Bias Detection and Generalization in AI Algorithms on Edge for Autonomous Driving

Bias Detection and Generalization in AI Algorithms on Edge for Autonomous Driving

Dewant Katare
Delft University of Technology
Delft, The Netherlands

Nicolas Kourtellis, Souneil Park, Diego Perino
Telefónica Research
Barcelona, Spain

Marijn Janssen, Aaron Yi Ding
Delft University of Technology
Delft, The Netherlands

***Abstract*—A machine learning model can often produce biased outputs for a familiar group or similar sets of classes during inference over an unknown dataset. The generalization of neural networks have been studied to resolve biases, which has also shown improvement in accuracy and performance metrics, such as precision and recall, and refining the dataset's validation set. Data distribution and instances included in test and validation-set play a significant role in improving the generalization of neural networks. For producing an unbiased AI model, it should not only be trained to achieve high accuracy and minimize false positives. The goal should be to prevent the dominance of one class/feature over the other class/feature while calculating weights. This paper investigates state-of-art object detection/classification on AI models using metrics such as selectivity score and cosine similarity. We focus on perception tasks for vehicular edge scenarios, which generally include collaborative tasks and model updates based on weights. The analysis is performed using cases that include the difference in data diversity, the viewpoint of the input class and combinations. Our results show the potential of using cosine similarity, selectivity score and invariance for measuring the training bias, which sheds light on developing unbiased AI models for future vehicular edge services.**

***Index Terms*—Biases, Data Diversity, Feature Similarity, Generalization, Selectivity Score**

I. Introduction

Private organizations, government agencies, and public entities have widely deployed artificial intelligence (AI) algorithms and models to make decisions or automate a manual process [10]. The automation and decision-making process influences current and future users by offering solutions and services. However, there is a possibility of offering false classifications, predictions, and denial of services based on the AI model's data processing and decision-making ability. The perceptive impact of AI-based decision-making has been observed in specific services and use cases for certain population groups [1], [27], [34]. Examples of gender and racial biases in healthcare, banking, and the hiring process have been discussed by providing potential mitigating strategies [27]. In another use case, self-driving car object detection algorithms failed to predict specific user groups as the datasets used for training the AI model consisted of class representation from humans with white race [1]. Similarly, self-driving vehicles may also show biased results during the classification and detection of women and mobility-impaired individuals, as the datasets, especially the validation set, lack such representation of classes [34]. These observations have led to the inclusion of bias mitigation strategies while developing an AI model and for the datasets used for training.

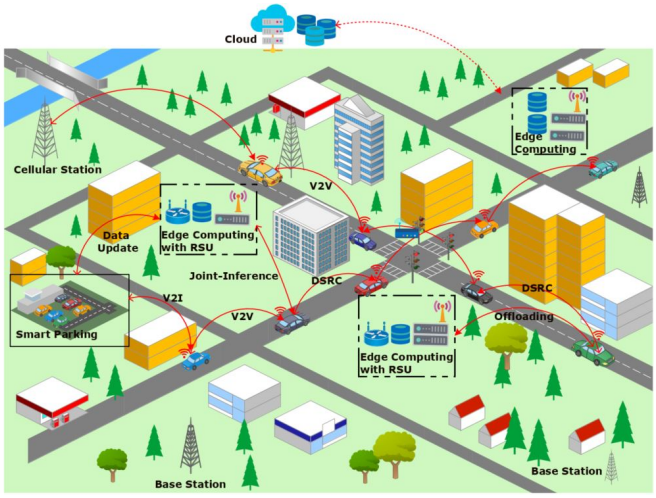

Fig. 1. Future vehicular ecosystem with edge-cloud infrastructure

Biases in the decision-making process are mainly inherited in AI models because of close-world assumption (CWA), which is generally used for knowledge representation in the datasets [34]. In CWA, the test/validation set samples are assumed to be similar to the training set. However, it has been observed that AI models trained with these assumptions show degradation in performance when tested in a real-world environment. This drop in performance for the AI models may be tolerable for applications and services, such as recommendations of products/features and personalization strategy. However, it can have a negative effect by increasing the existing bias when deployed for the intolerable application and domains, such as manufacturing, robotics, medicine, and fully autonomous systems. These concerns have also led to the design of AI models by addressing fairness [11], [12], [39].

