# Aggressive driving behaviour classification using smartphone's accelerometer sensor


Sanjay Kumar Sonbhadra
*Department of IT*
*IIIT, Allahabad*
Prayagraj, India
rsi2017502@iiita.ac.in

Sonali Agarwal
*Department of IT*
*IIIT, Allahabad*
Prayagraj, India
sonali@iiita.ac.in

Mohammad Syafrullah
*Program of master in CS*
*Universitas Budi Luhur*
Jakarta, Indonesia
mohammad.syafrullah@budiluhur.ac.id

Krisna Adiyarta
*Program of master in CS*
*Universitas Budi Luhur*
Jakarta, Indonesia
krisna.adiyarta@budiluhur.ac.id



***Abstract*—Aggressive driving is the most common factor of road accidents, and millions of lives are compromised every year. Early detection of aggressive driving behaviour can reduce the risks of accidents by taking preventive measures. The smartphone's accelerometer sensor data is mostly used for driving behavioural detection. In recent years, many research works have been published concerning to behavioural analysis, but the state of the art shows that still, there is a need for a more reliable prediction system because individually, each method has it's own limitations like accuracy, complexity etc. To overcome these problems, this paper proposes a heterogeneous ensemble technique that uses random forest, artificial neural network and dynamic time wrapping techniques along with weighted voting scheme to obtain the final result. The experimental results show that the weighted voting ensemble technique outperforms to all the individual classifiers with average marginal gain of 20%.**



***Index Terms*—Driving pattern; Dynamic time wrapping (DTW); Sliding window; Random forests; Accelerometer; Time series classification.**


## I. Introduction

A recent study shows that in India, the road accidents cost around INR 3.8 lakh crores [1] and the number of deaths due to road accidents in India will be around 2,50,000 by the year 2025 [2]. It is also observed that in every 4 minutes, one person becomes the victim of road accidents in India. Every year it costs around 3% of country's GDP. Moreover, aggressively driving may increase the fuel consumption rate by upto 255%, which also increases the amount of emission of greenhouse gases responsible for global warming [3].

By analysing a driver's behaviour, the risk of accidents can be reduced. The smartphone's accelerometer sensor data is mostly used to classify the behaviour as aggressive braking, aggressive acceleration, aggressive left turn, aggressive right turn or non-aggressive. Dynamic time warping (DTW) [4], random forest (RF) [5], and neural network methods (DNN) [6] are individually good enough to classify the aggressive behaviour. DTW is a distance-based method, whereas RF and DNN are feature-based methods and the joint operation of these models may give better performance. In this research, a novel ensemble model is proposed, where the above three classifiers are first individually trained and later, the final result is calculated using weighted voting, where weights are proportional to the accuracy of a classifier when evaluated individually.

For experiments, the driving behaviour dataset [6] has been used. This dataset is in the form of time series, where the data is labelled with five different types of driving behaviour: aggressive left turn, aggressive right turn, aggressive braking, aggressive acceleration, and non-aggressive. There are two ways to deal with time series data. The first way is to generate the feature vectors of a time series data to train feature based model (RF and DNN) whereas the second way is to calculate the distance between two time series using dynamic time warping (DTW). All the above mentioned keywords are discussed in proposed methodology section.

The rest of the paper is organized as follows: section 2 contains the pre-existing work on driving patterns analysis. Section 3 presents the proposed work whereas section 4 contains the brief discussion of dataset used and experimental results. The last section contains conclusion and future scope.

