2020-01-5150 Published 30 Dec 2020

Road-Users Classification Utilizing Roadside Light Detection and Ranging Data

Yanjie Song Shandong University, China

Jun Tian Shandong Expressway Co., Ltd., China

Tao Li Shandong Hi-Speed Construction Management Group Co., LTD., China

Renjuan Sun Shandong University, China

Han Zhang Shandong Hi-Speed Construction Management Group Co., LTD., China

Jianqing Wu and Xiuguang Song Shandong University, China

Citation: Song, Y., Tian, J., Li, T., Sun, R. et al., "Road-Users Classification Utilizing Roadside Light Detection and Ranging Data," SAE Technical Paper 2020-01-5150, 2020, doi:10.4271/2020-01-5150.

Abstract

Road-users classification plays a critical role in efficient transportation planning and management. It is also essential for a number of applications, including electronic tolling systems, roadway design, and intelligent transportation systems. This paper proposed a method to categorize road-users into distinct types using point-cloud data from light detection and ranging (LiDAR) sensors mounted on the roadside (e.g., signal or road poles). Not only motor and non-motor vehicles but also pedestrians were considered as classified objects (i.e., road-users). Firstly, data from the multi-layer LiDAR sensor were pre-processed. Then, six features representing vehicles' profiles were manually extracted from 3D cloud points as input variables for the classification procedure. Support Vector Machines (SVM), Random Forest (RF), Back Propagation Neural Network (BPNN), and Probabilistic Neural Network (PNN) supervised machine learning algorithms were employed to construct classification models. Based on the pre-defined 8-class scheme within the Federal Highway Administration (FHWA) and two types (i.e., bicycles and pedestrians) defined by authors, road-users were distinguished into ten groups. Compared with the accuracy used by most existing studies on vehicle classification, F_1 value, which is popular among the machine learning field, was taken as an indicator to evaluate the performance of classifiers. Finally, the number of features was reduced as an attempt to reduce the dimensionality of sample data for obtaining more accurate classification. The results show that the SVM model utilizing the Gaussian kernel function has the best ability to class road-users with a high F_1 value of 84%, and reducing the dimensionality of features can improve the object classification (F_1 value increased from 84% to 87.15%).

Introduction

Vehicle classification has a wide range of applications in the transportation field (e.g., roadway design, traffic planning, traffic operation, and transportation safety) [2]. With the road reconstruction and expansion prevailing worldwide, it is vital for determining appropriate roadway levels and pavement design parameters (turning radius, etc.) to know the main classes of vehicles that travel on the designed section of the road. If engineers grasp the frequency of large vehicles using road sections or the preference of heavy vehicles occupying selected lane, key reinforcement positions that roadway infrastructure safety needs to concern could be accessed readily. Under traffic management applications, the passage circumstance of each vehicle subcategory can help the surveillance department preserve traffic to flow unimpeded. Vehicle classification data are also crucial for the intelligent transportation system (ITS). For example, the automatic toll system collects the fees based on vehicle classification data, and the smart parking system also needs the classification results to assign vehicles suitable parking space. On the basis of those applications in ITS, an efficient and accurate classification method should be employed to meet actual requirements. A lot of studies have been conducted for vehicle classification using different methods. And various methods are distinguished by different data collection ways. Meta and Cinsdikici [3] developed a waveform-based method utilizing the time-variable signal collected by loop detectors to classify the vehicles into five groups. The testing results showed that the classification system could achieve the recognition rate of 94.21% in real-time traffic conditions. Obertov and Andrievsky [4] developed an axle-based method with the help of vibration detectors for four types of vehicles (car, truck, the car with trailer, and truck with trailer) classification. An accuracy of more than 89% can be achieved, and vehicle speeds could also be detected. Fuerstenberg and Willhoeft [5] developed a speed-based method using laser scanners to track and classify pedestrian or non-pedestrian. The purpose of this method was to provide an active safety system for all road-users. Odat et al. [6] developed a temperature/waveform-based method by combining passive infrared sensors (PIRs) and ultrasonic rangefinder to accomplish detecting and classifying vehicles. Though the testing results show a 99% accuracy in vehicle detection, the performance of classification was not fully analyzed. Due to the cameras are easily available and have a facilitative installation, a lot of efforts have been conducted to use the image data collected from cameras for road-users classification. Chen et al. [7] used images created by extracting pixels obtained from roadside CCTV cameras to detect and classify road-users into four types: car, van, bus, motorcycle. Kalman filter and SVM algorithms were integrated to improve accuracy. The results can achieve an accuracy of 94.69% for road-users classification under varying weather conditions. Mithun et al. [8] employed multiple time-spatial images collected from the video streams for vehicle classification. It was found that the k nearest neighborhood (kNN) algorithm could help the classification model achieve high accuracy of more than 88%. Dong et al. [9] used vehicle frontal-view images and convolutional neural network (CNN) to establish a classification system. The overall accuracy of 96.1% and 89.4% can be achieved on daylight and nightlight images, respectively. Hasnat et al. [10] presented a method by combining a camera with other optical sensors to address the problem that vehicle classification accuracy cannot satisfy the actual demand for daily applying. It should be noticed that the experimental results served for the practical use-case of automatic toll collection. Chang et al. [11] developed an image-based method using a camera mounted on the signal poles to classify vehicles into five groups. The devised CNN classifier was constructed to improve performance. The results showed that the accuracy of 97.62% could be achieved in the test. Weather conditions and vehicle occlusion problems, which have a great impact on the classification accuracy, were exposed in the above-mentioned camera-based studies. However, since the limitation of technology, these two main problems still cannot be solved well. Each data collection methods has its own weaknesses and advantages regarding performance, weather and illumination restriction, costs [12]. In order to complete the road-users classification work accurately, researchers are looking for other sensors.