Autonomous vehicle tasks include sensing, perception, localization, path planning, control, and acceleration [2], [16], [19]. A fully-connected autonomous vehicle in the future can be envisioned using a vehicle-edge-cloud setting [31], as shown in Figure 1. Vehicular tasks can be performed using distributed devices, transmitting data and model weights in these settings [8], [36]. Federated learning (FL) and communication-efficient approaches have been proposed to

facilitate the deployment of the AI model at the Edge while countering the computation and latency demands. As these methods also promote privacy-preservation by processing the data near the source, and sometimes in the presence of ground truth labels, it is necessary to consider the negative effects, measurement methods and mitigation approach for the data and algorithmic bias in a vehicle-edge-cloud setting using bias and similarity metrics [26], [29], [44]. These issues can be exacerbated in the future FL-as-a-Service platforms (e.g., [18], [20]) that are expected to enable third-party developers to jointly build FL models on locally stored data. Such data, can be bias-prone due to different generation and quality assurance processes taking place on each device, due to diversity in type of device (vehicles, etc.), sensors, third-party applications, environmental conditions, etc.

Object classification and detection are fundamental perception tasks performed using convolutional neural networks, deep neural networks, and recurrent neural networks [2], [17]. Some filters/channels perform better than others in these neural networks, and the reason behind this is unknown, as neural networks are generally described as black-box models. The performance of these neural networks is measured based on the accuracy, precision score, and loss measured against a validation set/sample of the images or test data. From these statements, the following concern arises: If the training set contains imbalanced distributed images of cars, buses, trucks, and trailers, can the AI model accurately classify images with out-of-distribution? Secondly, what is the activity of the neurons during such classification?

Detecting biases in an AI model performing object detection and classification in an imbalanced dataset is challenging, as some objects may have many "common" features and can influence the feature recognition. Therefore, choosing a neural network that can generalize and is unbiased when trained on an imbalanced dataset is essential. To address the existing biases on similar sets of classes, we study the following: bias existing or occurring in the popular AI models that are trained on familiar datasets, trained using transfer-learning, or cross-validated on the driving datasets. The datasets used in this study are biased cars dataset [24], nuScenes [2], and MNIST dataset [37], as these datasets can be grouped with sets of classes that have common features. In this work:

- We investigate the AI models used for object classification/detection tasks, and we provide insights on the accuracy in relation to the data diversity.
- We investigate selectivity score, accuracy, and minimum average precision metrics over the data diversity.
- We analyze the bias metrics and compare them with the model performance parameter using cross-validation and transfer-learning approach.

II. Motivation and Background

A. Object Detection using Edge in Autonomous Vehicles

An autonomous vehicle's essential task is object detection, which helps the vehicle to identify (classify) and estimate the other objects in the surrounding area [3], [17]. These objects can be other vehicles, pedestrians, traffic signs, signals, lane markings, and other roadside information [2], [38]. Based on the abilities of the sensors to imitate the vehicle's environment, current possible approaches include 2D, and 3D object detection [2], [17]. In 2D object detection, a rectangle-shaped bounding box is estimated across the classified object, and in 3D object detection, a cube-shaped bounding box is predicted for the classified object. Several driving datasets have been released in recent years to perform classification and detection for objects possibly present in the vehicle's environment [14].

Connected Vehicles: In future autonomous vehicles, an object detection task can be performed collaboratively using distributed computing or joint training and inference [18], [20]. Methods proposed in this category include deploying AI models through edge computing, fog computing, cloud computing or their respective combinations [3], [22], [31], [33], [36]. As the AI model, such as DNN, can be dense in size and may require high computational resources, it is practically challenging to deploy them on Edge devices. To counter this challenge, federated learning methods that focus on resource allocation schemes, split model training, heterogeneous computing, data aggregation, and privacy-preserving techniques have also been proposed for vehicular applications [41]. In such methods, the AI model is generally distributed between the participating edge devices as local models, and the incoming data is processed at these distributed devices to calculate the weights (locally). The model weights are then sent to the central device (consisting of global model and weights) that updates the parameters of the model [32].

Bias Perspective: Edge-cloud computing, federated learning, and other distributed computing approaches bring scalability to applications such as object detection by solving the computational, energy consumption, privacy, and security challenges [32], [41]. However, the disparity of class within a dataset, imbalanced labeling, and the presence of biased ground truth labels, which can further result in biased object detection within a vehicle-edge-cloud environment, has been so far overlooked. An empirical study to observe and understand the performance of convolution layers in the neural network was carried out by Rafegas et al. [28]. The authors studied the selectivity of specific properties (colour and class) of the input images by the convolution layers. The experiments in this study show that with the increase in convolution layers, the colour selectivity decreases, and the class selectivity increases. An investigation using bias metric and model performance parameter is performed by Madan et al. [23]. The authors studied the problem of generalization of neural networks for distributions and different view-point combinations using a newly proposed photo-realistic biased car dataset [24]. Leavitt et al. investigated the object/class selectivity of neurons in deep neural networks to improve the DNN performance, especially the test accuracy of these models [21]. However, based on the data samples, it was observed that class selectivity during training could further degrade the model performance.