## II. Related work

This section includes pre-existing work related to the driving behaviour analysis using the smartphone's sensors. Castignani et al. [7] developed a mobile application that detects acceleration, braking, over-speeding events using motion sensors and GPS, and generates a score for drivers using a fuzzy system that uses real time information like route topology and weather conditions. The score generation method is independent of the vehicle used. Later, Hamdy et al. [8] proposed k-nearest neighbour (KNN) and dynamic time warping (DTW) based methods. These methods can be used to identify the aggressive behaviour of a driver. The Accelerometer, GPS and gyroscope sensors are used to collect the data using a smartphone. Dynamic time warping identifies the similarity between two different time series, whereas k-NN finds the road anomaly. Later, Dai et al. [9] presented a work that detects unusual turning and abruptly changes in speed under the influence of drunk and driving. The smartphone's accelerometer and orientation sensors were used to analyze the lateral and longitudinal acceleration. When the difference of maximum and minimum lateral acceleration exceeds the pre-calculated threshold value then unusual turning is detected. When the maximum and minimum value of longitudinal acceleration exceeds then abruptly changes in speed is detected. Rui et al. [10] proposed a work to detect driving patterns using vehicle sensors and guide the driver to reduce the fuel consumption. It collects the data from vehicle sensors such as speed, acceleration and rotations per minute. The collected data is transferred to a smartphone which does rest of processing. Authors also showed the important relationship between fuel consumption and driving behaviour.

Afterwards, Juan Carmona et al. [11] proposed a driver behaviour analysis tool using a less expensive hardware. This tool uses vehiclel's built-in sensors using CAN-BUS. The velocity, steering wheel angle, brake frequency, linear acceleration and GPS data is provided as an input to the data fusion module that classifies the driving event into aggressive or normal. The system was tested in urban and highway areas. Later, Han et al. [12] presented a pattern recognition method that uses a kernel density function and euclidean distance with the Bayesian theory to calculate the probability of aggressive and non-aggressive events. This method was compared with fuzzy logic system and it is observed that this methodology is more robust as compared to the fuzzy logic based approach. It is also evident that the use of this method increases the precision of detection of aggressive event by 3%, and non-aggressive events by 22%.

In the same context, Saiprasert et al. [13] presented three different methods using smartphone's GPS and accelerometer sensor to identify the aggressive behaviour. Each method is suited for different types of applications. The first method uses a pre-calculated threshold to classify the behaviour whereas the second method applies the pattern matching algorithm by analyzing and comparing the time series data. The third algorithm includes a self triggering method, to make it more efficient. These methods are fully adjustable that makes it suitable for variety of applications. Chen et al. [14] worked on the detection of drunk driving using support vector machine (SVM). Various measurements such as maximum speed, minimum speed, average speed, steering wheel rotation angle etc. were used for training the model. The SVM trained with all these features resulted in an accuracy of upto 70%. Further principal component analysis (PCA) was used to obtain the subset of important features. Johnson et al. [15] implemented a system known as MIROAD. This system uses the fusion of multiple sensors such as gyroscope, magnetometer, accelerometer, GPS and video data along with dynamic time warping (DTW) to identify the various aggressive events. The proposed approach does not requires huge computation and can be used in any smartphone without any outside processing. The DTW method performs accurately even with small size of dataset. Meanwhile, Chen et al. [16] worked on identifying specific types of abnormal driving events such as weaving, sideslipping, fast u-turn, aggressive braking, swerving, abnormal turning. Smartphone's accelerometer and orientation sensors were used to extract total 16 discriminating features. Then SVM was used as a classifier, which resulted in an average accuracy of around 90%.

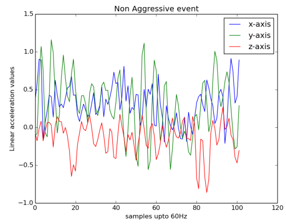


Fig. 1. Linear acceleration values of a non aggressive event (length 2 seconds)

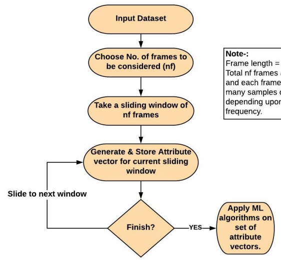


Fig. 2. Steps to Machine learning method.

