

Article

Data-Driven Clustering and Classification of Road Vehicle Radar Scattering Characteristics Using Histogram-Based RCS Features

Aysu Coşkun *  and Sándor Bilicz 

Department of Broadband Infocommunications and Electromagnetic Theory, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3, H-1111 Budapest, Hungary; bilicz.sandor@vik.bme.hu

* Correspondence: aysucoskun@edu.bme.hu

Abstract: This paper presents the clustering and classification of the radar scattering characteristics of vehicles under real-world driving conditions. The classification of 14 distinct vehicle types is achieved through statistical features derived from their radar cross-section (RCS) characteristics, represented as histograms. Various machine learning classification techniques are applied, and their performance is evaluated across different clustering scenarios. The results of the clustering algorithm are in line with the physics-based expectations on the scattering from different vehicle types. The classification results demonstrate the effectiveness of the proposed algorithm, validating the histogram-based feature method as a novel and promising approach for vehicle identification and detection. In addition, the results highlight the potential applications of our methods in millimeter-wave (mmWave) radar technology, illustrating their capability to improve feature extraction by means of RCS histograms and ensure robust classification in diverse and challenging environments.

Keywords: sensor data classification; clustering; histogram-based features; machine learning; radar cross-section; supervised learning; support vector machine



Academic Editors: Marcin Witzak and Jose Luis Calvo-Rolle

Received: 15 December 2024

Revised: 29 January 2025

Accepted: 13 February 2025

Published: 15 February 2025

Citation: Coşkun, A.; Bilicz, S. Data-Driven Clustering and Classification of Road Vehicle Radar Scattering Characteristics Using Histogram-Based RCS Features.

Electronics **2025**, *14*, 759. <https://doi.org/10.3390/electronics14040759>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Millimeter-wave (mmWave) radar systems are highly effective for precise, real-time object recognition and motion tracking due to their short wavelengths. Their high resolution and ability to operate in various environmental conditions make them ideal for target detection and identification. The radar cross section (RCS), representing scattered power, plays a key role in performance, and accurate analysis across angles and distances enhances object differentiation in cluttered environments. Recent advances in machine learning (ML) and deep learning (DL) have significantly improved radar systems' efficiency, reducing computational resource demands while enhancing classification, tracking, and detection in complex settings.

MmWave radar systems provide high-resolution target classification and tracking in challenging conditions (e.g., dust, fog), making them ideal for air and ground applications. Their accurate, real-time tracking of multiple objects meets the different sector's high expectations for reliable functionality [1,2] in complex environments, multi-target detection [3], indoor real-time crowd counting and tracking [4], and gesture recognition [5].

Object detection using mmWave radar and AI techniques offers significant advancements, such as improved road user tracking for automated driving [6] and enhanced accuracy in low-light and small object scenarios [7]. Applications include a deep learning-based classification for vehicle and pedestrian recognition with reduced computational

complexity [8], ground target tracking [9], and detecting small, distant objects effectively in challenging environments [10,11], such as airborne targets [12]. In [13,14], a novel target detection method is introduced that integrates neural networks with histogram statistics, reducing computational complexity while maintaining high detection accuracy and addressing traditional algorithm limitations and the high demands of deep learning.

New contributions are being made by introducing fusion solutions such as radar and camera data [15] or using kinematic features (e.g., velocity, acceleration, turn rate), and RCS statistics [16] are employed to enhance the classification performance. Histogram statistics are becoming a hot topic as they reduce the computation associated with deep learning algorithms [13]. Balancing computational efficiency and classification accuracy under real-world conditions still requires methods that are as yet underdeveloped. Our work addresses this challenge by introducing a new integration of histogram-based features and a clustering-based superclass strategy for RCS histogram classification. This work's histogram-based feature classification method is tailored to a smaller dataset, with a comprehensive evaluation of its performance using real-world mmWave radar data collected under controlled slalom driving scenarios. This study investigates the statistical properties of RCS, which are influenced by factors such as target size, shape, orientation, and material composition. RCS is typically analysed using histograms to capture its distribution characteristics. Recent research has introduced more efficient methods for target classification based on RCS distributions [17,18]. Building on these advancements, this work explores the RCS properties of 14 distinct vehicle types (see Figure 1) under real-world driving conditions, employing a controlled slalom driving scenario to ensure consistent reproducibility across various angles. The dataset is openly available and presented in detail in [19]. The studied set of vehicles forms a sort of representative collection of typical motor vehicles.

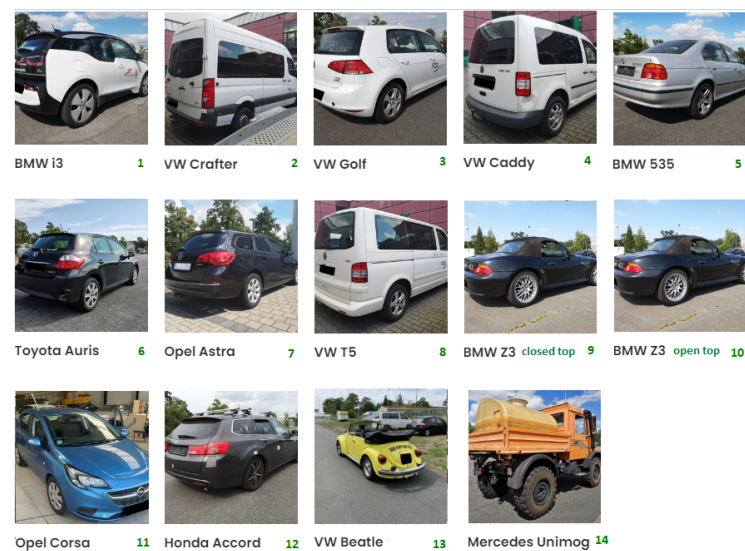


Figure 1. Dataset objects used for histogram-based feature classification, representing 14 distinct classes. The BMW Z3 is included in two variants within the dataset: one with an open top (9) and the other with a closed top (10) [19].