B. Dataset, Models and Biases

Dataset Bias is an overlooked problem for real-world environment AI applications, such as autonomous driving [34]. Bias in datasets can be caused due to unfair distribution of classes within the training and testing sets, out-of-distribution samples, limited availability of samples in different conditions (e.g., weather, daylight), inaccurate labeling, and viewpoint combinations [24]. An example of bias in datasets in the form of annotation bias is explained in [4]. In this work, authors described label bias as an issue that can further magnify the algorithm and model bias. To mitigate such biases during the training of DNN, a re-weighting scheme can be used [15]. Training the AI model with the re-weighted scheme implies using the pre-processing approach on the dataset, such that the unobserved and unbiased labels are used to train a DNN model. Robinson et al. [30] proposed a balanced dataset with sub-group specific threshold. The authors measured the verification performance through subgroups studies that created biases in the datasets. An approach to tackle bias in the dataset from labels is proposed by Cui et al. [5]. The authors proposed a Bayesian architecture to learn label generation for the dataset by using MAP inference to improve data annotation on the local and global levels (e.g., object and frame).

Another example of bias in the dataset can be realized from the Berkeley autonomous driving dataset [42]. It consists of visual annotations, namely classification, segmentation, and bounding boxes for 40 different classes. The visuals are collected from four different cities across the United States. Considering the large demographics of the country, an object detection model trained on such a dataset is likely to suffer selection bias when tested in unfamiliar conditions (e.g., urban vs. rural, with socio-economic differences). Here, the real-world conditions may have a different representation than the trained set of classes. The mentioned selection bias within a dataset can also have an adverse effect in an AI model designed for pedestrian detection or mobility-impaired individuals [27]. A framework to identify and mitigate bias for object detection applications in autonomous vehicles is proposed by Marathe et al. [25]. The authors proposed an approach to detect bias using a two-step process and transfer learning. In the first step, the AI model is trained on the perfect weather dataset, and evaluation is performed on the respective validation set. In the second step, the baseline model is again trained on the perfect weather conditions. However, the evaluation is performed on the validation set consisting of adverse weather conditions. AI model performance of both steps is compared, and if the model performs poorly for the second step, robustness in the training set and re-training for the second step is recommended. The bias for object detection is measured using each object's minimum average precision and average precision score.

AI Model Bias can be described as unfair classification or prediction for certain groups of classes due to unreliable modeling practices. An example of such modeling practices creating bias is using a biased estimator, which helps minimize variance on a small sample size of data by providing robustness for future use cases [6]. Another example of creating bias in an AI model is using the overlaying data that is not directly related to classification or prediction and sometimes not using the significant data from the annotated ground truth crucial for regression or prediction. An example of such a case is a forward collision warning application based on radar, camera or a combination of both. Here a neural network or prediction algorithm may use input values other than separation distance, acceleration, and velocity [16]. This scenario can also be studied with a dataset consisting of object detection/tracking using camera and radar values by experimenting with static (parked) and dynamic (moving vehicles). In AI models, high bias occurs because of missing connections between input data features and the predicted output, described as underfitting. Comparably, high variance in an AI model shows the model's ability to perform well on the training data but shows poor performance on validation or new data, also described as overfitting [44]. To overcome such a dilemma, balancing the dataset with equal class representation and training the AI model/learning algorithm by observing the neurons and layer activity is essential [28].

Generalization is an approach to enable AI model learning across unseen samples of classes. AI models are generally trained with the assumption that the classes present in the training and testing distributions are similar. Preparing a test dataset with all possible distribution combinations is an intensive process. Therefore, neural networks and AI models are generalized to overcome existing bias from the dataset and perform fair for practical applications [35]. Several generalization approaches to overcome dataset bias have been proposed in recent years, which include transfer learning, covariate shift, domain adaptation, adversarial mitigation, and out-of-distribution generalization [35]. Hardt et al. [9] proposed a fairness measure with demographic parity to train an AI classifier over a dataset using supervised learning. In this work, unfairness measurement during the model training and post-processing step is proposed, which can also be applied as privacy-preserving methods in AI applications.