Light Detection and Ranging (LiDAR) is an active sensor based on the principle of emitting laser beams and receiving echo signals. The performance of the LiDAR is not affected by illumination conditions. Compared with cameras, infrared sensors, radar, and other sensors, the point cloud data obtained from LiDAR are more accessible for computers to process. The LiDARs are now widely applied to autonomous vehicles for object perception and classification. Gao et al. [13] developed a CNN deep learning-based method by fusing pixel-level image and point cloud extracted from on-board camera/LiDAR for road-users (including pedestrian, cyclist, car, truck, and others) classification. Experimental results demonstrated that the average accuracy of 96% could be achieved using the proposed approach. Zhang et al. [14] used 64-HDL LiDAR mounted on a vehicle to classify moving objects around the test vehicle into four types. Gini index criterion was employed to select efficient features, including the number of points features, statistical features, and shape features. The results showed that the partial object occlusion problems could be solved, and the object classification accuracy can reach a value of 93.25%. A real-time object classification approach based on the Real AdaBoost algorithm was presented by Yoshioka et al. [15]. The shape-based features, intensity and velocity-based features, layered features were selected as the basis for classification. The time of training and computing the classifier was also provided for assessing the method's performance. Though results claimed that a high accuracy (over 90%) could be achieved, this approach can only classify objects into three groups. Kurnianggoro and Jo [16] developed an intelligent system to perform recognition of road-users utilizing and comparing several machine learning algorithms. There were 50 features encoded into smaller dimensions being extracted to train the efficient classification system. The testing results showed that the neural network method has the best performance with high accuracy of more than 91.64%. Since the LiDAR cost is now significantly reduced, it becomes a reality to deploy LiDAR in a large-scale distribution as the traffic roadside monitoring equipment. There are already some researchers who have attempted to employ the LiDAR at the roadside for the object classification [17, 18, 19, 20, 2, 12]. Nezafat et al. [17] used VLP-16 LiDAR installed on a roadside pole as the traffic data collection sensors for trucks traveled on two lanes classification. The CNN was used to extract the needed features for classifying the trucks into four groups (defined by FHWA). The overall correct classification rate was more than 90%. A lidar-based classification system was developed by Lee and Coifman [2] using the LiDAR sensors mounted on the roof of the test vehicle. Though the authors claimed that the vehicle classification accuracy could be as high as 99%, the occlusion issue that hinders road-users being distinguished accurately was not considered. Sahin et al. [19] advocated a truck trailer types classification method with the help of a 16-beam roadside LiDAR sensor. The kNN, SVM, Adaboost, and Multilayer Perceptron (MLP) algorithms were used to train classifiers for sorting the truck trailers into night types. It was found that the roadside LiDAR can provide much richer information compared with other sensors for truck body types classification. Wu et al. [12] constructed a Random Forest (RF) classification system by extracted six features of the road-users points from roadside LiDAR data. The overall correct classification rate can reach a value of more than 92%. The height-profile was used to improve the performance of the RF classifier. However, the reasons for feature selection were not fully explained. Asborno et al. [20] applied two low-cost single-beam LiDAR installed on the side-fire configuration to implement the truck body-types identification and classification. The true positive rate (TPR) and accuracy were used as the evaluation indicators of classification performance. The presented method can only classify the trucks' body-types with a low-speed. Though there have been many studies for road-users classification adopting the roadside LiDAR data, some limitations in the previous studies still need to be broken. First, the calculation time consumed by the classification model was not sufficiently considered to estimate whether the proposed approach meets the requirements of real-time data processing. Second, the evaluation index used in previous studies was mostly accurate. Road-users have multiple subcategories, which is challenging to maintain a balanced volume of collected data. However, the value of accuracy is not suitable for samples (including training and testing samples) imbalance issues and cannot comprehensively evaluate multi-category classification performance. In the machine learning field, the F_1 score is a matching indicator for multi-class problems. The definition and application will be elaborated in the later sections of this paper.

The primary objective of this paper is to develop a method using the point cloud data accessed from roadside LiDAR to distinguish road-uses (vehicles, bicycles, and pedestrians are contained) into ten detailed groups accurately and rapidly. In this article, the point cloud shape-based features of road-users were extracted from the raw LiDAR data draw on a previously developed LiDAR data processing system. We tested and compared the performance of different methods on road-users classification by adopting the F_1 score. It should be noted that the calculation time each classifier consumes is also be considered. The remainder of this article is organized as follows.

Road-User Point Cloud Data Extraction

Considering the real-time LiDAR data comprises other abundant information except for the road-user points, the establishment procedure of the road-users classification model utilizing the LiDAR point cloud data requires two main measures, including point cloud data extraction and classification model establishment. Those two measures can be divided into six steps: data collection, data processing, feature selection (finish building a database), model training, parameter adjustment, and classification output. The specific building process is shown in Figure 1.

Data Collection

The RS-16 LiDAR manufactured by Robosense was adopted for analysis in this paper. Table 1 shows the major parameters of the RS-LiDAR-16. The LiDAR sensor was installed on a tripod at the roadside by the authors for data collection. Compared with fixing the LiDAR on the signal or light poles, the tripod can be moved flexibly to choose the collecting location at will, which means that data for more diversified road-user types can be obtained. Since the purpose of this paper is to perform a tentative study, installing the LiDAR on the tripods are easier to operate. Three different scenes, including internal campus street (rich in pedestrians and bicycle), bus stop (rich in buses), major urban road (rich in trucks and other vehicles), were selected for data collection to serve for different demands. The height of the LiDAR sensor location is 1.6m above the ground, and the spherical camera is used for validating the collected results. Figure 2 shows the three chosen sites.

