding step, where each extracted bit from the received LiDAR frame, $\hat{m}_i$, is compared with the embedded message bit, m_i, where $i = 1, 2, \cdots, N$. The BER is calculated as follows:

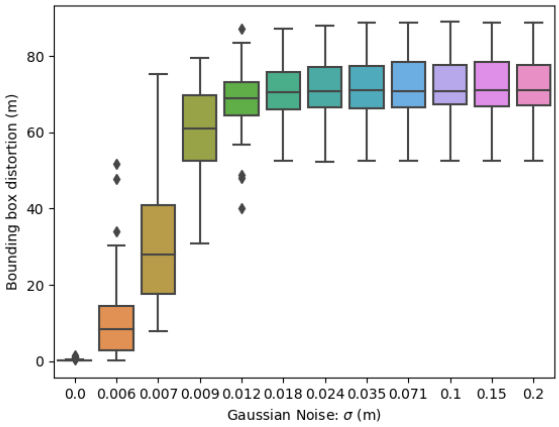
$$BER = \frac{\sum_{i=1}^{N} 1_{\hat{m}_i \neq m_i}}{N}$$

where 1 is the indicator function and N is the size of the decoded message bit stream.

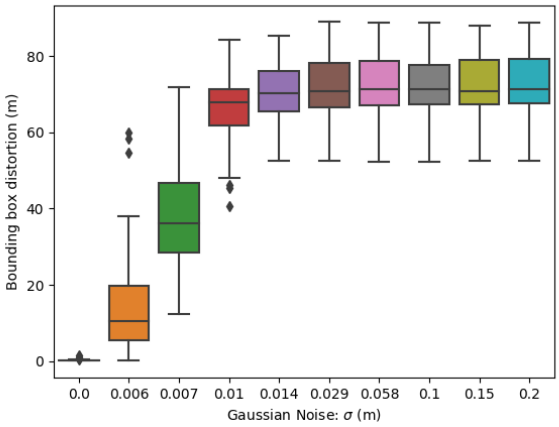
To achieve this goal, Gaussian noise with different standard deviations and uniform noise with different upper bounds are added into watermarked point-cloud frames separately, and the impact of the additive noise attack on BER performance is evaluated. For a clean QIM-modulated frame, the BER is close to 0%. As we tamper the data by the attack vectors FOI and TOD, a constant BER of close to 2% is observed at zero added noise. As the added noise value increases, the proposed three-bit QIM quantization method maintains a bit error rate of less than 2%, for an added uniform noise of $\sigma < \Delta/6.93$). This noise threshold as defined by Eq. (7) is $\{0.7, 2.9, 5.1\ cm\}$, respectively, for step-sizes of $\Delta \in \{5, 20, 35\ cm\}$. It can be observed from Fig. 10 that