III. Bias Detection in AI Models

The success of convolutional and deep neural networks for perception has led to the belief that AI models should be designed to achieve high accuracy and precision over the dataset. This direction has led to the development of benchmark architectures with several model performance parameters. However, the understanding of variables used for classification or prediction in the training/testing process is unknown. Believing the performance and results of such models bring entrust to the statistical equation the algorithm uses, and also to the data and annotation the model is trained with [28]. These practices can result in biased AI models trained on a biased dataset. As discussed in Section II, having a most complete dataset is an expensive process; therefore, the focus should be given to bias detection and mitigation approaches during the AI model development process.

A. Bias Identification

AI models, especially deep neural network evaluation and validation, generally measure performance parameters. Therefore to study the biases in an AI model, it is essential to consider the explainable metrics that provide information about the learning process of the neurons or layer over the object feature. This process can also be expressed as neuronal activity. Selectivity of neurons towards the *class-colour and class-label* have been investigated to understand the model learning ability against the input class/object [28]. Here selectivity can be defined as a measure of identifying the image and features that are not transformations of each other. If a neuron is activated for a particular class category, it can be assumed that it will have the maximum sum for this particular category. This property is expressed as the preferred category for the neuron for learning [45]. The selectivity score "$S(m,n)$" thus computes the difference between the average activity ($\bar{x}_m^n$) in the neuron for the preferred category with the average activity of the remaining categories ($\bar{x}_m^{-n}$). In this paper, the selectivity score has been used and integrated with the DNN models, namely ResNet, DenseNet, and SqueezeNet [13], [43]. Inspired by Zhou et. al [45], the selectivity score is expressed as:

$$S(m,n) = \frac{\bar{x}_m^n - \bar{x}_m^{-n}}{\bar{x}_m^n + \bar{x}_m^{-n}} \tag{1}$$

The value of "Selectivity score: $S(m,n)$" is generally measured between $(0,1)$. If the measure is 1, the neuron is *active* for a single class, and if the measured value is 0, the neuron is *identically active* for all category-class. Embedding this approach for object detection can help to understand individual class comparisons (e.g., truck vs trailer) for a particular class's features that may influence the prediction or classification of a similar present class or distribution. When embedded within the deep neural network model training process, selectivity returns a vector that has a dimension equal to a number of classes [23], [45]. Another approach that can help measure the two objects' common/similar features is cosine similarity. This metric has regularly been used in statistics to measure the distance between high dimensional feature-based vectors [7]. The general expression for the cosine similarity is:

$$Cos_{sim}(a,b) = \frac{a.b}{|a|\,.\,|b|} \tag{2}$$

As the representation of image or video frames in matrix form consist of high-dimensional feature vector maps, soft cosine similarity or soft similarity can be used. This is generally described as follows:

$$softcos_{sim}(a,b) = \frac{\sum_{i,j}^{n} s_{i,j}.A_i.B_j}{\sqrt{\sum_{i,j}^{n} s_{i,j}A_iA_j}\sqrt{\sum_{i,j}^{n} s_{i,j}B_iB_j}} \tag{3}$$

Here *A* is the un-normalized (raw vector) of the classification, *B* is the mean vector also embedding the most likely prediction of class according to the estimator or distribution, and $s_{i,j}$ is the similarity of features. If there exists a similarity between two feature vector maps, then equation 3 gives an equal output as equation 2. Comparable images with many similar features (e.g., red colour SUV and Sedan car with the same viewpoint) will have a high score, i.e. in the range of 0.6 - 0.9, and in case of no similarity between images (e.g., car, pedestrian), the score will be in the range of 0.0 - 0.1.

Model: ResNet and DenseNet are amongst the most popular deep learning-based object detectors. Several recent deep learning models also use their architecture as a backbone [43]. ResNet has been explored in several variants, thus providing options to implement the architecture with different memory and computational abilities. The most popular variants used in autonomous driving applications are ResNet-18 and ResNet-34. The number (18 and 34) represents the layers in the neural network [17], [43]. Compared to the previous benchmark architectures, ResNet additionally consists of an "identity connection" between two layers and the standard convolutional, pooling, and activation layer. It is also described as a neural network with a residual block.

On the contrary, DenseNet is also a feed-forward neural network with fully connected connections between the layers. The modules in the DenseNet are described as dense blocks. All layers with matching feature-map sizes in DenseNet are connected with each other [43]. These connections provide a gradient flow amongst the layer by ensuring feature reuse within the dense blocks. SqueezeNet is a relatively newer architecture as compared to ResNet. It became prevalent because of its smaller model size, with fewer parameters, and high accuracy with operability on embedded devices. The development of SqueezeNet also provided direction towards deep compression strategies for neural networks by still maintaining the benchmark accuracy [13]. Such compression strategies have been widely adopted for deploying large AI models on the Edge, and embedded devices [32], [33]. As the nuScenes dataset is complex and consists of annotation from several sensors such as (camera + LiDar + Radar), BIRANet [40] architecture is used to train and test the nuScenes dataset. This architecture uses a region proposal network approach for detection using camera and radar data, which also allows the analysis of other data (in this case, radar values) influencing an object detection for a particular class.

