Driving Maneuvers Recognition and Classification Using A Hyprid Pattern Matching and Machine Learning

Munaf Salim Najim Al-Din
Department of Electrical and Computer Engineering, University of Nizwa, Birkat Al-Mouz, Nizwa, P.C. 616, Oman

***Abstract*—Since most of the road and traffic accidents are related to human errors or distraction, the study of irregular driving behaviors is considered one of the most important research topics in this field. To prevent road accidents and assess driving competencies, there is an urgent need to evaluate driving behavior through the design of a driving maneuvers assessment system. In this study, the recognition and classification of highway driving maneuvers using smartphones' build-in sensors are presented. The paper examines the performance of three classical machine learning techniques and a novel hybrid system. The proposed hybrid system combines the pattern machining Dynamic Time Warping (DTW) technique for recognizing driving maneuvers and the machine learning techniques for classification. Results obtained from both approaches show that the performance of the hybrid system is superior to that obtained by using classical machine learning techniques. This enhancement in the performance of the hybrid system is due to the elimination of the overlapping in the target classes due to the separation, the recognition and the classification processes.**

Keywords*—*Driving behavior classification; driving maneuvers; pattern matching; machine learning

I. Introduction

According to previous studies in the field of traffic safety and road accidents, abnormal or irregular driving behaviors have been considered to be one of the main factors that greatly contribute to road accidents [1]. With the increase of vehicles all over the world, abnormal driving patterns detection and monitoring will most defiantly contribute to the reduction of road accidents. In addition to the above benefits, studies of driving patterns and behaviors have been instrumental in the development of advanced driver assistance systems (ADAS) and autonomous vehicles (AVs) [2, 3]. Driving behaviors can be assessed from two different perspectives namely; drivers' actions or the vehicle's dynamic state. In the first approach the driver is considered as the focal element where a set of parameters that affect the driver's vigilance and attention are continuously observed to predict his/her competence to achieve the driving course in a robust and safe manner [4]. Drivers' state monitoring systems may contain different modules, such as facial recognition systems, physiological signals monitoring and drivers' interaction and control. For example, drivers' interaction and control, combined with facial recognition, have been shown to be effective in detecting driver fatigue, drowsiness, and distraction [4].

In the second approach, the dynamic state of the vehicle, such as longitudinal and lateral accelerations, braking, is monitored to detect and classify abnormal driving patterns or maneuvers. In general, signals captured through the vehicle's built-in sensors captured through the CAN-BUS [5, 6], or external sensors such as accelerometers and gyroscopes, and GPS [7, 8], or a combination of in-vehicle and external sensors [9], can be used for the aforesaid purposes. In the past ten years, smartphones have emerged as an efficient and very reliable tool in this field, since they have powerful computational capabilities, richness and variety of built-in sensing devices and ability to have multiple ways of communication with external devices connected to the OBD-II port. Furthermore, smartphones especially with the emergence of 5G technology have been enabled to play cooperative coordinator between vehicles through vehicle-to-everything networks. With all the above listed features provided by smartphones, attention has been immensely focused on the utilization of smartphones in monitoring and analyzing driving behaviors.

The analysis of driving behavior is dependent on the maneuvers to be analyzed as well as the collected data or estimated parameters used to describe them. Various methods were proposed in the literature to perform this task. The simplest approach considers the driving process as a rule-based or fuzzy classification problem. A set of thresholds are defined or extracted, based on experience or trial-and-error, to assess the driving parameters and then classify driving maneuvers [7, 10-18]. In general, these methods are not reliably accurate because the thresholds, fuzzy sets and rules, as well as the classification results, are all based on presuppositions. The second approach is based on pattern matching and recognition techniques, such as Dynamic Time Warping (DTW) [19]. This approach is based on measuring the level of similarity between captured signals and standard patterns. The disadvantage of using classical DTW is the heavy computational burden especially when dealing with multivariate time series.

Recent research has demonstrated that machine learning techniques are capable of identifying and classifying irregular driving patterns using models and rules that evaluate driving maneuvers and then driving behavior. Machine learning approaches are generally classified into supervised learning approaches and unsupervised learning approaches. Various supervised learning approaches, such as K-Nearest Neighbor (KNN), Naive Bayes, Decision Trees, and Random Forest (RF)

[20], linear regression [21], Support Vector Machines (SVM) [22, 23], and Neural Networks [24] require the extraction of features, such as statistical values, time domain parameters, and frequency domain parameters, for training. On the other hand, unsupervised learning approaches, such as K-means clustering [25] and Principal Component Analysis algorithms [21] can infer and generate rules and threshold-based discriminators for clustering purposes. During the past decade, different methodologies and techniques have been proposed and implemented successfully in the field of driving behavior classification [4, 8].

In this paper, three classical machine learning techniques namely Random Forest (RF), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) were used to recognize and classify six highway driving maneuvers. The data required for training and testing the machine learning models were collected through smartphones' accelerometers and gyroscope sensors. Furthermore, a novel hybrid approach based on the integration of pattern matching DTW and machine learning approaches has also been proposed and investigated in this study. The basic idea of this hybrid approach is to separate the recognition process from the classification process. The DTW developed in this study is used to provide signal similarity measures for the input signals, while the three above-mentioned machine learning techniques were utilized for classification.

The rest of the paper is organized as follows: Section II introduces briefly three machine learning techniques, the RF, the SVM and the KNN. Section III provides a brief description of the structure and workflow of the system. In Section IV, the maneuver detection unit is described with emphasis on the implementation of an adaptive sliding window. In section V, the structure and implementation of the driving maneuvers identification unit is presented. Evaluation of the performance of the two approaches is presented and compared in Section VI. The conclusions are drawn in Section VII.