## III. Proposed Methodology

The problem of drivers' behaviour detection can be viewed as a time series classification problem, where a series of accelerometer data is given. This dataset contains five different driving behaviour classes: aggressive left turn, aggressive right turn, aggressive braking, aggressive acceleration, and non-aggressive. In this paper, two different approaches have been proposed for the time series classification. The first one is a machine learning based approach where the classification models are trained with extracted features from a time series data. The features are some of the important statistical parameters such as mean, median, standard deviation etc. The detailed explanation of extracting the features and training the classifier is provided in the upcoming subsections. The second type of method is a distance-based method that actually uses dynamic time warping (DTW) to calculate the distance between two different time series samples. The distance between two time series is inversely proportional to the similarity between them. The Fig. 1 shows the non-aggressive event plotted for 2 seconds time window, and it is visible that acceleration values are not going beyond 1.5 and even a few goes beyond 1.0.

### A. Machine learning based method

Fig. 2 shows the working of machine learning approach. It requires a set of feature vectors as an input for training. For every time series data available in the dataset, a set of feature vectors is obtained, which is taken as input to the machine learning models. Random forest and neural network are used as classifiers in this paper.

*1) Generation of feature vector:* In order to obtain a set of feature vectors from the available dataset, first a window of fixed length is chosen in range of 2 to 4, as the duration of all aggressive driving events is also in the range of 2 to 4. After

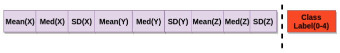


Fig. 3. An Attribute vector instance with labeled class

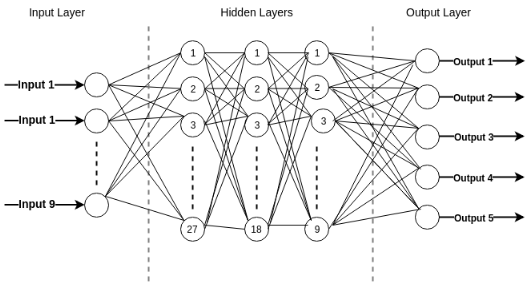


Fig. 4. Architecture of neural network used for classification

fixing the size of the window, we chose the first window as the starting time series sample. After extracting the features from this, the window slides by 1 second.

To generate a feature vector of a time series data; mean, standard deviation and median are calculated for individually $x$, $y$, and $z$-axis; hence, there are total 9 features generated. In supervised learning a class label is also required, so the class label is augmented (which is provided as the ground truth). Fig. 3 shows an instance of an attribute vector, where Mean is the mean of all sampled points, Med is the median of all sampled points, SD is the standard deviation of all sampled points for a particular window, and $x$, $y$, and $z$ refer to all three axes.

*2) Artificial neural network:* Artificial neural network (ANN) is a part of machine learning, where mainly three kinds of layers: input, hidden, and output layers. Each layer consists of multiple nodes that takes input from the previous layer and passes onto the next layer after applying activation function over it. It can be used to learn complex patterns; hence, used in many applications nowadays. In this research the neural network architecture is sequential and has 1 input layer, 3 hidden layers, and 1 output layer. The output layer outputs 5 values, showing probability to belong to a class. As shown in Fig. 3 there are 9 features extracted from a 3-dimensional time series data sample. The Fig. 4 shows architecture of ANN used that takes 9 features as input, which is densely connected to next layers. The hidden layers contains 27, 18 and 9 neurons respectively. The last layer is for output having 5 neurons. Relu(z) = max(0, z), is the activation function for initial 3 layers then whereas the softmax is the activation function for final output layer that converts numerical values to probability. The loss function for the architecture used is cross-entropy [18] that needs to be minimised for each subsequent epochs.

*3) Random forests:* Random forest is an ensemble learning method for classification where set of decision trees collectively vote to classify. The trees are constructed by randomly choosing the subset of features from a subset of training data points. This method removes the chances of over-fitting problems while training. In random forest, the number of decision trees is fixed to 100 and maximum depth of a decision tree is kept limited to 20.