Dataset: The model evaluation is performed on the biased-cars, and nuScenes dataset [2], [24]. The biased-car dataset consists of approximately 30000 images of cars (five different models) with different car colours having a common background in the frame. It further includes different viewpoints combinations, allowing out-of-distribution generalization by studying the measure metrics. nuScenes is an extremely large and one of the most complete datasets, developed using several vehicular sensors across Boston (USA) and Singapore [2]. As the dataset is extensive and consists of several classes, a split approach is used to train and test the model. Classes such as bicycle, bus, car, construction vehicle, motorcycle, and pedestrian are used for measuring selectivity. Since the nuScenes dataset consists of 3D bounding box annotations, to have a comparative study, they are transformed into 2D bounding

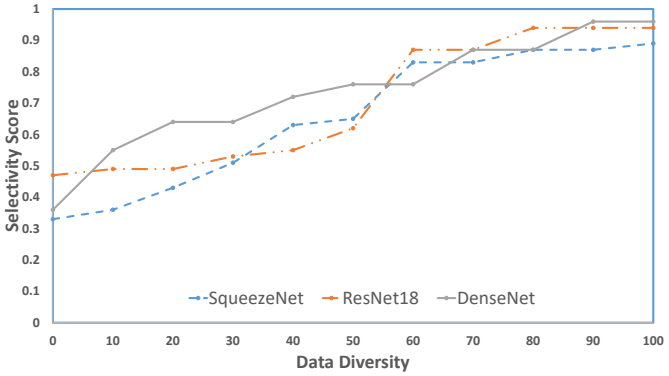

Fig. 2. Selectivity Score on Biased-Car dataset wrt data diversity

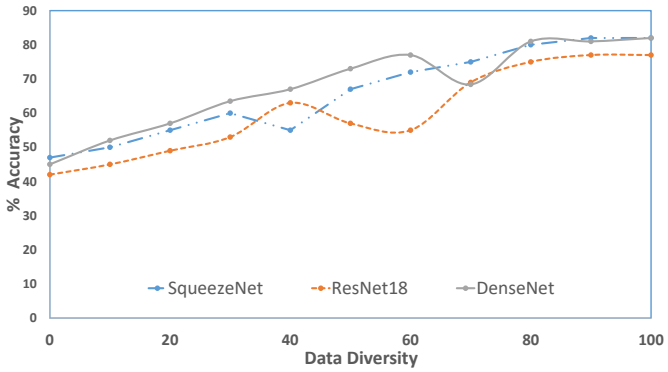

Fig. 3. Accuracy on Biased-Car dataset wrt data diversity

boxes before being utilized on the BIRANet architecture. The model performance is recorded for class: car and pedestrians.

IV. Experiments and Results

This section describes the model training process and testing results. All three models were first trained on the biased-cars dataset, which was followed by training on the nuScenes dataset for the classes mentioned in Section III. For a fair comparison of all models on the biased-car dataset, the models adapted to a common development environment using the PyTorch framework and are trained using the same loss function, ReLu activation function, learning rate, and optimizer. Adam optimizer with a learning rate of 0.001 is used for training these architectures, with a cross-entropy loss, which can be described as:

$$L_{ce} = -\sum_{i}^{j} \mathcal{P}(x_i) log(q(x_i)) \tag{4}$$

Here p is the probability distribution of the object label, q is the prediction, and j represents the number of samples present. The models are trained with 3000 train images, 390 validation, and 750 test images. All models are trained for 150 epochs. As the nuScenes dataset is relatively large and consists of several test images, the approach proposed by Caesar et al. is adapted to train the models [2]. The model is trained for 300 epochs with the learning rate and optimizer mentioned above. In the case of the biased-cars dataset, data diversity with respect to distribution combinations has been considered to train the AI model, which allows capturing the selectivity and model performance measure during the model training.