FIGURE 1 Classification establishment procedure

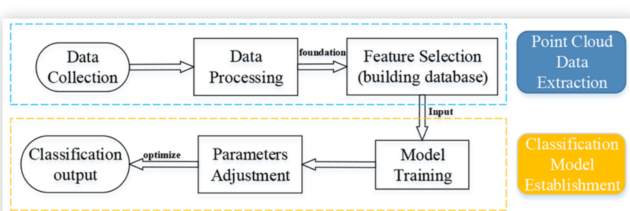


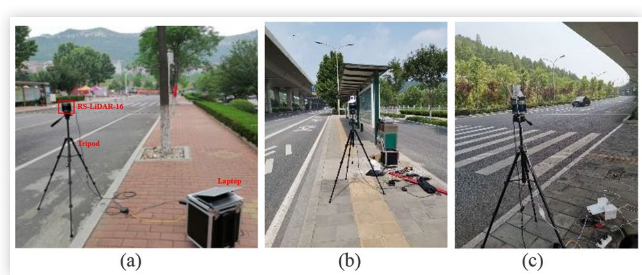
TABLE 1 Major Parameters of RS-16

Time of Flight (ToF) Distance Measurement	16 Channels
Measurement Range	40cm to 150m
Accuracy	±2cm
Field of View (Vertical)	±15°
Angular Resolution (Vertical)	2°
Field of View (Horizontal)	360°
Angular Resolution (Horizontal/Azimuth)	0.1°(5Hz) to 0.4°(20Hz)
Rotation Rate	5/10/20 Hz (300/600/1200 rpm)
Wavelength	905m
Full Beam Divergence Horizontal, Vertical	7.4 mrad, 0.7 mrad
Power Consumption	12 W
Data Rate	~300000 points/second

Data Processing

Much information, such as trees, buildings, roads, road-users, and so forth, is flooded with the raw LiDAR data. The road-user points make up only a small part of the huge-size data. For classification issues, only the road-user point cloud is worthy of attention. In order to reduce the amount of calculated data and improve the efficiency of the classification, it is necessary to precisely extract the road-user point cloud priority with the help of data processing. The applied data

FIGURE 2 Three selected collection sites: (a) campus internal street, (b) bus stop, and (c) urban major road.



processing approach in this paper was an exhaustive procedure developed by previous studies [21, 22, 23, 24]. The processing of data contains three main steps: background filtering, object clustering, and target tracking, which are briefly introduced as follows.

Background Filtering Due to the background point cloud in the raw LiDAR data making the road-user point cloud difficult to identify, a background filtering algorithm needs to be employed to filter the irrelevant points. The 3D-density-statistics-background-filtering (3D-DSF) algorithm divides the point-cloud space into multiple subspaces based on density statistics. It can achieve background filtering by deleting the subspace regarded as a background point space. The 3D-DSF method first merges multi-frame point cloud data. The difference between the moving object point cloud and background point cloud is that the latter has a higher point cloud density than the former after merging the multiple frames. This method is based on this characteristic to set a pre-defined value to filter out the background points.

Object Clustering After the background filtering, there were still some noise points that had not been filtered out, and the road-user point clouds could not yet be grouped for classification. The density-based spatial clustering of applications with noise (DBSCAN) uses the principle that the density of noise point clouds is much smaller than road-user was applied for clustering the road-user point cloud correctly. The DBSCAN method has two important parameters: epsilon (Eps) and minimum points (MinPts). Eps means that the radius of the neighborhood. MinPts represents the minimum number of neighbors within Eps. Since the noise point cloud density is smaller than the object point cloud's, the noise points can be deleted by setting the pre-defined density value. Moreover, the object points can be clustered by selecting appropriate values for these two parameters. The specific application about the DBSCAN method refers to Zhao et al. [25].

Target Tracking The moving process of the road-user is composed of multiple continuous frames in the point cloud data. For preventing misclassification caused by repeated recognition of the same target, the Global Nearest Neighbor (GNN) that identifies the main points of the same object using location information was employed as a tracking algorithm to associate the same target.

Feature Selection (Building Database)

LiDAR sensors with different numbers of beams (e.g., 16, 32, 128-beams) can generate different abundant levels for object point cloud information. The purpose of this paper is only to recognize the object's types on the road. RS-16 LiDAR is fully competent to accomplish the task, with both cost and performance being considered. The road-user point cloud information represents the contour characteristics of road-user depending on the shape-based features selection. Further, the road-users can be distinguished into different groups by using their respective shape features. For the same road-user, the point cloud information fed back by LiDAR at different detection distances is diverse. Consequently, the distance from the target to the LiDAR sensor should be employed as a selected feature. In order to separate the big-size vehicle from the small-size vehicle or non-vehicle, the numbers of points in the frame with max target length were also be selected. The six adopted features are listed as follows.

A. Max-length in the trajectory
B. Number of points in the frame with max length
C. Nearest distance from object points to LiDAR
D. Max-height in the trajectory
E. The difference between length and height
F. Target height profile

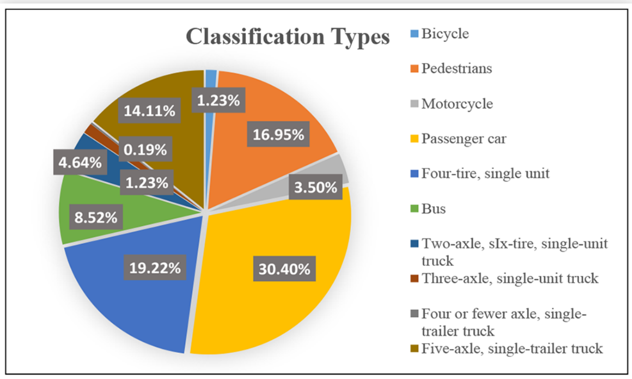
Feature A is the maximum length of the road-user's point cloud along the trajectory direction. Because the length of the road-user's point cloud along the trajectory is different in different frames, feature B is defined as the number of points in the frame with the largest length. Feature C is the nearest distance from the road-user's point to LiDAR. The definition of feature D is similar to feature A It is the maximum height along the trajectory. Feature E is defined as the difference between the maximum length and maximum height of the object's point cloud. The target height profile is defined by segmenting the road-user point cloud data into n number of bars along the driving track. A bar includes the point cloud information with the maximum and minimum z value of this small bar (the maximum z value minus the minimum z value is the bar height). The height profile of the road-user was conducted by recording each bar's height. The value of n was set to 10 in the paper. For the detailed definition and application of the features above, we refer the readers to the references [12].

