UDC 519.8

Daniel Fritz[1], Steffen Schober[2], Götz G. Grimmer [3], Clemens Klöck[4]

DETECTION AND CLASSIFICATION OF ROAD USERS BASED ON RADAR POINT CLOUDS

Introduction. A substantial amount of data is essential for the efficient training of a neural network, as it ensures sufficient variability in the dataset and enables a representative evaluation of the trained model. However, when a custom dataset is generated, such as one obtained from radar sensors, a labeling process is required. This process requires assigning class labels to individual data points, a task that is both labor-intensive and time-consuming when performed manually. Accurate labeling necessitates a deep understanding of the underlying data, as determining which data points correspond to relevant objects is not always straightforward, often due to factors such as noise. To mitigate these challenges, an open-source dataset is used to perform the required task in this work.

The dataset utilized in this study comprises point clouds generated by radar sensors, with each data point characterized by various raw features. This study builds on established techniques for clustering and classifying road users using radar point clouds and investigates the impact of different loss functions, with a particular focus on addressing the class imbalance in the dataset. An overview of the process is presented in Figure 1.

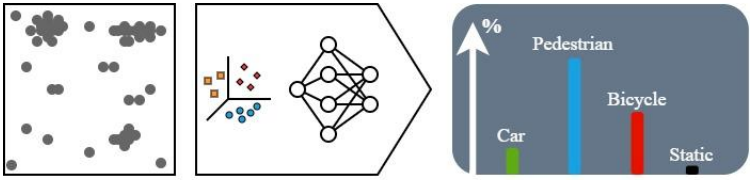


Figure 1 – Radar point clouds are pre-processed, including clustering and feature extraction. Subsequently, these clusters are passed to a neural network for training.

[1] Student, B.Eng., Esslingen University
[2] Prof., Dr.-Ing., Esslingen University
[3] PhD Candidate, M.Sc., Esslingen University
[4] Prof., Dr.-Ing., Esslingen University

Dataset. The dataset RadarScenes [1] is derived from data collected by four radar sensors mounted on the front of a vehicle. Each sensor operates at a frequency of 77 GHz, with a maximum range of 100 meters and a field of view extending from -60° to 60°. The sensors offer a range resolution of 0.15 meters and a radial speed resolution of 0.1 km/h. Measurements were recorded with an average cycle time of 60 milliseconds between consecutive readings. The dataset is provided as point clouds, where each data point is accompanied by a timestamp and additional features, including distance, angle, radar cross-section (RCS), radial velocity, ego-motion-compensated velocity, and Cartesian coordinates.

Methods. Clustering is performed using the DBSCAN algorithm. Challenges arise due to the presence of less representative points, which can lead to suboptimal clustering outcomes. Objects detected at greater distances often result in fewer data points, further degrading the clustering performance. To address this issue, a modified version of DBSCAN is employed [2]. In its original implementation, DBSCAN uses a hyperparameter to define the minimum number of neighbors required for a point to be considered a core point. In the modified approach, this hyperparameter is not applied uniformly across all points. Instead, it is computed individually for each point as [2]

$$N_{min}(r) = N_{\mathrm{min,50\ m}} \cdot \left(1 + \alpha_r \cdot \left(\frac{clip(r, 25\ m, 125\ m)}{50\ m} - 1\right)\right), \quad (1)$$

with $N_{\mathrm{min,50\ m}}$ being the baseline of the minimum number of points and α_r represents the slope for the relation, which includes the clip-function preventing very low or high numbers for $N_{min}(r)$.

After the clustering process, features are extracted for each cluster using a sliding time window of 150 ms, which moves across the clusters with a step size of 60 ms [3]. These features are derived from the measured values of individual points within the cluster, including radar cross-section (RCS), radial distance, angle, and velocity. Additional features are computed from Cartesian coordinates and ego-motion-compensated velocity. Data points within a time window can originate from different timestamps and thus distinct coordinate system origins. Therefore, it is crucial to transform all spatial features within the time window into a unified coordinate system. This transformation ensures accurate representation of clusters through the feature vectors and prevents distortions.

For classification, the feature vectors of each cluster are input into the neural network for training. The accuracy of a multi-class classifier is compared with that of a combination of multiple binary classifiers. For the binary classification task, models are trained using the one-vs-one (OVO) and one-vs-all (OVA) approaches, and the results of each binary model are

subsequently combined to achieve a more representative outcome for the final classification. Given the results of all OVO and OVA classifiers, the class i is predicted with the following expression [4]:

$$\mathrm{id}(\boldsymbol{x}) = \underset{i\in\{1,\dots K\}}{\arg\max}\ \ p_{i,\mathrm{OVA}}(\boldsymbol{x}) \sum_{j=1,i\neq j}^{N} p_{ij,\mathrm{OVO}}(\boldsymbol{x}) \qquad (2)$$

For model training, the feature vectors are grouped into equally sized sequences. If the sequence length is not met, the sequences are padded with neutral feature vectors. Each sequence contains only feature vectors from the same cluster instance. The model architecture, as illustrated in Figure 2, includes an LSTM layer with 80 cells with the hyperbolic tangent as the activation function followed by a fully connected layer with four cells, which is the number of classes to predict. The activation function after the fully connected layer is a softmax function for multi-class classification or a sigmoid function for binary classification.