Methodology Outline

The methodology is outlined as follows, which is in line with the structure of this manuscript:

- The field measurements performed by the mmWave radars are used to generate histograms for each vehicle class, i.e., a *feature extraction* is carried out. The measurement and the obtained histograms are presented in Section 2.

- The classification problem is assigned, and a preliminary analysis is carried out. The set of histograms are pre-processed: vehicle classes are aggregated into “superclasses”, with the aim of forming a lower number of categories that are easier to separate by the subsequent classification algorithms. This *pre-processing* step is detailed in Section 3.
- Various classification algorithms are thoroughly tested for the solution of the targeted problem, and the results are discussed in Section 4.

Finally, Section 5 concludes the work, along with a discussion of future perspectives. To define the main contribution of this work, the context of the recent research presented in [17,18] is to be considered. In those works, a coherent path is built from mmWave scattering, via a specific type of its statistical representation, to learning methods applied therein. Now, the present manuscript takes one step forward by considering real measured data. It is shown that the previous algorithms, which showed great performance with synthetic data, are still able to cope with real measurements, provided that appropriate pre-processing is carried out first.

2. Feature Extraction from Measured RCS Data

2.1. Accessing the Raw Data

In previous works [17,18], it has been shown that histograms generated from the RCS samples collected over a time interval can be used as features that are meaningful, with respect to the target’s properties. Therefore, the first step of the present work is to generate histograms from the field measurements. The experimental setup is presented in detail in [19]. The most relevant attributes from our viewpoint are summarised as follows:

- Thirteen vehicles are considered; one of them has two variants, hence fourteen types of vehicle are given.
- For each of the 14 types, $n_t = 10$ test runs are performed on the same vehicle slalom trajectory.
- Each slalom trajectory consists of $n_p = 10$ periods of sinusoidal shape.
- Six commercial 77 GHz FMCW radars (Continental ARS408) are mounted at different heights, aligned in a vertical line on a dedicated vehicle to perform the measurements. This car follows the one actually measured at a speed of 30 km/h. In most cases, the $n_r = 5$ of the six radars provide meaningful RCS measurements.
- Determined by the sampling frequency, the speed of the vehicle and the length of the trajectory, $n_s \approx 1100$ RCS samples are registered during each test run. Note that the total RCS of the object is meant here, as defined in Equation (2) of the source work [19]. The total RCS is the linear sum over the of the detections sensed by the FMCW radar.

As an illustration, we present one time series of the RCS record for two vehicles in Figure 2. For No. 6, the periodicity is clearly seen. Note that the number of peaks is 20, i.e., twice the number of sinusoidal periods, since the vehicle is seen at similar view angle twice per period.

2.2. Generating the Histograms and Defining Classes

From the measurement procedure outlined above, $n_t \times n_r \times n_s \approx 55,000$ RCS samples are collected for each vehicle type. We partitioned these samples into segments so that each segment approximately corresponds to one period of the slalom. There were $n_t \times n_p = 100$ periods, and $n_r = 5$ sensors performed measurements (in most cases), hence ≈ 550 of such segments are formed, each containing ≈ 100 RCS samples.

To generate histograms, $n_b = 30$ equi-spaced bins are defined within the $[0, 30]$ dBsm range, and the samples within each segments are partitioned accordingly; the count values are normalised so that they sum up to 1, i.e., the histogram directly approximates the probability density function. In this way, one histogram of the n_b bins is made from each

segment of RCS data, and the ≈ 550 of such histograms become available for each of the $N = 14$ vehicle types. The RCS histograms within the k -th class are $\mathbf{h}_i$, and $i \in \mathbb{I}_k$, where $\mathbb{I}_k$ denotes the index set of the k -th class, for $k = 1, 2, \dots, N$. In the following section, these histograms are considered as features on which the classification algorithms are based. A vehicle *class* is defined as the ensemble of the histograms corresponding to this type of vehicle. In illustration form, some histograms are shown in Figure 3. Note that the difference between the time series in Figure 2 was obvious in some respects, but it was not straightforward to describe. However, the histograms in Figure 3 apparently provide a compact description of the RCS statistics, which also makes the difference between Class 6 and 1 easy to quantify.

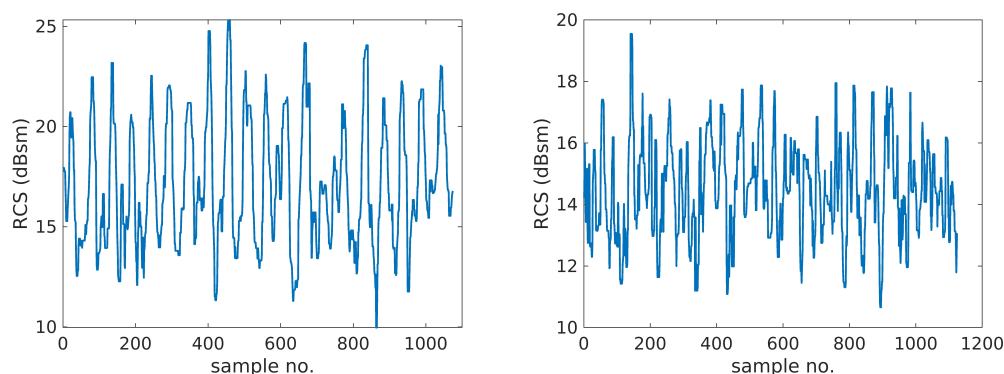


Figure 2. RCS record by one sensor in one test run for vehicle No. 6 (left) and No. 1 (right).