### *B. Dynamic time warping method*

DTW aims to find the optimal alignment between two different time series, efficiently with the help of dynamic programming approach. Suppose there are two different time series $A$ and $B$ with length $L$, so there are total $L$ points in both time series and the objective is to find the best alignment to compare them. DTW allows similar shapes to match even if they are out of phase in the time axis and it can even skip out some distortions. An alignment between two different time series will be a set of pair of points such as $(a_i, b_j)$, where $a_i$ is from series A and $b_j$ is from series $B$. One point can also match from two or more points. Optimal alignment should minimise the Eq. 1) given below.

$$\sum_{(a_i,b_j)\epsilon Alignment} Euclidean\_Dis(a_i, b_j) \qquad (1)$$

An alignment is represented like this $\{(a_0,\ b_0),\ (a_1,b_0),\ (a_1,b_1),\ .....\}$. It shows the mapping of points between two series $A$ and $B$. To find the optimal alignment, it is initially required to calculate distance matrix. The distance matrix is a 2-dimensional matrix of size $L \times L$ ($L$ is the length of time series $A$ and $B$), representing Euclidean distance of every point of series $A$ from every point of series $B$. Afterwards, accumulated cost matrix is calculated, so that it can minimize the Eq. 1. Each cell of the accumulated cost matrix represents the minimum summation of the distance of all points from the starting point of series. This is calculated using recurrence relation as follows:

$$\begin{aligned} AC[i,j] = minimum(AC[i-1,j-1], AC[i-1,j],\\ AC[i,j-1]) + dist[i,j] \end{aligned} \qquad (2)$$

where $AC$ is accumulated cost matrix, $dist$ is Distance matrix calculated using Euclidean distance formula between every points of series $A$ and $B$.

The accumulated cost matrix is filled until the top right corner starting from bottom left, which ensures that both series are compared from start to the end point. In the final step, the DTW distance between two different time series $A$ and $B$ is calculated via value of the topmost right cell of accumulated cost matrix i.e. $AC[L-1][L-1]$. The distance is lower when two time series are more similar otherwise they are different.

There are various instances of time series samples for each class and for labelling any time series sample, 1-Nearest Neighbour method can be applied. The DTW distance is calculated of the time series data samples in the dataset. The class label nearest to the time series data is assigned to it. It is a lazy learning method because every time it compares the distance between multiple time series. One can average the time series of each class in an efficient way so that resulting time series is the best representation of its class. Algorithm 1 describes the method to find the average template for each of the five classes. The medoid of a set of sequences is the one series ($X$) from the set which minimizes the Eq. 3, where $S$ is the set of sequences of a class and $X \in S$.

$$\sum_{Y \in S} (DTW - Distance(X, Y))^2 \quad (3)$$

**Algorithm 1** Algorithm to calculate the average of set of sequences.

1: Initially start with medoid so, $T_{avg}$ = medoid(S)
2: A = [$\phi$ ,$\phi$ ,$\phi$ ,... ] the set of n empty sets to store alignments.
3: for each sequence x in S :
4: AlignX = $DTW_{Path}$($T_{avg}$, x)
5: for i = 1 to n :
6: A[i] = AlignX[i] ∪ A[i]
7: for i from 1 to n :
8: $T_{avg}$ [i] = mean(A[i])

In Algorithm 1, $n$ is the length of both time series and $T_{avg}$ is the average sequence which keeps on improving in each iteration. It can average the set of sequences of a class. It starts with medoid and iterates from step no. 2 to 8 till convergence. In many cases, few hundred of iterations are enough to obtain the average sequence. $DTW_{Path}$ function returns the optimal alignment found using DTW between two-time sequences, that minimises the Eq. 1.

For labelling of any test series, it is needed to find the DTW distance of the series from all five average sequences obtained earlier. The series will be assigned the label of that class whose average sequence gives the minimum DTW distance from it. Minimum DTW distance indicates more similarity (or nearest point). In this way, it reduces the time complexity of the solution proposed, as it is invoking DTW subroutine only five times, which is much smaller than the total number of time sequences given in the dataset.