II. Classical Machine Learning

A wide range of techniques have been developed to recognize and classify driving maneuvers in the literature. In recent years, driving maneuver classification using machine learning techniques has received increasing attention for the evaluation of driving patterns and drivers' profiling [4]. Three machine learning techniques have been used in this study, namely RF, SVM, and KNN, for recognizing and classifying driving maneuvers. These three techniques will be discussed briefly:

A. Random Forest Technique

A random forest classifier is an ensemble classifier that is made up of a set of decision trees trained on different sub-sets of the training data and then their predictions are aggregated to improve prediction accuracy and control over-fitting. An RF classifier usually uses bootstrap aggregation and boosting, in which random samples of the training dataset are selected with replacement and trained independently. The use of bagging and feature randomness to generate a set of decision tree classifiers typically results in high variance and low correlation. As a solution to this problem, these decision trees are usually connected in parallel and by using majority voting the variance is minimized and thus the prediction is improved. The implementation of RF classifier is summarized as follows: [26]:

1) Select M random samples from the labeled training set using the bootstrapping technique.

2) Construct a RF with N parallel decision trees.

3) Form N samples to train the N parallel decision tree models as follows:

a) For each feature x in a given feature set N_i calculate the Information gain from the entropy of the classes and the entropy of the feature x.

b) Find the node with the maximum information gain and split it into sub-nodes.

c) Iterate through a and b to form the tree until reaching the lowest amount of samples nedded to split.

4) Repeat steps (1) and (2) to get N tree classifiers.

5) For testing data, find the prediction of each decision tree, and allocate the new data to the category that wins the majority votes using the following formula:

$$P^*(x) = max_y \sum_{t=1}^{N} I\big(N_t(x) = P(x)\big) \qquad (1)$$

In the formula, $P^*(x)$ is the classification result of random forest, $N_t(x)$ is the classification result of each classification tree, $P(x)$ is a classification target, and $I(\cdot)$ is an indicator function which returns 1 if the condition in the argument is true, 0 otherwise.

B. Support Vector Machine

The main function of the SVM algorithm is to find the finest hyperplane in an N-dimensional space that separates the data and clusters them based on classes by using a kernel function. The SVM is in fact a binary classifier but can be extended to handle multi-class classification by training a series of binary SVMs or by solving a single optimization problem. A high classification rate can be achieved if the optimal selected hyperplane has the largest functional margin. This margin is represented by the distance of the hyperplane to the nearest training data points of any class. For the learning process of the SVM algorithm, constrained nonlinear optimization is used to obtain an optimal hyperplane. In general, a SVM classifier uses a nonlinear mapping function that maps the data into a high-dimensional feature space to distinctly classify the data points as follows:

$$P(x) = \sum_i \lambda_i K_i \langle x \cdot x_i \rangle + b \qquad (2)$$

Where, λ_i is support vector, x_i is data sample, i = 1, 2,…, C; C number of classes and $K_i\langle x \cdot x_i \rangle$ are a set of kernel functions defined by:

$$K_i\langle x \cdot x_i \rangle = \begin{cases} e^{-z_1 h(x)^2} & if\ x \in X^1 \\ \vdots & \vdots \\ e^{-z_C h(x)^2} & if\ x \in X^C \end{cases} \qquad (3)$$

In the above equation h(x) is a binary decision function expressed as:

$$h(x) = \sum_j \lambda_j y_i \langle x \cdot x_i \rangle + b \qquad (4)$$

While x_i is the i[th] sample of the training dataset, which includes the N number of samples with C categories and the value of the parameter z_i can be computed from the chi-square test [27]. The final classification decision is made according to a rule of the form:

$$P^*(x) = arg\,max_C(w_c K_i\langle x \cdot x_i\rangle + b) \quad (5)$$

The weighting factor appearing in Eqn. 8 is defined as:

$$w_i = \frac{N/n_i}{\sum_{i=1}^{C} N/n_i} \quad (6)$$

Where, N and C denote the training sample size and category size, respectively. n_i indicates the sample size of every category with i = 1, 2,…, C.

The implementation of the SVM classifier is summarized as follows:

1) Select M random samples from the labeled training set using the 5-fold technique and initialize the kernel matrix K_i.

2) For each sample x calculate:

a) Calculate the distance $h_j(x)$, {j = 1, 2, …., C}; C number of classes.

b) Calculate the value of the weighting factor wj and parameter z_j for every support vector.

3) Find P(x) from Eqn.(2).

4) Find the new Kernel matrix from P(x) and from the previous Kernel matrix.

5) Repeat steps (2) to (4) until finding the optimal hyperplane, i.e $h_j(x)$ with optimal functional margin.

6) For testing data, find the prediction from equation (4).

C. K-Nearest-Neighbors Technique

The KNN is a supervised machine learning algorithm for classifying classes based on their feature similarity to other classes. In the KNN the classification of a certain testing sample depends on its distance with respect to other samples in the training dataset. The distance between two samples is employed to measure their similarity [28]. The distance is calculated using different measures such as the Chebyshev distance, the Euclidean distance, and more generally the Minkowski distance. In this paper the Minkowski distance between two feature vectors is used. Where the Minkowski distance is a distance measured between two points in N-dimensional feature space by the following formula:

$$d(x_i - x_j) = \left(\sum_{i=1}^{C} |x_i - x_j|^p\right)^{\frac{1}{p}} \quad (7)$$

Where x_i and x_j are two features vectors and p is an integer value.

The implementation of the KNN classifier is summarized as follows:

1) Select M random samples from the labeled training set using the 5-fold technique.

2) Set the value of the nearest data points K which can be any integer preferably to be odd integer.

3) For every point in the testing data do the following:

a) Compute the distance between the test data and each sample in the training data as in Eqn(7).

b) Sort the distances obtained in (a)in an ascending order.

c) Select the first K rows from the sorted distances array.

d) Assign a class to the test point depending on most frequent class of these rows.

III. System Structure

Fig. 1, shows the general workflow of the proposed system. The system consists of four main interrelated units namely, data collection unit, data processing unit, maneuver recognition unit, and finally, maneuver classification unit. In this section, the functions of the first two units are briefly introduced. A detailed description of the operation of these units can be found in [29].

