Performance Analysis of Object Classification System for Traffic Objects Using Various SVM Kernels

Madhura M. Bhosale, Tanuja Satish Dhope (Shendkar), Akshay P. Velapure, and Dina Simunic

Abstract In the area of autonomous vehicle, object classification is an important task. Different classification algorithms are available. In this work, we have focused on different kernels of support vector machine (SVM) to analyze the performance of traffic object classification system. We have performed experimentation on open-source database had been carried out on the basis of performance metrics such as recall, precision, F1, and accuracy. Our classification system provides maximum accuracy 69% with the help of RBF kernel.

Keywords Linear SVM · Machine learning · Classifier

1 Introduction

Object classification plays important role in the field of autonomous vehicle. Different classification algorithms are present in market such as SVM, KNN, random forest, and decision tree. In this paper, we have focused on performance of different kernels of SVM for the classification system of traffic objects. Non-parametric supervised learning model is a support vector machine (SVM) [1]. Figure 1 shows the decision function for a linearly separable problem, with three samples on the margin boundaries, called 'support vectors'. SVM is used for outlier's detection, regression, and

M. M. Bhosale (✉)
Department of Electronics and Telecommunication Engineering, JSPM's Rajarshi Shahu College of Engineering, Pune 411033, India

Tanuja Satish Dhope (Shendkar)
Professor, Department of Electronics and Communication, College of Engineering, Bharati Vidyapeeth (Deemed to be University), Dhankawadi, Pune 411043, Maharashtra, India
e-mail: tsdhope@bvucoep.edu.in

A. P. Velapure
Pune, India

D. Simunic
Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia
e-mail: Dina.Simunic@fer.hr

P. Verma et al. (eds.), *Advances in Data Computing, Communication and Security*, Lecture Notes on Data Engineering and Communications Technologies 106,
https://doi.org/10.1007/978-981-16-8403-6-66

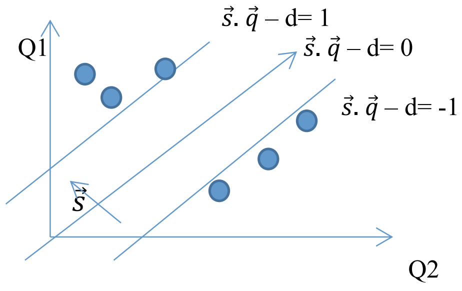


Fig. 1 Graph of SVM

classification. Multiple categorical data are handled with SVM. Separation of two classes based on feature set is the main aim of SVM. As graph shown in Figure 1 have three lines, one is marginal $\vec{s}.\vec{q} - d = 0$, and other two lines $\vec{s}.\vec{q} - d = 1$ and $\vec{s}.\vec{q} - d = -1$ indicate the position of closest data points of both the classes.

2 Literature Review

There is a huge research has been going on object classification as specified in [2–5]. This paper investigates the application of support vector machines (SVMs) in texture classification [6]. In [7], it gives you idea about multiclass classification using SVM. In [8], it gives idea about non-linear kernel of the SVM and provides you survey about various SVM concept and some real-time applications using SVM. In [9], paper reviews various concepts of random forest and SVM for remote sensing image classification and provides you comparative analysis using different parameters. In [10], SVM with k-fold algorithm is used to predict and evaluate degradation of concrete strength in a complicated marine environment. After some experimentation, average relative error is reduced from 34.8 to 27.6%, and median-relative error declines from 24.7 to 20.8%. In [11], it introduces least square support vector machine for regression into reliability analysis to overcome the shortcomings present in support vector regression. In [10], the influence of vehicle, traffic, people, and traffic management on driverless vehicle and constructs a highway safety evaluation model based on support vector machine. In [12], this paper presents performance of SVM for recognizing road images. Images divided into four classes turn left, right, forward, stop. Accuracy of this model is 70.77%. In [13–19], it compares independent component analysis, Fourier transform principal component analysis, and independent component analysis for data preprocessing with the help of machine learning method of SVM.

3 Methodology

The proposed methodology, overall algorithm has been splinted into two parts; one is training, and other is testing.

1. Model Training: Overall training model has been splinted into two steps. As shown in Figure 2, a. first step is preprocessing and feature extraction of images [20]. First, we resize the image and convert RGB to gray. We have used histogram of gradient HOG [20] features. The feature set was used to train machine learning models using different kernels of SVM.