(a) One-bit embedding

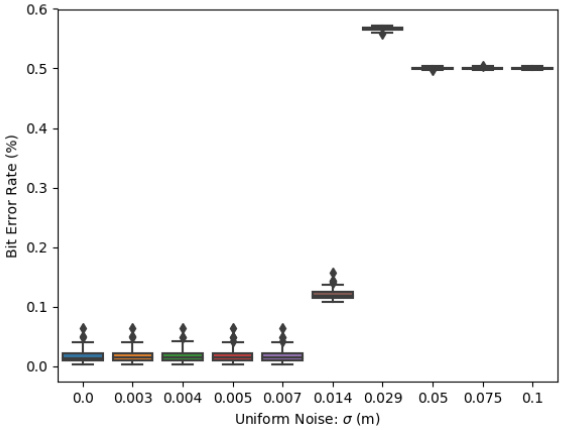


(b) Two-bit embedding

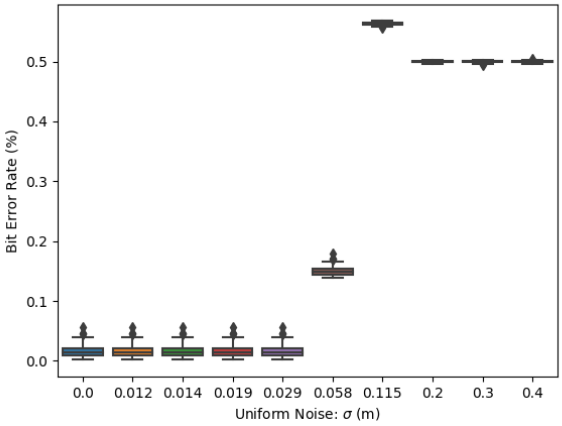


(c) Three-bit embedding

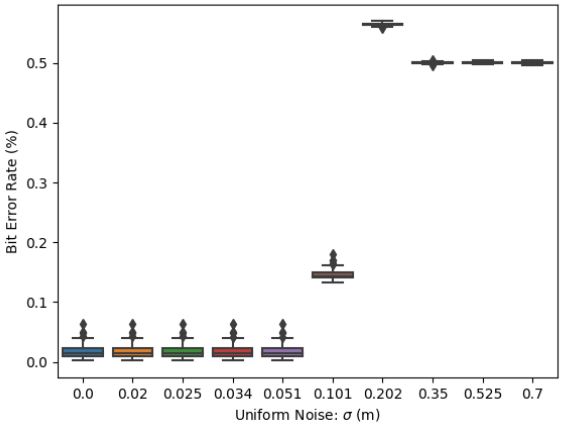
FIGURE 9. Bounding box distortion analysis for different bit-embedding schemes under Gaussian additive noise attack.



(a) $\Delta = 5\ cm$



(b) $\Delta = 20\ cm$



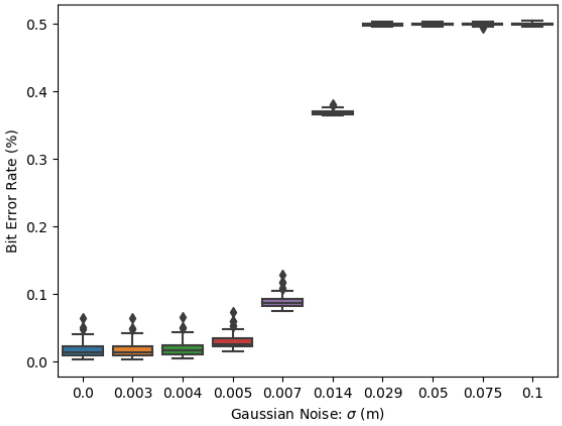
(c) $\Delta = 35\ cm$

FIGURE 10. Bit error rate of decoded code book for different step sizes and added uniform noise.

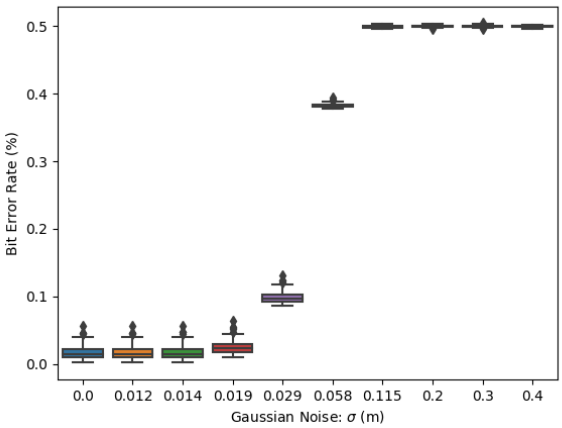
the BER values are within acceptable bounds until the noise levels exceed the threshold defined by Eq. (7). The BER values go high at levels below the threshold bounds defined by Eq. (7) for added Gaussian white noise, as observed in Fig. 11. This change can be attributed to the 32% noise values falling outside the 1σ range in the Gaussian distribution. Figures. 10, and 11 show the BER for one of the attack vectors, FOI. A similar trend is observed for the TOD attack vector.