### C. Weighted voting ensemble to improve accuracy

There are two different types of methods to solve the time series classification problem: distance based and features based, as already discussed. In distance-based method, DTW is used to find the distance (or similarity) between two time sequences. In feature-based methods, there is a need to obtain a feature vector of a time series, then model is trained using these feature vectors. Random forests and neural network architecture are used for experiments. In ensemble technique, combining two or more learners improves the accuracy of the resulting classification model. In this paper, a weighted voting ensemble(Fig. 5) model is used which is combination of three different classifiers weights are assigned proportionally to the average cross-validation scores. So a strong learner will have a higher weight than the weak learner.

In Random forest and neural network methods, there are five probability values as an output to an instance series.

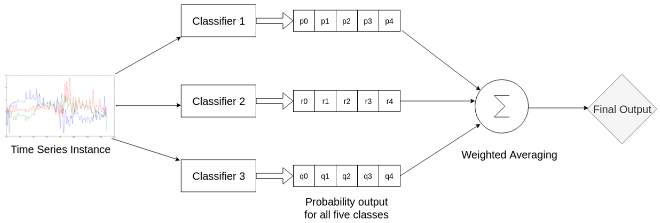


Fig. 5. Weighted voting ensemble for classification.

These values show probability of each class; the class with higher probability value will be assigned to the instance series. But DTW provides only the distance from all five templates representing classes individually. To convert the distance into probability a modified formula of softmax function [19] is used as represented in Eq. 4. This function assigns probability taking distances as input, and probability assignment is inversely proportional to DTW-distance as distance between two series is inversely related to the similarity between them.

$$r_i = \frac{e^{-d_i}}{\sum_x e^{-d_x}} \quad (4)$$

In Eq. 4, $d_i$ is the distance from the $i^{th}$ class series, $x$ iterates through all five classes(0 to 4), then $r_i$ provides the probability values for the $i^{th}$ class obtained using DTW. RF classifier outputs $p_0$, $p_1$, $p_2$, $p_3$, $p_4$ as the class probabilities for a time series instance. Similarly, NN outputs $q_0$, $q_1$, $q_2$, $q_3$, $q_4$ and DTW provides $r_0$, $r_1$, $r_2$, $r_3$, $r_4$ as its probability output, using Eq. 4.

$$C_i = \frac{p_i * W_1 + q_i * W_2 + r_i * W_3}{W_1 + W_2 + W_3} \quad (5)$$

In Eq. 5, $C_i$ is the final class probability using weighted voting ensemble, and $W_1$, $W_2$, and $W_3$ are weights assigned to the classifiers RF, NN, and DTW respectively. After calculating the distance for $i$ = 0 to 4, which is for all five classes respectively, then assign the instance to a class whose probability is maximum.

## IV. Experimental Results

### A. Dataset description

This section describes the driving behaviour dataset used for the experiments [17]. In this work, the experiment has been done in 4 car trips on an average of an hour each. The smartphones sensors used in this paper provides up to 60 samples per second. The sensor provides the values in 3 dimensions $x$, $y$ and $z$, and noted time-stamp when the data was taken and stored in CSV file as shown in Fig. 6.

The ground truth values of events with starting and ending of the event (in seconds) are stored in another CSV file as shown in Fig. 7. Enumeration of events are as follows: '0' as non-aggressive, '1' as right turn aggressive, '2' as left turn aggressive, '3' as aggressive braking, and '4' as aggressive acceleration.