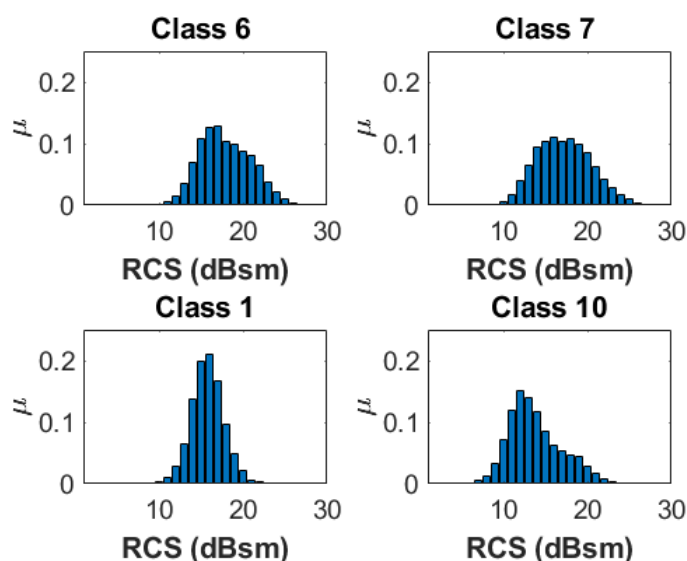


Figure 3. Examples for histogram distribution from different vehicle classes. Horizontal axes: RCS (in dBsm), vertical axes: empirical probability density.

3. Data Pre-Processing and Clustering for Enhanced Vehicle RCS Analysis

The main objective of this work is to construct a classification algorithm that is able to return the class label (that is, the type of vehicle) based on an RCS histogram resulting from a measurement performed under similar conditions, as seen above. The sought classifier will use the histograms resulting from the procedure in the previous section (in total $n = 6730$ histograms) as training data. A naive approach may apply a supervised learning method directly on this dataset, aiming at solving a 14-class classification problem. However, we could not reach an acceptable accuracy in this way. Our studies resulted in

the conclusion that there is a very high similarity between certain classes; therefore, they cannot be separated by classifiers.

3.1. Reduction of the Complexity of the Classification Problem

In Figure 3, the high similarity between Class 6 and 7 that hinders their separation is clearly seen. However, Class 1 and 10 are expected to be separable. This inspired the following three-step heuristics that we apply in this work.

Step (1) Clustering all histograms as they were unlabelled, by using the k-means algorithm. Let the number of classes required from k-means be set to $N = 14$, i.e., same as the number of original classes. This algorithm yields the index sets $\mathbb{J}_k$, each containing the indices of the samples within the k -th cluster ($k = 1, 2, \dots, N$). The goal of this step is to explore the inherent property of the dataset in terms of separability. In the very best case, the partitioning defined by the k-means clusters would perfectly match the original labelling of the histograms, meaning that the classes are easily separable.

Note that the $L1$ -norm is used as a metric in the space of the histograms, both in the k-means algorithm and whenever *distance* is referred to in the following. $L1$ -norm reads as

$$\text{dist}(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=1}^{n_b} |h_1^{(i)} - h_2^{(i)}|, \quad (1)$$

where the superscript refers to the i -th bin of the histogram.

Step (2) Mapping the clusters to their corresponding vehicle classes. The k-means clusters are to be labelled so that the i -th cluster matches the i -th vehicle class as much as possible ($i = 1, 2, \dots, N$).

A mapping is formally defined by a permutation σ , which assigns a label l to the k -th cluster: $l = \sigma(k)$, $k, l = 1, 2, \dots, N$, so that each cluster has a different label.

The quality of a certain mapping is evaluated by means of the confusion matrix C , that is defined in the standard way

$$C_{ij} = \#(\mathbb{I}_i \cap \mathbb{J}_{\sigma(j)}), \quad (2)$$

where the symbol $\#$ stands for the cardinality. The goodness of the mapping is defined as the ratio of true positive matches and the total number of samples:

$$Q = Q(\sigma) = \frac{1}{n} \sum_{i=1}^N C_{ii}. \quad (3)$$

The best permutation yields the maximal goodness:

$$\hat{\sigma} = \arg \max Q(\sigma). \quad (4)$$

Since $N!$ number of different permutations exist, an exhaustive search for solving (4) is intractable (in our case, $N = 14$, $N! = 8.7 \times 10^{10}$). Therefore, a heuristic optimisation strategy is applied to find an approximate solution for (4). This algorithm is given as follows.

- (a) Calculate the matrix D of mean distances between classes and clusters. Each element of this matrix is given as

$$D_{ij} = \frac{1}{(\#\mathbb{I}_i)(\#\mathbb{J}_j)} \sum_{p \in \mathbb{I}_i} \sum_{q \in \mathbb{J}_j} \text{dist}(\mathbf{h}_p, \mathbf{h}_q). \quad (5)$$

- (b) Find the indices $\hat{i}, \hat{j}$ which correspond to the minimum distance:

$$[\hat{i}, \hat{j}] = \arg \min_{(i,j) \in \{1,2,\dots,N\}^2} D_{ji}. \quad (6)$$

- (c) Define the rule $\sigma(\hat{j}) = \hat{i}$, i.e., the permutation will assign the label $\hat{i}$ to the $\hat{j}$ -th cluster.
 (d) Cancel out the $\hat{j}$ -th cluster and $\hat{i}$ -th label from the subsequent iterations as they are now taken. Then, set the $\hat{j}$ -th column and $\hat{i}$ -th row of D to zero.
 (e) Go back to step (ii) until all elements of D are zero.

In Figure 4, the result of this optimisation procedure is given by the confusion matrix. Some interesting properties of the classification problem are readily available based on this matrix; just to name a few:

- Classes 6 and 7 are quite confused, i.e., hard-to-separate. On the other hand, very little overlap is experienced between Class 1 and 10. Both findings are in line with the histograms shown in Figure 3.
- Classes 9 and 10 are also very confused. Note that they correspond to the same vehicle (BMW Z3), with closed and open tops; hence, indeed, little difference in the RCS is expected.
- Class 14 has little overlap with any other class; it is well-separated. Note that this corresponds to the Mercedes Unimog, with its size and shape significantly different from the other vehicles under test.