A **SVM Parameters**:

(i) **C Parameter**: This parameter is regularization parameter. The C parameter is trade off correct classification of training examples against maximization of decision function margin. Classifying all training data points correctly there need to be smaller margin accepted which is only possible for the larger values of C [21].

(ii) **Gamma Parameter**: the gamma parameter defines how far the influence of a single training example approaches, high values meaning 'close' and with low values meaning 'far'.

(iii) **Kernel**: Selection of the kernel is totally depends on whether your data points distribution. Figure 3 gives you clear idea about types of the kernels.

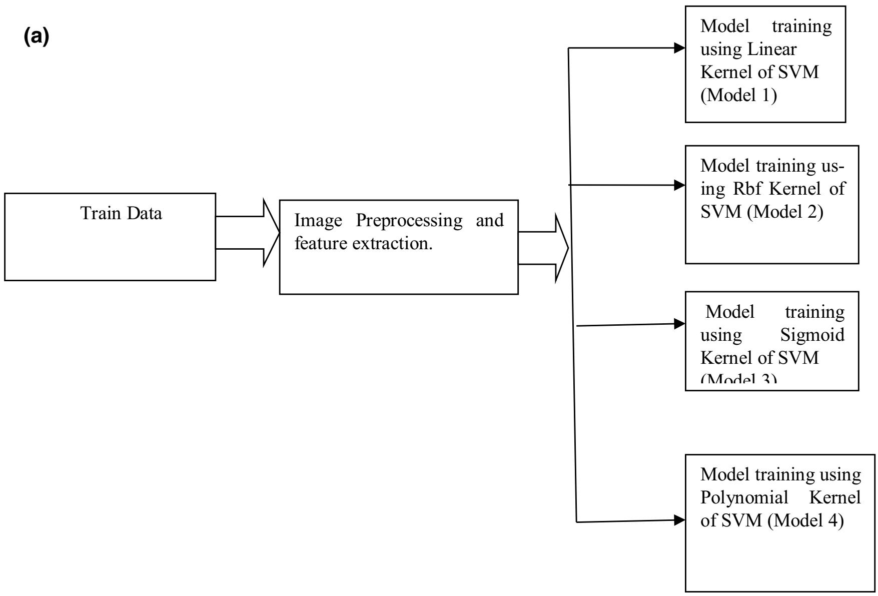


Fig. 2 **a** Block diagram of training of machine learning model using different SVM. **b** Block diagram of testing system kernels

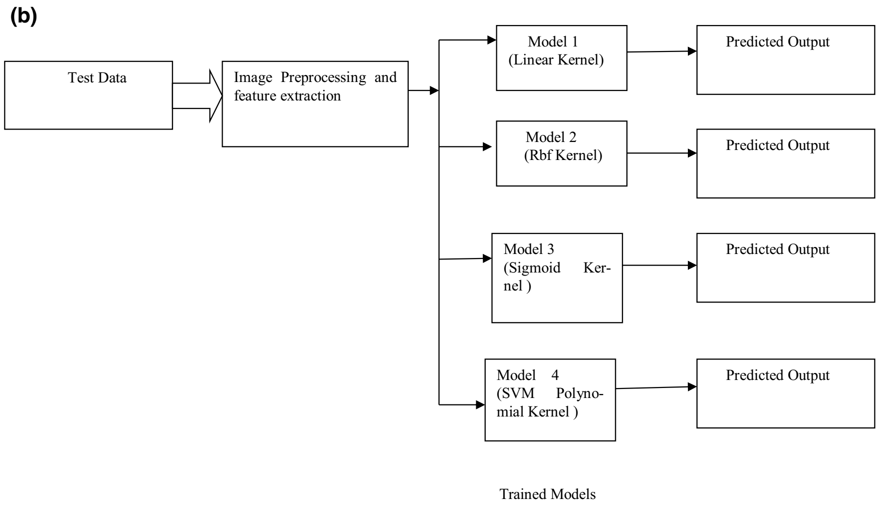


Fig. 2 (continued)