| timestamp | x | y | z |
|---|---|---|---|
| 0 | -0.1616024371 | 0.1201739673 | -0.2098931518 |
| 0 | -0.1226284383 | 0.3156377901 | -0.3809964702 |
| 0 | -0.1787770614 | 0.3301805262 | -0.360695758 |
| 0 | 0.016043462 | 0.0387586297 | -0.2782036481 |
| 0 | 0.1417157333 | -0.1624922219 | -0.0497964292 |
| 0 | 0.0947013699 | -0.2086841363 | 0.0609764263 |
| 0 | -0.0937785397 | 0.1269107899 | -0.190099904 |
| 0 | -0.20495549 | 0.1981129375 | -0.3375213472 |
| 0 | -0.0703803654 | 0.0812154204 | -0.2736953278 |
| 0 | 0.047233676 | -0.2192403246 | -0.0182329735 |
| 0 | 0.0853048782 | -0.3227521316 | -0.1067391428 |
| 0 | 0.0219364705 | 0.0108426618 | -0.0879220141 |
| 0 | -0.0456436358 | 0.0900660833 | -0.2529803809 |
| 0 | -0.0750211332 | 0.2064443884 | -0.1626452998 |
| 0 | 0.0485853556 | -0.0335742881 | -0.0741314969 |
| 0 | 0.0366045312 | -0.1146583636 | -0.0637145104 |
| 0 | 0.0605672988 | 0.0103378178 | -0.1459923438 |
| 0 | 0.0174135822 | 0.0979623191 | -0.271338463 |
| 0 | -0.0866376581 | 0.0725138137 | -0.1780290326 |
| 0 | 0.0125661597 | -0.0279836434 | -0.1894627966 |
| 0 | 0.0319553934 | -0.0517540606 | -0.0567802701 |
| 1 | 0.0687299766 | -0.1510330524 | -0.1260529092 |
| 1 | -0.035639516 | 0.0885218293 | -0.0985758927 |
| 1 | -0.0759690177 | 0.0101036887 | -0.1848568883 |
| 1 | -0.0138792613 | 0.0018049745 | -0.0700439592 |
| 1 | -0.0307783335 | -0.0746561184 | -0.0785854609 |
| 1 | -0.0005406739 | -0.0319050016 | -0.1509209207 |
| 1 | 0.0476584592 | 0.1027951101 | -0.1936119301 |
| 1 | 0.1477709152 | 0.10727161 | -0.2511411824 |
| 1 | -0.0206243105 | 0.0542387616 | -0.1413978567 |
| 1 | 0.0360858868 | 0.0289639512 | -0.0477697795 |
| 1 | 0.0047232437 | -0.0183581916 | -0.0917035688 |

Fig. 6. Few dataset instances with timestamp, x, y, z axis , and linear acceleration values

| event | start | end |
|---|---|---|
| 0 | 2 | 6.5 |
| 1 | 19.5 | 23.5 |
| 3 | 30 | 33.5 |
| 1 | 95 | 98 |
| 4 | 247 | 251.5 |
| 2 | 348.7 | 352.3 |
| 0 | 485 | 489 |
| 4 | 496 | 499.5 |
| 1 | 587 | 590 |
| 3 | 750 | 753.8 |
| 1 | 840.7 | 844 |
| 1 | 980 | 983.2 |
| 4 | 1087.4 | 1090.9 |
| 1 | 1139.8 | 1142 |
| 2 | 1201 | 1202.9 |
| 1 | 1211.4 | 1213.5 |

Fig. 7. Ground Truth Events with start and end timestamp of event.

### B. Discussion

For evaluating the performance of the proposed model, the dataset is splitted randomly into train and test set in 70% and 30% respectively. This process is iterated several times (approx. 100 iterations) to find the average accuracy of a model. Various performance metrics for classification such as accuracy, precision, and recall have been calculated in experiments (Eqs 6, 7 and 8). The confusion matrices of is also calculated for every case.

Initially, various cases of random forest (a feature-based method) are compared, by varying sliding window size from 2 to 4. In Fig. 8, the confusion matrix is shown of the random forest when the sliding window size = 3. The Diagonal values show the total number of correct classifications.

$$Accuracy = \frac{\text{Sum of diagonal elements of CM}}{\text{Sum of all elements of CM}} \tag{6}$$

| True / Pred | c-0 | c-1 | c-2 | c-3 | c-4 |
|---|---|---|---|---|---|
| class-0 | 204 | 0 | 0 | 0 | 0 |
| class-1 | 0 | 357 | 0 | 52 | 0 |
| class-2 | 0 | 204 | 205 | 0 | 0 |
| class-3 | 0 | 0 | 0 | 49 | 0 |
| class-4 | 102 | 0 | 0 | 0 | 408 |

