Modelling Perception in Autonomous Vehicles via 3D Convolutional Representations on LiDAR

Hafsa Iqbal[1,2], Damian Campo[1],Pablo Marin-Plaza[2], Lucio Marcenaro[1], David Martin[2] and Carlo Regazzoni[1]
Department of Engineering and Naval architecture (DITEN), University of Genoa, Italy[1]
Intelligent systems lab, Carlos III University of Madrid, Spain[2]
email addresses: {hafsa.iqbal, damian.campo}@edu.unige.it
{lucio.marcenaro, carlo.regazzoni}@unige.it
{pamarinp, dmgomez}@ing.uc3m.es

***Abstract*—This paper proposes an algorithm to model and process streams of LiDAR data under an autonomous vehicle framework. LiDAR is assumed to be an exteroceptive sensor that allows the vehicle to have dynamic 3D scene perception of its surroundings. We employ an encoder-decoder architecture based on 3D-Convolutional layers called 3D Convolution Encoder-Decoder (3D-CED), together with a transfer learning strategy to extract a set of features from point clouds, which are relevant in the context of autonomous driving. The resulting features allow to make predictions of the future point cloud data and detect multiple abstraction level anomalies in controlled scenarios by utilizing a probabilistic switching dynamic model called High Dimensional Markov Jump Particle Filter (HD-MJPF). Moreover, a comparison is provided between piecewise linear, piecewise nonlinear, and nonlinear predictive models at multiple abstraction levels of anomaly detection. Our approach is evaluated with data collected from the LiDAR sensors of the autonomous vehicle while performing certain tasks in a controlled environment.**

***Index Terms*—3D-Convolutional Encoder Decoder, Transfer learning, High Dimensional Markov Jump Particle Filter, Anomaly detection, LSTM, Hierarchical Generalize Dynamic Bayesian Network.**

I. Introduction

Intelligent vehicles that attempt to minimize (or even eliminate) human intervention are known as autonomous or driverless vehicles. These vehicles are endowed with exteroceptive and proprioceptive sensors that allow them to monitor surrounding and internal information, respectively. The usage of heterogeneous sensors, e.g., cameras, LiDAR, Radar, GPS, etc., facilitates the learning of different tasks and leveraging the vehicle's understanding of the context in which it is involved [1], [2], [3]. Recent developments in image processing techniques have enabled the industries to resolve some of the practical complexities of autonomous vehicles (AV) [4], [5]. A vehicle cannot be fully autonomous with a single front camera [6]. It will be like to drive a car without side and back mirrors, which induces a high probability of encountering accidents. However, those type of accidents can be prevented by considering a broader perception of the environment [7]. To drive safely on the road, drivers must maintain a safe distance from the surrounding vehicles/objects to avoid collisions. Therefore, intentionally or unintentionally, drivers perceive a circular region of interest around the vehicle. A similar approach is adapted to bring a higher level of autonomy in vehicles, where the side and back mirrors need to be replaced with multi-cameras (depending on the architecture of vehicles) to provide full coverage around the vehicle [8]. Using more than one camera may introduce overlapping regions among images. Additionally, images may be highly affected by shadowy areas, and illumination levels. These problems can be diminished with LiDAR, which provides a 360-degree coverage around the vehicle and is less affected by weather conditions than conventional images [9]. In 2009, Waymo bet heavily on the use of LiDAR for self-driving vehicles. And in 2019, introduced the first fully self-driving taxi service, which consists of three sensors, i.e., LiDAR, Radar, and a front camera, which has proven to offer high levels of safety in the real world situations [10].

LiDARs are expensive as compared to standard cameras but are becoming more affordable. Recently, Apple has introduced sophisticated LiDAR technology [11] as one of the features in their products, which motivates industries to launch LiDAR sensors at accessible prices, which triggers the research towards such a type of technology. Advances in LiDAR provide a more precise real-time 3D map of the environment, which in-turn enhances the decision-making ability of AVs under challenging conditions.

This paper uses deep learning techniques together with classical machine learning (ML) and signal processing algorithms to develop a self-aware model for AVs. Our model allows a vehicle to make inferences on LiDARs' point cloud and detect multiple abstraction level anomalies w.r.t previously seen experiences. The detection of anomalies is the primary step for autonomous agents to perceive the environment and perform actions accordingly.

The proposed approach is evaluated with the real-time LiDAR data collected from the AV. Apart from AVs, LiDAR has wide applications in forestry, ecological, land classification, and geological applications. The proposed methodology is applicable in these fields with slight modifications. However, the focus of this work is to enable the AV to detect anomalies (unknown segments of point clouds (objects) or unknown dynamics) by using probabilistic inferences. The code for the proposed methodology is shared on "$https : //github.com/Hafsa -$

$Iqbal/LiDAR\_Anomaly\_Detection$".

Detailed contribution. *i)* This paper proposes an algorithm that uses LiDAR sensory data to make inferences about future changes in the surrounding environment and detect anomalies.
ii) Multiple abstraction level anomalies, i.e., Continuous Level (CL), Discrete Level (DL), and Voxel Level (VL), are detected and perform the comparison between piecewise linear (PL), piecewise nonlinear (PNL), and nonlinear (NL) models.
iii) Our approach employs deep learning techniques that allow us to take advantage of 3D CNN architectures together with transfer learning and fine-tuning and adapt them for 3D Reconstruction and an anomaly detection task.
iv) This work aims to develop a framework to analyze the high dimensional LiDARs' point cloud and potentially combine it with approaches already developed for positional [12], video [13], and control [14] data in autonomous transportation systems.
v) An ablation study and additional results (from the KITTI dataset) for anomaly detection are provided to analyze the validity of the proposed methodology.

Paper organization. The rest of the paper is organized as follows: Section II discusses the state of the art that deals with the modeling and understanding of sensory data in the context of autonomous driving. Section III describes in detail the proposed methodology that consists of two phases; training and testing of the proposed methodology, which allows us to perform anomaly detection in LiDAR data. Section IV explains the experimental dataset used for evaluating the proposed methodology. Section V shows the results and provides discussion on the performance of the proposed methodology. Section VI presents the conclusion with the future extensions and applications of the proposed method.

II. Related works

AVs should be capable of identifying new situations and update their predictive models accordingly to make decisions [15]. Many advanced autonomous systems have emerged in the past decades with the ability of detection and mapping in the real environment [16], handover transitions from automation to manual driving [17], overtaking [18] and detection of cracks on roads [19]. Although AVs are constantly improving, there are still some unanswered gaps and open issues preventing the full automation of self-driving vehicles. Recently, many signal processing techniques, together with classical ML and deep learning algorithms, have drawn the attention of the research community and established themselves as a strong contestant against statistical models for developing fully autonomous systems [20], [21]. Information of the depth in 3D point clouds made LiDAR more efficient as compared to cameras [22]. Recently, researchers introduce methods to fuse the LiDAR with cameras to enhance the performance efficiency of the autonomous agents [23], [24], [25].

In recent works [11],[12], techniques based on probabilistic reasoning and ML have been demonstrated to be useful for modeling and estimating the low-dimensional data, e.g., vehicle's positional information and steering angle data. Some works have used LiDAR data to dynamically detect features of surrounding areas in an autonomous driving context. In [26], researchers use LiDAR together with a frontal camera to assess the quality of the roads. Authors in [27], use LiDAR to evaluate the impact of the work zone geometry on traffic operations. The work in [28] proposes a way for estimating lane widths in work zones based on LiDAR information. Additionally, several works [29], [30], [31] have tackled the problem of classifying point clouds by extracting features through Convolutional Neural Networks (CNN). In literature, there are some well-known approaches such as; Volumetric CNNs (e.g., 3D-CNN, FPNN, Vote3D), Multiview CNNs, Spectral CNNs, Feature-based DNNs, and PointNet [32], [33], [34], [35]. However, these approaches suffer from certain limitations, i.e., data sparsity, computational cost, sparse volumes, nontrivial to scene understanding, constrained on power or meshes. These works do not focus on dynamic anomaly detection and tracking of relevant features coming from LiDAR streams in the context of autonomous driving. Therefore, we utilize the 3D-Convolutional feature extraction method to make inferences of future states (features) of LiDAR data in a vehicular context.

