International Journal of Future Generation Communication and Networking
Vol. 13, No. 4, (2020), pp. 86 –98

# A Novel Methodology to Recognize Drivers' Distraction using Support Vector Machine Classifier

J. Mary Dallfin Bruxella[1], Dr. J.K.Kanimozhi[2]

[1]*Assistant Professor, Department of Computer Science, K.S.Rangasamy College of Arts and Science, Tiruchengode*

[2] *Assistant Professor, Research and PG Department of Computer Science, Sengunthar Arts and Science College, Tiruchengode*

***Abstract***

*Monitoring driver fatigue, inattention, and lack of sleep is very important in preventing motor vehicles accidents. An efficient system for automatic driver vigilance should make use of physiological, behavioral and car measurements. The driver distraction system is often performed by supervised classifiers, which require an adequate amount of labeled instances to train the classifier. All of these classifiers depend upon the quality and quantity of the training set used to train the classifier, whose reliability is a fundamental issue for an accurate mapping of the investigated area. Support Vector Machines (SVMs) are one of the successful classifier applied for driver inattention detection applications. SVMs are nonparametric statistical approaches for addressing supervised classification and regression problems. Therefore, there is no assumption made on the underlying data distribution. In the original formulation of SVMs, the method is presented with a set of data samples, and the SVM training algorithm aims to determine a hyperplane that linearly divides the data set into two classes. The term optimal separating hyperplane is used to refer to the decision boundary that minimizes misclassification attained during the training phase. Learning refers to finding an optimal decision boundary to separate the training patterns and then to separate test data under the same configuration. The crucial part for any kernel-based technique, including SVMs, is the proper definition of a kernel function that accurately reflects the similarity among samples. This research work addresses two problems of SVM, one is the kernel function selection and the other is the training time. A convex-hull and geometry based SVMs is proposed here for driver distraction detection. The proposed Convex-hull & Geometry based SVM (CG-SVM) doesn't require a kernel function and a training time too. In this way, the complexity of SVM is much reduced while preserving the driver distraction detection accuracy. The performance of the proposed CG-SVM is studied with the Ford challenge driver distraction database received from 2011 International Joint Conference on Neural Networks.*

***Keywords***
*Drivers' Fatigue, SVM, CG-SVM, Hyperplane, Kernel function*

## 1. INTRODUCTION

Driver distraction is one of the major risk factor of motor accidents. Recent statistics show that 16% of the accidents were happen due to driver inattention or distraction. Nowadays the usage of in-vehicle information systems such as navigation systems, smart phones, audio systems and other smart devices are increasing and they distract the drivers and cause the problem of inattention. A study from NHTSA shown that the drivers are distracted by these information devices approximately for about 10 seconds, and nine out of ten drivers are using mobile phones while driving. Though the information devices are helpful to the drivers, it is also difficult for drivers to avoid distraction and direct an adequate level of attention to the road. One simple solution is to develop intelligent in-vehicle systems, which would assist the drivers as well as produce context based feedback to avoid distraction (Lee, 2009; Toledo et al., 2008). For example, in a heavy traffic situation, the intelligent system could hold the mobile calls, until the driver can get off the road. Development of such system must monitor the driver behavior to



ISSN: 2233-7857 IJFGCN

identify the safe driving behavior, and then assist them accurately and non-intrusively (Lee et al., 2008). Distraction can be defined as a deviation of a driver's attention away from the actions precarious for safe driving toward a competing activity. The distraction could be either visual distraction or cognitive distraction, represented as "eye-off-road" and "mind-off-road" (Liang & Lee, 2010; Victor et al., 2015). A typical algorithm that monitors driver's glance behavior to detect the visual distraction, whereas driver's expression and behavior could be studied to identify the cognitive distraction (Liang et al., 2012).

A drivers' behavior can be measured by a set of psychological, physiological and physical parameters. Various algorithms has been developed to detect driver inattention, most of them are classification approaches, where the driver's state is identified as drowsy or not, distracted or not. However, this problem could also be viewed as a regression problem, where the driver's state could be measured with various continuous behaviors. Moreover, the previous methods make use of single source of driver's state to detect their inattention. In this paper, the combination of different sources: physiological, behavioral and psychological data are received from driver's state, in addition to that the vehicle information also considered for the driver inattention classification. In previous studies, two data mining methods—Support Vector Machines (SVMs) and Dynamic Bayesian Networks (DBNs)—successfully detected cognitive distraction from driver visual behavior and driving performance (Liang et al., 2007a, b). Since the driver's data would be over-fitting and noisy, this paper proposed a machine-learning method based on Support Vector Machine (SVM) for driver distraction recognition.

The rest of the paper is organized as follows: the following section presents the related work on SVM and driver inattention detection. Section 3 describes the convention SVM classifier. Section 4 illustrates the proposed CG-SVM classifier. Section 5 presents the experimental setup and discusses the results. Section 6 concludes the paper.

## 2. RELATED WORKS

This section presents a study on SVM and a brief literature on driver inattention detection algorithms.

### 2.1 Study on Support Vector Machine (SVM)

There are numerous SVM variants have been proposed in the literature to tailor SVM fit for higher dimensional data. These methods can be divided into four types: (a) reducing training data sets (data selection), (b) using geometric properties of SVM, (c) modifying SVM classifiers, (d) decomposition, and (e) the other methods. The data selection method chooses the objects which are possible to be the Support Vectors (SV) (Li, 2011). These data are used for SVM training. Comparatively, the number of support vectors is much smaller than the complete data. Clustering is another effective tool to reduce data set size, for example, hierarchical clustering (Yu et al., 2003) and parallel clustering (Pizzuti & Talia, 2003).