2) TAMPER DETECTION AND LOCALIZATION

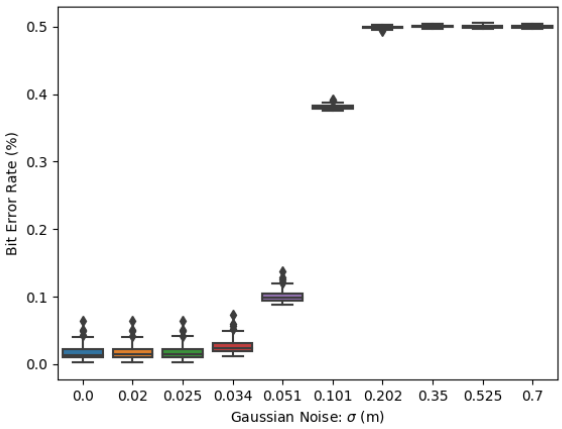
This experiment aims to measure the accuracy of the tamper localization feature of the proposed method and how the accuracy is affected by added channel noise. As part of the



(a) $\Delta = 5\ cm$

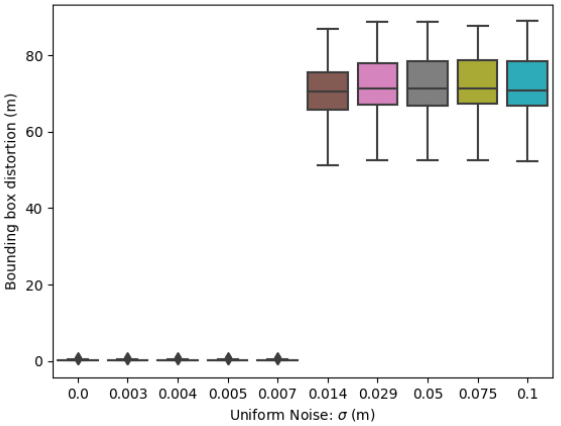


(b) $\Delta = 20\ cm$

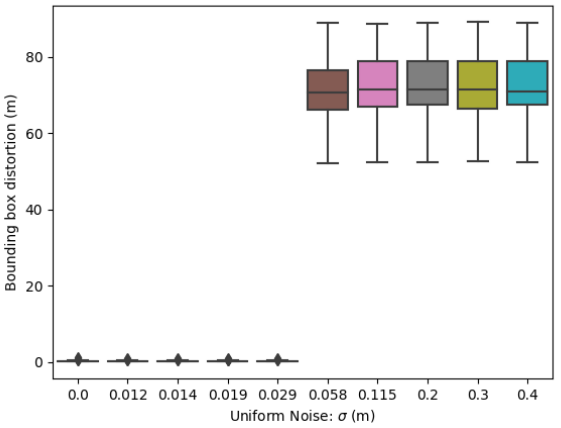


(c) $\Delta = 35\ cm$

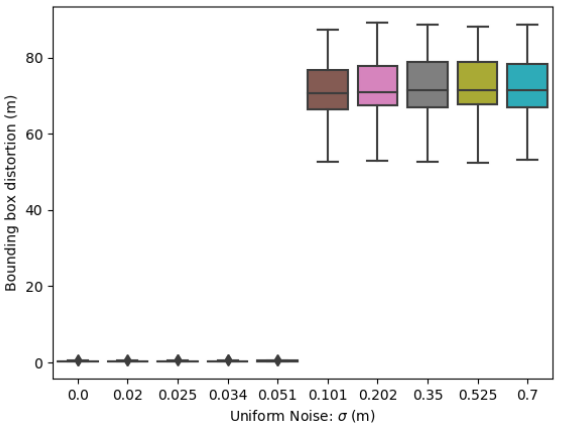
FIGURE 11. Bit error rate of decoded code book for different step sizes and added Gaussian noise.