Q = 0.39837

1	156	6	8		6		4	4	10	6	49	143	87	
2	79	238	90		17		6	26	7		13	8	16	
3	6	57	160	5	52	29	23	116			41	3	8	
4		2	16	380	13	15	2	22			5			15
5	1		3			1	45	30	55	132	226		7	
6	11	18	62	1	131	170	51	8			7	23		18
7	8	4	41	4	73	116	116	22	21	3	31	42	10	9
8		3	58	28	4	11	32	209	2	11	141			1
9	3					1	36	2	182	186	28	3	59	
10	1					1	14	1	110	132	13		28	
11	1	1	43				45	61	35	40	201		13	1
12	77	9	36		65	64	95	4	10	2	10	90	37	1
13	71		2		2		38	1	154	23	24	24	201	
14				2	41	11								446
	1	2	3	4	5	6	7	8	9	10	11	12	13	14

cluster label (j)

Figure 4. Confusion matrices after mapping the clusters to their corresponding vehicle classes.

Step (3) Merging classes. The quality of the match between clusters and classes has been used to characterise the complexity of the classification problem. The confusion matrix provides us with important knowledge on how this complexity can be reduced if certain classes are merged. It is clear that by merging those classes that are highly confused with each other, on the one hand, we accept that it is impossible to separate them; on the other hand, the remaining classification problem (with one less class) can obviously be solved with higher accuracy.

The implementation of the merging is a combination of both an exhaustive and greedy search. Each iteration consists of two stages:

- (i) All possible merging is exhaustively considered: two rows and two columns of the confusion matrix are to be added to form a new row and a column, while the original ones are suppressed, and Q is calculated for the new confusion matrix. The merging providing the highest Q is effectively performed (greedily).

- (ii) The merging providing the highest Q is effectively performed (greedily). We call *superclass* the result of a merging. It may happen that a superclass will be merged with an original class or with another superclass. In this case, we go back to (i).

These steps are repeated until the number of superclasses reaches a pre-defined lower limit. Constraints may be set for the maximum size of superclasses, i.e., how many of the original classes are allowed to be included at most.

3.2. Results of the Complexity Reduction

The results of the heuristics presented in Section 3.1 are discussed herein. The algorithm has been run with three different settings to obtain 4, 5 and 7 superclasses at the end, respectively. The output is summarised in Table 1, with the corresponding confusion matrices shown in Figure 5. It is important to note that as the final number of superclasses increases, meaning more vehicle categories need to be distinguished, the classification problem becomes more challenging, as indicated by decreasing goodness of match Q . Also, the remarks on the typical class merging (reflecting the underlying physical phenomena) made in Step 2 in Section 3.1 still apply to the results seen in Table 1.

Table 1. Summary of complexity reduction results: Case 1 represents the 4 superclass scenario, Case 2 represents the 5 superclass scenario, and Case 3 represents the 7 superclass scenario, along with the class distribution within each superclass.

Case	Superclass Label	Classes Within the Superclass
Case 1	1	1, 6, 7, 12
	2	2, 9, 10, 13
	3	3, 5, 8, 11
	4	4, 14
Case 2	1	1, 2, 12
	2	3, 6, 7
	3	4, 14
	4	5, 8, 11
	5	9, 10, 13
Case 3	1	1, 12
	2	2, 13
	3	3, 8
	4	5, 11
	5	4, 14
	6	6, 7
	7	9, 10

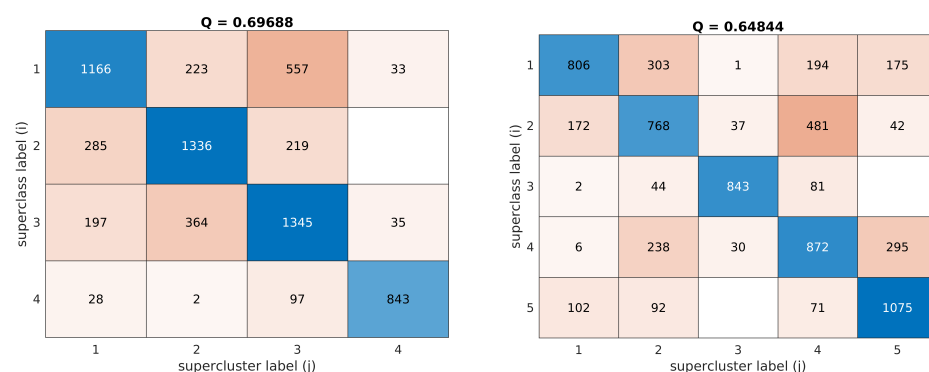


Figure 5. Cont.

Q = 0.56419

1	466	139	52	1	130	163	28
2	182	455	119		56	44	184
3	9	68	543	34	238	95	13
4		2	38	843	59	28	
5	2	21	137	1	427	91	262
6	84	32	133	32	242	453	24
7	7	87	3		41	52	610
	1	2	3	4	5	6	7

supercluster label (j)

superclass label (i)

Figure 5. Confusion matrices corresponding to the cases presented in Table 1.

4. Comprehensive Analysis of Classification Results

In this section, we present the classification results for the real-life problem investigated using the histogram-based feature set, as described in detail previously. In [18], a similar study was conducted using synthetic data, which allowed us to control various parameters and conditions to demonstrate that our *histogram-based feature* classification method provides an innovative solution to real-world classification problems.

In the current work, the focus has shifted to measured field RCS data, which present significantly greater challenges. Unlike synthetic data, where conditions are idealised, measured data introduce real-world complexities such as noise, variability, and unpredictable external factors. This transition represents a critical advancement in our research, as it allows us to evaluate the system's performance in practical environments, providing more accurate and applicable insights.

The dataset, comprising 6730 observations with 30 histogram bins, was divided into three subsets: 20% for testing, 20% for validation, and the remaining 60% for training. Each of the 30 bins was individually driven into the classification algorithm to validate the proposed concept of *histogram-based feature* classification.

The initial evaluation assessed the classification accuracy of various machine learning models using ten-fold cross-validation. The models examined include Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Gaussian Naive Bayes (NB), Support Vector Machine (SVM) [20], and Decision Tree (CART) [21], all applied with their default configurations.