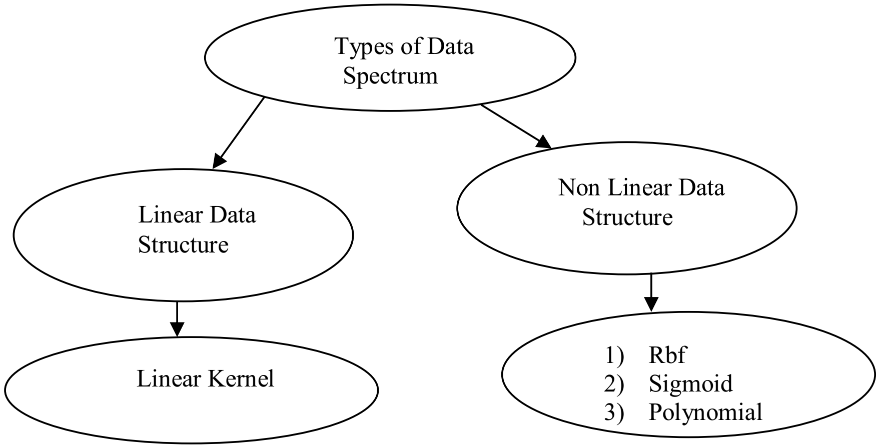


Fig. 3 Types of kernels

(a) **Linear kernel**: In linear classifier as shown in Fig. 1, n dimensional points are separated with $(n-1)$ dimensional hyperplane [22]. The best hyperplane is one which separate two classes with maximum marginal value. Suppose we have n number of points $\left(\overrightarrow{q1}, y1\right) \ldots \left(\overrightarrow{qn}, yn\right)$. Here, $y(j)$ is 1 or -1 representing the class of point $q(j)$. Now, we are interested in exploring maximum margin hyperplane separating class of point $q(j)$ having $y(j) = 1$ from class of $y(j) = -1$. Distance between hyperplanes and nearest $q(j)$ point should be maximum. Equation of the hyperplane is as follows

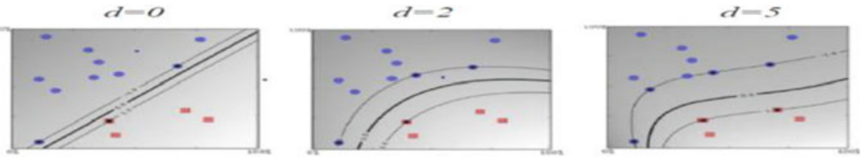


Fig. 4 Response of the kernel functions on different degrees [8]

$$\vec{s}.\vec{q} - d = 0 \tag{1}$$

where $\vec{s}$ = normal vector to the hyperplane, $d/\overrightarrow{\|q\|}$ = offset of hyperplanes from origin along $\vec{s}$.

(b) **Polynomial kernel**: SVM represents the similarity of the training samples in the feature space over polynomial of the original variables [23]. For the degree d, polynomial function is represented by following equation (Fig. 4):

$$K(x, y) - (xy + C) \tag{2}$$

(c) **Radial Basis Function**: This function is used Euclidean distance**.** In SVM, radial basis function is used to define the Gaussian radial basis function [22, 24]. Euclidean distance is calculated by following equation.

$$\Phi(x, c) = \Phi(\|x - c\|) \tag{3}$$

In radial basis function, there is free parameter σ present which help to calculate Euclidean distance between two landmarks. Figure 5 shows effect of σ on Rbf kernel.

This is most popular kernel method which is given by formula.

$$K\left(x, x'\right) = \exp\left(-\gamma \left\|x - x'\right\|2\right) \tag{4}$$

where γ given by,

$$\gamma = 1/2\sigma^\wedge 2 \tag{5}$$

(d) **Sigmoid kernel**: When the numbers of features are too large and non-linear, then sigmoid kernel is used. This kernel is basically inspired by neural network.