3D point clouds are challenging to organize in grid format due to specific characteristics such as; disordered, irregular, varying number of points, and invariant to point ordering. There are various approaches to encode the 3D point clouds into a dense and gird-like structure [32], [36]. In this paper, a simple and efficient approach called 3D-voxelization [37] is employed and provided as an input to the 3D-Convolutional Encoder (CE) and obtained as an output from the 3D-Convolutional Decoder (CD). The 3D-Convolutional Encoder-Decoder (CED) network is employed to reduce the dimensionality of LiDAR sensory data and extract the features. Transfer learning, which has already proven to be an important strategy in autonomous driving [38], is employed to take well-established features from point cloud classification tasks and use them to leverage features from an autonomous driving environment. Transfer learning potentially enables the algorithm to continuously learn from new situations and update the model without training the 3D-CED from scratch. These features are used in a probabilistic framework to make inferences of future states and detect multiple abstraction level anomalies. In surveillance systems and AVs [39], [40], the detection of anomalies is an interesting topic for researchers. Usually, anomaly detection is performed with low-dimensional data, e.g., positional or control data, and not much work has been done in the specific case of LiDAR for AVs. Existing methods based on deep learning techniques do not provide a solution that uses deep-features to make inferences on LiDARs' point clouds and detect anomalies probabilistically in the domain of autonomous driving. In this work, a clustering algorithm is used to learn a set of piecewise dynamical models, which allow us to represent the entire problem as a Hierarchical-Generalize Dynamic Bayesian Network (H-GDBN) that relates the current and future state at different levels of inference. Predictions of future state vectors are handled by a Markov Jump Particle Filter (MJPF), which is a type of *Switching Linear Dynamical Systems* [41], [42] that uses GSs to make multilevel (continuous and discrete) inferences by employing a Particle Filter (PF) coupled with a bank of Kalman Filters (KFs). In this work, a modified MJPF called High Dimensional

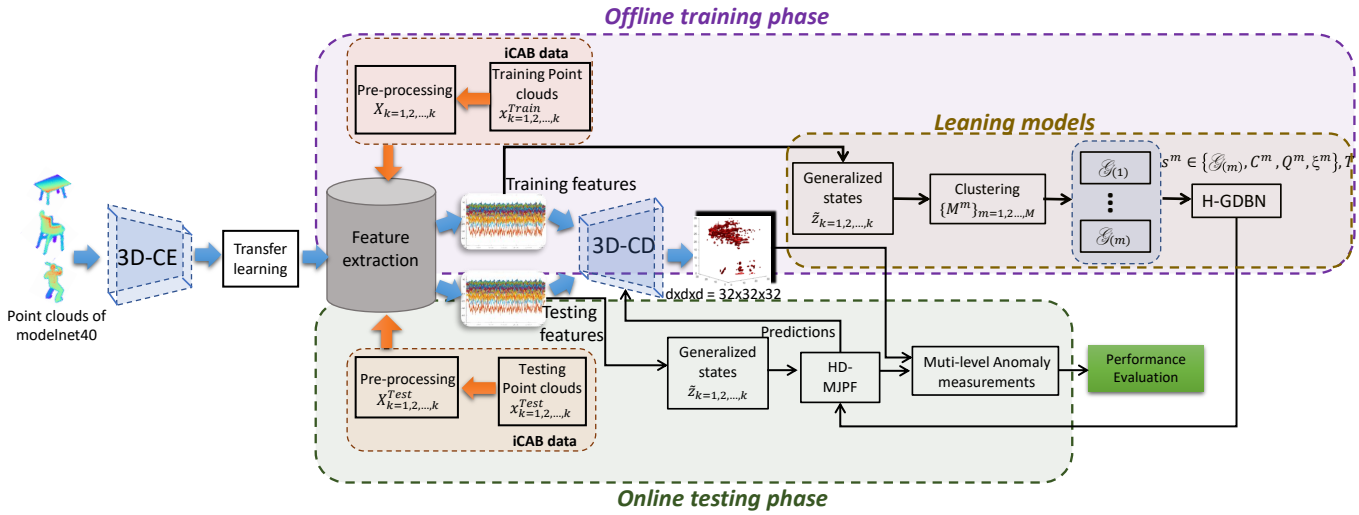


Fig. 1: Block diagram of proposed methodology. It consists of two phases, i.e., *offline training phase*: used to learn the information from the point clouds of LiDAR (training data sequences) and *online testing phase*: used to make inferences of future states, which helps to detect anomalies from the testing point clouds of LiDAR.

Markov Jump Particle Filter (HD-MJPF) is used for the high dimensional data of LiDARs' point clouds.

III. Proposed Methodology

This section is subdivided into two phases: *Offline training* and *Online testing* phase. *Training phase* consists of the 3D-CED to extract the features from the LiDARs' point clouds and represent it as an H-GDBN model to learn the scenario from these features. While in the *Testing phase*, HD-MJPF is proposed to make inferences of future states and detect anomalies. Block diagram of the proposed methodology is shown in Fig. 1.

A. Offline training phase

1) 3D-Convolutional Encoder-Decoder (3D-CED): As mentioned before, since point cloud representations collected from a LiDAR (sensor) are available, a 3D-CNN is adapted for extracting features to detect anomalies in time-series data.

This work uses an Encoder-Decoder architecture. The encoder allows us to represent the raw LiDAR data as a set of bottleneck features (states) that can be tracked through time by using learnable predictive models. Such predictive models are powerful as they allow an intelligent system to predict the future behavior of features coming from high-dimensional data, i.e., LiDAR observations, therefore having an idea about how the environment should look like in the future based on previously acquired experiences. On the other hand, 3D-CD (Convolutional Decoder) decoder allows us to transform the bottleneck features (encoder's output) into the 3D representation initially provided to 3D-CE (Convolutional Encoder). Such a transformation enables us to predict future LiDAR observations, facilitating a qualitative/quantitative comparison of our system's predictions with LiDAR's raw observations. Fig.2 shows the architecture of the proposed neural network. Its different layers are further explained and justified in the following subsections.

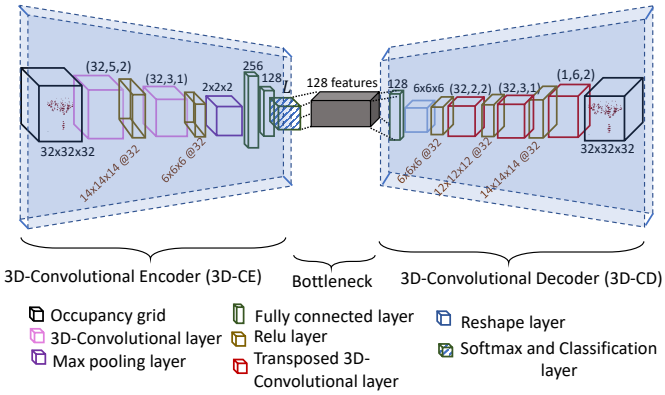


Fig. 2: Network architecture of 3D-Convolutional Encoder-Decoder. L in 3D-CE is composed of softmax and classification layer (removed while extracting features).

2) Transfer learning and Fine tuning: Although our objective is to generate models that can predict future behaviors and detect anomalies on data series coming from LiDAR, we take advantage of the architectures from the state-of-art that tackle the problem of feature extraction from the LiDAR's point clouds. Accordingly, we identified the benchmark dataset ModelNet40 [43], which consists of 12311 CAD models from 40 object categories; and another smaller dataset named the Sydney Urban Objects dataset [44], which contains a variety of common urban road objects scanned with a Velodyne HDL-64E LIDAR, collected in Sydney, Australia. Latter dataset includes the classes relevant to the transportation systems, e.g., vehicles, pedestrians, traffic signs and trees. Additionally, we replicate the training process shown in the work [34], which considers a classification network over LiDAR data and we demonstrate the robustness of their classifier in both datasets mentioned above.

The corpus of the ModelNet40 is larger than the Sydney Urban dataset as well as our data (introduced in Section IV). Therefore, we use two Transfer Learning (TL) processes, one for the encoder and another for the decoder, to adapt it. This leverages the information of vehicle data by using an

initial robust representation of point clouds of Modelnet40. Accordingly, both TL processes are explained as follows:

TL for the encoder. We use 3D-CE trained with ModelNet40 data and fine-tune it with data of transportation systems. Accordingly, we removed the last classification layer of 40 classes and replaced it with another one with three main classes observed in our own dataset focused on vehicle systems, namely pedestrian, tree, and building. The fine-tuning of the network is performed with labeled data from the Sydney Urban dataset and LiDAR observations coming from our dataset focused on vehicle navigation. The main idea is to take advantage of TL and use it in the context of AVs.

The classification approach described above is a fundamental step to verify the performance of the encoder. After the fine-tuning operation, we remove the soft-max classification layer (L) at the end of the neural network and then consider the 128 features extracted from the last convolutional layer (see bottleneck features in Fig.2) as a meaningful representation of point cloud data coming from a transportation context. The resulting network is here referred as 3D-CE.

TL for the decoder. The decoder takes 128 dimensional features extracted from the 3D-CE as an input and uses a set of transposed 3D-Convolution layers that resemble an inverted version of 3D-Convolution layers, see Fig. 2. Such an architecture is referred as a 3D-Convolutional Decoder (3D-CD) that outputs the 3D-voxel reconstruction of the point cloud. For the training of 3D-CD, a procedure similar to the training of 3D-CE is employed. Consistently, the 128-dimensional feature vectors of the ModelNet40 dataset are used as an input to train the decoder, which provides their respective 3D-Voxel representation as an output. After obtaining such an initial version of the 3D-CD, 128-dimensional feature vectors from the Sydney Urban dataset and from our own data(see Section IV) is employed to fine-tune the network, allowing it to reconstruct the realistic LiDAR observations (coming from a vehicular context) based on the proposed features.

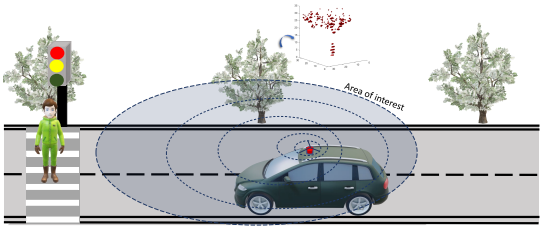


Fig. 3: LiDAR as perception sensor in the context of autonomous driving. Objects surrounding a vehicle and its dynamics can be used to define the normality of a given situation faced by a vehicle.