Initially, all algorithms were tested using their default parameters as a baseline to establish their performance without optimisation. This choice was made to evaluate the classification algorithms' effectiveness and provide a fair comparison under uniform conditions. The accuracy mean and standard deviation are computed using ten-fold cross-validation, allowing us to identify the most promising algorithms for further refinement. The default parameters are often designed to perform reasonably well across various benchmarking datasets, so the default parameter setting for initial configuration serves practicality and efficiency. Following the step after the baseline evaluation, a detailed parameter tuning process is used to reach the optimum performance by grid search with cross-validation to tune parameters of well-performed supervised algorithms, such as kernel types, along with their corresponding parameters to refine the model performance. This systematic approach of initial investigation starts with default settings and progresses to the optimum performance of the dataset and the applied methodologies. In preliminary studies, standardisation techniques such as StandardScaler and MinMaxScaler were explored. StandardScaler normalises features by removing the mean and scaling to unit variance, ensuring a mean of zero and a standard deviation of one, while MinMaxScaler rescales features to a [0, 1] range by subtracting the minimum and dividing by the feature range. However, neither

technique demonstrated significant improvements in model performance, as shown in Figure 6. Therefore, no scaling methods were applied in the subsequent analyses. The observed insensitivity may be due to inherited data properties; it is worth highlighting that the data themselves are normalised as the sum of all bins of the histogram is 1, referring to probability normalisation ensuring that each histogram is already scaled to a consistent range, and mitigating the requirements of additional scaling steps. In addition, the histogram-based feature classification's nature implies that the histogram bins' relative proportions carry the crucial classification information for the implementation, rather than absolute values. Since scaling techniques implemented in this work scale the data into their absolute values, their effect does not impact the performance of the classification algorithms used in this study effectively.

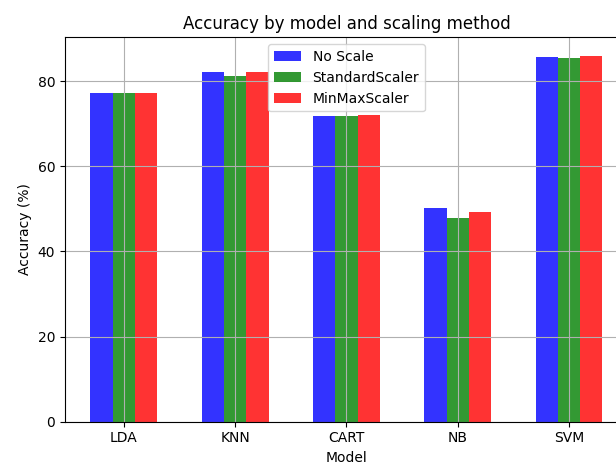


Figure 6. Evaluation of different standardisation techniques on 4 superclass clustering classification performance using various supervised algorithms, demonstrating that initial scaling methods are unnecessary for the dataset.

Upon analysis, SVM consistently demonstrated the strongest performance across all three distinct clustering scenarios. In contrast, the NB and CART algorithms showed lower accuracy than the other models, as detailed in Table 2.

Table 2. Algorithm performance across three clustering scenarios prior to parameter tuning: Case 1 (4 superclass), Case 2 (superclasses), and Case 3 (7 superclasses).

Case	Superclass Label	Classes Within the Superclass
Case 1	LDA	77.08
	KNN	81.54
	CART	71.91
	NB	49.69
	SVM	85.86
Case 2	LDA	77.39
	KNN	76.46
	CART	65.36
	NB	47.09
	SVM	81.01
Case 3	LDA	68.01
	KNN	68.73
	CART	57.67
	NB	39.67
	SVM	76.16

4.1. Algorithm Tuning and Classification Results

This section presents the fine-tuning of parameters for the SVM algorithm, identified as the most promising model in the previous phase. This tuning builds on the prior evaluation of SVM performance across three distinct clustering scenarios, with a pipeline implemented to integrate SVM within each scenario. Parameter tuning is conducted using grid search with five-fold cross-validation on the training data to identify optimal settings for constructing an effective SVM classification model. This approach also verifies whether the model overfits the training data. The grid search process evaluates multiple kernel functions, including ‘linear’, ‘polynomial’, ‘rbf’, and ‘sigmoid’ [22], along with their corresponding hyperparameters to refine model performance. The model performance was assessed by taking into account the following hyperparameter ranges:

- C and γ values are divided into 10 equal intervals between 0.1 and 1.0.
- $degree$ values of 2, 3, and 4 are considered.

For example, the refinement of ‘polynomial’ kernel for C , γ , and $degree$ and the optimisation of the ‘linear’ kernel for the C parameter could be given as an example of hyperparameter search. No significant performance difference was observed between the ‘rbf’ and ‘polynomial’ kernels in terms of accuracy; for this reason, the ‘rbf’ kernel was selected for modelling in each superclass classification scenario. The regularisation parameter C and kernel parameter γ were fine-tuned for each clustering scenario to refine model performance.

4.1.1. Case (1): 4 SuperClass

The optimal configuration for the 4 superclass scenario was achieved using the ‘rbf’ kernel with parameters $C = 1.0$ and $\gamma = 1.0$, yielding an accuracy of 83.06 % on the testing dataset. The decision function shape applied was one-vs-one (ovo), which is a robust approach for multi-class model training.

The classification performance of the SVM model was evaluated using several metrics, including precision, recall, and F1-Score across different classes. Precision represents the proportion of correct positive predictions, while recall (or sensitivity) assesses the model’s ability to identify all positive samples. The F1-Score, a harmonic mean of precision and recall, is particularly useful for handling imbalanced class distributions. The results for these metrics are summarised in Table 3.

Table 3. Classification performance metrics for the 4 superclass scenarios, achieved using the SVM model with the ‘rbf’ kernel and parameters $C = 1.0$ and $\gamma = 1.0$.

Class Labels	Precision	Recall	F1-Score	Support
0	0.77	0.80	0.79	414
1	0.82	0.79	0.80	377
2	0.86	0.87	0.86	375
3	0.94	0.92	0.93	180

The model’s capability to distinguish between superclasses is further assessed using the receiver operating characteristic (ROC) curve. In Figure 7, each curve represents the ROC for a specific class, labelled with integers from the test dataset. A higher Area Under the Curve (AUC) value signifies stronger model performance, with class label 3 achieving an AUC of 1.00, indicating perfect classification for this class.