The selected six features were manually marked by the authors from each road-user point cloud, and the feature information was normalized as sample data to build the database for classification. Every feature sample was labeled by actual road-user types with the help of the spherical camera validation results. A shape-based classification database containing 1,093 sample points with the matched labels was built for the input of the classifiers. The proportion distribution of the collected sample types is shown in Figure 3. Figure 3 also displays the percentage of the ten classes (defined by FHWA or authors), including bicycle (code a), pedestrians (b), motorcycle (c), passenger car (d), four-tire/single unit (e), bus (f), two-axle/six-tire/single-unit truck (g), three-axle/single-unit truck (h), four or fewer axle/single-trailer truck (i), five-axle/single-trailer truck (j).

Object Classification Model Establishment

Four machine learning algorithms, including SVM, RF, BPNN, and PNN, widely used in previous studies to classify vehicles [26], were employed to train the road-user classifiers.

FIGURE 3 The proportion distribution of the classification types.



The programming code of the classification model was achieved by Python Language and the Scikit-Learn open source library. All classifiers' testing and training sets were divided according to the ratio of 0.25. This section introduces the detailed establishment process and parameter adjustment of the four classifiers.

SVM Classifier

SVM originated in 1964 and was developed based on the VC dimension theory of statistical learning theory. Its core idea is to find an optimal hyperplane that meets the classification requirements by ensuring the hyperplane is at the maximum distance from the nearest sample point. The SVM constructs complex algorithms under simple theory and addresses difficult problems using a straightforward method. The traffic data has the characteristics of nonlinearity and instability. Since SVM is good at handle the problems with nonlinearity and small-size sample data, it is one of the most prevailing methods in the vehicle classification field.

For linear problems: if the training sample set is $\{(x_i, y_i) | i = 1, 2\ldots, n\}, x_i \in R^n$, where x_i is the input sample value and y_i is the output, then the hyperplane is expressed as:

$$(w \cdot x_i) + b = 0 \quad (1)$$

The classification interval is $\frac{2}{\| w \|}$. Introducing the Lagrange function to solve this optimization problem, then the hyperplane formula is transformed into:

$$L(w,a,b) = \frac{1}{2} \| w \| - a \cdot \left(y\left((w \cdot x) + b \right) - 1 \right) \quad (2)$$

Calculate the partial derivative of formula (2) for w and b, and let the value of partial derivative equal zero. The optimal function is:

$$f(x) = \operatorname{sgn}\left\{ \left(\sum_{i=1}^{n} a_j^* y_j \left(x_j \cdot x_i \right) \right) + b* \right\}, x \in R^n \quad (3)$$

For nonlinear problems, the solution of SVM is to introduce a kernel trick. Its principle is to map the input vector to a high-dimensional feature vector space and construct the optimal classification surface in the space. The optimal function at this time can be written as:

$$\min f(w) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i \quad (4)$$

C is the penalty factor, ξ_i is a slack variable, and both of them are optimizable hyperparameters. The role of C is to adjust the error for affecting the model's loss due to outliers. It should be pointed out that SVM only involves inner product operations between samples, so there is no need to perform complicated high-dimensional operations in high-dimensional space.

Linear and Gaussian kernel functions were applied in the SVM classifier, and the corresponding classifiers were defined as SVM-linear and SVM-RBF (they were regarded as two different classifiers in the following sections), respectively. After inputting samples and training, the model was optimized. Grid searching that is to operate cross-validation for searching the most optimal parameter was used to adjust the parameters in the training model process. Then the optimal parameters were obtained by setting c = 100 in SVM-linear and set c = 110 and gamma = 1/10 in SVM-RBF.

RF Classifier

Random forest is an algorithm proposed by Breiman [27]. Once born, it received widespread attention and quickly became a hot algorithm in the area of machine learning. It is generated based on the decision tree model, taking each decision tree as a base classifier, and integrating them to form an ensemble learning model. The "random" in the RF algorithm is reflected in the randomness of feature selection and the randomness of training set extraction when constructing the base classifier. After the base classifier was established, the best result is obtained by "voting", so the obtained prediction result is more accurate and stable than the single tree model. The RF algorithm improves the prediction accuracy without significantly increasing the amount of calculation. It has good scalability and parallelism for high-dimensional data classification problems. For the unstable characteristics of traffic data, the RF algorithm is very suitable.

According to the generalization error formula proposed by Breiman [27]:

$$err \leqslant \frac{\rho\left(1 - s^2\right)}{s^2} \quad (5)$$

Where s is the classification strength of a single tree model and ρ is the correlation between trees. It is evident that the higher classification intensity or, the smaller correlation leads to a smaller generalization error bound (i.e., the higher classification accuracy). The above formula provided useful inspiration for us to improve the performance of the random forest classifier.

The optimal performance was obtained by setting the number of trees to 600.

BPNN Classifier

With the development of artificial neural network theory, the BPNN algorithm has an extensive application in the traffic issue. The standard BPNN consists of an input layer, a hidden layer, and an output layer. They are output from the output layer under the activation function processing after inputting the training samples from the input layer. During the training process, the sample's output error will also be output from the output layer. Suppose the error is higher than the set value (pre-set value). In that case, the output error will propagate back towards the input layer, and the error will be apportioned to each neuron so that the weight and threshold of each neuron are adjusted again until the output error is less than the set value. In the process, each neuron's weights and thresholds are continuously adjusted under the stimulation of errors. When the test input value is close to the trained network, the neural network can obtain the corresponding output result. BPNN has good data tolerance and is adept at processing nonlinear data. However, it has a slow convergence, and it is not easy to obtain a globally optimal solution.

The road-user classification is a multi-classification problem, and the solution of the BPNN classifier for the multi-class in our code was to perform a binary class on each category. We applied the grid-search and obtained the optimal number of hidden layers was 50.