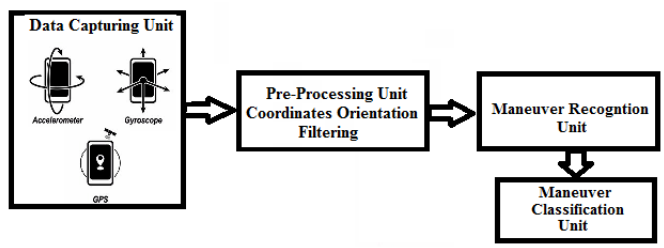


Fig. 1. The proposed system workflow.

Using calibrated Android smartphones with built-in accelerometers and gyroscopes, raw vehicle data was collected at a rate of 50 samples/second. The calibration method for the smartphones' IMUs sensors are adopted from [30]. As well as the data captured by the IMUs, the smartphones' GPS data was used for referencing the location of the vehicle.

The pre-processing unit is intended to achieve two main functions namely, signal filtering, and transformation of sensors data to the vehicle's coordinate system. The first problem is typically attributed to the fact that the IMUs in Smartphones are based on MEMS technology, thus they suffer from white Gaussian noise. Furthermore, the sensors are very sensitive hence they capture in addition to the variation of the dynamic parameters of the vehicle's vibration [30]. Fig. 2 shows instantaneous captured data for a sample maneuver. A locally weighted running line smoother (LOSS) filtering technique is used for removing this noise and smoothing the recorded signals. The use of this type of filter was investigated and its performance was compared with two other filters; the one-dimensional Kalman filter and the simple moving average filter. The LOSS filter was found to be the best effective filtering approach when compared with others, and Fig. 3 shows a sample of a smoothed signal [29-30].

A coordinate reorientation module is integrated with the pre-processing to correct the collected sensors' data by aligning the smartphone's coordinate system with the vehicle coordinate system. By presuming that the vehicle is driven on a horizontal road, during the initial calibration, the vehicle roll and pitch angles relative to a tangent frame both can be considered to be zero. Furthermore, if the vehicle does not experience any acceleration, the smartphone's roll and pitch angles can be estimated from accelerometer measurements of the gravity vector. This can be done using a set of geometrical rotations using Euler angles. The determination of Euler angles is fully explained in [31].

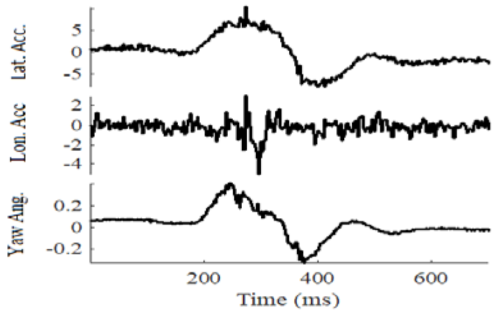


Fig. 2. Raw signals captured by the smartphone's IMUs.

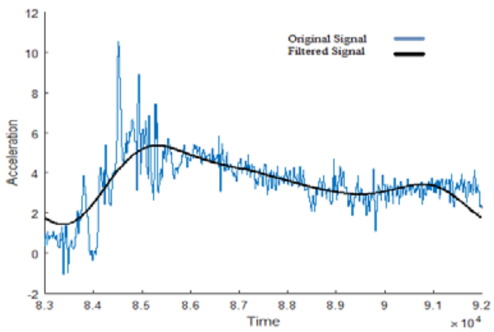


Fig. 3. Raw and filtered accelerometer signal.

IV. Maneuvers Detection

Table I, presents the list of maneuvers that can be detected by the proposed system. These maneuvers have been detected by an adaptive sliding window with a short-term energy endpoint detection algorithm.

TABLE I. Maneuvers Classes

1-	Acceleration straight road segment	2-	Braking straight road segment
3-	Left lane change straight road segment	4-	Right lane change straight road segment
5-	Merging into highway	6-	Exit from highway

Maneuvers are detected in three iterative stages. In the first stage a window of 100ms width is used to compute the short-term energy of the signal. Based on the fact that for an infinite sequence of a discrete signal the energy is defined by:

$$\hat{y}[n] = y[n]W[m-n], m - M + 1 \leq n \leq M \quad (8)$$

Where W is a window function given by:

$$W[m-n] = \begin{cases} 1 & 0 \leq m \leq M-1 \\ 0 & \text{Otherwise} \end{cases} \quad (9)$$

The energy contained in this short interval then can be computed by:

$$E_w = \sum_{n=m-M+1}^{m} (y[m]W[n-m])^2 \quad (10)$$

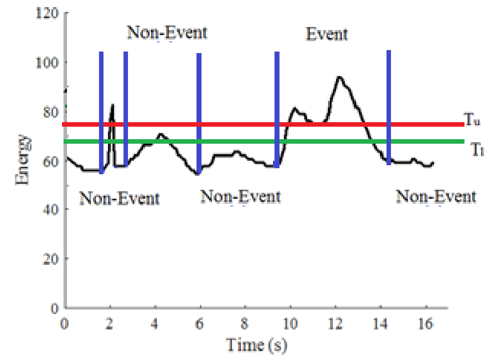


Fig. 4. Maneuver detection using short-term energy.

Once the short-energy is computed it will be compared with a set of pre-defined thresholds, as shown in Fig. 4. If this energy is less than a specific threshold T_l for the whole 100msec window, then this frame will be ignored and will be considered a non-event. Otherwise, if the energy is greater than T_l, then the starting time of the event detected is recorded and the short-term energy is computed for a sliding window as in Eq. (10). The width of the window will increase by 20msec and the short-term energy will be calculated over the whole interval of the extended window. For each step in this stage the following conditions will be checked:

- If the computed short-term energy remains less than the upper threshold T_u for 1 second or drops below T_l in a short time, then this segment will be considered a false event and the system will start with a new 100msec window as in the first stage.
- When the short-term energy for the extendable sliding window is computed to be higher than the upper threshold T_u for more than one second, then the system will consider this signal as a result of an event. If the system records the starting time of the event, it will continue to compute the short-term energy for the extendable sliding window and compare it with T_u. If the short-term energy drops below T_l for more than 100ms, the system will record the ending time of the event.