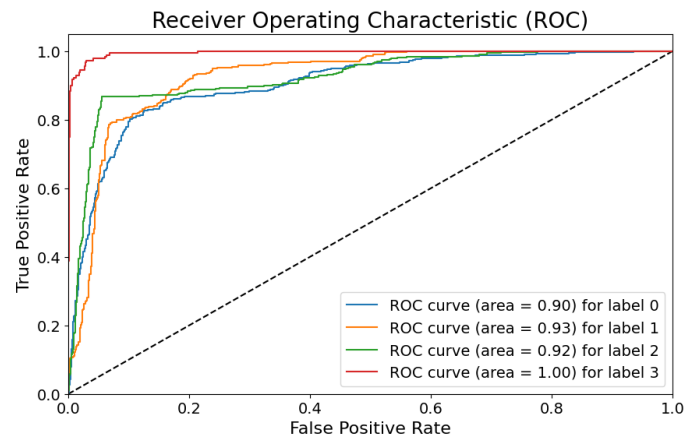


Figure 7. Receiver operating characteristic (ROC) curve analysis for the 4 superclass testing data. The AUC for class label 3 is 1.00, indicating perfect classification for this class.

Additionally, the model performance is further illustrated by a confusion matrix (see Figure 8), which illustrates the alignment between true and predicted labels on the test dataset. The model demonstrates robust performance, achieving an accuracy of 83.06 % on the test set.

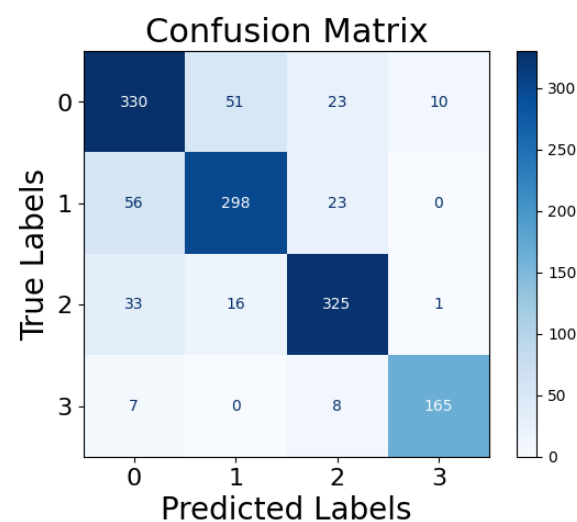


Figure 8. Confusion matrix analysis for the 4 superclass testing data, showing an overall accuracy of 83.06 % on the test set.

4.1.2. Case (2): 5 SuperClass

Another optimal classification performance using SVM was achieved with the 'rbf' kernel and parameters $C = 0.89$, yielding an accuracy of 77.11 % on the testing dataset. The classification performance metrics for the 5 superclass approach are detailed in Table 4.

Table 4. Classification performance metrics for the 5 superclass scenario, achieved using the SVM model with the 'rbf' kernel and parameters $C = 0.89$ and $\gamma = 1$

Class Labels	Precision	Recall	F1-Score	Support
0	0.75	0.74	0.74	314
1	0.71	0.66	0.69	298
2	0.97	0.91	0.94	206
3	0.74	0.79	0.76	267
4	0.76	0.80	0.78	261

Figure 9 illustrates the ROC curve for each superclass, labelled with integers from the testing dataset. Notably, for label 2, the AUC is 1.00, indicating perfect classification for this class.

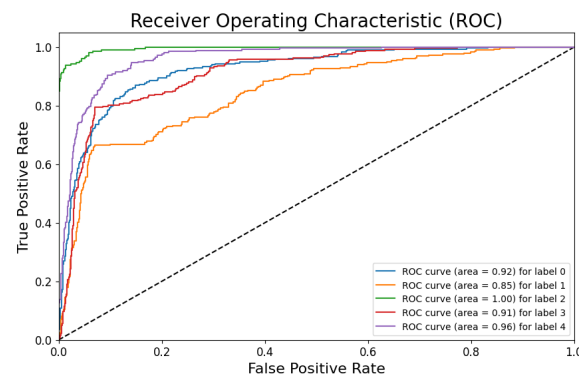


Figure 9. Receiver operating characteristics curve analysis for the 5 superclasses testing data. The AUC for class label 2 is 1.00, indicating perfect classification for this class.

The model's performance is also depicted by the confusion matrix in Figure 10. Specifically, the class labelled 0 exhibits moderate misclassification. Overall, the model performs well, achieving an accuracy of 77.11 % on the test set.

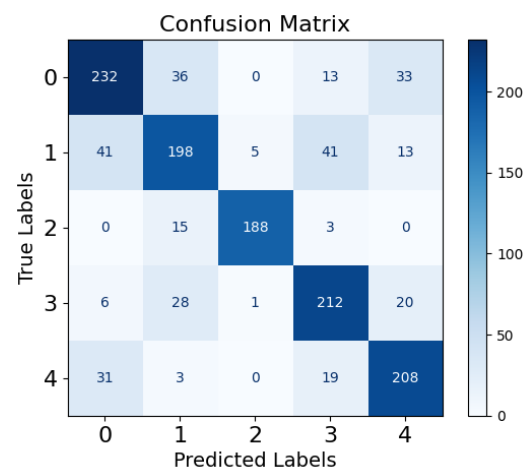


Figure 10. Confusion matrix analysis for the 5 superclass testing data, showing an overall accuracy of 77.11 % on the test set.

4.1.3. Case (3): 7 SuperClass

The last optimal classification performance with the SVM model was achieved with the 'rbf' kernel and parameters $C = 0.89$, yielding an accuracy of 77.71 % on the testing dataset. The classification performance metrics for the 7 superclass approach are detailed in Table 5.