(a) $\Delta = 5\ cm$



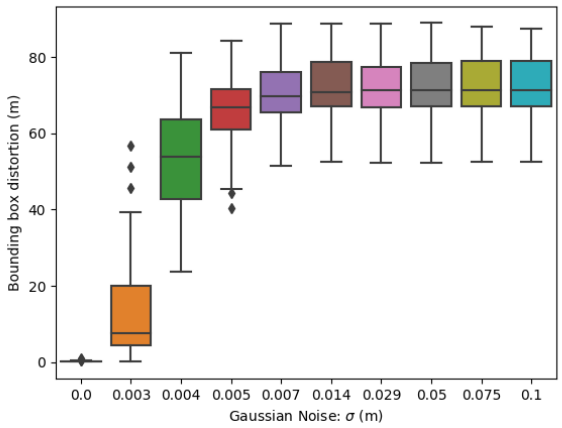
(b) $\Delta = 20\ cm$



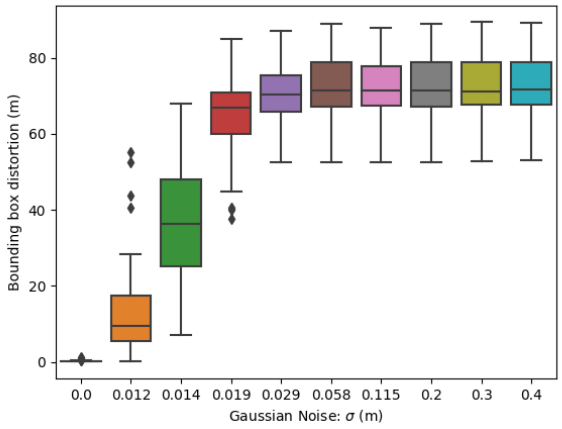
(c) $\Delta = 35\ cm$

FIGURE 12. Bounding box distortion in meters for different step sizes and added uniform noise.

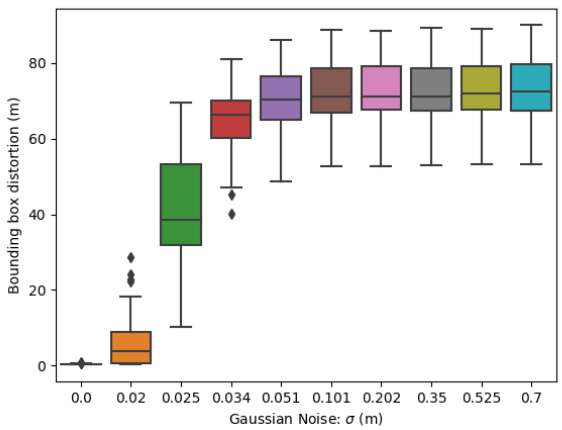
decoding step, the indices of the tampered points are extracted from the received frame. A bounding box enclosing these points is generated, and the corners of this bounding box are compared with the corners of the ground-truth bounding box provided by KITTI to get a measure of their proximity. We calculated the Hausdorff distance between the two corner sets to measure the maximum distance of a given vertex from the ground-truth bounding box to a similar vertex in the predicted bounding box. Smaller values of this localization-distortion attribute suggest higher accuracy of the prediction or, in other words, suggest that the proposed method is able to draw a boundary across the tampered points accurately. We added Gaussian noise with different standard deviations and uniform noise with different upper bounds to the



(a) $\Delta = 5\ cm$



(b) $\Delta = 20\ cm$



(c) $\Delta = 35\ cm$

FIGURE 13. Bounding box distortion in meters for different step sizes and added Gaussian noise.

QIM-modulated frames and measured the effect of added noise on the performance of the localization accuracy.

The added noise is tested at varying levels of σ in the range $\sigma \in \{0.0, \Delta/(10\sqrt{(N)}), \Delta/(8\sqrt{(N)}), \Delta/(6\sqrt{(N)}), \ldots, 2\Delta\}$ and step-size Δ in range $\Delta \in \{5, 10, 20, 30, 35, 40\ cm\}$, where $N = 3$ is the number of bits used for embedding.

Shown in Fig. 12 are the localization distortion box plots for the FOI attack vector measured on point clouds with added uniform noise. It is observed that the localization distortion is less than 2 cm as long as the noise level $\sigma < d_{min}/(2*\sqrt{(N)})$, where $d_{min} = \Delta/2$ and $N = 3$. These results are in sync with the theoretical limits given by Eq. (7) and demonstrate that the proposed method can localize tampering in the point cloud accurately in the presence of bounded noise. The performance of the proposed method under added Gaussian noise is shown in Fig. 10. For added Gaussian noise, we observed that the localization distortion is less than 2 cm as long as the noise level $\sigma < d_{min}/(5 * \sqrt{(N)})$, where $d_{min} = \Delta/2$ and $N = 3$. Again, this behavior of low robustness to noise can be attributed to the noise samples that fall outside the 1σ range in Gaussian noise. Similar trends were observed for the TOD attack vector.