The geometric properties of SVM can also be used to reduce the training data. In separable case, the maximum-margin hyperplane is equivalent to find the nearest neighbors in the convex hulls of each class (Bennett & Bredensteiner, 2000). Spatial neighborhood relations of the objects can be used to detect support vectors. Keerthi et al., (2000) proposed an active learning model of sample selection which is based on the measurement of neighborhood entropy. Keerthi & Gilbert (2002) proposed an algorithm that selects the patterns in the overlap region around the decision boundary. Cervantes et al., (2008) presented a SVM model that follows fuzzy C-means clustering to select samples on the boundaries of class distribution. The most popular geometric methods are the Nearest Points Problem (NPP) and the convex hull. NPP was first reformulated to solve SVM classification in Keerthi et al., (2000). The data which are close to those opposite label have high probability to be support vector. The Euclidean distance between these data and the opposite label are computed to find the shortest distance. The problem of this method is the computation time is $O(n^2)$. Gilbert's algorithm (1966) is one of the first algorithms for solving the Minimum Norm Problem (MNP) in NPP. The NPP method is also extended to solve the SMO-SVM optimization problem in the image processing (Keerthi & Gilbert,



ISSN: 2233-7857 IJFGCN

2002). Since the support vectors are usually located in local extremum or near extremum, using extremes examples from the full training set can reduce training set (Guo & Zhang, 2007).

The main objective of SVM is to find the margin to discriminate the dataset into classes. K-Nearest Neighbor (KNN) is one fundamental solution, in addition to that the local data distribution and local geometrical property were applied in (Li, 2011; Xia et al., 2006). Minimum enclosing ball is applied to SVM classification in Cervantes et al., (2008). Approximately optimal solutions of SVM are obtained by Core Vector Machine (CVM) in Tsank et al., (2005). The maximal margin classifier with a given precision for separable data was obtained. Convex hull has been applied in training SVM in Bennett & Bredensteiner (2000). There are some algorithms to compute the convex hull with finite points. The SVM's performance highly depends on the kernel selection, most of the time, the unsuitable kernel function leads to imperfect hyperplane for a nonlinear data samples. In these cases, the closest points in the convex hulls are no longer support vectors. In this case, the soft SVM. Although the slack variable can be applied to solve the soft margin optimization, it effects the optimal performance of SVM. It is evident to obtain the optimum margin with negligible error. Another method is to use the reduced convex hull. The intersection parts disappear by reducing the convex hulls. The inseparable case becomes separable case. The main disadvantage of the reduced convex hull is the convex hull has to be calculated in each reducing step. Variant SVM (v-SVM) can make the intersection set be empty by the choice of parameters (Crisp & Burges, 2000). The v-SVM is similar with the reduced convex hull, the computation process is complex. Yuille & Rangarajan (2003) proposed a Concave–Convex Procedure (CCP) that separates the energy function into a convex function and a concave (non-convex) function. By using a non-convex loss function, it forms a nonconvex SVM. But some good properties of SVM, for example the maximum margin, cannot be guaranteed (Collobert & Bengio, 2000), because the intersection parts of data sets are not satisfied convex conditions. The SVM training time is reduced by convex hull approach, however this method decreases the classification accuracy when there are outliers. Chau et al., (2013) introduced a novel method for SVM classification, called convex–concave hull. After grid preprocessing, the convex hull and the concave (non-convex) hull are found by Jarvis March method. Then the SVM training is performed with these convex-concave hull vertices. This approach is much suitable for higher dimension data, as it was shown in the simulation results that this SVM variant is able to achieve better accuracy rate with less computation effort for training. However, the training time is unavoidable for the supervised classifiers. Here, a Convex-Hull and Geometry based Support Vector Machine (CG-SVM) is proposed to improve the SVM classifier which doesn't require kernel methods and no training time. The proposed classifier is applied for driver inattention detection and the performance is studied to investigate the significance of CG-SVM.

## 3. INTRODUCTION TO SUPPORT VECTOR MACHINE CLASSIFIER

Vapnik (1995) introduce the concept of Support Vector Machine (SVM), a set of supervised learning model meant for classification and regression. SVM makes use of machine learning concepts to linearly separate the data with a hyperplane, while the margin maximizes the prediction accuracy. SVM are the systems based on hypothesis space of a linear function with high dimensional data, where the training is a learning model based on optimization theory. Support vector machine was initially popular with the NIPS community and now is an active part of the machine learning research around the world. SVM models outperforms the conventional ANN models particularly in handwriting recognition problem space. Vapnik (1995) developed the foundations for SVM, the efficiency make this model more popular, and successfully used for many applications in pattern recognition domain especially. SVM follows the approach of structural risk minimization which minimizes the expected risk, rather than empirical risk minimization as followed in conventional neural networks which reduces the error on the training data. This property makes SVM to outperform with better generalization model that is the main objective of statistical learning. Originally SVMs are meant for solving classification problems, however in recent times they have been applied for regression problems too (Vapnik et al., 1997).

Consider M training points, where each input $x_i$ has N attributes (i.e. is of dimensionality D) and is in one of two classes $y_i$ = -1 or +1, i.e the training data is of the form



ISSN: 2233-7857 IJFGCN

$$\{x_i, y_i\} \; where \; i = 1, \dots, M, y_i \in \{-1, 1), x \in \Re^N \tag{1}$$

Here it is assumed that the data is linearly separable, meaning that, drawing a straight line on a graph of $x_1$ vs $x_2$ would separate the two classes when N = 2 and a hyperplane on graphs of $x_1$, $x_2$, … , $x_D$ when N > 2.

This hyperplane can be defined by $w.x + b = 0$ where:

- w is normal to the hyperplane
- $\frac{b}{\|w\|}$ is the perpendicular distance from the hyperplane to the origin.