PNN Classifier

The core idea of the probabilistic neural network (PNN) algorithm is to estimate the probability distribution function of the overall distribution with the Parzen window estimation method based on the ANN model and Bayesian decision theory [28]. Compared with BPNN, it only has a forward calculation and does not has a feedback process. PNN uses Bayesian discriminant function to minimize the error rate. As long as the number and quality of training samples are guaranteed, it could obtain the optimal classification results based on the Bayesian minimum risk decision. Its advantages include fast training speed, easy classification process, and good fault tolerance. However, its shortcomings are also apparent, such as the accuracy for small-scale data is low, and large-scale data's computation burden is weighty.

There was only one smooth parameter σ that could be adjusted in the whole classification process. The optimal performance of the PNN classifier was obtained by setting the smooth parameter σ to 0.1.

The Evaluation and Optimization of Classification Performance

The Evaluation of Classification Performance

There are many evaluation indicators for classification issues. Accuracy is a common indicator that has a poor performance for unbalanced samples data. The F_1-score indicator is the reconciled average of precision and recalls ratio, which means that the precision ratio and recall ratio are equally important. Its expression is as follows:

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{7}$$

Considering the F_1-score for each sub-category of the classifier, the classifier's predictive ability for the sub-category can be clearly obtained. The F_1-macro is an evaluation index that considers the overall classification performance of all categories. It averages the accuracy and the recall rate of sub-categories, respectively. Then perform F_1-score calculation on these two averages. The formula is as follows:

$$F_{1-macro} = \frac{2\left(p_{average} \times r_{average}\right)}{p_{average} + r_{average}} \tag{8}$$

For multi-classification problems, F_1-macro could evaluate performance very well. It is worth emphasizing that F_1-macro is very suitable for classifiers with an unbalanced sample size, which is easy to know according to its definition. This unbalanced data feature is consistent with the road-user classification database used by the authors, as shown in Figure 3. Based on the above, this article mainly used F_1-macro to make an overall evaluation of the classifier and used F_1-score to evaluate the prediction ability of each sub-category. The Accuracy value was used as an auxiliary evaluation.

Because the sample points size of category i (four or fewer axle/single-trailer truck) was too small (only 2) to divide the training and testing set, the F_1 value of class i was not be analyzed in the later section.

The F_1-macro of each classifier is shown in Figure 4. The results indicate that the road-user classification outcome using SVM and RF algorithms were relatively good, and the SVM-RBF classifier was the best among these five classifiers (F_1-macro = 84%, accuracy = 91.67%). Considering the non-linear characteristics of traffic data and the Gaussian kernel function SVM algorithm trait that was adept at dealing with non-linear problems, the SVM-RBF classifier with better performance results were not difficult to predict. It was also found that the BPNN and PNN classifiers have low F_1-macro

FIGURE 4 The F_1-macro and accuracy of each classifier.

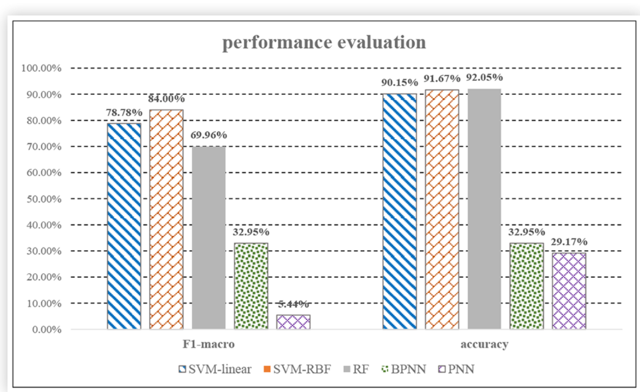


TABLE 2 The F_1-score and F_1-macro of nine classes (except for category i)

		F_1 score				
Labels	**Code**	**SVM-Linear**	**SVM-RBF**	**RF**	**BPNN**	**PNN**
Bicycle	a	57.14%	75.00%	0.00%	0.00%	0.00%
Pedestrians	b	94.51%	95.45%	95.65%	0.00%	0.00%
Motorcycle	c	57.14%	58.82%	88.89%	0.00%	0.00%
Passenger car	d	93.59%	96.15%	98.04%	49.57%	44.97%
Four-tire, single unit	e	85.71%	86.27%	89.52%	0.00%	4.00%
Bus	f	100.00%	100.00%	96.00%	0.00%	0.00%
Two-axle, six-tire, single-unit truck	g	75%	72.00%	66.67%	0.00%	0.00%
Three-axle, single-unit truck	h	50%	75.00%	0.00%	0.00%	0.00%
Five-axle, single-trailer truck	j	95.89%	97.30%	94.87%	0.00%	0.00%
F_1-macro		78.78%	84.00%	69.96%	32.95%	5.44%

values. In Table 2, BPNN and PNN had apparent correct classification behavior only in the passenger car. As shown in Figure 3, the number of samples belong to the passenger car was the largest among all categories. It meant that the low performance of BPNN and PNN classifiers might be caused by the small sample size of the database. This also confirmed that the BPNN and PNN algorithms are not skilled in predicting small sample data. It was worth noting that the performance of the PNN classifier in the classification task was unsatisfactory, and the adjustment of the parameter σ could not improve the performance very well. The high-dimensional (multi-feature) of the sample data may also be a significant factor that interferes with the performance of the BPNN and PNN classifiers, which will be further discussed in the later section.

From Table 2, it can also be found that the classification of bicycle and category h (three-axle, single-unit truck) in the SVM-RBF classifier had a much higher F_1-score value than the other four classifiers. This shows that it is more efficient to use the SVM algorithm for the classification of bicycles and category h. The accuracy of the RF classifier was the highest, reaching 92.05%, but the performance of the F_1 -macro value was not very well (only 69.96%). Although the RF classifier's overall classification capability was not as good as that of the SVM-RBF classifier, it had a better classification effect for category c, d, e (motorcycles; passenger car; four-tire single unit), especially the classification performance of category c was much better than other algorithms.