3) FALSE-ALARM RATE ANALYSIS

This experiment aims to measure the false-alarm rate of the proposed method in detecting the tampered point cloud in the presence of additive noise. In this experiment, we tracked the number of frames that our algorithm falsely detected as tampered when it was given a clean frame. The tamper detection false-alarm rate (f_{ar}) is calculated as the ratio of number of clean frames falsely detected as tampered ($N_{falseTamper}$) to total number of clean frames (N_{clean}) given as

$$f_{ar} = N_{falseTamper}/N_{clean}$$

In this test, f_{ar} stayed at 0% when there was no added channel noise. In other words, the proposed model detected the presence of both the FOI and TOD attack vectors accurately when there was no added channel noise. When noise was added along with the attack vectors, for added uniform noise, f_{ar} stayed at 0% for $\sigma < d_{min}/(2 * \sqrt{(N)})$, where $d_{min} = \Delta/2$, $N = 3$ and $\Delta \in \{5, 10, 20, 30, 35, 40\ cm\}$.

As the noise σ level increased beyond that threshold, the f_{ar} value jumped to 100%, as shown in Table 3. It is also observed that the f_{ar} value increased to 100% at lower thresholds of $\sigma > d_{min}/(10 * \sqrt{(N)})$ in the case of Gaussian added noise, and this could be attributed to the noise values grater than 1σ.

It can be observed from Table 3 that at a given Δ within acceptable distortion bound, the proposed method can

TABLE 3. False alarm rates at different added noise levels for different step-sizes.

$\Delta = 5$	$\sigma(\ cm)$	0	0.3	0.4	0.5	0.7	1.4
	Gaussian	0	0.95	1.0	1.0	1.0	1.0
	uniform	0	0	0	0	0	1.0
$\Delta = 20$	$\sigma(\ cm)$	0	1.2	1.4	1.9	2.9	5.8
	Gaussian	0	0.94	1.0	1.0	1.0	1.0
	uniform	0	0	0	0	0	1.0
$\Delta = 35$	$\sigma(\ cm)$	0	2.0	2.5	3.4	5.1	10.1
	Gaussian	0	0.55	1.0	1.0	1.0	1.0
	uniform	0	0	0	0	0	1.0

achieve 100% accurate detection and localization. A similar trend was observed for the TOD attack vector. In automotive applications, the ethernet and other local networks are not susceptible to high channel noise; hence, our proposed method is expected to achieve the desired performance.

VIII. CONCLUSION

Insider attacks on raw sensor data are a real threat to autonomous vehicles. In this paper, we chose the LiDAR sensor, which is extensively used in autonomous vehicles, and proposed a novel approach to detect and localize tampering of the raw data from the LiDAR sensor. We demonstrated that the proposed method can detect and localize tampering to the real-world benchmark KITTI dataset with a 100% success rate as long as additive noise is less than the quantization step-size. We also established the QIM embedding-induced distortion thresholds for proper detection using 3D FCN and VoxelNet deep learning models. This work can be extended to other 3D point-cloud generating sensors as well and can be further fortified by using dynamic message embedding techniques.

REFERENCES

[1] S. Checkoway, D. McCoy, D. McCoy, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. 20th USENIX Conf. Secur.*, Aug. 2011, p. 6.

[2] A. M. Wyglinski, X. Huang, T. Padir, L. Lai, T. R. Eisenbarth, and K. Venkatasubramanian, "Security of autonomous systems employing embedded computing and sensors," *IEEE Micro*, vol. 33, no. 1, pp. 80–86, Jan. 2013.

[3] H. Malik, K. P. Subbalakshmi, and R. Chandramouli, "Nonparametric steganalysis of QIM data hiding using approximate entropy," *Proc. SPIE*, vol. 6819, no. 2, Mar. 2008, Art. no. 681914.

[4] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[5] 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* (Lecture Notes in Computer Science), vol. 10529, W. Fischer and N. Homma, Eds. Cham, Switzerland: Springer, 2017, pp. 445–467.

[6] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[7] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and LiDAR," in *Proc. Black Hat Eur.*, Nov. 2015, pp. 1–13.

[8] Y. Chan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," in *Proc. ACM DEFCON*, Aug. 2016, pp. 1–70.