Support Vectors are the examples closest to the separating hyperplane and the objective of Support Vector Machines (SVM) is to estimate this hyperplane, which optimally separates both classes while considering the possible closest members of each class.

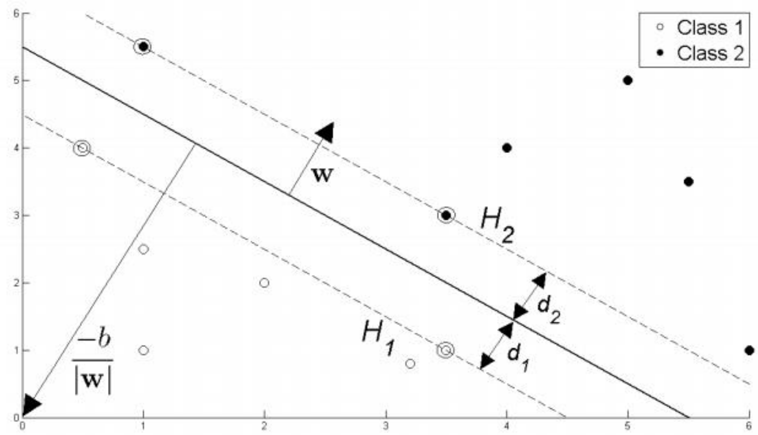


Figure 1. Hyperplane through two linearly separable classes

With reference to Figure 1, applying a SVM means selecting the variables w and b then the training data can be described by:

$$x_i.w + b \geq +1 \quad for \; y_i = +1 \tag{2}$$

$$x_i.w + b \leq -1 \quad for \; y_i = -1 \tag{3}$$

These two equations can be combined and rewritten as:

$$y_i(x_i.w + b) - 1 \geq 0 \; \forall_i \tag{4}$$

When the points closest to the separating hyperplanes alone considered, i.e. the Support Vectors (points encircled in Figure 1), then the two hyperplanes $H_1$ and $H_2$ that these points lie on can be defined by:

$$x_i.w + b = +1 \quad for \; H_1 \tag{5}$$

$$x_i.w + b = -1 \quad for \; H_2 \tag{6}$$

Based on Figure 1, the distances $d_1$ and $d_2$ could be defined from hyperplanes $H_1$ and $H_2$ respectively. The distance between the hyperplane's $H_1$ and $H_2$ refers that $d_1 = d_2$, a metric known as SVM's margin. In order to place the hyperplane to be as far from the support vectors as possible, the margin has to be maximized.

Simple vector geometry shown that the margin is equal to $1/\|w\|$ and maximizing it subject to the constraint in eq.(3) is same as to finding

$$\min\|w\|, \quad such \; that \quad y_i(x_i.w + b) - 1 \geq 0 \; \forall_i \tag{7}$$

Minimizing $\|w\|$ is equivalent to minimizing $\frac{1}{2}\|w\|^2$ and the use of this term makes it possible to perform Quadratic Programming (QP) optimization further. Hence it is necessary to estimate

$$\min\frac{1}{2}\|w\|^2, \quad such \; that \quad y_i(x_i.w + b) - 1 \geq 0 \; \forall_i \tag{8}$$

In order to satisfy the minimization constraints, it is evident to allocate them Lagrange multipliers $\alpha$, where $\alpha_i \geq \forall_i$:

$$L_P \equiv \frac{1}{2} \|w\|^2 - \alpha[y_i(x_i.w + b) - 1 \; \forall_i] \tag{9}$$



ISSN: 2233-7857 IJFGCN

$$\equiv \frac{1}{2}\ \|w\|^2 - \sum_{i=i}^{L} \alpha_i [y_i(x_i.w+b) - 1\ ]$$
$$\equiv \frac{1}{2}\ \|w\|^2 - \sum_{i=i}^{L} \alpha_i y_i(x_i.w+b) + \sum_{i=1}^{L} \alpha_i\ ]$$

It is likely to estimate the w and b which minimizes, and the α which maximizes eq(9), while satisfying the constraint $\alpha_i \geq 0\ \forall_i$. This is possible by differentiating $L_P$ with respect to w and b and setting the derivatives to zero:

$$\frac{\partial L_P}{\partial w} = 0 \Rightarrow\ w = \sum_{i=1}^{L} \alpha_i y_i x_i \tag{10}$$
$$\frac{\partial L_P}{\partial b} = 0 \Rightarrow\ \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{11}$$

Substituting equations (10) and (11) into (9) gives a new formulation which, being dependent on α, which is to be maximized

$$L_D \equiv\ \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i.x_j\ \ s.t.\ \alpha_i \geq 0\ \forall_i, \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{12}$$
$$\equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i H_{ij} \alpha_j\ ,\ \ where\ \ H_{ij} \equiv y_i y_j x_i.x_j$$
$$\equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T H \alpha \quad s.t. \alpha_i \geq 0\ \forall_i,\ \sum_{i=1}^{L} \alpha_i y_i = 0$$

This new definition $L_D$ is known as the Dual form of the Primary $L_P$. It is worth noting that the Dual form requires only the dot product of each input vector xi to be calculated, this is important for the Kernel Trick termed in the following text. Having moved from minimizaing $L_P$ to maximiaing $L_D$, it is necessary to calculate:

$$\max_{\alpha}\left[\sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T H\ \alpha\right] \quad s.t. \quad \alpha_i \geq 0\ \forall_i \quad and \quad \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{13}$$

This is a convex quadratic optimization problem, and a QP solver is implemented and return α from equation (10) will give us w, after the value of b has to be estimated.