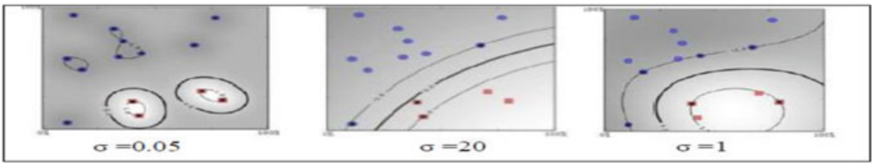


Fig. 5 Variation in the Gaussian RBF kernel with variation in σ [8]

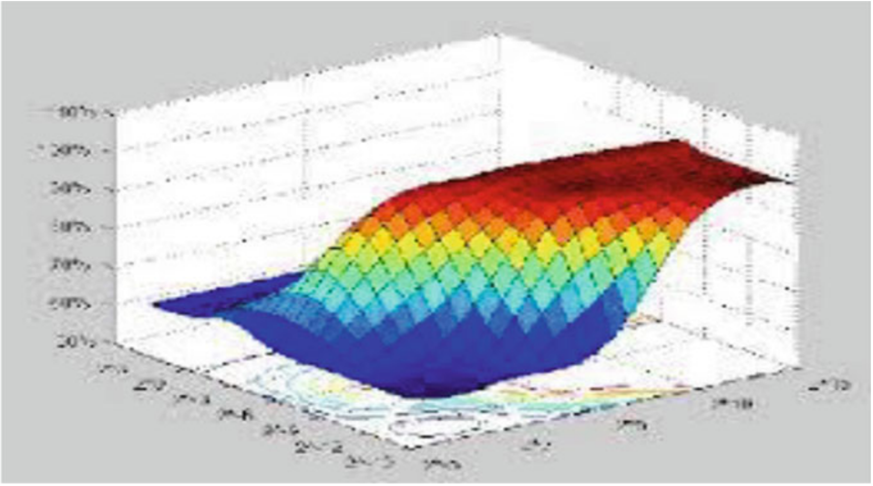


Fig. 6 Sigmoid function plot using C-SVM [8]

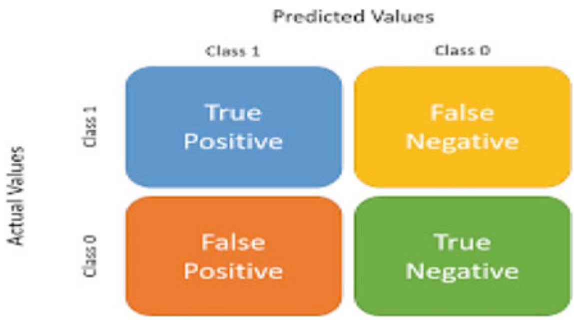
Sigmoid kernel function is given by (Fig. 6),

$$K(x, y) = \tan h(\propto xT\ \ y + c) \tag{6}$$

2. Second part of algorithm is testing. In this, we do same operation on testing data which we already performed on training data. And provide that data as an input to trained model and predict the output as shown in Fig. 2b.

B **Confusion matrix**: The confusion matrix is used to extract more information about model performance. The confusion matrix helps us visualize whether the model is "confused" in discriminating between the classes. The labels of the two rows and columns are *negative* and *positive* to reflect the two class labels as shown in Fig. 7.

Fig. 7 Confusion matrix [19]



4 Results

For our work, we used open-source database acquired from [25]. The image database consists of two classes of objects truck and car. We implemented this work using Python 3.8 programming language. We extracted total 16,384 HOG features from 782 images. First, we split dataset into two parts, 70% kept for training and 30% used for testing.

Figures 8, 9, 10, and 11 show classification results obtained from various models using linear, polynomial, RBF, and sigmoid kernels of SVM, respectively. Machine learning model using SVM linear kernel true positive, false negative, false positive,

```
[Status] Loaded features of shape (782, 16384)
[Status] Loaded labels of shape (782,)

Enter 1 for linear kernel
Enter 2 for polinomial kernel
Enter 3 for RBF kernel
Enter 4 for Sigmoid kernel
Enter your choice: 1

You have selected SVM Linear kernel

Confusion matrix:
 [[75 51]
 [44 65]]

Classification report
              precision    recall  f1-score   support

         car       0.63      0.60      0.61       126
       truck       0.56      0.60      0.58       109

    accuracy                           0.60       235
   macro avg       0.60      0.60      0.60       235
weighted avg       0.60      0.60      0.60       235
```

Fig. 8 Results of linear kernel of SVM

```
[Status] Loaded features of shape (782, 16384)
[Status] Loaded labels of shape (782,)

Enter 1 for linear kernel
Enter 2 for polinomial kernel
Enter 3 for RBF kernel
Enter 4 for Sigmoid kernel
Enter your choice: 2

You have selected SVM Polynomial kernel

Confusion matrix:
 [[94 32]
 [46 63]]

Classification report
              precision    recall  f1-score   support

         car       0.67      0.75      0.71       126
       truck       0.66      0.58      0.62       109

    accuracy                           0.67       235
   macro avg       0.67      0.66      0.66       235
weighted avg       0.67      0.67      0.67       235
```