3) Data Pre-Processing and voxelization: At each time-step, LiDAR provides a 3D map of the environment in the form of point clouds x_k. We follow pre-processing step to obtain 3D voxel representation X_k before the feature extraction such that;

$$X_k = \mathscr{F}(x_k), \tag{1}$$

where $\mathscr{F}$ is the pre-processing function of the point cloud, comprises of two steps; filtering and attention mechanism.

Filtering. This step includes fixing the plane, selecting of area of interest, removing ground points, inliers and outliers. To find and fit the plane to the 3D point cloud, we use M-estimator SAmple Consensus (MSAC) algorithm [45]. Once we define the point cloud in a fixed plane, $\approx 4m^2$ radius is selected around the vehicle as an area of interest, as shown in Fig. 3. A point in a point cloud is considered an outlier if its distance to the nearest points is above a threshold and removed from the point cloud. This threshold is selected based on the standard deviation from the mean distance to the neighbors.

Attention mechanism. This step includes the segmentation of point cloud, selection of the nearest segment, and 3D-voxelization. Filtered point cloud is segmented into different parts based on the clustering-based method. Distance between each segment is optimized (see, [46]) by considering the characteristics of the environment as well as the intensity of each segment of the point cloud. This optimization helps to avoid the clutter of objects in a single segment. Therefore, each segment contains specific information about the environment, such as tree, pedestrian, building/pillar, that comes inside the area of interest of vehicle. In this paper, we select the nearest segment to the vehicle as a potentially risky object/segment. Therefore, the decision of a normal/abnormal situation depends on the selected nearest segment and its dynamics. This choice is performed for the feasibility of the proposed approach but can be extended to multi clusters/objects. The nearest segment wrt vehicle's position is rasterized into 3D-voxel grid (array) [37] with the resolution defined as $d \times d \times d$. At each time-step, we have a 3D-voxel representation of the nearest segment X_k around the vehicle. X_k is provided as an input to the 3D-CE for feature extraction.

4) Learning predictive models: At each time-step, the 3D-CE is responsible for transforming X_k into a 128-dimensional feature vector that we used to make inferences of future states of LiDARs' observations by using a Bayesian model. Accordingly, we consider two types of predictive models to estimate future instances of LiDARs' point clouds:

Piecewise models. This model assumes that future vector states, i.e., dynamics, are related to current states linearly. Hence, it is proposed to find a set of N linear models that are employed to make inferences of future states. Such an approach is based on the concept of Generalized states (GSs) [47], which facilitates the learning of *Piecewise Linear* (PL) dynamic models in a data-driven way by considering an extended version of traditional states, which include their time-derivatives (motion) over time. Accordingly, by considering a memory of a single time-step, it is possible to write the GSs of the LiDAR data as the current 128-dimensional feature vector concatenated with its motion vector computed based on the precedent feature vector. Resulting in a 256-dimensional feature vector that takes the following form:

$$\tilde{z}_k = [z_k, \dot{z}_k]^\intercal, \tag{2}$$

where $z_k = \phi(X_k)$ and $\dot{z}_k = z_k - z_{k-1}$. The function $\phi(\cdot)$ corresponds to the 3D-CE, which takes the 3D voxel representation X_k, see Eq.(1); and outputs its encoded version (128 feature vector). Consequently, a clustering algorithm [48], [49] is used to group similar GSs into a set of M clusters, such that $\tilde{z} = \{\tilde{\mathsf{z}}_m\}_{m=1,\dots,M}$. Cluster of similar GSs, (from Eq.(2)) can be written as; $\tilde{\mathsf{z}}_m = [\mathsf{z}_m, \dot{\mathsf{z}}_m]^\intercal$.

Each resulting cluster encodes a region of the feature vectors where their motion is expected to be quasilinear. Accordingly, a linear dynamic model can be associated with each cluster m, such that:

$$\begin{bmatrix} z_{k+1} \\ \dot{z}_{k+1} \end{bmatrix} = \begin{bmatrix} z_k + \mathbb{E}(\dot{\mathbf{z}}_{m(k)}) + w_k \\ \mathbb{E}(\dot{\mathbf{z}}_{m(k)}) + w_k \end{bmatrix}, \tag{3}$$

where $\mathbb{E}(\cdot)$ is the expected value operator, which provides the mean of the respective clusters m, and w_k represents the Gaussian noise. $m(k)$ indexes the active cluster at time k; and it corresponds to the closest cluster to the current GS $\tilde{z}_k$. The linear dynamical model in Eq.(3) can be employed to estimate the next GS, i.e., $p(\tilde{z}_{k+1}|\tilde{z}_k)$. The effectiveness of the proposed piecewise model is compared with a piecewise nonlinear version; which uses a Long Short-Term Memory (LSTM) for each cluster m that approximates the following GSs given the history of the k past GSs, such that:

$$\tilde{z}_{k+1} \approx \mathscr{G}_m(\tilde{z}_\gamma), \quad \gamma = \{k-(\mathrm{k}-1), k-(\mathrm{k}-2), \ldots, k\}, \tag{4}$$

where $\mathscr{G}_m(\cdot)$ represents the LSTM network trained on GSs coming from the cluster m, where m is the closest cluster to $\tilde{z}_k$ and γ is the set of time-steps corresponding to the similar observations that are clustered together. Each cluster is defined by following dynamic switching parameters $\tilde{s}^m = \{\mathscr{G}_m, C^m, Q^m, \xi^m, T\}$: *i)* LSTM trained for each cluster $\mathscr{G}_m$, *ii)* centroid of clusters C^m, *iii)* covariance of cluster Q^m, *iv)* radius of cluster ξ^m and *v)* transition matrix encodes the probability of transitions between the clusters T. These switching parameters $\tilde{s}^m$ are used to learn probabilistic graphical model called Hierarchical Generalize Dynamic Bayesian Network (H-GDBN), i.e., a DBN using GSs [50]. H-GDBN provides the graphical representation of objects and their dynamics, which AV observes at each time-steps.

Eq.(3) and Eq.(4) show two different alternatives to estimate the future GSs from past observations by using a set of piecewise models. Such models represent the horizontal arrows in the intermediate level (orange block) in Fig.4, which encode inferences at the continuous level. On the other hand, inferences at the discrete level, i.e., cluster transitions (see blue horizontal arrows in Fig.4), is predicted by using a transition matrix T that encodes the probability of going from a single cluster $\tilde{s}^1$ to another cluster $\tilde{s}^2$. Accordingly, a particle filter uses T to model the discrete probability distribution of the following active cluster given the current one, i.e., $p(\tilde{s}_{k+1}|\tilde{s}_k)$.

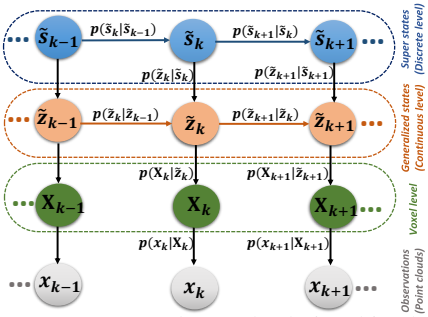


Fig. 4: H-GDBN: Proposed causal relationships between variables involved in the modeling and inferences of LiDAR data.

Nonlinear (NL) model. This model assumes that current and feature vector states are related through a single nonlinear function. In this case, an LSTM network is directly considered to estimate the next GSs, i.e., $\tilde{z}_k$, given the history of k past GSs. In this case, we do not consider any clustering algorithm over GSs and use a single nonlinear model for predicting the future GS (see Fig.5(b)). Such a model can be written as follows:

$$\tilde{z}_{k+1} \approx \mathscr{L}(\tilde{z}_k), \quad k = \{k-(\mathrm{k}-1), k-(\mathrm{k}-2), \ldots, k\}. \tag{5}$$

Note that Eq.(5) resembles Eq.(4); however, the former considers all training data to learn a single model, whereas the latter learns a set of M piecewise models for predicting the next GS. Although this approach does not include a discrete level of inference (blue level in Fig.4), we propose that a single LSTM network trained with enough data may capture the nonlinear dynamics of data series coming from the 3D-CE. This paper compares the single LSTM approach in Eq.(5) with the piecewise models described before in Eq.(3) and Eq.(4) based on their capabilities of detecting anomalies and predicting future LiDAR instances.

B. Online testing phase

In the online testing phase, each point cloud passes through the pre-processing step and attention mechanism to encode the features of the nearest segment of the point cloud to the vehicle, at each time-step. An extended version of the Markov Jump Particle Filter (MJPF) for high dimensional data is proposed here to make inferences of future states and anomaly detection by employing learned H-GDBN.

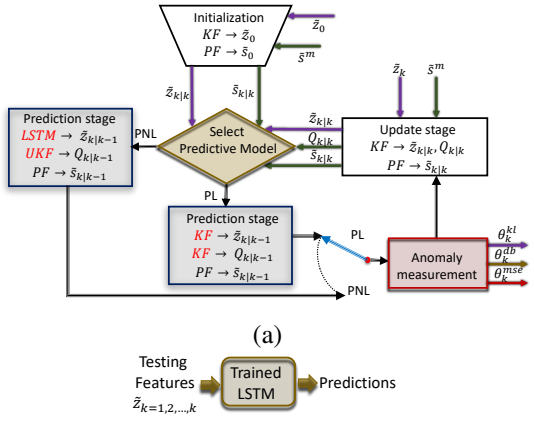


Fig. 5: Block diagram of the predictive models: (a) Piecewise linear (PL) and Piecewise nonlinear (PNL), (b) Nonlinear (NL) models.