Any data sample satisfying equation (11) which is a Support Vector $x_s$ will have the form:

$$y_s(x_s.w+b) = 1 \tag{14}$$

Substituting in equation (10):

$$y_s(\sum_{m\in S} \alpha_m y_m x_m.x_s + b) = 1 \tag{15}$$

where S denotes the set of indices of the support vectors, and it is estimated by finding the indices $i$ where $\alpha_i > 0$. Multiplying through by $y_s$ and then using $y_s^2 = 1$ from equations (1) and (2):

$$y_s^2(\sum_{m\in S} \alpha_m y_m x_m.x_s + b) = y_s \tag{16}$$
$$b = y_s -\ (\sum_{m\in S} \alpha_m y_m x_m.x_s + b) \tag{17}$$

Instead of taking an arbitrary support vector $x_s$, it is superior to yield an average of all the support vectors in S:

$$b = \frac{1}{N_s}\sum_{s\in S}(y_s - \sum_{m\in S} \alpha_m y_m x_m\ .x_s) \tag{18}$$

Now the variables w and b that defined the separating hyperplane's optimal orientation and hence the Support Vector Machine.

**SVM Kernels**

SVM is much more suitable for linear data, constructing a hyperplane is simpler here. This is not the case with most of the data, it might be inseparable or non-linear. Kernel functions are equipped at this



ISSN: 2233-7857 IJFGCN

situation to map non-linear data such data into a high-dimensional space, to make it linearly separable. Figure 2 shows one such data mapping from non-linear to linear space.

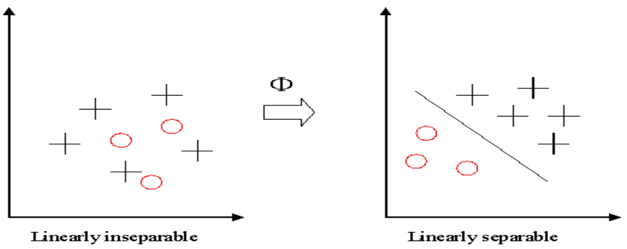


Figure 2. Kernels in SVM

The problem space mapping is achieved by defining the Kernel:

$$K(x,y) = \Phi(x).\Phi(y) \tag{19}$$

Mapping the data into feature space allows us to estimate a similarity measure on the basis of the dot product as shown in Figure 3. The classification performance highly depends on careful selection of feature space.

$$\langle x_1, x_2 \rangle \leftarrow K(x_1,x_2) = \langle \Phi(x_1).\Phi(x_2) \rangle \tag{20}$$

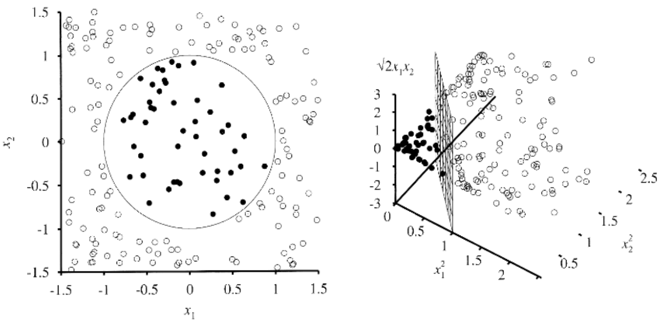

Figure 3. Feature Space Representation

For a conventional SVM with linear data, one major focus is to find w & b and to mean that the problem is solved. For the non-linear problem space, the kernel tricks are used to find the support vectors in nonlinear boundaries.

Choosing a precise kernel could be done by trial and error on the test set, moreover, the kernel selection is application specific, influences the SVM's performance at the most. The proposed CG-SVM resolves this problem using geometry approach.

## 4. THE PROPOSED CONVEX-HULL & GEOMETRY BASED SUPPORT VECTOR MACHINE

Consider a binary classification problem with the training data given in the form $\{x_i, y_i\}, i = 1, \ldots, M, y_i \in \{-1, +1\}, x_i \in \Re^N$. To separate classes, SVM classifier finds a separating hyperplane that maximizes the margin, which is defined as the distance between the hyperplane and closest samples from the classes. To achieve this, at first, each class is approximated with a convex hull (Bennett & Bredensteiner, 2000). A convex hull consists of all points that can be written as a convex combination of the points in the original set, and a convex combination of points is a linear combination of data points where all coefficients are nonnegative and sum up to 1. More formally, the convex hull of samples $\{x_i\}_{i=1,\ldots,L}$ can be written as

$$H^{convex} = \left\{ x = \sum_{i=1}^{M} \alpha_i x_i \mid \sum_{i=1}^{M} \alpha_i = 1,\ \alpha_i \geq 0 \right\} \tag{21}$$

Figure 4 depicts convex hulls of two classes. Following this approximation, SVM finds the closest points in these convex hulls. Then, these two points are joined with a line fragment. The plane, orthogonal to the line fragment that bisects the line, is selected to be the separating hyperplane as shown in Figure 4.