Table 5. Classification performance metrics for the 7 superclass scenario achieved using the SVM model were achieved with the 'rbf' kernel and parameters $C = 0.89$, resulting in an accuracy of 77.71%.

Class Labels	Precision	Recall	F1-Score	Support
0	0.71	0.69	0.70	94
1	0.78	0.66	0.71	106
2	0.87	0.78	0.83	106
3	0.76	0.84	0.80	104
4	0.92	0.90	0.91	91
5	0.66	0.75	0.70	95
6	0.75	0.84	0.79	77

Figure 11 presents the ROC curves for each superclass, labelled with integers from the test dataset. This is complemented by the confusion matrix analysis shown in Figure 12, which provides further insights into the model's classification performance across the superclasses.

4.2. General Discussion of the Classification Results

The SVM model demonstrated the strongest classification performance across all superclass scenarios, with the following highlights:

- The highest accuracy performance, 83.06 %, was achieved in the 4 superclass cases, with particularly excellent classification performance exhibited in the class labelled '3'.
- Moderate accuracy performance was observed for the 5 superclass (77.11 %) and 7 superclass (77.71 %) scenarios; both performed well in terms of AUC values across most classes.

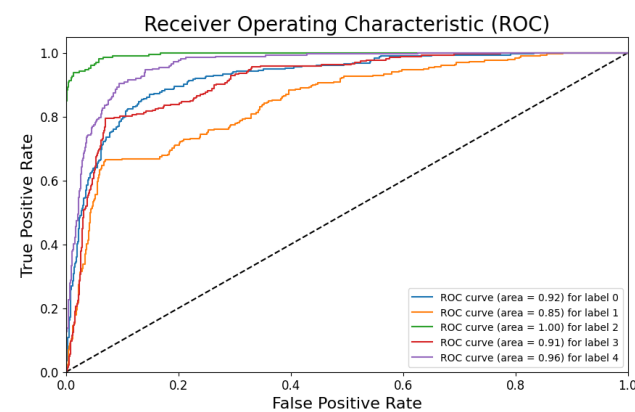


Figure 11. Receiver operating characteristics curve analysis for the 7 superclasses testing data.

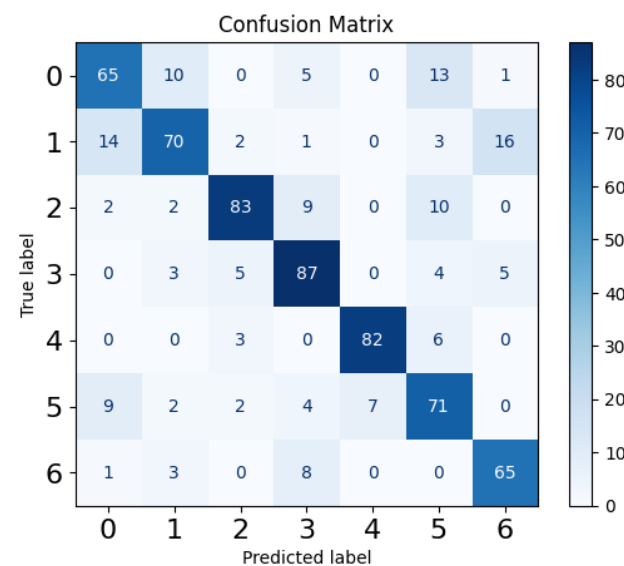


Figure 12. Confusion matrix analysis for the 7 superclass testing data, presenting an overall score of 77.71 %.

Overall, good classification performance has been demonstrated, proving the effectiveness of varying levels of class granularity. Additionally, the impact of the distribution of vehicles within each superclass on performance has been clearly illustrated, as previously discussed in Section 3.2.

Traditional ML methods, specifically SVM, have been shown to perform comparably to neural networks in similar contexts, as demonstrated in previous work [18]. This applies

especially when dealing with structured data, where efficiency and interoperability are critical. Therefore, it is crucial to highlight the study's context and constraints to justify the preference for traditional ML tools over neural networks. This choice is driven by the following two main factors:

1. **Data Limitations:** Larger datasets and extensive parameter tuning are essential when using neural networks, which is where data limitations become problematic particularly due to their reliance on large-scale data for generalisation. The available dataset is relatively small and has limited diversity in vehicle types and driving scenarios, so it is better suited for traditional ML models like SVM, which provide robust and stable performance without the risk of overfitting, provided the correct hyperparameters are adjusted. Comprehensive evaluation and improved generalisation across diverse scenarios, such as urban roads and highways, can be performed by extending the dataset to offer a broader variety of vehicles and their driving conditions.
2. **Comparable Performance:** The performance of traditional ML models and neural networks can be closely matched, especially with structured information and smaller datasets, as demonstrated in [18]. For this reason, given the benefits of reduced computational costs, traditional ML models offer a more practical and efficient solution for this study and the current dataset.

In today's neural network-dominated landscape, it is essential to justify this choice, highlighting that traditional ML approaches remain highly relevant, particularly when data size and resource constraints limit the feasibility of neural networks. This emphasises the practical advantages of traditional ML models for our study.

5. Conclusions

This study introduces a robust data-driven classification method using the *histogram-based features* for a real-world vehicle dataset consisting of 14 distinct radar cross-section (RCS) classes. The data were collected from vehicles navigating a slalom course, observed by multiple radar sensors. The primary goal is to mitigate uncertainties in radar perception caused by human driving behaviour. The *histogram-based feature* classification approach analyses RCS characteristics using histograms and clustering, integrating a superclass strategy alongside supervised machine learning techniques. Each method was thoroughly tested and evaluated.

The output of the clustering algorithm is in line with certain physics-based expectations on the scattering from different vehicle types. In particular, the large-sized truck seems to be easily separable from the smaller passenger cars, and when the same car appears with top open or closed, this makes little difference in the RCS feature. Furthermore, the classification results demonstrate that the proposed data-driven approach significantly improves classification performance under varying noise conditions, simulating real-world driving scenarios. As the number of superclasses increases, requiring the distinction of more vehicle categories, the classification challenge becomes more pronounced, as evidenced by the detailed analysis provided. These findings offer valuable insights into target classification and identification, presenting potential advancements for radar-based vehicle perception systems.