TABLE I
Performance of SOTA AI model on the biased-car dataset

SOTA AI Model	Data Distr.	Average-Similarity	Accuracy(%)
ResNet18-cos	OOD	.38	74.1
SqueezeNet-cos	OOD	.36	74.8
DenseNet-cos	OOD	.57	71.5

Data Diversity

As both datasets are exclusive with different distributions and labels for classes, an independent approach is used for varying data diversity in the datasets. The biased car dataset can be separated into category and viewpoint combinations for each class. For a fair comparison, the number of images is kept constant during the training of all models. Experiments are carried out by alternatively varying the category and viewpoint combinations for these classes. This process is carried out to check the selectivity of neurons and the model's accuracy, which is also very useful for understanding the model's generalization ability over an unseen dataset. Figure 2 shows the selectivity score of neurons on the biased-cars dataset with respect to the data diversity of classes. As the distribution combinations are increased for a fixed number of train-test-validation images, data diversity and class distribution make it difficult for the neural network to learn a challenging category or viewpoint. When the combination of train-test images is diverse, it leads to a drop in the neural network's performance (accuracy). The selectivity score shows a similar trend for SqueezeNet and ResNet once the data diversity is varied more than 50%. A little drop in accuracy (Figure 3) is observed for all neural networks between a data diversity of 40 - 60 %.

Table I shows the cosine similarity analysis over the biased car dataset. For the equally distributed samples, the cosine similarity results for ResNet, SqueezeNet, and DenseNet were 0.38, 0.36, and 0.57. Depending on the test sample and model's selectivity towards a category and class labels, SqueezeNet and ResNet could generalize better for the given input data. Figure 4 shows the selectivity score of neurons on the nuScenes dataset. To vary data diversity in the nuScenes dataset using existing annotations, AI model [40] is trained in different conditions, which include day, night and rain situations. This approach is used to ensure the presence of data diversity for the above-mentioned classes while capturing selectivity scores and average precision (also shown in Table II). Similar to the biased car dataset, as the distribution combinations from different weather conditions are increased for a fixed number of train-test images, data diversity leads to a significant drop in the current model performance (precision). The selectivity

TABLE II
PERFORMANCE OF THE MODEL ON THE NUSCENES DATASET

Model	Dataset	AP(car)	AP(ped)	mAP(0.5)
BIRANet1	nuScenes	64.1	46.4	55.2
BIRANet1	nuScenes (Night)	49.7	32.5	40.8
BIRANet1	nuScenes (Rain)	60.2	44.3	52.1
BIRANet2	nuScenes	64.8	46.1	56.3
BIRANet2	nuScenes (Night)	48.5	31.7	39.2
BIRANet2	nuScenes (Rain)	59.8	42.4	51.6

score shows a similar trend for the two versions of the BIRANet architecture. BIRANet1 uses cross-entropy loss, and BIRANet2 uses joint loss. It is important to note that the selectivity score shows a similar trend for the two versions in data diversity, irrespective of separate loss functions. The models are trained using a train-validation split of 52-48%. The models show performance degradation in adverse conditions compared to normal conditions.

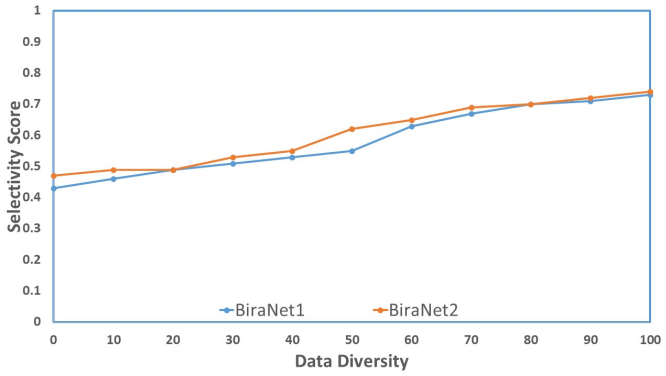

Fig. 4. Selectivity on nuScenes dataset wrt data diversity

Edge AI and Bias Mitigation

Investigating metrics such as the selectivity of the neurons (to a particular category, class, or viewpoint) and similarity score can provide insights to prevent bias in an AI model. However, the challenge remains of adding bias through selective and imbalanced distribution while developing a dataset. This problem intensifies when AI practices and techniques such as Edge-AI and federated learning are proposed collaboratively (i.e., to be used while processing raw data in conjunction with already present ground truth labels at the device). A common framework measuring bias parameters and selectivity of neurons towards unseen data can be proposed for the vehicle-edge-cloud environment by using efficient communication and computation. To prevent bias inheritance from ground truth labels in an AI model at the Edge, diverse data near the edge devices is essential, and hybrid learning approaches combining supervised and active learning can be used. This process can help in bias mitigation by updating the ground truth with newly generated labels.