ISSN: 2233-7857 IJFGCN

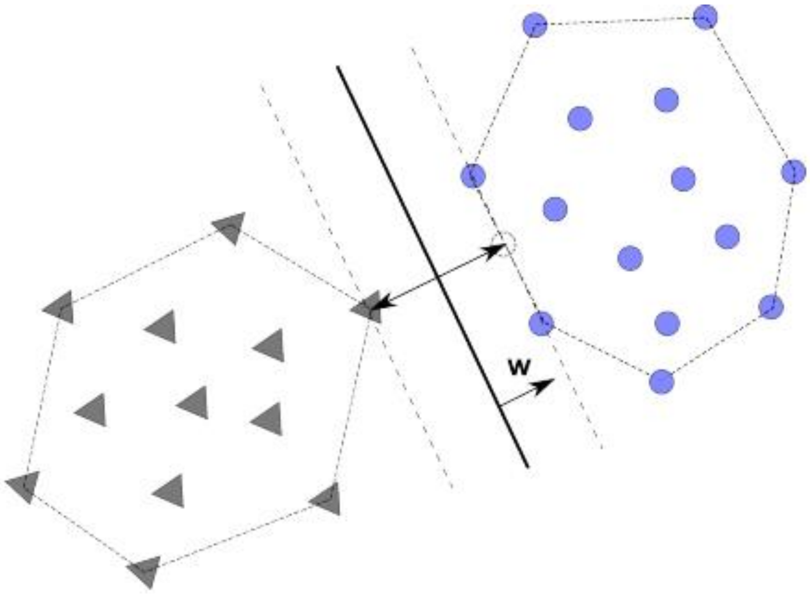


Figure 4. Two closest points on the convex hulls determine the separating hyperplane

For the proposed Convex-Hull & Geometry based SVM (CGSVM), initially the data points are plotted and a convex hull for each class $H_1$ and $H_2$ has been constructed as shown Figure 5, one class samples has been shown with red points and the other with blue points.

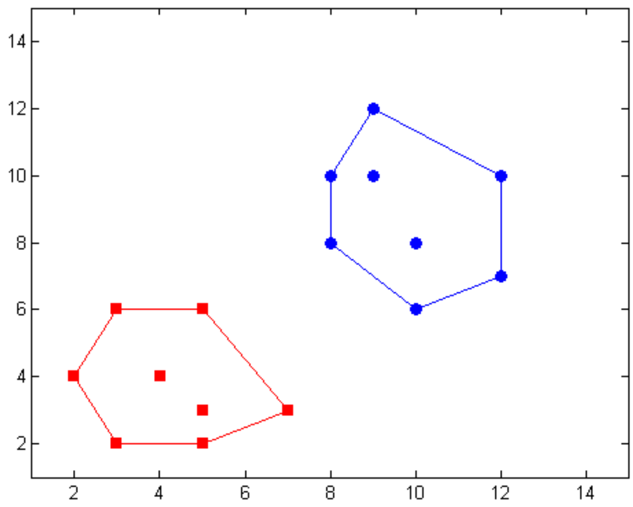


Figure 5. Data points with Convex Hull

In the next step, the centroid for each convex hull $H_{c1}$ & $H_{c2}$, are estimated and connected throw a straight line. Figure 6 shows the output of convex hull centroids, the green markers and connecting them with a straight line, this is to identify the direction of each convex hull.



ISSN: 2233-7857 IJFGCN

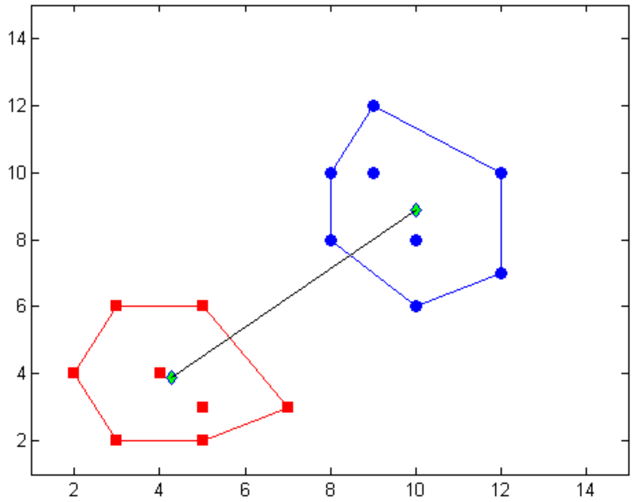


Figure 6. Convex Centroids

With this, the hyperplane, wx+b=0, could have been made perpendicular to the line which connects the centroids. However, the slope won't be appropriate when the size of the convex hulls varies a lot. Hence, the hyperplane is estimated based on geometry approach. Here, after connecting the convex hull centroids, the edge from both convex hull which intersect the connecting line is identified and their end-points are stored .Then each end-point from $H_1$ is connected to the other end-point from $H_2$, whichever is closer, this would introduce two new line segments $L_1$ and $L_2$, presented as dashed line in Figure 7.

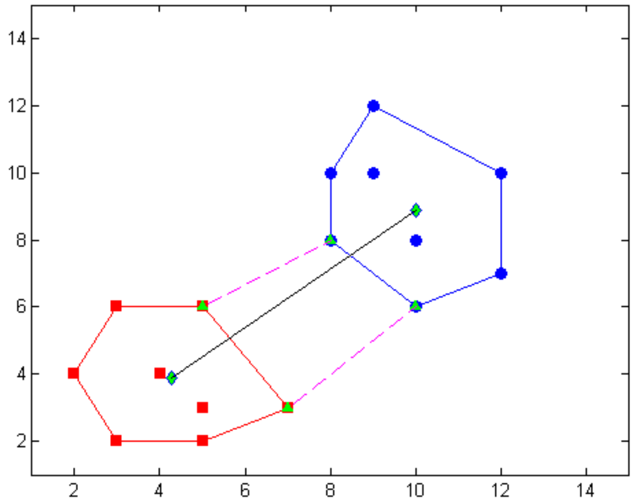


Figure 7. Connecting the vertex points from $H_1$ and $H_2$

Further, a straight line connecting mid-points of the lines $L_1$ and $L_2$, derives the hyperplane as illustrated in Figure 8.