Fig. 9 Results of polynomial kernel of SVM

```
Enter 1 for linear kernel
Enter 2 for polinomial kernel
Enter 3 for RBF kernel
Enter 4 for Sigmoid kernel
Enter your choice: 3

You have selected SVM RBF kernel

Confusion matrix:
 [[81 45]
 [27 82]]

Classification report
              precision    recall  f1-score   support

         car       0.75      0.64      0.69       126
       truck       0.65      0.75      0.69       109

    accuracy                           0.69       235
   macro avg       0.70      0.70      0.69       235
weighted avg       0.70      0.69      0.69       235
```

Fig. 10 Results of RBF kernel of SVM

```
[Status] Loaded features of shape (782, 16384)
[Status] Loaded labels of shape (782,)
Enter 1 for linear kernel
Enter 2 for polinomial kernel
Enter 3 for RBF kernel
Enter 4 for Sigmoid kernel
Enter your choice: 4
You have selected SVM Sigmoid kernel
Confusion matrix:
 [[68 58]
 [52 57]]
Classification report
              precision    recall  f1-score   support

         car       0.57      0.54      0.55       126
       truck       0.50      0.52      0.51       109

    accuracy                           0.53       235
   macro avg       0.53      0.53      0.53       235
weighted avg       0.53      0.53      0.53       235
```

Fig. 11 Results of sigmoid kernel of SVM

and true negatives values is 75, 51, 44, and 65, respectively. Precision values are 63% for car and 56% for truck. Recall values are 60% for car and 60% for truck. $F1$-score is 61% for car and 58% for truck, and overall accuracy of system using linear SVM kernel is 60%.

Machine learning model using polynomial kernel true positive, false negative, false positive, and true negatives values is 94, 32, 46, and 63, respectively. Precision values are 67% for car and 66% for truck. Recall values are 75% for car and 58% for truck. $F1$-score is 71% for car and 62% for truck, and overall accuracy of system using linear SVM kernel is 67%.

Machine learning model using RBF kernel true positive, false negative, false positive, and true negatives values is 81, 45, 27, and 82, respectively.

Precision values are 75% for car and 65% for truck. Recall values are 64% for car and 75% for truck. $F1$-score is 69% for car and 69% for truck, and overall accuracy of system using linear SVM kernel is 69%.

Machine learning model using sigmoid kernel true positive, false negative, false positive, and true negatives values is 68, 58, 52, and 57, respectively. Precision values are 57% for car and 50% for truck. Recall values are 54% for car and 52% for truck. $F1$-score is 55% for car and 51% for truck, and overall accuracy of system using linear SVM kernel is 60%.

5 Conclusion

In this paper, we have discussed various object classification papers in the literature review. We have presented comparative study of different kernels of SVM. From Table 1, we have observed that for RBF kernel precision, recall, and F1 values for car are 75%, 64%, 69%, and for truck are 64%, 75%, 69%. The precision values mean true positive values and recall which means ratio of true positive and false negative is high with RBF kernel. From result, we have concluded that SVM classifier with RBF kernel provides best accuracy for this set of parameters.

Table 1 Comparison between different kernels of SVM

Kernel	Confusion matrix parameters	Car (%)	Truck (%)	Accuracy (%)
Linear SVM	Precision Recall F1	63 60 61	56 60 58	60
Polynomial	Precision Recall F1	67 75 71	66 58 62	67
RBF	Precision Recall F1	75 64 69	64 75 69	69
Sigmoid	Precision Recall F1	57 54 55	50 52 51	53

6 Future Scope

In future scope, focus needs to be given on other parameters of SVM such as C, gamma. Also, different classifiers can be utilizing for classifying multiple classes. Based on our current results and future experimentations, results will be helpful in the area of autonomous vehicle.