1) High Dimensional Markov Jump Particle Filter (HD-MJPF): MJPF is proposed to make inferences of the future states by employing H-GDBN (see Fig. 4) at continuous as well as the discrete level and detect the multiple abstraction level anomalies. In our work, we utilize the *Piecewise Linear* (PL) and *Piecewise Nonlinear* (PNL) version of MJPF for high dimensional data called HD-MJPF. HD-MJPF is a *Switching Dynamical Model* which constitutes a bank of KFs at the continuous level (CL) and a Particle Filter (PF) at the discrete level (DL). HD-MJPF comprised of two main stages, i.e., *prediction* and *update* [50] at both levels, i.e., CL and DL.

Prediction stage. The **PL model** of HD-MJPF uses learned H-GDBN comprised of dynamic switching parameters

$\tilde{s}^m = \{C^m, Q^m, \xi^m, T\}$ to make inferences over the futures states (continuous and discrete) of LiDARs' observations (see Fig.5(a)). At CL, GSs are predicted $p(\tilde{z}_{k+1}|\tilde{z}_k, \tilde{s}_k)$ for each particle, as shown in Fig. 4 and coupled with a PF to make inferences at DL, i.e., $p(\tilde{s}_{k+1}|\tilde{s}_k)$ (see Fig. 4). The predictions at the DL are performed by using the transition matrix T in each particle. In the **PNL model** of HD-MJPF, predictions at the DL are computed similarly to the PL model. On the other hand, predictions at the CL are performed with LSTM, i.e., $\mathscr{G}_m$, associated with the selected cluster, i.e., m from the discrete state $\tilde{s}_{k+1}$ (see Fig.5(a)). Since LSTMs have nonlinear characteristics, therefore, KFs at the CL cannot be used. To tackle the nonlinearity, we used Unscented Kalman Filter (UKF) [51] to make inferences of the future states.

Update stage. It is performed when a new observation (point cloud from LiDAR) at time-step k is available, so the KFs and PF update bases on the information at $k-1$ to make predictions at k. At the CL, the update is performed by KF and UKF of PL and PNL models, respectively. At the DL, re-sampling of the particles is performed based on the anomaly measurements, i.e., CL, DL and VL for PL and PNL models.

2) Anomaly measurement: The difference between the predictions made by learned predictive models and the actual encoded features obtained from the future LiDAR observation; facilitates the definition of temporal anomaly measurements. Here, the definition of anomalies are two-fold: *i)* Anomalies come when prediction do not fall inside the range of the threshold of any learned clusters of trained features, i.e., $\xi^m = 2\sigma(\mathsf{z}_m)$ (see Section III-A4) [50], *or ii)* Anomaly related to the dynamics arise when different dynamics are observed with respect to the learned dynamics within a clusters that are obtained from the training features (blue shaded time-steps in color-coded ground-truth Fig.7 and 9(i)) [52]. Therefore, when an anomaly is detected among the dynamic features, it can be described as the deviation of the prediction with respect to the current observation and its dynamics. The proposed methodology allows us to compute the multiple abstraction-level anomalies, explained as follows:

Continuous level (CL). At CL, anomalies are influenced only by GSs; therefore, it can be considered a local measurement of anomalies without taking into account the relevant active clusters. It is estimated by taking Bhattacharya distance [53] between the predicted GSs $\tilde{z}^{v,P}_{k|k-1}$ and the actual updated GSs $\tilde{z}^{v,P}_{k|k}$ (from Fig.4 and 5), related to the vehicular features v and particle P of the PF, at each time-step. Mathematically, it can be defined as:

$$\theta^{db}_k = \min_P \frac{\sum_{v=1}^{\mathcal{L}} \mathcal{D}_{bh}(\tilde{z}^{v,P}_{k|k}, \tilde{z}^{v,P}_{k|k-1})}{\mathcal{L}}, \tag{6}$$

where θ^{db}_k provides the CL anomaly measurement, $\mathcal{L}$ is the total dimension of the vehicular features and $\mathcal{D}_{bh}$ is the Bhattacharya distance. θ^{db}_k shows high peaks when observations and their dynamics are different from the learned dynamics of training features.

Discrete level (DL). *Event level* anomalies are computed at the discrete level, which shows the errors in the learned transitions between different clusters. If the predicted transition is different from the learned transitions, this indicates the false event that our model does not learn before and vice versa. For the computation of false events, we use Kullback-Leibler Divergence [54]. Mathematically, it is defined as:

$$\theta^{kl}_k = \min_P \frac{\sum_{v=1}^{\mathcal{L}} \mathcal{D}_{kl}(T(m_{\tilde{s}_{k|k}})^{v,P}, \mathcal{O}(m_{\tilde{s}_{k|k-1}})^{v,P})}{\mathcal{L}}, \tag{7}$$

where $\mathcal{O}(m_{\tilde{s}_{k|k-1}})$ is the predicted transition and estimated as;

$$\mathcal{O}(m_{\tilde{s}_{k|k-1}}) \sim \begin{cases} 0 & \text{if } \frac{|\tilde{s}_{k-1}, \tilde{s}_k|}{\xi^{s^m}} > 1 \\ 1 - \frac{|\tilde{s}_{k-1}, \tilde{s}_k|}{\xi^{s^m}} & \text{Otherwise,} \end{cases}$$

and θ^{kl}_k provides the discrete level anomaly measurement, $T(m_{\tilde{s}_{k|k}})^{v,P}$ is the learned transitions computed from the transition matrix T when the cluster m is active at time-step k related to the vehicular feature v and particle P. Therefore, at DL, anomalies come due to the presence of outliers in generalized coordinates, and the predicted transitions become different from the learned transitions between the cluster.

Voxel level (VL). 3D-CD allows to compute the anomalies at the VL. For the computation of the VL anomalies, continuous level prediction of HD-MJPF, i.e., $\tilde{z}_{k|k-1}$, is passed through the 3D-CD. The decoder reconstructs the predictions into corresponding voxels, i.e., $\hat{\mathrm{X}}^{reconst}_k$. Here, the MSE is used for the computation of VL anomalies between the voxels of the reconstructed predictions and the observed voxel X^{test}_{k-1}. Mathematically, it can be defined as:

$$\theta^{mse}_k = mse(\mathrm{X}^{test}_{k-1}, \hat{\mathrm{X}}^{reconst}_k), \tag{8}$$

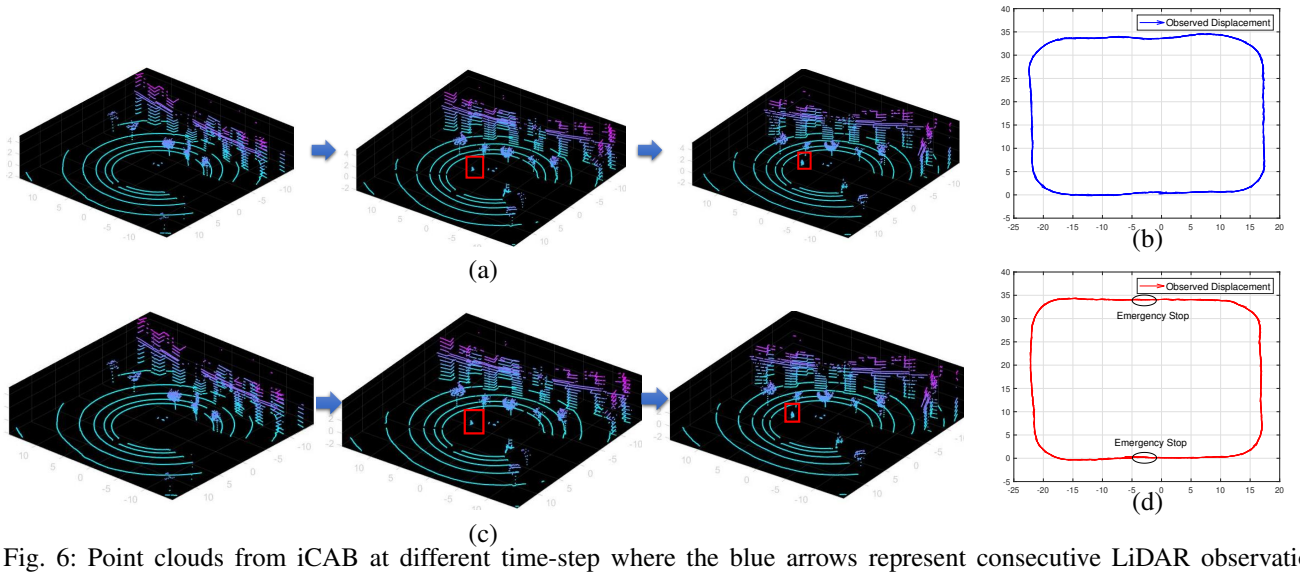
where θ^{mse}_k provides the VL anomaly measurement and $\hat{\mathrm{X}}^{reconst}_k$ are the voxels obtained from the reconstructed continuous level predictions of HD-MJPF by utilizing the 3D-CD. VL anomalies are analogous to the CL anomalies; however, it refers to the observation model instead of the dynamic model. Therefore, an anomaly is detected if the observation is fallen inside a cluster whose distribution is different from the observation.