ISSN: 2233-7857 IJFGCN

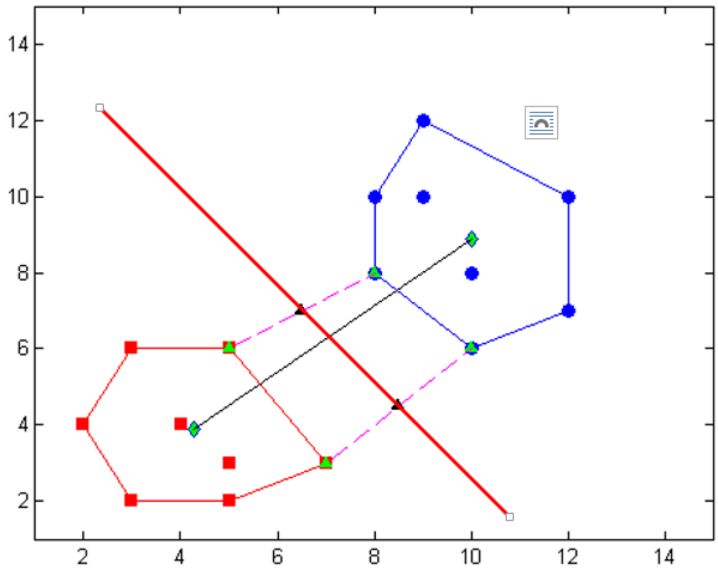

Figure 8. SVM Hyperplane

In the last step, the support vectors are drawn parallel to the hyperplane, which passes through a vertex from $H_1$ and $H_2$ as shown in Figure 10. According to the support vectors, the hyperplane is adjusted to make the distance equal between the hyperplane and the soft margin.

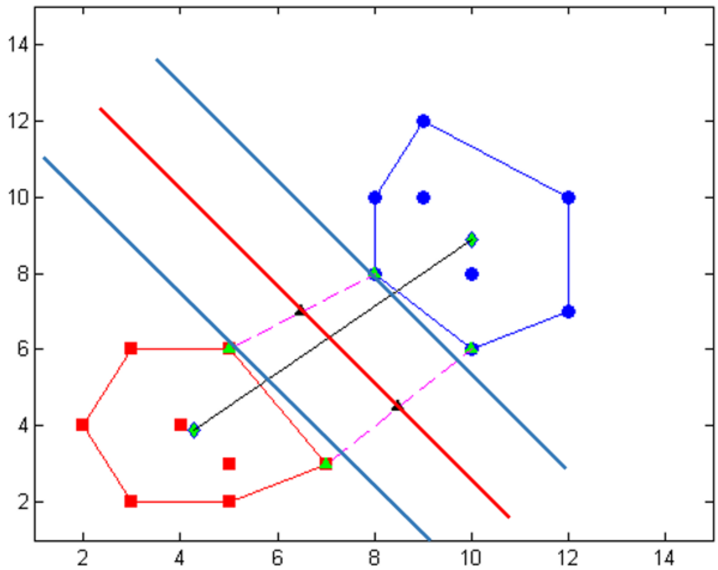

Figure 9. Support Vectors and the Margin

The advantage of the proposed CG-SVM over conventional SVM is the time complexity, and the kernel function selection. The convex-hull and geometry approach doesn't take any iterative or training procedures, hence it takes O(n) time complexity. Also it doesn't require to maintain the data samples, whereas it is enough to hold the convex hull vertices, hence it won't require more than O(n) space complexity. Comparatively the conventional SVM needs O($n^3$) and O($n^2$) time and space complexity respectively, the CG-SVM reduces these complexity significantly with O(n) for both time & memory requirements. Hence, the proposed CG-SVM is faster, memory efficient and significantly improves the SVM classification's performance as discussed in the experimental results section.

## 5. RESULTS AND DISCUSSIONS



ISSN: 2233-7857 IJFGCN

The performance of the proposed CG-SVM is studied with the Ford's stay alert driver's dataset. Each sample is representing a sequential data, recorded at every 100ms during a driving session on the road. The sample consists of 100 participants of different age level, genders and ethnic backgrounds.

There were 33 columns of data, collected from 610 drivers, and 1210 trials for each. In total the dataset has the dimension of 738100 samples with 33 measures, where the first two columns could be ignored as they maintain the sequential numbers, hence, the dataset dimension is reduced to 738100×31. The dataset is divided into two sets, 510 drivers' trials training and the rest of 100 drivers' trails for testing.

The classification performance is studied with the following parameters. The notations used in the formulae are expressed as follows: True Positive (TP) is the number of distracted instances which are correctly detected by the proposed classifier, False Positive (FP) means the number of non-distracted (normal) instances which are incorrectly detected as distracted, True Negative (TN) is the number of normal samples correctly classified, and False Negative (FN) is the number of distracted samples which are imperfectly identified as normal.

**Accuracy**

It is the percentage of correctly classified records and expressed as (TP + TN)/N. That is, the accuracy is the proportion of true results (both true positives and true negatives) in the population. To make the context clear by the semantics, it is often referred to as the "Rand Accuracy". Figure presents the classification results based on accuracy, comparatively the CG-SVM outperforms the other classification algorithms with the maximum accuracy of 0.9129, and the same has been represented in Figure 10.