V. Maneuvers Identification

Generally, supervised machine learning techniques such as decision trees, support vector machines, neural networks, and many others are used to identify and classify types of driving maneuvers in a single process. All these techniques require a set of features to represent the input signals such as time, frequency or statistical features, for training and testing. In this study, sixteen time and statistical features listed in Table II were used to train and test the recognition and classification performance of the first approach. It should be noted that classical machine learning techniques, when trained with time and statistical features, cannot provide a clear description of how patterns of signals behave. In this regard, it is difficult to draw any conclusions from the parameters of the systems. Additionally, errors resulting from recognition and classification will accumulate and affect the performance of these techniques.

TABLE II. SELECTED STATISTICAL FEATURES

1	Mean	9	Peak to peak value
2	Median	10	Peak to RMS value
3	Maximum value	11	Root-sum-of-squares
4	Minimum value	12	Skewness
5	Standard of deviation	13	Kurtosis
6	Mode	14	Range of values
7	Variance	15	Interquartile range values
8	RMS value	16	The mean absolute deviation

Due to the fact that time-varying signals are required to recognize the types of driving maneuvers, it has been shown that pattern recognition or matching techniques, such as the DTW technique, are superior in this regard. DTW identifies the types of maneuvers by comparing input patterns against standard templates and calculating the similarity level between them. As a result of using the DTW method, incoming signals can be compared with a predefined standard template regardless of any differences in their amplitudes or durations. Therefore, it would be likely to have a set of standard templates to measure the similarity of maneuvers for different drivers [32-35]. It should be noted that the main disadvantage of the DTW approach is its extreme computational requirements, since it computes the similarity level between all the possible patterns in the input signals. In the case that multi-signal identification is required, this problem will become more complex. Furthermore, a considerable amount of work is required to select and compute the reference templates because it is very difficult to collect all possible templates that would cover all driving styles and behaviors of drivers [19].

Fig. 5 shows the basic structure of the DWT unit. The DTW technique utilizes discrete dynamic programming to determine the similarity between two signals, regardless of any difference in time, frequency, or deformation related effects to dynamic spatiotemporal differences. In a previous study [29], the recognition unit was implemented using (n × m) DTW units, where (n) represents the number of signals and (m) represents the number of standard patterns for each signal. As a consequence, for every detected event, i.e. driving maneuver, a (n × m) matrix containing warping cost is derived by comparing all the signals with all the stranded templates. This study has reduced the amount of computation required by the classical DTW technique by reducing the number of signals used to recognize driving maneuvers, as well as by utilizing energy activation units. In this study, the implementation of the DWT technique is based on two facts, which have been demonstrated in previous studies. The first is that there are only three signals, longitudinal acceleration, lateral acceleration, and yaw angle. The second fact is that the signals vary according to certain patterns, so their energy depends on these variations, see Table III.

When using the DWT technique to identify the type of any signal, a set of standard signals, or templates, are required to compare the unknown input signal with them and measure the similarity level. The selection of these reference signals for each specific class is not a straight forward task since the set of the collected signals, for each maneuver class, have different time durations and amplitudes. There are three different approaches in electing a suitable reference signal from a set of measurements namely; the longest common sequence approach, the medoid sequence approach, and the average sequence approach. In this paper, for a specific DWT unit, the signal that has the minimum average of distances with all the signals in that set is extracted and elected to be the reference signal or template. The details of this novel methodology are given in [35].

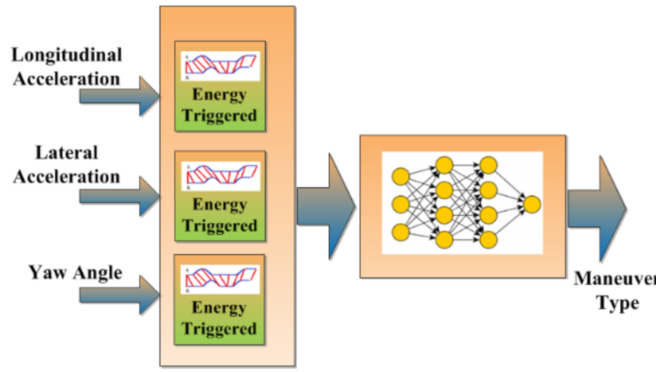
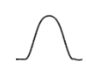
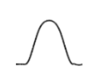


Fig. 5. Maneuvers recognition unit.

TABLE III. TYPICAL SIGNALS' PATTERNS FOR DIFFERENT MANEUVER

Maneuver Type	*Longitudinal Acceleration*	*Lateral Acceleration*	*Yaw Angle*
Acceleration			
Break			
Left-Lane Change			
Right-Lane Change			
Exit			
Merge			

VI. EXPERIMENTS AND RESULTS

In this work two different approaches were used to recognize and classify highway driving maneuvers. The first approach utilizes the classical machine techniques described in Section II, while in the second approach the DTW and the aforementioned techniques were integrated to create a hybrid system. With this system, the DTW method will be used for the recognition process while classical machine learning techniques will be applied for the classification process.

A. Experimental Data

Before exploring the analysis and results, it is worth mentioning that the development of the system progressed through two levels, the development level and the naturalistic driving testing level.

At the development level ten drivers with different types of vehicles and experience were volunteered to drive through a 16km highway road segment that has different configurations and conditions as shown in Fig. 6. Each driver was asked to execute the driving maneuvers listed in Table I with different categories; i.e. Light, Normal and Hard. All of the vehicles were equipped with smartphones that were programmed to collect sensor data at a rate of 50Hz and four cameras that recorded the surrounding vehicles. Every class of driving maneuver was performed by each driver at least five times, so the total number of driving maneuvers gathered in this phase was 900 samples. This part of the dataset was then presented to experts to obtain their judgment and to build the knowledge base that is required for labeling the maneuvers. This initial dataset was used to train and test the two suggested systems.

Fig. 6. The route used to collect initial dataset.