Fig. 8. Confusion matrix of RF for sliding window length = 3.

$$Precision_i = \frac{CM(i,i)}{\sum_j CM(j,i)} \tag{7}$$

$$Recall_i = \frac{CM(i,i)}{\sum_j CM(i,j)} \tag{8}$$

*where CM is the confusion matrix.*

Based on the confusion matrix shown in Fig. 8, RF is good at separating class-0 (which is non-aggressive behaviour) as it accurately classified every instance belonging to that class. Using Eqs. 6, 7 and 8, the following values are found: $Accuracy$ = 0.77, $Precision_0$ = 0.67, $Precision_1$ = 0.64, $Precision_2$= 1.0, $Precision_3$= 0.50, $Precision_4$ = 1.0, $Recall_0$= 1.0, $Recall_1$= 0.87, $Recall_2$= 0.5, $Recall_3$= 1.0, and $Recall_4$= 0.8

Similarly, the confusion matrices of every algorithm with a varying sliding window of length 2, 3 and 4, are calculated. Table I shows the value of average accuracy obtained of each algorithm by repeating the experiments several times.

TABLE I
AVERAGE ACCURACY OF DIFFERENT CLASSIFIERS INDIVIDUALLY

| Accuracy shown in percentage | | | |
|---|---|---|---|
| Classifiers used | Sliding window size = 2 | Sliding window size = 3 | Sliding window size = 4 |
| RF method | 56% | 72% | 62% |
| NN method | 49% | 77% | 61% |
| DTW method | 58% | 86% | 70% |

It is evident from the Table I that all the algorithms performed well while taking the sliding window length 3. Hence, for ensemble model sliding window size is chosen as 3, and weights are proportional to the average accuracy shown in the Table I for each of the three methods. This paper also demonstrates some variations of averaging the set of time sequences for the use of DTW. Table II shows that comparison of the performance of the DTW method.

TABLE II
DTW WITH DIFFERENT VARIATIONS OF AVERAGING THE SEQUENCES.

| **Method of averaging all sequences of a class** | **Accuracy in percentage** |
|---|---|
| Direct one to one averaging of sequences. | 56% |
| Choosing medoid as an average. | 49% |
| Aligned averaging using DTW. | 58% |

Comparison of weighted voting ensemble technique with other base methods is shown in Fig. 9. It shows that the weighted voting ensemble technique has outperformed the RF, DTW, NN individually with a margin of 22%, 7%, and 17% respectively.

## V. CONCLUSION AND FUTURE SCOPE

This paper proposes a robust weighted voting ensemble technique for driving behaviour analysis, which is more promising than individual classifiers. The experimental results show that the weighted voting ensemble technique outperforms to all the individual classifiers with average marginal gain of

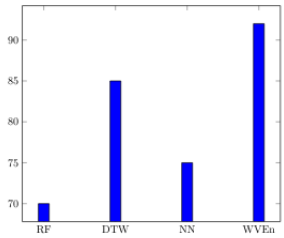


Fig. 9. Average accuracy comparison of individual classifiers with the weighted voting ensemble technique, taking window size as 3.

20%. The results also show that irrespective of using any of the methodologies, sliding window size of 3 provides better results. When comparing individual classifiers, DTW outperforms other methods. Recently, gyroscope, magnetometer and GPS are common in smartphones. Fusion of these along with accelerometer may bring out more accurate model.

## REFERENCES


[1] J. M. Kumar, R. Mahajan, D. Prabhu and D. Ghose, ”Cost effective road accident prevention system,” 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Noida, 2016, pp. 353-357. doi: 10.1109/IC3I.2016.7917988

[2] Singh, Sanjay Kumar. ”Road traffic accidents in India: issues and challenges.” Transportation research procedia 25 (2017): 4708-4719.