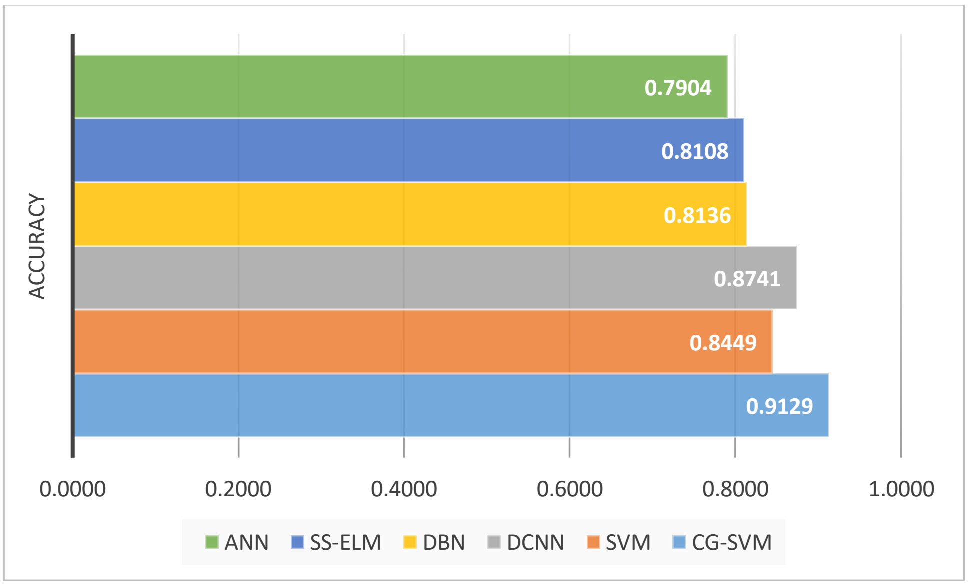


Figure 10. Classifiers Performance with Accuracy Measures

The proposed Driver's Distraction Detection performance is further analyzed with more parameters as given below.

**Classification Success Index (CSI)**

The individual class-specific classification performance can be measured by using Individual Classification Success Index (ICSI) based on Positive Predicted Value (or Precision) and True Positive Rate (TPR) (or Sensitivity) defined as:

$$ICSI_i = 1 - PPV_i + 1 - TPR_i = PPV_i + TPR_i - 1 \qquad (23)$$



ISSN: 2233-7857 IJFGCN

Where, the terms $1 - PPV_i$ and $1 - TPR_i$ represents the type I and II errors for the corresponding class, respectively. Hence, the range for ICSI is from -1 (maximum error) to +1 (minimum error), but the result 0 doesn't have any specific meaning. This measure is symmetric, and linearly connects to the average of TPR and PPV, which is itself known as Kulczynski's metric. The Classification Success Index (CSI) is the overall mean value of ICSI over all classes. Figure 11 presents the classification results based on CSI, the proposed CG-SVM significantly improves CSI to 0.9378, which is greater than the other classification algorithms.

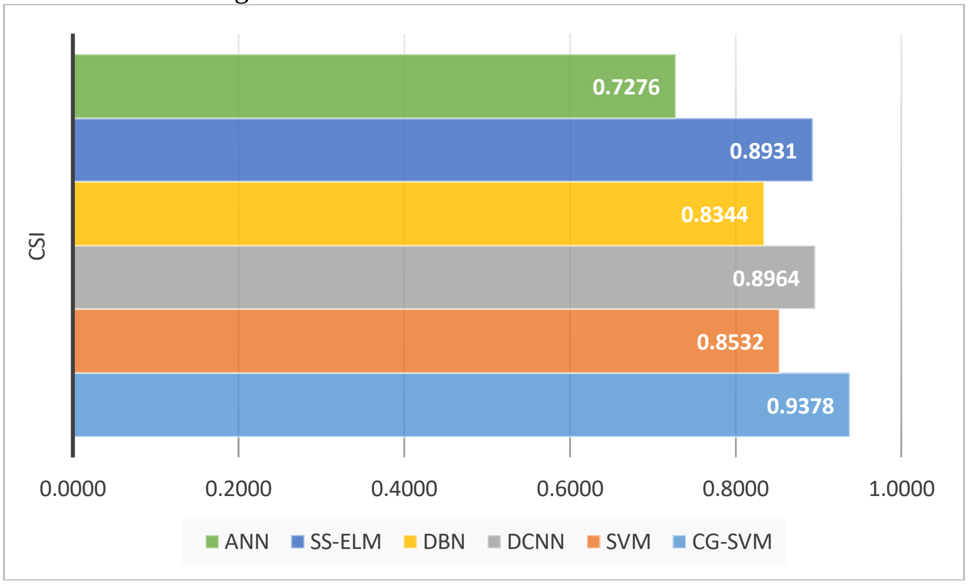


Figure 11. Classifiers Performance with CSI Measures

**Geometric-Mean (GM)**

This measurement is used to maximiz the True Positive (TP) rate and True Negative (TN) rate, and simultaneously keeping both rates relatively balanced. It is defined as

$$GM = \sqrt{tp * tn} \tag{25}$$

Figure 12 presents the classification results based on GM, the proposed CG-SVM significantly improves the performance by achieving higher GM of 0.9528.