IV. Experimental dataset

Velodyne LiDAR Puck (VLP-16) is used to collect the point cloud dataset from the AV called as "iCAB" [48]. It provides 360° real-time coverage around the vehicle with 3D distance and calibrated reflectively measurements. VLP-16 supports 16 channels which provide approximately 300,000 points/second, 360° horizontal and 30° vertical field of view, with the ±15° precision range. Experiments are performed in a close environment [48], [55], [52], [56]. Two different scenarios are considered for the training and testing purposes, described as follows:

A. Scenario I: Rectangular maneuver

In this scenario, the vehicle follows a rectangular maneuver, see Fig.6(b) in a closed environment. 3D point clouds collected from the iCAB at different time-step is shown in Fig.6(a). iCAB does not encounter any anomalous situation during this experiment therefore, the LiDARs' point clouds related to this experiment are used for the training and learning of the dynamic models.



(a) (b) (c) (d)

Fig. 6: Point clouds from iCAB at different time-step where the blue arrows represent consecutive LiDAR observations. In these point clouds, vehicle is moving straight and then starts taking turn towards the left side (a) Scenario I; Rectangular maneuver (c) Scenario II; Emergency stop while pedestrian comes in front of the vehicle; where the red square encodes the presence of a pedestrian. (b, d) Positional plots of the Scenario I and Scenario II, respectively used to evaluate the proposed methodology.

B. Scenario II: Emergency Stop maneuver

In this scenario, while performing a rectangular maneuver, a pedestrian comes in front of the vehicle. Therefore, the vehicle performs an emergency stop and waits for the pedestrian to pass and then starts to continue maneuvering, see Fig.6(d). The obtained 3D point clouds for this scenario at different time-step, are shown in Fig.6(c). iCAB encounter an anomalous situation in this experiment; therefore, LiDAR's point clouds from this experiment are used to test the proposed algorithm.

V. Experimental Results

In this section, we present the results of the proposed algorithm. Scenario I (Rectangular maneuver) is used for training, and Scenario II (Emergency stop) is used to test the proposed algorithm. The main idea consists of first collecting and processing LiDARs' point cloud to learn from Scenario I. Then, observe the new experiences from Scenario II in the testing phase and are detected as an anomaly. We include results regarding the reconstruction of voxel representation of Point clouds from the 3D-CD and the accuracy of the entire network after transfer learning. Additionally, we compare different models for making inferences of future states in an HD-MJPF framework through ROC curves.

A. Anomaly detection

Testing data from scenario II (see, Section IV-B and Fig. 6(c)) are utilized for the testing of the proposed methodology. First, the H-GDBN (see, Fig. 4) is learned from the normal scenario (see, Section IV-A and Fig. 6(a)). This learned model is used in the testing phase and detect anomalies (see, Fig.7(a-h)) from the testing dataset by using the proposed predictive models, i.e., PL, PNL, and NL model. Blue shaded area in Fig.7 indicates the time-steps when iCAB performs curves, while the pink shaded area indicates the time-step when iCAB encounters the pedestrian, i.e., an unknown segments of point cloud wrt learned segments which encodes in H-GDBN. High peaks at these time-steps depict the presence of an anomalous situation. Anomaly measurement in Fig. 7 shows how well the presence of pedestrian (pink area) and its dynamics can be detected from the LiDARs' point cloud.

DL anomaly. The PNL and the PL models enable us to detect anomalies at DL (see, Fig. 7(a, b)). Anomalies at DL come when the learned transitions $T(m_{\tilde{s}_{k|k-1}})$ between the clustered GSs (from H-GDBN) are not as same as the predicted transitions $\mathcal{O}(m_{\tilde{s}_{k|k-1}})$. In the NL model, predictions are made over the vehicular features from the LSTM; therefore, it cannot provide us DL anomalies.

CL anomaly. CL anomaly can be computed for all three predictive models, i.e., PNL, PL, and NL (see Fig. 7(c, d, e)). θ_k^{db} shows high peaks when predicted GSs $\tilde{z}_{k|k-1}$ are different from the actual GSs $\tilde{z}_{k|k}^{test}$ (see, Section III-B2). High peaks at curves (blue shaded area) depict that the curves performed by AV during the training experiment are different from the testing experiments, which indicates the presence of unknown dynamics within learned clusters.

VL anomaly. Fig. 7(f, g, h) exhibits the anomaly measurement at VL where high peaks are obtained in the shaded area, i.e., pink and blue. The high peaks in the shaded area indicate that the learning model encodes the necessary features/information from the environment that required for anomaly detection. θ_k^{mse} (see, Eq.(8)) shows the presence of pedestrians (pink area), and its dynamics can be decoded from the predicted features by utilizing the 3D-CD. VL anomaly provides an advantage to go back to the voxel level and observe the elements that cause the anomaly, whether it is an unknown segment of point cloud or unknown dynamics.

Computation of multiple abstraction level anomalies provides us a set of anomaly measurements that can better explain the reason of the anomaly. It allows incremental learning of the LiDAR's point clouds by employing GSs features

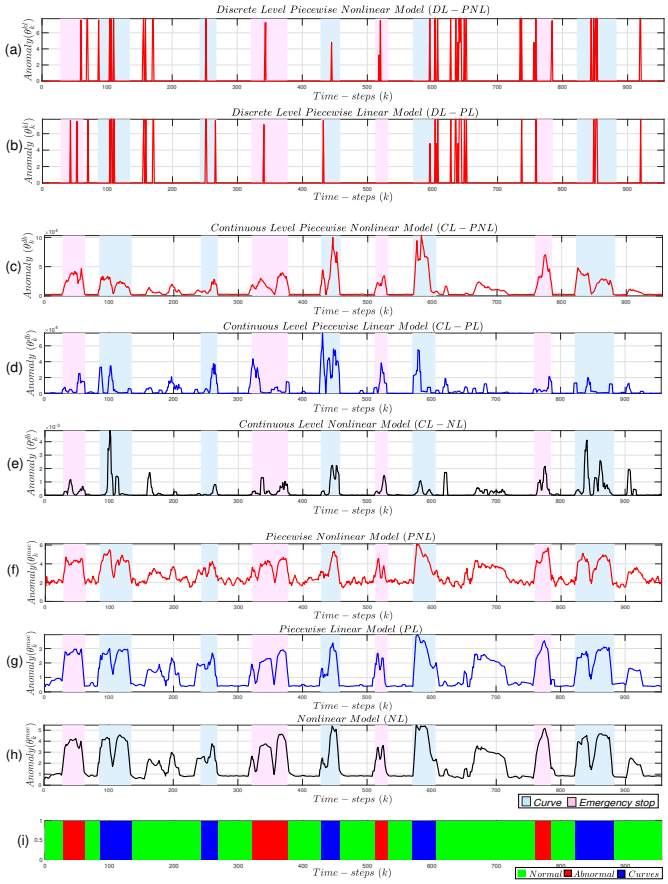


Fig. 7: Anomaly signals from the *online testing phase*: (a, b) DL anomalies for PNL and PL models, respectively. (c, d, e) CL anomalies for PNL, PL and NL models, respectively. (f, g, h) VL anomalies for PNL, PL and NL models. (i) color-coded ground-truth label anomalies where {green, blue, red} = {normal, unknown dynamics, unknown segments wrt learned H-GDBN}, respectively.

associated with the anomalies [56]. Fig.7(e) shows the color-coded ground truth for the anomaly detection with respect to the rectangular maneuver. Green color shows the presence of the normal point cloud, whereas the blue and red indicate the presence of anomalies wrt dynamics and unknown segments from the learning models, respectively.

B. Performance Evaluation

True Positive Rate (TPR) and False Positive Rate (FPR) of θ_k^{db} and θ_k^{mse} are used to compute the ROC curves, shown in Fig. 8. The color-coded anomaly (see, Fig.7(i)) are used as ground-truth, where blue and red region indicates the presence of anomalies. Table I contains the Area under the curve (AUC), Accuracy (ACC), and Precision measurements of the proposed methodology, which shows the quantitative analysis of the PNL, PL and NL models. The PNL model outperforms the PL and NL models. In the PNL model, LSTMs are trained for each cluster and used together with the UKF to make inferences of future states at the CL. This improves the prediction at the CL-PNL. Additionally, VL-PNL outperforms the CL-PL and CL-NL (see, Table I).

In the NL model, a single LSTM trained with the vehicular features from Scenario I and used to make inferences over Scenario II. The NL model shows less accuracy as compared to the piecewise models. Because NL requires a large amount of data for the initial training to attain high performance and accuracy, which is not the case for the piecewise models.

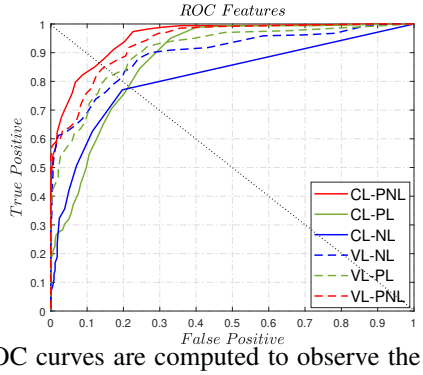


Fig. 8: ROC curves are computed to observe the performance analysis of the proposed methodology.

TABLE I: Quantitative analysis of the proposed methodology.