[3] Faria, Marta V., et al. ”How do road grade, road type and driving aggressiveness impact vehicle fuel consumption? Assessing potential fuel savings in Lisbon, Portugal.” Transportation Research Part D: Transport and Environment 72 (2019): 148-161.

[4] Kate, Rohit J. ”Using dynamic time warping distances as features for improved time series classification.” Data Mining and Knowledge Discovery 30.2 (2016): 283-312.

[5] T. Gindele, S. Brechtel and R. Dillmann, ”Learning Driver Behavior Models from Traffic Observations for Decision Making and Planning,” in IEEE Intelligent Transportation Systems Magazine, vol. 7, no. 1, pp. 69-79, Spring 2015. doi: 10.1109/MITS.2014.2357038

[6] Wang, Zhiguang, Weizhong Yan, and Tim Oates. ”Time series classification from scratch with deep neural networks: A strong baseline.” 2017 International joint conference on neural networks (IJCNN). IEEE, 2017.

[7] G. Castignani, T. Derrmann, R. Frank and T. Engel, ”Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring,” in IEEE Intelligent Transportation Systems Magazine, vol. 7, no. 1, pp. 91-102, Spring 2015. doi: 10.1109/MITS.2014.2328673

[8] Hamdy, Aya & Atia, Ayman & Mostafa, Mostafa-Sami. (2017). Recognizing Driving Behavior and Road Anomaly using Smartphone Sensors. International Journal of Ambient Computing and Intelligence. 8. 22-37. 10.4018/IJACI.2017070102.

[9] J. Dai, J. Teng, X. Bai, Z. Shen and D. Xuan, ”Mobile phone based drunk driving detection,” 2010 4th International Conference on Pervasive Computing Technologies for Healthcare, Munich, 2010, pp. 1-8. doi: 10.4108/ICST.PERVASIVEHEALTH2010.8901

[10] R. Araújo, . Igreja, R. de Castro and R.E. Araújo, ”Driving coach: A smartphone application to evaluate driving efficient patterns,” 2012 IEEE Intelligent Vehicles Symposium, Alcala de Henares, 2012, pp. 1005-1010

[11] Carmona, Juan, et al. ”Data fusion for driver behaviour analysis.” Sensors 15.10 (2015): 25968-25991.

[12] Han, Wei, et al. ”Statistical-based approach for driving style recognition using Bayesian probability with kernel density estimation.” IET Intelligent Transport Systems 13.1 (2018): 22-30.

[13] Saiprasert, C., Pholprasit, T., & Thajchayapong, S. (2017). Detection of driving events using sensory data on smartphone. International journal of intelligent transportation systems research, 15(1), 17-28.

[14] Chen, H., & Chen, L. (2017). Support vector machine classification of drunk driving behaviour. International journal of environmental research and public health, 14(1), 108.

[15] D. A. Johnson and M. M. Trivedi, ”Driving style recognition using a smartphone as a sensor platform,” 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, 2011, pp. 1609-1615.

[16] Z. Chen, J. Yu, Y. Zhu, Y. Chen and M. Li, ”D3: Abnormal driving behaviors detection and identification using smartphone sensors,” 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Seattle, WA, 2015, pp. 524-532. doi: 10.1109/SAHCN.2015.7338354

[17] Ferreira J, Ju´nior, Carvalho E, Ferreira BV, de Souza C, Suhara Y, Pentland A, et al. (2017) Driver behavior profiling: An investigation with different smartphone sensors and machine learning. PLoS ONE 12(4): e0174959. https://doi.org/10.1371/journal.pone.0174959

[18] Kurata, Gakuto, Bing Xiang, and Bowen Zhou. ”Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence.” Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.

[19] Okuda, H., Ikami, N., Suzuki, T., Tazaki, Y., and Takeda, K. (2012). ”Modeling and analysis of driving behavior based on a probability-weighted ARX model.” IEEE Transactions on Intelligent Transportation Systems, 14(1), 98-112.