Author Contributions: Conceptualization, A.C. and S.B.; Methodology, A.C. and S.B.; Software, A.C.; Validation, A.C.; Formal analysis, A.C.; Investigation, A.C.; Writing—original draft, A.C. and S.B.; Visualization, A.C.; Supervision, S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jiuming, L.; Jingqing, L. Millimeter-wave Radar Application in Tracking Maneuvering Target. In Proceedings of the 2006 CIE International Conference on Radar, Shanghai, China, 16–19 October 2006; pp. 1–4. [\[CrossRef\]](#)
2. Russell, M.; Crain, A.; Curran, A.; Campbell, R.; Drubin, C.; Miccioli, W. Millimeter-wave radar sensor for automotive intelligent cruise control (ICC). *IEEE Trans. Microw. Theory Tech.* **1997**, *45*, 2444–2453. [\[CrossRef\]](#)
3. Shamsfakhr, F.; Macii, D.; Palopoli, L.; Corrà, M.; Ferrari, A.; Fontanelli, D. A multi-target detection and position tracking algorithm based on mmWave-FMCW radar data. *Measurement* **2024**, *234*, 114797. [\[CrossRef\]](#)
4. Li, S.; Hishiyama, R. An Indoor People Counting and Tracking System using mmWave sensor and sub-sensors. *IFAC-PapersOnLine* **2023**, *56*, 7096–7101. [\[CrossRef\]](#)
5. Yan, B.; Wang, P.; Du, L.; Chen, X.; Fang, Z.; Wu, Y. mmGesture: Semi-supervised gesture recognition system using mmWave radar. *Expert Syst. Appl.* **2023**, *213*, 119042. [\[CrossRef\]](#)
6. Tilly, J.F.; Haag, S.; Schumann, O.; Weishaupt, F.; Duraisamy, B.; Dickmann, J.; Fritzsche, M. Detection and Tracking on Automotive Radar Data with Deep Learning. In Proceedings of the 2020 IEEE 23rd International Conference on Information Fusion (FUSION), Rustenburg, South Africa, 6–9 July 2020; pp. 1–7. [\[CrossRef\]](#)
7. Song, Y.; Xie, Z.; Wang, X.; Zou, Y. MS-YOLO: Object Detection Based on YOLOv5 Optimized Fusion Millimeter-Wave Radar and Machine Vision. *IEEE Sens. J.* **2022**, *22*, 15435–15447. [\[CrossRef\]](#)
8. Zhang, L.; Xu, S.; Li, J. CNN Based Target Classification in Vehicular Networks with Millimeter-Wave Radar. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 19–22 June 2022; pp. 1–6. [\[CrossRef\]](#)
9. Vega, D.; Newey, M.; Barrett, D.; Axelrod, A.; Myne, A.; Wollaber, A. STAP-informed Neural Network for Radar Moving Target Indicator. In Proceedings of the 2024 IEEE Radar Conference (RadarConf24), Denver, CO, USA, 6–10 May 2024; pp. 1–6. [\[CrossRef\]](#)
10. Amiriara, H.; Andargoli, S.M.H.; Meghdadi, V. Efficient Radar-Target Assignment in Low Probability of Intercept Radar Networks: A Machine-Learning Approach. *IEEE Open J. Commun. Soc.* **2023**, *4*, 2165–2175. [\[CrossRef\]](#)
11. Das, N.; Swarna, R.N.; Hossain, M.S. Deep learning-based circular disk type radar target detection in complex environment. *Phys. Commun.* **2023**, *58*, 102014. [\[CrossRef\]](#)
12. Holt, D.; Guo, W.; Sun, M.; Panagiotakopoulos, D.; Warston, H. Deep Learning for Radar Classification. In Proceedings of the 2024 International Conference on Unmanned Aircraft Systems (ICUAS), Chania, Greece, 4–7 June 2024; pp. 1208–1215. [\[CrossRef\]](#)
13. Shuai, J.; Yalong, P.; Luyuan, W.; Jiyang, Y.; Bowen, C.; Zongling, L. A Target Detection Algorithm of Neural Network Based on Histogram Statistics. In Proceedings of the IGARSS 2020–2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020; pp. 1628–1631. [\[CrossRef\]](#)
14. Cai, X.; Giallorenzo, M.; Sarabandi, K. Machine Learning-Based Target Classification for MMW Radar in Autonomous Driving. *IEEE Trans. Intell. Veh.* **2021**, *6*, 678–689. [\[CrossRef\]](#)
15. Wu, Y. Fusion-based modeling of an intelligent algorithm for enhanced object detection using a Deep Learning Approach on radar and camera data. *Inf. Fusion* **2025**, *113*, 102647. [\[CrossRef\]](#)
16. Başkaya, K.; Kurt, E. Aerial Target Identification Based on Classification Learner. In Proceedings of the 2024 4th Interdisciplinary Conference on Electrics and Computer (INTCEC), Chicago, IL, USA, 11–13 June 2024; pp. 1–6. [\[CrossRef\]](#)
17. Coskun, A.; Bilicz, S. Target Classification Using Radar Cross-Section Statistics of Millimeter-Wave Scattering. *COMPEL* **2023**, *42*, 1197–1209. [\[CrossRef\]](#)
18. Coskun, A.; Bilicz, S. Analysis of data-driven approaches for radar target classification. *COMPEL* **2024**, *ahead-of-print*. [\[CrossRef\]](#)
19. Elster, L.; Holder, M.F.; Rapp, M. A Dataset for Radar Scattering Characteristics of Vehicles Under Real-World Driving Conditions: Major Findings for Sensor Simulation. *IEEE Sens. J.* **2023**, *23*, 4873–4882. [\[CrossRef\]](#)
20. Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
21. Geron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2019.
22. Cristianini, N.; Shawe-Taylor, J. *Support Vector Machines and Other Kernel-Based Learning Methods*; Cambridge University Press: Cambridge, UK, 2000.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.