	Models	AUC (%)	ACC (%)	Precision (%)
Continuous Level	CL-PL	84.84	80.29	67.34
	CL-PNL	**91.85**	**86.18**	**74.29**
	CL-NL	78.16	77.06	63.66
Voxel Level	VL-PL	88.77	80.74	62.94
	VL-PNL	**91.50**	**85.15**	**73.62**
	VL-NL	91.52	84.56	76.70

Ablation Study. An ablation study is performed to analyze the impact of transfer learning and fine-tuning on feature space, which implicitly allows to observe the performance of the predictive models, as shown in Table II. This table provides the quantitative analysis of the anomaly detection by training the 3D-CED with ModelNet40 and Sydney Urban dataset. For this purpose, 3D-CED is trained with the ModelNet40 and Sydney Urban datasets, respectively. Features related to the iCAB dataset (explained in Section IV) are extracted without transfer learning. The learning of the predictive models, i.e., PNL, PL, and VL, is performed to make inferences of the future states.

TABLE II: Ablation study is performed based on training of 3D-CED with ModelNet40 and Sydney Urban dataset. Moreover, the comparison of the performance analysis of proposed methodology with transfer learning is provided.

Network	Models	AUC (%)	ACC (%)	Precision (%)
Train 3D-CNN with ModelNet40 dataset [43]	CL-PL	55.82	55.29	55.58
	CL-PNL	60.81	51.41	51.12
	CL-NL	50.52	51.76	50.22
	VL-PL	46.82	52.06	49.77
	VL-PNL	64.89	63.09	53.11
	VL-NL	66.00	62.94	51.95
Train 3D-CNN with Sydney Urban dataset [44]	CL-PL	58.11	60.32	58.71
	CL-PNL	60.68	59.21	53.43
	CL-NL	52.35	52.94	50.62
	VL-PL	64.93	62.79	53.02
	VL-PNL	76.67	69.12	55.29
	VL-NL	68.74	64.85	53.24
Proposed methodology with Transfer Learning	CL-PL	**84.84**	**80.29**	**67.34**
	CL-PNL	**91.85**	**86.18**	**74.29**
	CL-NL	**78.16**	**77.06**	**63.66**
	VL-PL	**88.77**	**80.74**	**62.94**
	VL-PNL	**91.50**	**85.15**	**73.62**
	VL-NL	**91.52**	**84.56**	**76.70**

The table shows that the performance of both networks is not well; however, the network trained with ModelNet40

performs worse than the Sydney Urban dataset. It is because of the unknown segments of point cloud, which 3D-CED does not knows and confuses them with other objects. While the 3D-CED network trained with the Sydney Urban dataset performed better than the 3D-CED trained with Modelnet40. However, the performance is still not good. This is due to the small training dataset; therefore, the inferences are extremely noisy. Moreover, the comparison between the performance shown in Table I and II indicates that the transfer learning and fine-tuning of the network adversely impacts the performance of the predictive models. This manifests the fact that the fine-tuning refines the feature space and increases the intra-features distances, which encodes better the different segments/objects and their relevant dynamics within the clusters and implicitly refines the probabilistic graphical representation i.e., H-GDBN.

C. Computational Sources

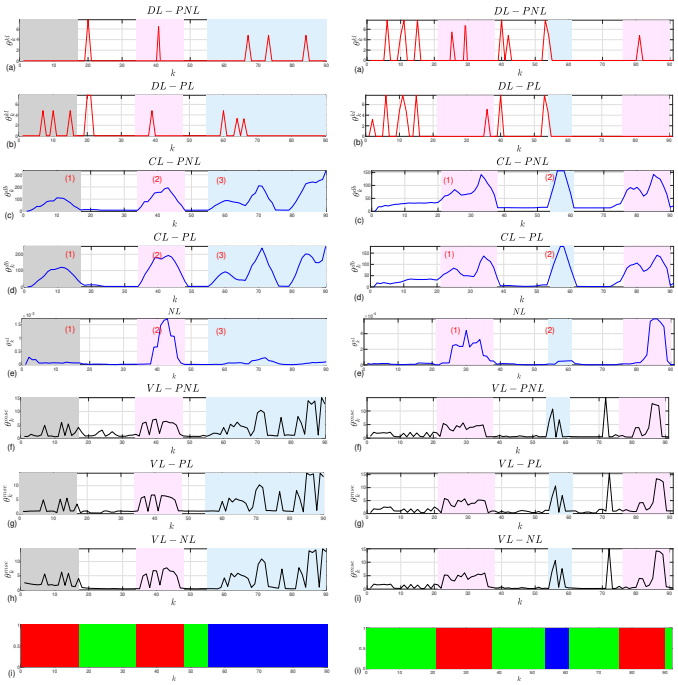
The proposed methodology is implemented in Python and MATLAB. A GPU (dual NVIDIA® GeForce® GTX 1080 Ti with 8 GB RAM GDDR5X each) is used only to obtain the pre-trained weights of the network. The online testing is performed with a CPU (Intel® CoreTM i7-8700 Processor with 16 GB RAM). The computational time in Table III depicts the time required for the online testing phase with CPU, which includes perception and prediction at a time-step k. Specifically, a point cloud at time-step k passes through the data pre-processing, attention mechanism, feature extraction, and predictive modeling steps of the proposed methodology. The computational time required to extract the bottleneck features at a time-step k is fixed, i.e., $\mathcal{O} \sim 5.337ms$. Table III shows that the increases in the number of particles P in HD-MJPF have linear relation with time, i.e., $P \times \mathcal{M}$, and consumes more time to make inferences while it does not bring much improvement in the accuracy. Therefore, total time consumption can be estimated as $\mathcal{O} + (P \times \mathcal{M})$ where $\mathcal{M} \sim 0.086\ ms/particle$. Results presented in Fig.7 and 8 are obtained with $P = 10$.

TABLE III: Execution time of the proposed methodology with varying number of particle P in HD-MJPF with accuracy measurement of PNL at CL and VL.

P	time (***ms***) [bottleneck + prediction]	ACC at CL (%)	ACC at VL (%)
5	$\mathcal{O}$ + 0.43	85.88	83.64
10	$\mathcal{O}$ **+ 0.86**	**86.18**	**85.15**
15	$\mathcal{O}$ + 1.30	86.61	85.66
20	$\mathcal{O}$ + 1.72	86.98	85.90
30	$\mathcal{O}$ + 2.54	87.10	86.34

D. Additional Result

This section provides additional results to validate the proposed methodology in a complex/urban environment. The pre-trained network is fine-tuned with the Sydney Urban dataset [44] that includes all classes, i.e., bicycle, biker, building, bus, car, scooter, pillar, etc. Training features are extracted from the iCAB dataset (see Fig.6(a)), which comprised of the tracking of trees, buildings, or pillars (normal situation). The well-known KITTI dataset [57] is employed for the online testing phase. For this purpose, two sequences of 3D point clouds, i.e., one is from the city, and the other is from the campus environment, are utilized. This also proves the validation of the proposed methodology in an unknown environment (different from the learned). Here, the objective is to detect the anomalies, i.e., unknown observations (different from the training features) and the unknown dynamics of the known segments of point clouds. Multiple abstraction level anomalies, i.e., DL, CL, and VL, are computed from the 3D point cloud sequences (see Fig.9). Quantitative analysis is performed with the ROC curves (as explained in section V-B), and comparison is provided between the predictive models, i.e., PNL, PL, and NL (see Table IV). Fig.9(i) is the color-coded ground-truth that shows the green shaded time-steps as the normal time-steps, whereas blue and red color depicts the presence of unknown dynamics or segments of point clouds wrt learning model, respectively.



(A) City [57] (B) Campus [57]

Fig. 9: Anomalies from the KITTI dataset in two different environment (A) 2011-09-26, drive 0113, City [57] (B) 2011-09-28, drive 0037, Campus [57]. Discrete level anomaly signals are; (a) DL-PNL and (b) DL-PL, Continuous level anomaly signals are; (c) CL-PNL, (d) CL-PL, (e) CL-NL, the voxel level anomaly signals are; (f) VL-PNL, (g) VL-PL, (h) VL-NL models and (i) color-coded ground-truth label.

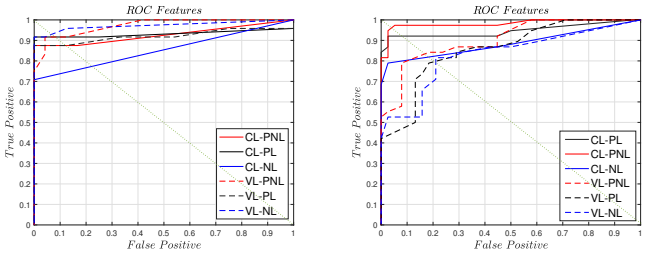
City dataset. For testing purposes, the KITTI dataset is employed from the city (2011-09-26, drive 0113). In Fig.9, the grey shaded area depicts the time-steps when AV observes static cars. The pink shaded area indicates the time-steps when AV confronts bicyclists, which came from the front of the AV and AV turn right to avoid it while the blue shaded time-steps indicates when AV turn left after avoiding the bicyclists and take the left turn to enter in a left road. Fig.9(c,d) shows approximately similar accuracy in the detection of the anomalies. Fig.9(e) shows that NL cannot be able to capture the information of the static cars and the left turn of AV (unknown dynamics); however, at VL, 3D-CD successfully reconstructs the voxels and detect the corresponding anomalies

(see Fig.9(h)). This depicts the advantage of the three-level architecture when employed to make inferences of the future states. Fig.9(i) is the color-coded ground-truth used for the quantitative analysis of the anomalies. Green color indicates the normal time-steps, whereas blue and red color depicts the presence of unknown dynamics or segments of point clouds (objects) wrt the learning model, respectively.