Among all categories, the classification effects of categories b, d, f, and j (pedestrians and skateboarders; passenger car; bus; five-axle, single-trailer truck) were relatively sound. Category d was a common type of road-users. The original sample data volume was also the largest, so it had a good classification result that is not difficult to understand. Category b, f, and j were easy to distinguish because of their prominent shape characteristics, and they had better F_1-score value than other categories. Category e (four-tire, single unite) had a relatively large sample size, but the classification performance was not excellent (F_1-score = 85.71%). The reason may be that it was mistakenly recognized as category d and category g (passenger car; two-axle, six-tire, single-unit truck) with the F_1-score = 93.59% and 75%, respectively.

The Optimization of Classification Performance

The dependence of the six selected features in this paper was to try to describe the vehicle's side profile as detailed as possible based on the extracted road-user point cloud. Though only six features have been proposed, the actual model input feature dimension was 15, with the existence of n (n = 10) in the height profile. Considering the high dimension may cause poor predictive ability of the classifier, the authors implemented the sample dimensionality reduction to further strengthen the performance. Feature A and F (max-length in the trajectory; object height profile) described the height profile of the target along the moving track and played a vital role in the detailed description of the side profile. Besides, in order to enrich the details of road-user's contour, it was also essential to keep the parameter n = 10. For feature C (nearest distance from target points to LiDAR), the distance of the road-user from the LiDAR affected the depiction of the side profile. The closer the distance was, the side profile of the road-user was clearer. Without this feature, it was likely to cause the classification model to be misidentified due to the difference in actual monitoring distance, so feature C could not be discarded. The original consideration of selecting feature B and E (number of points in the frame with max length; the difference between length and height) was that those types of road-users with much similar side profiles are more likely to be correctly classified, but the actual results need to be further discussed. Feature D (max-height in the trajectory) was already included in the feature F. Based on the above, the features A, C, and F were considered to be retained, and the features B, D, and E were deleted. The sensitivity of the classifier performance for feature selection was studied by reducing the sample data's dimensions.

The dimensionality reduction processing of deleting features BD, BE, DE, B, D, E, BDE, was implemented on the database, respectively, and the classifier was retrained. As shown in Table 3, by comparing the classification results of the five classifiers, no matter which dimensionality reduction method was selected, the SVM-RBF classifier had significantly better classification performance than other classifiers. This also thoroughly verifies the above conclusion that the SVM

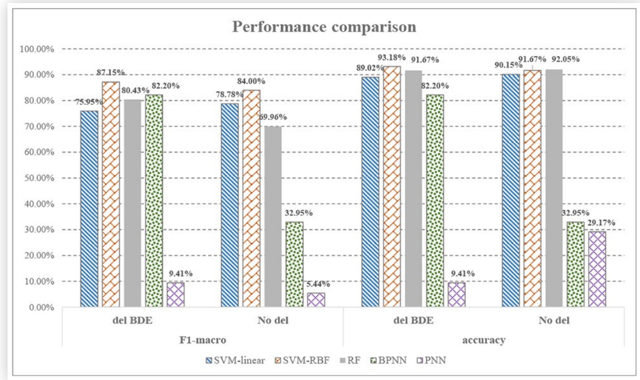
TABLE 3 The F_1-macro value of classifiers after reducing the feature dimensionality

	F_1-macro				
Features	**SVM-linear**	**SVM-RBF**	**RF**	**BPNN**	**PNN**
del BD	79.92%	86.71%	69.41%	85.61%	8.38%
del BE	79.03%	86.88%	76.36%	81.82%	9.32%
del DE	77.15%	84.00%	73.32%	32.20%	5.00%
del B	79.19%	86.51%	68.75%	82.58%	8.64%
del D	80.75%	84.96%	70.63%	32.95%	5.44%
del E	78.57%	83.29%	78.57%	32.95%	5.00%
del BDE	**75.95%**	**87.15%**	**80.43%**	**82.20%**	**9.41%**
del F	54.67%	60.61%	63.76%	27.27%	5.00%
No del	78.78%	84.00%	69.96%	32.95%	5.44%

algorithm using the Gaussian kernel function had the best classification results. Table 3 added the F_1-macro with deleting feature F, and all features were retained (No del) for comparison. The results indicate that deleting feature F had a high impact on each classifier. It can be found that it is imperative to select a height profile as the classification feature.

The F_1-score value of the category prediction under each dimensionality reduction action of the SVM-RBF classifier is shown in Table 4. The results show that the input samples deleting feature BDE had the highest F_1-macro value (F_1-macro=87.15%), that is, the performance of this SVM-RBF classifier reduced feature dimensionalities was the best. As far as feature B was concerned, the F_1-macro values of the four dimensionality reduction methods (del BD, BE, B, BDE) removing feature B were larger than retaining all features. That is, the classification was more accurate than before. This indicates that feature B was not beneficial for correct classification. Similarly, for feature D, by the dimensionality reduction way (del BD, DE, D, BDE), the classifier performance was better, so it was necessary to delete feature D. There was a particular case about the dimension reduction of feature E. Deleting feature E alone did not improve the F_1-macro but deleting it with other features together could improve performance. It can be found from Table 4 that with the dimensionality reduction method of deleting E, the F_1-macro values of categories a, b, and c were the smallest compared with other

FIGURE 5 The comparison of classification performance (F_1-macore and accuracy) by deleting the feature BDE and retaining all features.