References

1. V. Vapnik, *Statistical learning theory* (Wiley-Interscience Publication, New York, 1998)
2. C. Papageorgiou, T. Poggio, A trainable system for object detection. IJCV **38**(1), 15–33 (2000)
3. P. Viola, M.J. Jones, D. Snow. Detecting pedestrians using patterns of motion and appearance, in *The 9th ICCV, Nice, France*, vol. 1 (2003), pp. 734–741
4. H. Schneiderman, T. Kanade, Object detection using the statistics of parts. IJCV **56**(3), 151–177 (2004)
5. H. Ling, C. Qian, W. Kang, C. Liang, H. Chen, Combination of support vector machine and K-fold cross validation to predict compressive strength of concrete in marine environment. Constr. Build. Mater. **206**, 355–363 (2019)
6. K.I. Kim, K. Jung, S.H. Park, H.J. Kim, Support vector machine for texture classification, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, No. 11 (2002)
7. C.-W. Hsu, C.-J. Lin, A comparison on methods for multi-class support vector machines, in *Technical report, Department of Computer Science and Information Engineering, Nat'l Taiwan University* (2001)
8. S. Ghosh, A. Dasgupta, A. Swetapadma, A study of support vector machine based on linear and non-linear pattern classification, in *International Conference on Intelligent Sustainable Systems (ICISS 2019)*
9. M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, S. Homayouni, Support vector machine vs. random forest for remote sensing image classification: a meta analysis and systematic review, in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2020)
10. W. Gang, Safety evaluation model for driverless car using support vector machine. J. Intell. Fuzzy Syst. **37**, 433–440 (2019)

11. Z. Guo, L. Zhang, D. Zhang, Rotation invariant texture classification using LBP variance (LBPV) with global matching. Pattern Recogn. **43**(3), 706–719 (2010)
12. I. Thammachantuek, S. Kosolsombat, M. Ketcham, Support vector machine for road image recognition in autonomous car, in *18th International Symposium on Communications and Information Technologies (ISCIT 2018)*
13. H. Qian, Y. Ou, X. Wu, X. Meng, Y. Xu, Support vector machine for behaviour based driver identification system. *Hindawi Publishing Corporation, Journal of Robotics* (2010)
14. S. Gupta, M. Sameer, N. Mohan, Detection of epileptic seizures using convolutional neural network. Int. Conf. Emerg. Smart Comput. Inform. (ESCI) **2021**, 786–790 (2021)
15. A. Mahajan, K. Somaraj, M. Sameer, Adopting artificial intelligence powered ConvNet to detect epileptic seizures. IEEE-EMBS Conf. Biomed. Eng. Sci. (IECBES) **2021**, 427–432 (2020)
16. S.M. Beeraka, A. Kumar, M. Sameer et al., Accuracy enhancement of epileptic seizure detection: a deep learning approach with hardware realization of STFT. Circuits Syst. Signal Process. (2021)
17. M. Sameer, A.K. Gupta, C. Chakraborty, B. Gupta, "Epileptical seizure detection: performance analysis of gamma band in EEG signal using short-time Fourier transform, in *2019 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC)* (2019), pp. 1–6
18. M. Sameer, B. Gupta, Beta band as a biomarker for classification between interictal and ictal states of epileptical patients, in *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)* (2020), pp. 567–570
19. https://ailearnerhub.com/2020/05/10/what-is-the-confusion-matrix/
20. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 05 (2005), pp. 1063–6919
21. H. Jiang, K. Huang, R. Zhang, Field support vector regression, in *Proceedings of the International Conference on Neural Information Process* (2017)
22. W. Shang, K. Sohn, D. Almeida, H. Lee, Understanding and improving convolutional neural networks via concatenated rectified linear units, in *Proceedings of 33rd International Conference on Machine Learning* (2016), pp. 2217–2225
23. Y. Chang, C. Hsieh, K. Chang, M. Ringgaard, C. Lin, Training and testing low-degree polynomial data mappings via linear SVM. J. Mach. Learn. Res. **11**, 1471–1490 (2010)
24. K. Huang, H. Yang, I. King, M.R. Lyu, Maximin margin machine: learning large margin classifiers locally and globally, in *IEEE Transaction on Neural Network*, vol. 19, no. 2 (2008), pp. 260–272
25. https://www.kaggle.com/enesumcu/car-and-truck
26. D.K. Agarwal, R. Kumar, Spam filtering using SVM with different kernel functions. Int. J. Comput. Appl. **136**(5) (2016)