Campus dataset. KITTI dataset, collected from the campus road (2011-09-28, drive 0037), is employed for testing purposes. In Fig.9, the pink shaded area indicates the time-steps when bicyclists pass in-front (horizontal) of the AV. The blue shaded area indicates the time-steps when AV turns right to move to the right road. The pink shaded time-steps (at the end of the anomaly signal) indicate when AV again observes bicyclists moving in the opposite direction in the second lane of the road. Although it is not an anomalous situation but the learning model, i.e., H-GDBN, does not know the dynamics of the bicyclists; therefore, it is captured as an anomaly. PNL model at DL anomaly signal (see Fig.9(a)) shows high peaks even in the starting time-steps because AV is static, i.e., unknown dynamics. While DL-PL (Fig.9(b)) does not show peaks in pink shaded time-steps. CL anomalies Fig.9(c,d) from predictive models, i.e., PNL and PL, show approximately the same performance. However, NL is not able to detect the unknown dynamics of the AV; therefore, it does not have peaks in the blue shaded area. While the anomalies at VL (see Fig.9(e,f,g)) shows approximately a similar performance with a bit of variation in the fluctuation of anomaly signal. Similar performance at VL depicts the importance of the reconstruction of voxels/point clouds in the field of automation.

TABLE IV: Quantitative analysis of the proposed predictive models when tested with the KITTI [57] dataset.

Data	Models	AUC (%)	ACC (%)	Precision (%)
KITTI City dataset [57]	CL-PL	92.71	93.75	72.64
	CL-PNL	**93.23**	**95.83**	**87.05**
	CL-NL	85.42	85.42	62.85
	VL-PL	92.19	93.75	74.33
	VL-PNL	93.23	93.75	82.87
	VL-NL	93.06	**95.83**	72.97
KITTI Campus dataset [57]	CL-PL	94.14	96.05	70.35
	CL-PNL	**95.57**	**96.15**	**80.12**
	CL-NL	86.01	88.16	75.69
	VL-PL	80.47	80.26	70.89
	VL-PNL	87.74	85.53	74.46
	VL-NL	85.63	80.26	67.04



(a) City [57]. (b) Campus [57].

Fig. 10: ROC curves corresponding to the anomaly measurements at CL and VL from the KITTI [57] Campus and City dataset.

Fig.9(i) shows the color-coded ground-truth labels for the KITTI dataset employed for the quantitative analysis of the anomaly detection. Table IV and Fig.10, show the AUC, accuracy, and precision measurements corresponding to the anomaly measurements at CL and VL.

VI. Conclusion

3D-CED is employed with a transfer learning that allows us to pretrain the network with the ModelNet40 dataset and use it with fine-tuning in the context of the transportation system to detect anomalies. We generate meaningful features for the tracking and representation of point clouds from vehicular technology to bring the high dimensional data of LiDAR to a low dimensionality. A probabilistic switching dynamic model called HD-MJPF is utilized to make an inference of future states and detection of multiple abstraction level anomalies. Three different prediction models, i.e., piecewise linear (PL), piecewise nonlinear (PNL), and nonlinear (NL) models, are proposed for anomaly detection. Training and testing of the proposed methodology are performed with the real-time data of the AV. This shows that the PNL outperforms PL and NL by attaining **86.18%** and **85.15%** accuracy for anomaly detection at the CL and VL, respectively. Some additional results with KITTI dataset are provided to validate the proposed methodology.

For future work, feature-level fusion between LiDAR, camera, and odometry trajectories can be performed to improve the accuracy of the learning models to tackle complex situations, such as weather conditions. The proposed approach also has an application in the field of cooperative dynamic learning models, as the features of an agent encode the dynamics of the other agent and vice versa. The proposed methodology can be extended for the incrementally learning [58] of the complex situations with slight modification, such as clustering of the similar objects/features corresponding to the anomalies, can be employed for the fine-tuning of the network which implicitly enriches the knowledge embedded in H-GDBN; this leads to the continual learning of the AV from surroundings.

VII. Acknowledgment

This work is partially funded by Spanish Government (PID2019-104793RB-C31 and RTI2018-096036-B-C21), UC3M (PEAVAUTO-CM-UC3M) and Comunidad de Madrid (SEGVAUTO-4.0-CM P2018/EMT-4362).

References

[1] Amir Rasouli and John K Tsotsos, "Autonomous vehicles that interact with pedestrians: A survey of theory and practice," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 900–918, 2019.

[2] Christian Kuka, Andre Bolles, Alexander Funk, Sönke Eilers, Sören Schweigert, Sebastian Gerwinn, and Daniela Nicklas, "Salsa streams: Dynamic context models for autonomous transport vehicles based on multi-sensor fusion," in *2013 IEEE 14th International Conference on Mobile Data Management*. IEEE, 2013, vol. 1, pp. 263–266.

[3] Gian Luca Foresti and Carlo S Regazzoni, "Multisensor data fusion for autonomous vehicle navigation in risky environments," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 5, pp. 1165–1185, 2002.

[4] Roi Mit, Yoav Zangvil, and Dror Katalan, "Analyzing tesla's level 2 autonomous driving system under different gnss spoofing scenarios and implementing connected services for authentication and reliability of gnss data," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 621–646.

[5] Nuksit Noomwongs, Ambar Bajpai, Prin Phutthaburee, Lattapol Wongpiya, Anuparp Skulthai, Tay Zar Bhone Maung, Ye Moe Myint, Irfan Ullah, Lunchakorn Wuttisittikulkij, and Muhammad Saadi, "Design and testing of autonomous steering system implemented on a toyota ha: mo," in *2020 International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE, 2020, pp. 1–5.

[6] S. Campbell, N. O'Mahony, L. Krpalcova, D. Riordan, J. Walsh, A. Murphy, and C. Ryan, "Sensor technology in autonomous vehicles : A review," in *2018 29th Irish Signals and Systems Conference (ISSC)*, 2018, pp. 1–4.

[7] Jianqing Wu, Hao Xu, Yichen Zheng, and Zong Tian, "A novel method of vehicle-pedestrian near-crash identification with roadside lidar data," *Accident Analysis & Prevention*, vol. 121, pp. 238–249, 2018.

[8] L. Heng, B. Choi, Z. Cui, M. Geppert, S. Hu, B. Kuan, P. Liu, R. Nguyen, Y. C. Yeo, A. Geiger, G. H. Lee, M. Pollefeys, and T. Sattler, "Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4695–4702.

[9] A. M. Wallace, A. Halimi, and G. S. Buller, "Full waveform lidar for adverse weather conditions," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7064–7077, 2020.

[10] Matthew Schwall, Tom Daniel, Trent Victor, Francesca Favaro, and Henning Hohnhold, "Waymo public road safety performance data," *arXiv preprint arXiv:2011.00038*, 2020.

[11] Richard E Bills, Micah P Kalscheur, Evan Cull, and Ryan A Gibbs, "Remote sensing for detection and ranging of objects," July 14 2020, US Patent 10,712,446.

[12] Damian Campo, Mohamad Baydoun, Pablo Marin, David Martin, Lucio Marcenaro, Arturo de la Escalera, and Carlo Regazzoni, "Learning probabilistic awareness models for detecting abnormalities in vehicle motions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1308–1320, 2019.

[13] M. Ravanbakhsh, M. Baydoun, D. Campo, P. Marin, D. Martin, L. Marcenaro, and C. Regazzoni, "Learning self-awareness for autonomous vehicles: Exploring multisensory incremental models," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2020.

[14] D. Kanapram, D. Campo, M. Baydoun, L. Marcenaro, E. L. Bodanese, C. Regazzoni, and M. Marchese, "Dynamic bayesian approach for decision-making in ego-things," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 909–914.

[15] Dinithi Nallaperuma, Rashmika Nawaratne, Tharindu Bandaragoda, Achini Adikari, Su Nguyen, Thimal Kempitiya, Daswin De Silva, Damminda Alahakoon, and Dakshan Pothuhera, "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, 2019.

[16] Jaewoong Choi, Junyoung Lee, Dongwook Kim, Giacomo Soprani, Pietro Cerri, Alberto Broggi, and Kyongsu Yi, "Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 974–982, 2012.

[17] Jianqiang Nie, Jian Zhang, Wanting Ding, Xia Wan, Xiaoxuan Chen, and Bin Ran, "Decentralized cooperative lane-changing decision-making for connected autonomous vehicles," *IEEE Access*, vol. 4, pp. 9413–9420, 2016.

[18] Shilp Dixit, Umberto Montanaro, Mehrdad Dianati, David Oxtoby, Tom Mizutani, Alexandros Mouzakitis, and Saber Fallah, "Trajectory planning for autonomous high-speed overtaking in structured environments using robust mpc," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2310–2323, 2019.

[19] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen, "Automatic road crack detection using random structured forests," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, 2016.

[20] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford, "A hybrid discriminative/generative approach for modeling human activities," 2005.

[21] Shi Zhong and Joydeep Ghosh, "Generative model-based document clustering: a comparative study," *Knowledge and Information Systems*, vol. 8, no. 3, pp. 374–384, 2005.

[22] Alan Lukezic, Ugur Kart, Jani Kapyla, Ahmed Durmush, Joni-Kristian Kamarainen, Jiri Matas, and Matej Kristan, "Cdtb: A color and depth visual object tracking dataset and benchmark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[23] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li, "Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 444–453.