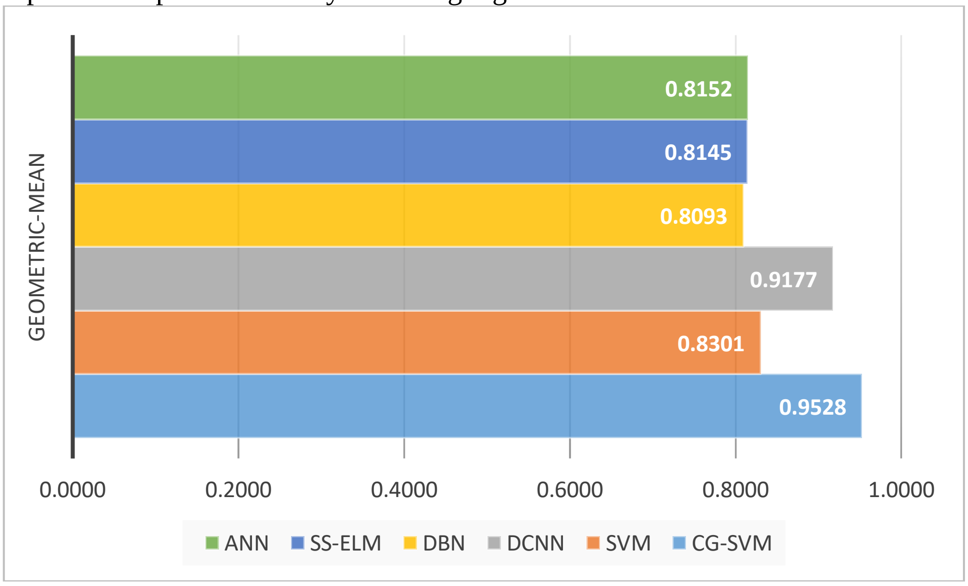


Figure 12. Classifiers Performance with GM Measures



ISSN: 2233-7857 IJFGCN

## 6. SUMMARY

Driver distraction detection system helps to avoid motor vehicle road accidents. There are various classifiers are developed to support the drivers while using in-vehicle systems. Most of these classifiers makes use of the features like physiological, environmental and vehicular data independently. However, it is suggested to detect the driver's distraction with hybrid features. In this paper, a Convex-hull and Geometry based Support Vector Machine (CG-SVM) classifier is proposed for driver inattention detection with all three set of features. The distraction detection performance is compared with five other classifiers and demonstrated that the proposed CG-SVM is able to detect the driver's distraction better than any other classification methods.

## REFERENCES

1. Lee, J. D. (2009). Can technology get your eyes back on the road?. Science, 324(5925), 344-346.
2. Toledo, T., Musicant, O., & Lotan, T. (2008). In-vehicle data recorders for monitoring and feedback on drivers' behavior. Transportation Research Part C: Emerging Technologies, 16(3), 320-331.
3. Victor, T. W., Harbluk, J. L., & Engström, J. A. (2005). Sensitivity of eye-movement measures to in-vehicle task difficulty. Transportation Research Part F: Traffic Psychology and Behaviour, 8(2), 167-190.
4. Liang, Y., Lee, J. D., & Yekhshatyan, L. (2012). How dangerous is looking away from the road? Algorithms predict crash risk from glance patterns in naturalistic driving. Human Factors, 54(6), 1104-1116.
5. Liang, Y., Reyes, M. L., & Lee, J. D. (2007b). Real-time detection of driver cognitive distraction using support vector machines. IEEE transactions on intelligent transportation systems, 8(2), 340-350.
6. Vapnik, V. (1995). The nature of statistical learning theory. Springer science & business media.
7. Li, Y. (2011). Selecting training points for one-class support vector machines. Pattern recognition letters, 32(11), 1517-1522.
8. Collobert, R., & Bengio, S. (2000). Support vector machines for large-scale regression problems. Institut Dalle Molle d'Intelligence Artificelle Perceptive (IDIAP), Martigny, Switzerland, Tech. Rep. IDIAP-RR-00-17.
9. Keerthi, S. S., & Gilbert, E. G. (2002). Convergence of a generalized SMO algorithm for SVM classifier design. Machine Learning, 46(1-3), 351-360.
10. Guo, G., & Zhang, J. S. (2007). Reducing examples to accelerate support vector regression. Pattern Recognition Letters, 28(16), 2173-2183.
11. Xia, C., Hsu, W., Lee, M. L., & Ooi, B. C. (2006). BORDER: efficient computation of boundary points. IEEE Transactions on Knowledge and Data Engineering, 18(3), 289-303.
12. Bennett, K. P., & Bredensteiner, E. J. (2000, June). Duality and geometry in SVM classifiers. In ICML (Vol. 2000, pp. 57-64).
13. Yu, H., Yang, J., & Han, J. (2003, August). Classifying large data sets using SVMs with hierarchical clusters. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 306-315). ACM.
14. J. Mary Dallfin Bruxella & Dr. J.K. Kanimozhi (2019). Review on identification of the drivers' distraction while driving, The International Journal of analytical and experimental modal analysis.(pp. 1676-1690)
15. Liang, Y., Reyes, M. L., & Lee, J. D. (2007). Real-time detection of driver cognitive distraction using support vector machines. IEEE transactions on intelligent transportation systems, 8(2), 340-350.
16. Young, K., Regan, M., & Hammer, M. (2007). Driver distraction: A review of the literature. Distracted driving, 2007, 379-405.



ISSN: 2233-7857 IJFGCN

17. Zhang, G. Y., Cheng, B., Feng, R. J., & Li, J. W. (2008, July). Real-time driver eye detection method using Support Vector Machine with Hu invariant moments. IEEE Proceedings of the International Conference on Machine Learning and Cybernetics (Vol. 5, pp. 2999-3004).
18. Dong, Y., Hu, Z., Uchimura, K., & Murayama, N. (2011). Driver inattention monitoring system for intelligent vehicles: A review. IEEE transactions on intelligent transportation systems, 12(2), 596-614.
19. Kang, H. B. (2013). Various approaches for driver and driving behavior monitoring: a review. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 616-623).
20. Yan, C., Coenen, F., & Zhang, B. (2015). Driving posture recognition by convolutional neural networks. IET Computer Vision, 10(2), 103-114.
21. Liao, Y., Li, S. E., Li, G., Wang, W., Cheng, B., & Chen, F. (2016, June). Detection of driver cognitive distraction: An SVM based real-time algorithm and its comparison study in typical driving scenarios. Proceedings of the IEEE Intelligent Vehicles Symposium (IV), pp. 394-399
22. Liu, T., Yang, Y., Huang, G. B., Yeo, Y. K., & Lin, Z. (2016). Driver distraction detection using semi-supervised machine learning. IEEE transactions on intelligent transportation systems, 17(4), 1108-1120.



ISSN: 2233-7857 IJFGCN