The 5-fold cross-validation technique was used from which 60% of the initial dataset were utilized to generate and extract the data and the features that are required in the computation of the DTW reference templates, define the lower and upper limits that define the range of values of each cluster, i.e. class and statistical features vectors for training the systems.

B. Models Evaluation

To assess the validation of the two approaches the remaining 40% of the initial dataset has been used to validate their performance.

The first assessment of the system was to test its capability to detect driving maneuvers, i.e. recording the starting and ending time. Fig. 7 shows a portion of a short trip conducted to cover some of the basic maneuvers. As shown in Fig. 7, these are the raw data that were captured directly from the calibrated smartphone’s sensors. Fig. 8 illustrates the pre-processed signals, i.e. after smoothing the signals of Fig. 7. As shown in the figure, the red rectangles represent the output of the maneuver detection unit. As it can be seen, the unit effectively detects the beginning and the end of any variation in the input signals. According to the testing of maneuver detection unit with manually registered maneuvers, the detection rate was more than 96%.

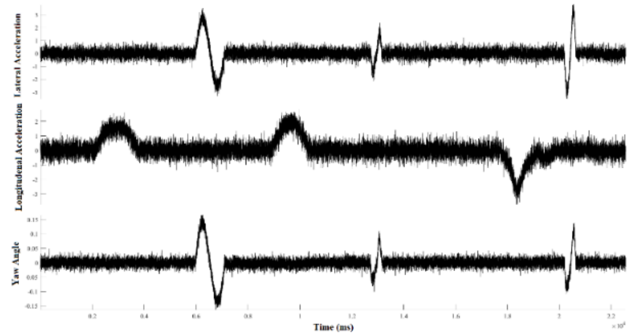


Fig. 7. Raw data captured from the smartphone’s sensors.

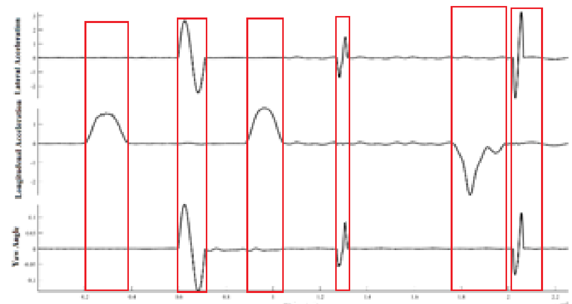


Fig. 8. Signals after filtration and maneuver detection.

Three evaluation metrics namely Precision (PR), Recall (RC), and F1-score (F1) have been used for evaluating the performance of each system in addition to the confusion matrix. Precision is generally defined as the probability that a certain class of maneuvers is correctly classified in either recognition or classification results. In contrast, recall is the probability that all maneuvers in a particular maneuver class are correctly identified. Finally, the F1-score is determined based on both precision and recall, as shown in Eq. (11), where a high F1-score indicates the system's overall performance quality.

$$F1 = \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \quad (11)$$

Where TP is true positive, FP is false positive and FN is false negative. All these three values can be found using the confusion matrix.

In the first approach, namely the three classical machine learning models, all the statistical features listed in Table III were obtained for each segmented maneuver. It should be noted that the models are performing both the recognition and classification processes. The confusion matrix for the RF model is shown in Fig. 9, and Table IV presents a comparison for each maneuver of the three models in terms of PR, RC and F1.

As it can be seen from Table IV, the performance of the RF model is the highest, where the average precision of the model is 0.84, the recall is 0.833 and the F1 score is 0.835. For the SVM the parameters are (PR = 0.783, RC = 0.772, F1 = 0.775) and for the KNN they are (PR = 0.749, RC = 0.736, F1 = 0.74). It is not an easy task to dig for the actual factors behind the low performance of the models when compared with the RF.

However, both SVM and KNN are not efficient algorithms when they deal with large data sets, and they do not function well when the target classes are overlapped. The RF model is able to handle large datasets because it is based on the bagging algorithm which generates as many trees as possible based on the testing data and generates an output combining the tree outputs. Therefore, the RF techniques can be considered as an ensemble learning approach, hence it would reduce the overfitting problem in decision trees, reduces the variance and improves the accuracy.

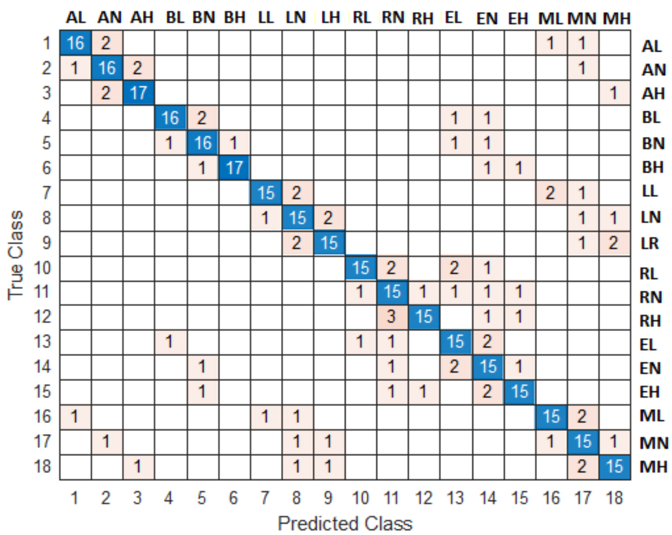


Fig. 9. Confusion matrix for the classical RF implementation.