[24] Li-Hua Wen and Kang-Hyun Jo, "Fast and accurate 3d object detection for lidar-camera-based autonomous vehicles using one shared voxel-based backbone," *IEEE Access*, vol. 9, pp. 22080–22089, 2021.

[25] Jing Ren, Hossam Gaber, and Sk Sami Al Jabar, "Applying deep learning to autonomous vehicles: A survey," in *2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD)*. IEEE, 2021, pp. 247–252.

[26] Binbin Li, Dezhen Song, Haifeng Li, Adam Pike, and Paul Carlson, "Lane marking quality assessment for autonomous driving," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[27] Michelle M Mekker, Yun-Jou Lin, Magdy KI Elbahnasawy, Tamer SA Shamseldin, Howell Li, Ayman F Habib, and Darcy M Bullock, "Application of lidar and connected vehicle data to evaluate the impact of work zone geometry on freeway traffic operations," *Transportation research record*, vol. 2672, no. 16, pp. 1–13, 2018.

[28] Radhika Ravi, Yi-Ting Cheng, Yi-Chun Lin, Yun-Jou Lin, Seyyed Meghdad Hasheminasab, Tian Zhou, John Evan Flatt, and Ayman Habib, "Lane width estimation in work zones using lidar-based mobile mapping systems," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[29] Masaru Yoshioka, Naoki Suganuma, Keisuke Yoneda, and Mohammad Aldibaja, "Real-time object classification for autonomous vehicle using lidar," in *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*. IEEE, 2017, pp. 210–211.

[30] Binbin Xiang, Jingmin Tu, Jian Yao, and Li Li, "A novel octree-based 3-d fully convolutional neural network for point cloud classification in road environment," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 10, pp. 7799–7818, 2019.

[31] Wei Song, Lingfeng Zhang, Yifei Tian, Simon Fong, Jinming Liu, and Amanda Gozho, "Cnn-based 3d object classification using hough space of lidar point clouds," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, pp. 1–14, 2020.

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[33] Larissa T Triess, Mariella Dreissig, Christoph B Rist, and J Marius Zöllner, "A survey on deep domain adaptation for lidar perception," *arXiv preprint arXiv:2106.02377*, 2021.

[34] Daniel Maturana and Sebastian Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.

[35] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer, "3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3145–3152, 2018.

[36] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12697–12705.

[37] Jing Huang and Suya You, "Point cloud labeling using 3d convolutional neural network," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2670–2675.

[38] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5419–5427.

[39] Waqas Sultani, Chen Chen, and Mubarak Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6479–6488.

[40] Yuan Yuan, Dong Wang, and Qi Wang, "Anomaly detection in traffic scenes via spatial-aware motion reconstruction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1198–1209, 2016.

[41] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski, "Bayesian learning and inference in recurrent switching linear dynamical systems," in *Artificial Intelligence and Statistics*, 2017, pp. 914–922.

[42] Josue Nassar, Scott W Linderman, Monica Bugallo, and Il Memming Park, "Tree-structured recurrent switching linear dynamical systems for multi-scale modeling," *arXiv preprint arXiv:1811.12386*, 2018.

[43] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[44] Mark De Deuge, Alastair Quadros, Calvin Hung, and Bertrand Douillard, "Unsupervised feature learning for classification of outdoor 3d scans," in *Australasian Conference on Robitics and Automation*, 2013, vol. 2, p. 1.

[45] Korawit Pleansamai1and and Krisada Chaiyasarn, "M-estimator sample consensus planar extraction from image-based 3d point cloud for building information modelling," *International Journal*, vol. 17, no. 63, pp. 69–76, 2019.

[46] Zhao Wang, Xing Wang, Bin Fang, Kun Yu, and Jie Ma, "Vehicle detection based on point cloud intensity and distance clustering," in *Journal of Physics: Conference Series*. IOP Publishing, 2021, vol. 1748, p. 042053.

[47] Karl Friston, Biswa Sengupta, and Gennaro Auletta, "Cognitive dynamics: From attractors to active inference," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 427–445, 2014.

[48] Hafsa Iqbal, Damian Campo, Mohamad Baydoun, Lucio Marcenaro, David Martin Gomez, and Carlo Regazzoni, "Clustering optimization for abnormality detection in semi-autonomous systems," in *1st International Workshop on Multimodal Understanding and Learning for Embodied Applications*, 2019, pp. 33–41.

[49] Hafsa Iqbal, Damian Campo, Giulia Slavic, Lucio Marcenaro, David Martin Gomez, and Carlo Regazzoni, "Optimization of probabilistic switching models based on a two-step clustering approach," in *2020 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, 2020, pp. 1–6.

[50] Hafsa Iqbal, Damian Campo, Lucio Marcenaro, David Martin Gomez, and Carlo Regazzoni, "Data-driven transition matrix estimation in probabilistic learning models for autonomous driving," *Signal Processing*, p. 108170, 2021.

[51] Daphne Koller and Nir Friedman, *Probabilistic graphical models: principles and techniques*, MIT press, 2009.

[52] Giulia Slavic, Mohamad Baydoun, Damian Campo, Lucio Marcenaro, and Carlo Regazzoni, "Multilevel anomaly detection through variational autoencoders and bayesian models for self-aware embodied agents," *IEEE Transactions on Multimedia*, 2021.

[53] Guy Barrett Coleman and Harry C Andrews, "Image segmentation by clustering," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 773–785, 1979.

[54] Ziad Rached, Fady Alajaji, and L Lorne Campbell, "The kullback-leibler divergence rate between markov sources," *IEEE Transactions on Information Theory*, vol. 50, no. 5, pp. 917–921, 2004.

[55] Hafsa Iqbal, Abdulla Al-Kaff, Pablo Marin, Lucio Marcenaro, David Martin Gomez, and Carlo Regazzoni, "Detection of abnormal motion by estimating scene flows of point clouds for autonomous driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2788–2793.

[56] Mahdyar Ravanbakhsh, Mohamad Baydoun, Damian Campo, Pablo Marin, David Martin, Lucio Marcenaro, and Carlo Regazzoni, "Learning self-awareness for autonomous vehicles: Exploring multisensory incremental models," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[57] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[58] Hassan Zaal, Hafsa Iqbal, Damian Campo, Lucio Marcenaro, and Carlo S Regazzoni, "Incremental learning of abnormalities in autonomous systems," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2019, pp. 1–8.

Hafsa Iqbal received her BSc degree in Electrical Engineering from University of Engineering and Technology Taxila, Pakistan in 2016 and MS degree (President's Gold Medal) in Electrical Engineering from National University of Sciences and Technology, Pakistan in 2018. She is PhD student and involved in the JD ICE program of University of Genoa,Italy and University Carlos III of Madrid, Spain. Her research interests include signal processing, machine learning and deep learning techniques for cognitive environments.

Damian Campo received a degree in engineering physics from EAFIT University, Medellin, Colombia, in 2014, and a Ph.D. degree in cognitive environments from the University of Genoa, Italy, in 2018. Since 2018, he has been a postdoc researcher in the Department of Engineering and Naval architecture (DITEN), University of Genoa. His interests include the use of machine learning techniques and probabilistic theory for modeling and predicting the states of multisensory data.

Pablo Marin-Plaza obtained the degree of Industrial Electronics and Automation Engineering from Universidad Carlos III de Madrid in 2011. In 2012 he joined the Department of Systems and Automation Engineering at Universidad Carlos III de Madrid, becoming a member of the Intelligent Systems Lab. In 2013, he received the Master degree in Robotics and Automation. He started working as an assistant lecturer in 2013 and his current research interests include computer vision and autonomous ground vehicles. In 2018, he obtained the title of Doctor Engineering in Electric, Electronic, and Automation. Currently, working on several projects attached to the University Carlos III related to autonomous vehicles and computer vision.

Lucio Marcenaro enjoys over 15 years experience in image and video sequence analysis, and authored about 100 technical papers related to signal and video processing for computer vision. He did an Electronic engineer in 1999 and received his PhD in Computer Science and Electronic Engineering in 2003 from University of Genoa. Currently, he is Assistant Professor in Telecommunications for the Faculty of Engineering at the Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture at the University of Genoa.

David Martin is graduated in Industrial Physics (Automation) from the National University of Distance Education (UNED, 2002) and got his Ph.D. degree in Computer Science from the Spanish Council for Scientific Research (CSIC) and UNED, Spain 2008. Currently, he is Professor and Post-Doc researcher at Carlos III University of Madrid and member of the Intelligent Systems Lab since 2011. In 2014, he was awarded with the VII Barreiros Foundation award to the best research in the automotive field. In 2015, the IEEE Society has awarded Dr. Martin as the best reviewer of the 18th IEEE International Conference on Intelligent Transportation Systems.

Carlo Regazzoni is full professor of Cognitive Telecommunications Systems at DITEN, University of Genoa, Italy. He has been responsible of several national and EU funded research projects. He is currently the coordinator of international PhD courses on Interactive and Cognitive Environments involving several European universities. He served as general chair in several conferences and associate/guest editor in several international technical journals. He has served in many roles in governance bodies of IEEE SPS and He is serving as Vice President Conferences IEEE Signal Processing Society in 2015-2017. He is author/co-author of more than 100 papers on International Scientific Journals and of more than 300 papers at peer reviewed International Conferences.