V. CONCLUDING REMARKS

The topics and methods covered in this paper show approaches to identify and detect bias in an AI model used for real-world applications, such as autonomous driving. The focus given to the biased-car dataset shows the requirement to have wide distributions of the objects in the training/testing set. It provides an unbiased dataset and further helps in the generalization of the AI model, thus preventing algorithmic bias. We investigated bias detection using metrics such as selectivity score and cosine similarity during the learning process by varying the data diversity of the test set. The learned model is further used on the nuScenes dataset to detect pedestrians and cars. A further investigation can be carried out using distributed machine learning at the vehicle-edge-cloud for the vulnerable pair of classes such as pedestrians and cyclists. With respect to the other class (such as vehicles or traffic signs), the vulnerable classes generally has less representation within the dataset. As the self-driving domain advances, studying metrics such as cosine similarity, selectivity score, and invariance can provide an approach to measure bias during the training process and further develop an unbiased AI model for inference. Exploring such metrics on a layer and block level for a neural network by having a direct comparison with a similar object (e.g., pedestrian, cyclist, motorcyclist) can help understand the interpretability of the neural network, which further helps in generalization and preventing biases.

ACKNOWLEDGMENT

The authors gratefully acknowledge funding from European Union's Horizon 2020 Research and Innovation programme under the Marie Skłodowska Curie grant agreement No. 956090 (APROPOS), SPATIAL project under grant agreement No 101021808, and CONCORDIA project under grant agreement No 830927.

REFERENCES

[1] Martim Brandao. Age and gender bias in pedestrian detection algorithms. *CoRR*, abs/1906.10490, 2019.

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11618–11628. IEEE, 2020.

[3] Yung-Yao Chen, Yu-Hsiu Lin, Yu-Chen Hu, Chih-Hsien Hsia, Yi-An Lian, and Sin-Ye Jhong. Distributed real-time object detection based on edge-cloud collaboration for smart video surveillance applications. *IEEE Access*, 10, 2022.

[4] Yunliang Chen and Jungseock Joo. Understanding and mitigating annotation bias in facial expression recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14980–14991, October 2021.

[5] Zijun Cui, Yong Zhang, and Qiang Ji. Label error correction and generation through label relationships. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3693–3700, 2020.

[6] David Danks and Alex John London. Algorithmic bias in autonomous systems. In *IJCAI*, volume 17, pages 4691–4697, 2017.

[7] Sunipa Dev, Tao Li, Jeff M Phillips, and Vivek Srikumar. On measuring and mitigating biased inferences of word embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7659–7666, 2020.

[8] Aaron Yi Ding, Ella Peltonen, Tobias Meuser, Atakan Aral, Christian Becker, Schahram Dustdar, Thomas Hiessl, Dieter Kranzlmüller, Madhusanka Liyanage, Setareh Maghsudi, Nitinder Mohan, Jörg Ott, Jan S. Rellermeyer, Stefan Schulte, Henning Schulzrinne, Gürkan Solmaz, Sasu Tarkoma, Blesson Varghese, and Lars Wolf. Roadmap for edge ai: A dagstuhl perspective. *ACM SIGCOMM Computer Communication Review*, 52(1):28–33, 2022.

[9] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3323–3331, 2016.

[10] Paul Henman. Improving public services using artificial intelligence: possibilities, pitfalls, governance. *Asia Pacific Journal of Public Administration*, 42(4):209–221, 2020.

[11] Wiebke Toussaint Hutiri and Aaron Yi Ding. Bias in automated speaker recognition. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pages 230–247. ACM, 2022.

[12] Wiebke Toussaint Hutiri and Aaron Yi Ding. Towards trustworthy edge intelligence: Insights from voice-activated services. In *2022 IEEE International Conference on Services Computing (SCC)*, pages 239–248. IEEE, 2022.

[13] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[14] Pengliang Ji, Li Ruan, Yunzhi Xue, Limin Xiao, and Qian Dong. Perspective, survey and trends: Public driving datasets and toolsets for autonomous driving virtual test. *CoRR*, abs/2104.00273, 2021.

[15] Heinrich Jiang and Ofir Nachum. Identifying and correcting label bias in machine learning, 2019.

[16] Dewant Katare and Mohamed El-Sharkawy. Collision warning system: embedded enabled (rtmaps with nxp blbx2). In *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 1–6. IEEE, 2018.

[17] Dewant Katare and Mohamed El-Sharkawy. Real-time 3-d segmentation on an autonomous embedded system: using point cloud and camera. In *2019 IEEE National Aerospace and Electronics Conference (NAECON)*, pages 356–361. IEEE, 2019.

[18] Kleomenis Katevas, Diego Perino, and Nicolas Kourtellis. Flaas: Enabling practical federated learning on mobile environments. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, MobiSys '22, page 605–606, New York, NY, USA, 2022. Association for Computing Machinery.