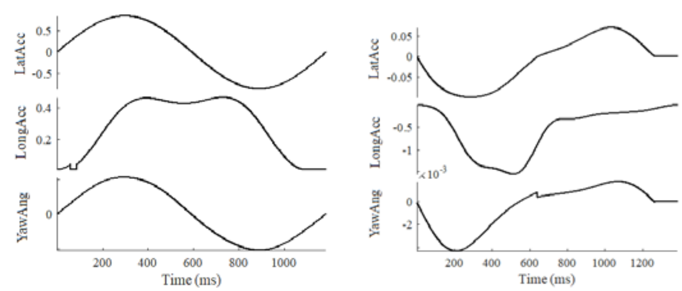
TABLE IV. COMPARISON OF THE MODELS FIRST APPROACH

	RF			SVM			KNN		
Class	***PR***	***RC***	***F1***	***PR***	***RC***	***F1***	***PR***	***RC***	***F1***
AL	0.85	0.85	0.85	0.89	0.80	0.84	0.83	0.75	0.79
AN	0.81	0.85	0.83	0.76	0.80	0.78	0.71	0.75	0.73
AH	0.94	0.85	0.89	0.85	0.85	0.85	0.84	0.80	0.82
BL	0.89	0.85	0.87	0.89	0.80	0.84	0.88	0.75	0.81
BN	0.77	0.85	0.81	0.76	0.80	0.78	0.71	0.75	0.73
BH	0.95	0.90	0.92	0.94	0.85	0.89	0.88	0.75	0.81
LL	0.89	0.85	0.87	0.88	0.75	0.81	0.83	0.75	0.79
LN	0.84	0.80	0.82	0.68	0.75	0.71	0.67	0.70	0.68
LH	0.89	0.85	0.87	0.79	0.75	0.77	0.78	0.70	0.74
RL	0.89	0.80	0.84	0.88	0.75	0.81	0.83	0.75	0.79
RN	0.76	0.80	0.78	0.65	0.75	0.70	0.68	0.75	0.71
RH	0.89	0.80	0.84	0.88	0.75	0.81	0.82	0.70	0.76
EL	0.81	0.85	0.83	0.68	0.75	0.71	0.70	0.70	0.70
EN	0.67	0.80	0.73	0.60	0.75	0.67	0.54	0.75	0.63
EH	0.89	0.80	0.84	0.79	0.75	0.77	0.70	0.70	0.70
ML	0.85	0.85	0.85	0.79	0.75	0.77	0.79	0.75	0.77
MN	0.70	0.80	0.74	0.63	0.75	0.68	0.56	0.70	0.62
MH	0.81	0.85	0.83	0.75	0.75	0.75	0.71	0.75	0.73

It should be mentioned here that a thorough analysis has been conducted in this study to identify the overlap in the target classes. It was found that there are two groups of maneuvers which could have a high similarity rate between their classes. The first group contains the Acceleration, Left-Lane change and the Merging maneuvers and the second group contains the other three maneuver classes. Fig. 10(a) illustrates a signal that was manually recorded as a left-lane change, while the system recognized it as a merging maneuver. On the other hand, Fig. 10(b) shows a break maneuver but has been recognized by the system as an exit maneuver. From the point of view of the author, this noise in the dataset needs careful analysis, hence it will be left to a future investigation.

In the second approach the same 5-fold cross-validation method was used to extract the DTW reference templates and again to train and validate the same models but for a specific maneuver type. As mentioned previously in this approach the DTW unit is acting as a recognition unit while the three classical machine learning models are acting as classifiers.

The performance of the DTW was first tested and it was found that the structure of the unit needs some modification to overcome the problem of overlapping classes. A simple two-hidden layers neural network was integrated into the unit, where the three measured distances obtained from each DTW are fed as an input to this neural network. Fig. 11 shows the confusion matrix for the predicted maneuvers. All the performance measures, precision, recall and F1-score were calculated for the recognition unit and they are equal to 0.95, which indicates an excellent validity of the recognition unit.



(a): LLC recognized merge. (b): Break recognized exit.

Fig. 10. Examples for misrecognized classes.

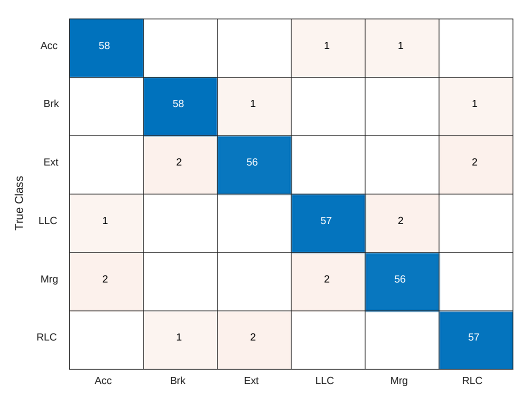
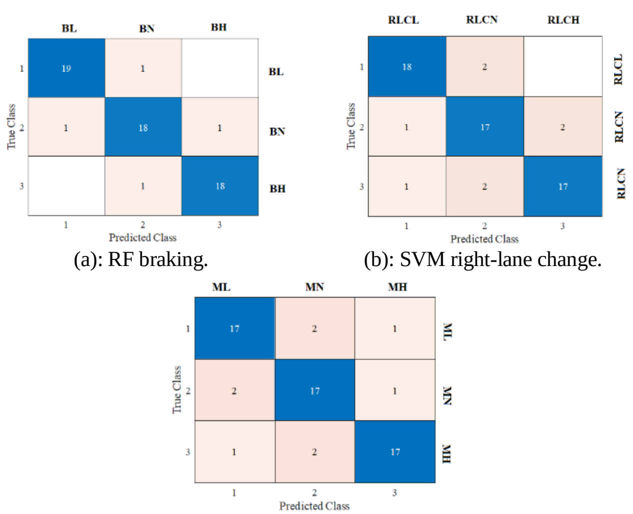


Fig. 11. Confusion matrix for the recognition unit.

Table V presents a comparison between the three models that perform a classification process for each maneuver separately. Fig. 12 shows samples of the confusion matrix for different cases. Again the performance of the RF model is the highest when compared with the others and still the KNN model has the lowest performance. The average precision of the RF model is 0.908, the recall is 0.905 and the F1 score is 0.91. These newly obtained results indicate an enhancement of 9% is achieved when using the second approach. Similar improvements were also noticed in the other models, where for the SVM the performance indicators are PR = 0.875, RC = 0.871, F1 = 0.87 and an average enhancement of 12.25%, while the performance indicators for the KNN model are PR = 0.838, RC=0.835, F1 = 0.84 and an average enhancement of 13.5%.



(a): RF braking. (b): SVM right-lane change.

(c): KNN merge.