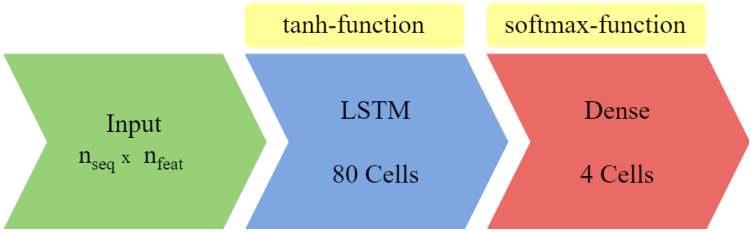


Figure 2 – Model Architecture based on a LSTM layer [3].

Results. Figure 2 illustrates the confusion matrix based on the combination of the results from OVO and OVA classifiers with the help of equation 2. The diagonal line represents recall. Despite the significant class imbalance, promising results are achieved. Although sequences of static objects were often padded with neutral feature vectors, the static class achieves nearly 99% recall. In contrast, the remaining classes exhibit lower recall, which can be partially attributed to the imbalance in the dataset.

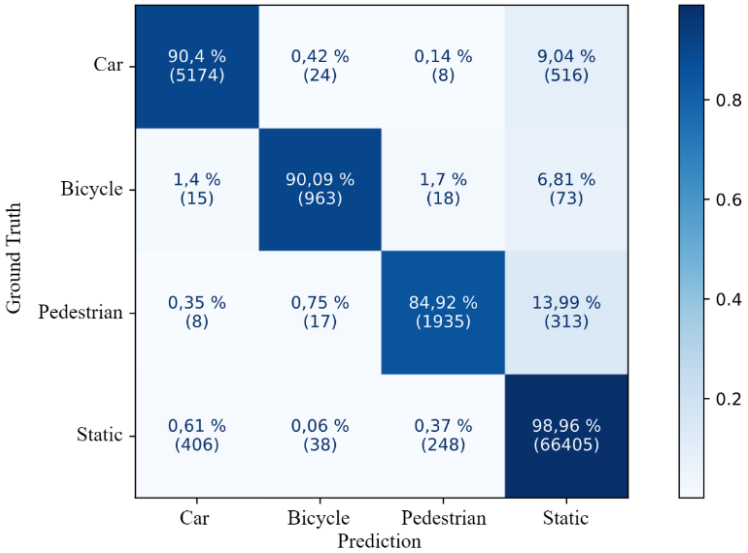


Figure 3 – Confusion Matrix for the combination of OVO and OVA classifiers.

Table I presents the results of various model configurations. In the first experiment, a multi-classifier was trained, which predicts all classes. The second experiment utilized the OVO+OVA approach, and the results were combined as described in equation 2. The next experiment included class weighting for each binary classifier based on the number of occurrences of each class. For class i the corresponding weight w_i can be computed as

$$w_i = \frac{N_{\text{total}}}{N_{\text{unique}} \cdot N_i}, \tag{3}$$

where N_{total} is the number of total samples within the dataset, N_{unique} is the number of distinct classes, and N_i is the number of data samples for the class i. However, the overall performance did not improve compared to the sole OVO+OVA approach. For the last experiment, instead of using the cross-entropy as the loss function, the focal cross-entropy is utilized to address data samples that are difficult to predict and thus apply more focus on classes with fewer occurrences. The focal cross-entropy is defined as [5]

$$FL(p_t) = -\alpha_t (1-p)^{\gamma} \log(p), \tag{4}$$

where $\gamma \geq 0$ is a tunable focusing parameter, α the weighting factor, and p_t represents the model's estimated probability. Nevertheless, the

OVO+OVA is still outperforming the remaining experiments regarding F.1 score and accuracy.

Table 1 – Results of multi-classifier and OVO+OVA classifier.

Model	***F.1-macro-avg***	***Accuracy***
Multi	0.91	0.975
OVO + OVA	0.919	0.978
OVO + OVA + class weights	0.903	0.973
OVO + OVA + focal cross entropy	0.875	0.964

Conclusion. In this work, an open-source radar point cloud dataset was utilized to tackle a multi-classification task. Initially, clustering was performed using a slightly modified version of DBSCAN, adapted to better suit the characteristics of radar point clouds. Next, features were manually extracted for each cluster. These feature vectors were then aggregated into sequences and fed into an LSTM-based model for training. To address class imbalance the OVO and OVA principles were applied. Compared to a model that directly predicts all classes, using a combination of binary classifiers led to more accurate predictions.

REFERENCES

1. Schumann O, Hahn M, Scheiner N, Weishaupt F, Tilly JF, Dickmann J, et al. RadarScenes: A real-world radar point cloud data set for automotive applications. IEEE 24th International Conference on Information Fusion. 2021; 1 – 8.

2. Scheiner N, Appenrodt N, Dickmann J, Sick B. A multi-stage clustering framework for automotive radar data. IEEE Intelligent Transportation Systems Conference, 2019; 2060 – 2067.

3. Schumann O, Wöhler C, Hahn M, Dickmann J. Comparison of random forest and long short-term memory network performances in classification tasks using radar. Sensor Data Fusion: Trends, Solutions, Applications, 2017; 1 – 6.

4. Scheiner N, Appenrodt N, Dickmann J, Sick B. Radar-based feature design and multiclass classification for road user recognition. IEEE Intelligent Vehicles Symposium, 2018; 779 – 786.

5. Lin T, Goyal P, Girshick R, He K, Dollár. Focal loss for dense object detection. Transactions on Pattern Analysis and Machine Intelligence. 2018; 42: 318-327.