[19] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296. IEEE, 2018.

[20] Nicolas Kourtellis, Kleomenis Katevas, and Diego Perino. Flaas: Federated learning as a service. In *Proceedings of the 1st Workshop on Distributed Machine Learning*, DistributedML'20, page 7–13, New York, NY, USA, 2020. Association for Computing Machinery.

[21] Matthew L Leavitt and Ari Morcos. Selectivity considered harmful: evaluating the causal impact of class selectivity in dnns. *arXiv preprint arXiv:2003.01262*, 2020.

[22] Junwon Lee, Kieun Lee, Aelee Yoo, and Changjoo Moon. Design and implementation of edge-fog-cloud system through hd map generation from lidar data of autonomous vehicles. *Electronics*, 9(12):2084, 2020.

[23] Spandan Madan, Timothy Henry, Jamell Dozier, Helen Ho, Nishchal Bhandari, Tomotake Sasaki, Frédo Durand, Hanspeter Pfister, and Xavier Boix. *arXiv preprint arXiv:2007.08032*, 2020.

[24] Spandan Madan, Timothy Henry, Jamell Dozier, Helen Ho, Nishchal Bhandari, Tomotake Sasaki, Fredo Durand, Hanspeter Pfister, and Xavier Boix. Biased-Cars Dataset, 2021.

[25] Aboli Marathe, Rahee Walambe, Ketan Kotecha, and Deepak Kumar Jain. In rain or shine: Understanding and overcoming dataset bias for improving robustness against weather corruptions for autonomous vehicles. *arXiv preprint arXiv:2204.01062*, 2022.

[26] Tomoya Murata and Taiji Suzuki. Bias-variance reduced local sgd for less heterogeneous federated learning. *arXiv preprint arXiv:2102.03198*, 2021.

[27] Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, et al. Bias in data-driven artificial intelligence systems—an introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1356, 2020.

[28] Ivet Rafegas, Maria Vanrell, Luís A Alexandre, and Guillem Arias. Understanding trained cnns by indexing neuron selectivity. *Pattern Recognition Letters*, 136:318–325, 2020.

[29] Mohammad Saidur Rahman, Ibrahim Khalil, Mohammed Atiquzzaman, and Xun Yi. Towards privacy preserving ai based composition framework in edge networks using fully homomorphic encryption. *Engineering Applications of Artificial Intelligence*, 94:103737, 2020.

[30] Joseph P. Robinson, Gennady Livitz, Yann Henon, Can Qin, Yun Fu, and Samson Timoner. Face recognition: Too bias, or not too bias? *CoRR*, 2020.

[31] Weisong Shi and Liangkai Liu. *Computing Systems for Autonomous Driving*. Springer, 2021.

[32] Yuanming Shi, Kai Yang, Tao Jiang, Jun Zhang, and Khaled B Letaief. Communication-efficient edge ai: Algorithms and systems. *IEEE Communications Surveys & Tutorials*, pages 2167–2191, 2020.

[33] Jie Tang, Shaoshan Liu, Liangkai Liu, Bo Yu, and Weisong Shi. Lopecs: A low-power edge computing system for real-time autonomous driving services. *IEEE Access*, pages 30467–30479, 2020.

[34] Antonin Vobecky, Michal Uricar, David Hurych, and Radoslav Skoviera. Advanced pedestrian dataset augmentation for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[35] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[36] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In *IEEE INFOCOM 2018-IEEE conference on computer communications*, pages 63–71. IEEE, 2018.

[37] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018.

[38] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[39] Jeannette M Wing. Trustworthy ai. *Communications of the ACM*, 64(10):64–71, 2021.

[40] Ritu Yadav, Axel Vierling, and Karsten Berns. Radar+rgb attentive fusion for robust object detection in autonomous vehicles. *CoRR*, abs/2008.13642, 2020.

[41] Yunfan Ye, Shen Li, Fang Liu, Yonghao Tang, and Wanting Hu. Edgefed: Optimized federated learning based on edge computing. *IEEE Access*, 8:209191–209198, 2020.

[42] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[43] Chaoning Zhang, Philipp Benz, Dawit Mureja Argaw, Seokju Lee, Junsik Kim, Francois Rameau, Jean-Charles Bazin, and In So Kweon. Resnet or densenet? introducing dense shortcuts to resnet. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3550–3559, 2021.

[44] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, 31, 2018.

[45] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Revisiting the importance of individual units in cnns via ablation. *arXiv preprint arXiv:1806.02891*, 2018.