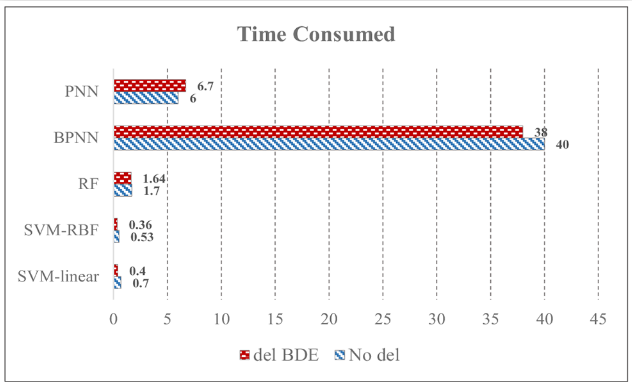
dimensionality reduction methods. This was also the reason why the F_1-macro value of only deleting E was the smallest. This phenomenon indicates that feature E can help class a, b, and e be correctly classified. In contrast, deleting feature BDE could get better classification results. It was found that the classification idea of portraying side profile can already meet the prediction accuracy of category a, b, and c. From the overall view to analyze Table 4, the classification prediction of each sub-category made by the sample dimensionality reduction method of deleting BDE maintained its F_1-score value at a very high level, and there was no other situation where the performance of one dimension reduction method was much better than it. This analysis results inspire us to use feature ACF (max-length; nearest distance from the object; height profile) to achieve better classification performance while classifying road-users.

Figure 5 shows the comparison of classification performance by removing feature BDE and maintaining all features. It was found that the F_1-macro value of other classifiers except SVM-linear improved significantly after deleting the feature BDE. Dimensionality reduction by removing the feature BDE had a significant effect on improving the performance of the classifier. The reason why the SVM-linear classifier's F_1-macro slightly became small may be the incompatibility of the linear kernel function SVM algorithm for the input samples. The

TABLE 4 The F_1-macro and F_1-score of subcategories after reducing the feature dimensionality

	F_1 score							
Code	**del BD**	**del BE**	**del DE**	**del B**	**del D**	**del E**	**del BDE**	**No del**
a	85.71%	85.71%	75.00%	85.71%	75.00%	66.67%	85.71%	75.00%
b	96.63%	96.63%	95.45%	96.63%	95.45%	94.25%	96.63%	95.45%
c	66.67%	70.59%	58.82%	66.67%	58.82%	55.56%	66.67%	58.82%
d	96.77%	96.15%	96.15%	96.77%	96.77%	97.40%	96.77%	96.15%
e	86.87%	87.13%	86.27%	87.13%	88.24%	88.46%	88%	86.27%
f	100.00%	100.00%	100.00%	100.00%	98.04%	100.00%	100%	100.00%
g	74%	72.00%	72.00%	72.00%	80.00%	75.00%	76.92%	72.00%
h	75%	75.00%	75.00%	75.00%	75.00%	75.00%	75%	75.00%
j	98.67%	98.67%	97.30%	98.67%	97.30%	97.30%	98.67%	97.30%
F_1-macro	86.71%	86.88%	84.00%	86.51%	84.96%	83.29%	**87.15%**	84.00%

FIGURE 6 Calculation time consumption (retain all features and delete the feature BDE) by each classifier.



accuracy value of SVM-RBF also improved slightly after reducing the features. The SVM-RBF classifier was still the performance winner among the five vehicle classifiers. The RF classifier also had a clear trend of performance enhancement throughout the dimensionality reduction process. This shows that the RF classifier was relatively sensitive to the reduction of dimensions. It can be surprisingly found from Figure 5 that the BPNN classifier had a qualitative improvement in the classification ability (F_1-macro value reach 82.2%) after removing feature BDE and even surpassed the RF classifier.

The calculation time consumption of each classifier is shown in Figure 6. The SVM-RBF classifier had the most rapid calculation process (0.53s consumed) among five classifiers. Furthermore, the calculation time became shorter than before, with the feature BDE being deleted (0.36s consumed). The BPNN classifier consumed more than 40s to complete the calculation process.

Conclusion

This paper presented a rapid and accurate method using roadside LiDAR data to classify the road-users into different groups. After extracting the road-user point cloud from the raw data, four machine learning algorithms, including SVM, RF, BPNN, PNN, were applied for model training. Then the parameters were adjusted to obtain the optimal classification results. Finally, the linear kernel and Gaussian kernel function SVM classifiers were regarded as two distinct classifiers, together with other classifiers, for comparison and discussion. Given the characteristics of multi-classification and unbalanced sample size, F_1-score and F_1-macro were mainly used to evaluate the classification performance.

Six features were initially selected to build the classifier. The results show that the SVM-RBF classifier had the best ability to classify road-users. A comparison of the classifier's performance and detailed classification evaluation of each sub-category was given. To further improve the classifier's performance, the authors proposed a way to reduce the number of selected features. The SVM-RBF classifier achieved the best performance among the five methods. It was found that the performance of the classifier built by retaining the features ACF has been significantly improved. The RF classifier can identify motorcycles better than the other methods. For the rest types, especially bicycles and large-size vehicles, the SVM-RBF classifier had an outstanding performance. Feature F (height profile) played an essential role in maintaining the vehicle classifier's high performance. This result had proper enlightenment for the feature selection in the future road-user classification task. BPNN has made significant progress in the classification work after dimensionality reduction, and the results show that it was sensitive to the sample size.