[9] H. Shin, Y. Son, Y. Park, Y. Kwon, and Y. Kim, "Sampling race: Bypassing timing-based analog active sensor spoofing detection on analog-digital systems," in *Proc. Woot*, 2016, pp. 1–11.

[10] K. Bahirat and B. Prabhakaran, "A study on lidar data forensics," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 679–684.

[11] B. Glas, J. Guajardo, H. Hacioglu, M. Ihle, K. Wehefritz, and A. Yavuz, "Signal-based automotive communication security and its interplay with safety requirements," in *Proc. Embedded Secur. Cars Conf.*, Nov. 2012, pp. 1–5.

[12] P. Agarwal and B. Prabhakaran, "Robust blind watermarking of point-sampled geometry," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 36–48, Mar. 2009.

[13] H. Malik, K. Subbalakshmi, and R. Chandramouli, "Steganalysis: Trends challenges," in *Multimedia Forensics and Security*, Calgary, AB, Canada: Idea Group Inc., 2008.

[14] B. Chen and G. W. Wornell, "Quantization index modulation methods for digital watermarking and information embedding of multimedia," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 27, nos. 1–2, pp. 7–33, Feb. 2001.

[15] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2008.

[16] E. Joachim and G. Bernd, *Informed Watermarking*. New York, NY, USA: Kluwer Academic Publishers, 2002.

[17] B. Chen and G. W. Wornell, "Digital watermarking and information embedding using dither modulation," in *Proc. IEEE 2nd Workshop Multimedia Signal Process.*, Dec. 1998, pp. 273–278.

[18] L. D. L. Perera, W. S. Wijesoma, S. Challa, and M. D. Adams, "Sensor bias correction in simultaneous localization and mapping," in *Proc. 6th Int. Conf. Inf. Fusion*, vol. 1, Jul. 2003, pp. 151–158.

[19] B. Li, "3D fully convolutional network for vehicle detection in point cloud," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*,Sep. 2017, pp. 1513–1518.

[20] Y. Tsuji, "3D CNN tensorflow," GitHub, Tech. Rep., 2018. [Online]. Available: https://github.com/yukitsuji/3D_CNN_tensorflow

[21] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," *CoRR*, 2017, *arXiv:1711.06396*. [Online]. Available: https://arxiv.org/abs/1711.06396

[22] Q. Huang, "Voxelnet," GitHub, Tech. Rep., 2018. [Online]. Available: https://github.com/qianguih/voxelnet

[23] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915.

RAGHU CHANGALVALA received the B.S. degree in electrical and communications engineering from Jawaharlal Nehru Technological University, India, in 2002, and the M.S. degree in electrical engineering from the University of Alaska Fairbanks, in 2005. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Michigan-Dearborn, USA. Since 2005, he has served various organizations in the automotive industry, as a Software Designer and Architect. He is also a Senior Software Engineer with the Autonomous Vehicle Engineering Group, FCA, Auburn Hills, MI, USA. His research interests include V2X communications, cyber-security, and deep learning and its applications to autonomous driving.

HAFIZ MALIK is currently an Associate Professor with the Department of Electrical and Computer Engineering (ECE), University of Michigan-Dearborn. He has published more than 90 articles in leading journals, conferences, and workshops. His research interests include cybersecurity, deepfakes, multimedia forensics, sensor security, information hiding, wireless sensor networks, steganography/steganalysis, pattern recognition, information fusion, and biometric security, which are funded by the National Academies, National Science Foundation, Ford Motor Company, and other agencies. He also organized a Special Session titled "Data Mining in Industrial Applications" within the IEEE Symposium Series on Computational Intelligence (IEEE SSCI), in 2013. He has served as the Vice Chair for the IEEE SEM, Chapter 16, from 2011 to 2016. He is also serving on several technical program committees, including the IEEE AVSS, ICME ICIP, MINES, ISPA, CCNC, ICASSP, and ICC. He is also on the Review Board Committee of the IEEE Technical Committee on Multimedia Communications (MMTC). He organized the Special Track on Doctoral Dissertation in Multimedia in the 6th IEEE International Symposium on Multimedia (ISM), in 2006. He has served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, since August 2014. He has been serving as an Associate Editor for the *Springer Journal of Signal, Image, and Video Processing* (SIVP), since May 2013.