Fig. 12. Confusion matrices samples for hybrid system initial dataset.

TABLE V. COMPARISON OF THE MODELS SECOND APPROACH

	RF			SVM			KNN		
Class	***PR***	***RC***	***F1***	***Class***	***PR***	***RC***	***F1***	***Class***	***PR***
AL	0.95	0.90	0.92	0.89	0.89	0.89	0.85	0.89	0.87
AN	0.86	0.90	0.88	0.81	0.85	0.83	0.80	0.80	0.80
AH	0.95	0.95	0.95	0.95	0.90	0.92	0.89	0.85	0.87
BL	0.95	0.95	0.95	0.90	0.90	0.90	0.86	0.90	0.88
BN	0.90	0.90	0.90	0.81	0.85	0.83	0.80	0.80	0.80
BH	0.95	0.95	0.95	0.95	0.90	0.92	0.89	0.85	0.87
LL	0.95	0.90	0.92	0.90	0.90	0.90	0.85	0.85	0.85
LN	0.82	0.90	0.86	0.81	0.85	0.83	0.80	0.80	0.80
LH	0.95	0.90	0.92	0.89	0.85	0.87	0.85	0.85	0.85
RL	0.95	0.95	0.95	0.90	0.90	0.90	0.85	0.85	0.85
RN	0.86	0.90	0.88	0.81	0.85	0.83	0.76	0.80	0.78
RH	0.95	0.90	0.92	0.89	0.85	0.87	0.89	0.85	0.87
EL	0.90	0.90	0.90	0.89	0.85	0.87	0.84	0.80	0.82
EN	0.81	0.85	0.83	0.74	0.85	0.79	0.70	0.80	0.74
EH	0.95	0.90	0.92	0.94	0.85	0.89	0.89	0.80	0.84
ML	0.95	0.90	0.92	0.90	0.90	0.90	0.85	0.85	0.85
MN	0.78	0.90	0.84	0.81	0.85	0.83	0.81	0.85	0.83
MH	0.94	0.85	0.89	0.95	0.90	0.92	0.89	0.85	0.87

C. Naturalistic Driving Testing

In the second stage of this study, a comprehensive dataset was collected by installing only the data collection app onto the smartphones of 25 drivers who drove frequently to and from various locations and the University of Nizwa, as shown in Fig. 13. Data collected in this phase is real naturalistic driving data based on different routes that are very dynamic and include many different types of roads.

Fig. 13. The routes used for collecting the naturalistic dataset.

After performing the necessary preprocessing for the data of each driver, the captured signals were analyzed by an off-line Matlab code to detect and extract driving actions by using the adaptive sliding window described in Section V. Table VI provides a list of the number of driving events obtained in this phase.

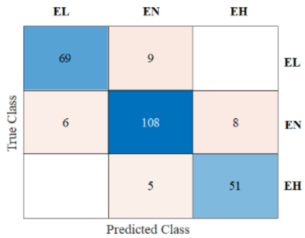
TABLE VI. COMPARISON OF THE MODELS SECOND APPROACH

Type of Maneuver	#	Type of Maneuver	#
Acceleration	680	Right-Lane Change	495
Breaking	595	Exit	256
Left-Lane Change	508	Merge	255

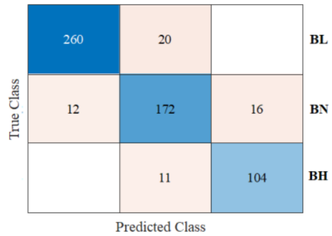
As a result of the high number of maneuvers obtained from smartphone sensors, a separate module was developed to label the maneuvers in addition to the suggested system. The module uses a semi-supervised labeling system based on the DTW technique. The module is similar to the DTW recognition unit with the exception that it was specifically designed to identify maneuver classes. The only difference between the two systems is that there are nine DTW units and each one is devoted to a single class of a certain maneuver. A detailed explanation of the implementation of this technique can be found in [36]. The distance between two time series signals is given by:

$$Distance(A,B) = \frac{DTW(A,B)}{ED(A,B)+\delta} \qquad (12)$$

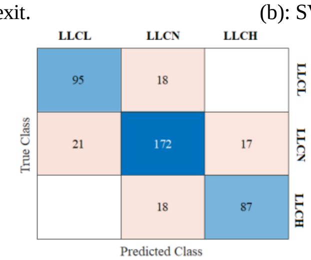
Where A is a standard reference signal, or template used by the DTW and Euclidian distance calculation, B is the signal that needs to be classified, DTW(A, B) is the distance measured by the classical constrained DTW algorithm, ED(A, B) is the classical Euclidian distance and d is an extremely small positive quantity used to avoid divide-by-zero error.



(a): RF exit.



(b): SVM break.



(c): KNN left-lane change.

Fig. 14. Confusion matrices samples for hybrid system naturalistic dataset.

Fig. 14 shows the confusion matrix for different cases. As it has been expected the RF model has the highest performance with respect to the SVM and KNN, while the KNN model is still showing the lowest performance. The average precision, recall and F1-score are all approximately 0.9, those for the SVM are 0.87 and finally those for the KNN are 0.834. As it can be seen, the results are almost the same for both datasets and this gives a positive indication that the suggested approach is stable and reliable.

VII. Conclusions

Two different approaches are presented in this paper for the recognition and classification of highway driving maneuvers using smartphone sensors. Raw data captured through smartphone's IMUs sensors are first pre-processed by transforming sensors' data from the smartphone's coordinates system to the actual vehicle coordinates system, then these data were smoothed by using the LOSS filter and finally, the longitudinal and lateral acceleration and the yaw angle are deduced from these data. Three parameters were found to be sufficient to recognize and classify driving maneuvers.

The first approach investigated in this paper utilizes three different classical machine learning techniques, namely RF, SVM and KNN techniques. Results obtained from this approach showed that RF had the highest performance when compared to SVM and KNN. This superiority of the RF model can be attributed to the fact that the RF model can handle large datasets efficiently. It's based on the bagging algorithm and uses the Ensemble Learning technique. Nevertheless, it was found that the classical implementation of machine learning techniques suffers from a serious problem in dealing with noisy data, i.e. overlapping in the target classes. It was found that there are two groups of maneuvers which could have a high similarity rate between their classes. The first group contains the Acceleration, Left-Lane change and the Merging maneuvers and the second group contains the other three maneuver classes.