References

1. Xiao, W., Vallet, B., Schindler, K., and Paparoditis, N., "Street-Side Vehicle Detection, Classification and Change Detection Using Mobile Laser Scanning Data," *ISPRS Journal of Photogrammetry* 114:166-178, 2016.
2. Lee, H. and Coifman, B., "Side-Fire Lidar-Based Vehicle Classification," *Transportation Research Record* 2308(1):173-183, 2012.
3. Meta, S. and Cinsdikici, M.G., "Vehicle-Classification Algorithm Based on Component Analysis for Single-Loop Inductive Detector," *IEEE Transactions on Vehicular Technology* 59(6):2795-2805, 2010, https://doi.org/10.1109/TVT.2010.2049756.
4. Obertov, D. and Andrievsky, B., "Vehicle Classification Using Measurements from Accelerometers Mounted on the Road Surface," in *International Conference on Methods and Models in Automation and Robotics*, 413-417, 2014.
5. Fuerstenberg, K. and Willhoeft, V., "Object Tracking and Classification Using Laserscanners-Pedestrian Recognition in Urban Environment," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings* (Cat. No. 01TH8585), 451-453, 2001, https://doi.org/10.1109/ITSC.2001.948699.
6. Odat, E., Shamma, J.S., and Claudel, C., "Vehicle Classification and Speed Estimation Using Combined Passive Infrared/Ultrasonic Sensors," *IEEE Transactions on Intelligent Transportation Systems* 19(5):1593-1606, 2017, https://doi.org/10.1109/TITS.2017.2727224.
7. Chen, Z., Ellis, T., and Velastin, S.A., "Vehicle Detection, Tracking and Classification in Urban Traffic," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 951-956, 2012, https://doi.org/10.1109/ITSC.2012.6338852.
8. Mithun, N.C., Ur Rashid, N., and Mahbubur Rahman, S.M., "Detection and Classification of Vehicles from Video Using Multiple Time-Spatial Images," *IEEE Transactions on Intelligent Transportation Systems* 13(3):1215-1225, 2012, https://doi.org/10.1109/TITS.2012.2186128.
9. Dong, Z., Wu, Y., Pei, M., and Jia, Y., "Vehicle Type Classification Using a Semisupervised Convolutional Neural Network," *IEEE Transactions on Intelligent Transportation Systems* 16(4):2247-2256, 2015, https://doi.org/10.1109/TITS.2015.2402438.
10. Hasnat, A., Shvai, N., Meicler, A., Maarek, P. et al., "New Vehicle Classification Method Based on Hybrid Classifiers,"

in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 3084-3088, 2018, https://doi.org/10.1109/ICIP.2018.8451642.

11. Chang, J., Wang, L., Meng, G., Xiang, S. et al., "Vision-Based Occlusion Handling and Vehicle Classification for Traffic Surveillance Systems," *IEEE Intelligent Transportation Systems Magazine* 10(2):80-92, 2018, https://doi.org/10.1109/MITS.2018.2806619.
12. Wu, J., Xu, H., Zheng, Y., Zhang, Y. et al., "Automatic Vehicle Classification Using Roadside LiDAR Data," *Transportation Research Record: Journal of the Transportation Research Board* 2673(6):153-164, 2019, https://doi.org/10.1177/0361198119843857.
13. Gao, H.B., Cheng, B., Wang, J.Q., Li, K.Q. et al., "Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment," *IEEE Transactions on Industrial Informatics* 14(9):4224-4231, 2018, https://doi.org/10.1109/tii.2018.2822828.
14. Zhang, M., Fu, R., Guo, Y., and Wang, L., "Moving Object Classification Using 3D Point Cloud in Urban Traffic Environment," *Journal of Advanced Transportation*1-12, 2020.
15. Yoshioka, M., Suganuma, N., Yoneda, K., and Aldibaja, M., "Real-Time Object Classification for Autonomous Vehicle Using LIDAR," in *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 210-211, 2017.
16. Kurnianggoro, L. and Jo, K.-H., "Object Classification for LIDAR Data Using Encoded Features," in *2017 10th International Conference on Human System Interactions (HSI)*, 49-53, 2017.
17. Nezafat, R.V., Sahin, O., and Cetin, M., "Transfer Learning Using Deep Neural Networks for Classification of Truck Body Types Based on Side-Fire Lidar Data," *Journal of Big Data Analytics in Transportation* 1(1):71-82, 2019.
18. Lee, H. and Coifman, B., "Using LIDAR to Validate the Performance of Vehicle Classification Stations," *Journal of Intelligent Transportation Systems* 19(4):355-369, 2015, https://doi.org/10.1080/15472450.2014.941750.
19. Sahin, O., Nezafat, R.V., and Cetin, M., "Methods for Classification of Truck Trailers Using Side-Fire Light Detection and Ranging (LiDAR) Data," *Journal of Intelligent Transportation Systems*1-13, 2020, https://doi.org/10.1080/15472450.2020.1733999.
20. Asborno, M.I., Burris, C.G., and Hernandez, S., "Truck Body-Type Classification Using Single-Beam Lidar Sensors," *Transportation Research Record* 2673(1):26-40, 2019.
21. Wu, J., Xu, H., and Zhao, J., "Automatic Lane Identification Using the Roadside LiDAR Sensors," *IEEE Intelligent Transportation Systems Magazine* 12(1), 2020, https://doi.org/10.1109/MITS.2018.2876559.
22. Wu, J., Xu, H., Sun, Y., Zheng, J. et al., "Automatic Background Filtering Method for Roadside LiDAR Data," *Transportation Research Record* 2672(45):106-114, 2018, https://doi.org/10.1177/0361198118775841.
23. Wu, J., "An Automatic Procedure for Vehicle Tracking with a Roadside LiDAR Sensor," *Institute of Transportation Engineers. ITE Journal* 88(11):32-37, 2018.
24. Wu, J., Xu, H., and Zheng, J., "Automatic Background Filtering and Lane Identification with Roadside LiDAR Data," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 1-6, 2017, https://doi.org/10.1109/ITSC.2017.8317723.
25. Zhao, J., Xu, H., Liu, H., Wu, J. et al., "Detection and Tracking of Pedestrians and Vehicles Using Roadside LiDAR Sensors," *Transportation Research Part C-Emerging Technologies* 100:68-87, 2019.
26. Won, M., "Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey," *IEEE Access* 8:73340-73358, 2020.
27. Breiman, L., "Random Forests," *Mach. Learn.* 45(1):5-32, 2001, https://doi.org/10.1023/a:1010933404324.
28. Specht, D.F., "Probabilistic Neural Networks," *Neural Networks* 3(1):109-118, 1990, https://doi.org/10.1016/0893-6080(90)90049-Q.

This paper is based upon a presentation at the *3rd International Forum on Connected Automated Vehicle Highway System through the China Highway & Transportation Society.*

ISSN 0148-7191