In this paper, a hybrid technique is used to overcome the overlapping between the classes. The recognition unit of this approach utilizes a novel DTW unit that demonstrates an excellent recognition rate with F1-Score of 0.95. The maneuver classifications are then obtained by machine learning techniques. When compared to the classical approach, the performance of the novel approach was significantly improved.

A large dataset was collected from naturalistic driving for 25 drivers on different highways. About 2800 maneuvers were obtained from this dataset. With such a high number of maneuvers a semi-supervised labeling system based on the DTW technique was used. The module is similar to the DTW recognition unit but was trained solely for labeling maneuver classes. The second approach was tested on the second dataset. Results obtained show a high rate of recognition and classification, nearly the same as that obtained with the first dataset.

References

[1] A. Haghi, D. Ketabi, M. Ghanbari and H. Rajabi, "Assessment of Human Errors in Driving Accidents; Analysis of the Causes Based on Aberrant Behaviors", Life Science Journal, vol 11, No. 9, pp. 414-420, 2014.

[2] Muhammad Qasim Khan and Sukhan Lee, "A Comprehensive Survey of Driving Monitoring and Assistance Systems", Sensors, vol. 19, issue 11, pp. 2574-2606, 2019.

[3] Clara Marina Martinez, Mira Heucke, Fei:Yue Wang, Bo Gao, Dongpu Cao, "Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey", IEEE Transactions on Intelligent Transportation Systems, vol. 19, issue 3, pp. 666–676, 2018.

[4] Dengfeng Zhao, Yudong Zhong, Zhijun Fu, Junjian Hou, Mingyuan Zhao, "A Review for the Driving Behavior Recognition Methods Based on Vehicle Multisensor Information", Journal of Advanced Transportation, vol. 2022, pp. 1-16, 2022.

[5] R. Kridalukmana, H. Y. Lu and M. Naderpour, "An object oriented Bayesian network approach for unsafe driving maneuvers prevention system", in the 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 1-6, 2017.

[6] Iván Silva and JoséEugenio Naranjo, "A Systematic Methodology to Evaluate Prediction Models for Driving Style Classification", Sensors, vol. 20, issue 6, pp. 1692-1713, 2020.

[7] H. Malik, G. S. Larue, A. Rakotonirainy and F. Maire, "Fuzzy Logic to Evaluate Driving Maneuvers: An Integrated Approach to Improve Training," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 1728-1735, 2015.

[8] T. K. Chan, C. S. Chin, H. Chen and X. Zhong, "A Comprehensive Review of Driver Behavior Analysis Utilizing Smartphones," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 10, 4444-4475, 2020.

[9] N. AbuAli, "Advanced vehicular sensing of road artifacts and driver behavior," the 2015 IEEE Symposium on Computers and Communication (ISCC), pp. 45-49, 2015.

[10] L. M. Bergasa, D. Almería, J. Almazán, J. J. Yebes and R. Arroyo, "DriveSafe: An app for alerting inattentive drivers and scoring driving behaviors" the 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 240-245, 2014.

[11] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya and M. C. González, "Safe Driving Using Mobile Phones," IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 3, pp. 1462-1468, pp. 1462:1468, 2012.

[12] Yang Zheng, A. Sathyanarayana and J. H. L. Hansen, "Threshold based decision-tree for automatic driving maneuver recognition using CAN-Bus signal," the 2014 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 834-2839, 2014.

[13] T. Pholprasit, W. Choochaiwattana and C. Saiprasert, "A comparison of driving behaviour prediction algorithm using multi-sensory data on a smartphone," the 2015 IEEE/ACIS 16th International Conference on

Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp. 1-6, 2015.

[14] Hamid Reza Eftekhari and Mehdi Ghatee, “Hybrid of discrete wavelet transform and adaptive neuro fuzzy inference system for overall driving behavior recognition”, Transportation Research Part F: Traffic Psychology and Behaviour, vol. 58, pp. 782-796, 2018.

[15] Hamid Reza Eftekhari, Mehdi Ghatee,: “A similarity:based neuro:fuzzy modelling for driving behavior recognition applying fusion of smartphone sensors”, Journal of Intelligent Transportation Systems, vol. 23, no. 1, pp. 72-83, 2019.

[16] C. Arroyo, L. M. Bergasa and E. Romera, "Adaptive fuzzy classifier to detect driving events from the inertial sensors of a smartphone," the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 1896-1901, 2016.

[17] A. Aljaafreh, N. Alshabatat and M. S. Najim Al-Din, "Driving style recognition using fuzzy logic," the 2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012), pp. 460-463l, 2012.

[18] Munaf Najim Al:Din, Ahmad Aljaafreh, Nashat Albdour and Maen Saleh:, “Driving Styles Recognition Using Decomposed Fuzzy Logic System”, International Journal of Electrical, Electronics and Computer Systems, vol. 16, issue 1, pp. 820-824, 2013.

[19] Munaf S. Najim Al-Din. “Real-Time Identification and Classification of Driving Maneuvers using Smartphone.” Advances in Science, Technology and Engineering Systems Journal, vol. 5, pp. 193-205, 2020.

[20] Wu, M.; Zhang, S.; Dong, Y., “A Novel Model-Based Driving Behavior Recognition System Using Motion Sensors”, Sensors, vol. 16, issue 10, pp. 1746-1769, 2016.

[21] Chen Chen, 1 Xiaohua Zhao, "Driver’s Eco-Driving Behavior Evaluation Modeling Based on Driving Events”, Journal of Advanced Transportation, vol. 2018, pp. 1-12, 2018.

[22] J. Yu, Z. Chen, Y. Zhu, Y. Chen, L. Kong and M. Li, "Fine-Grained Abnormal Driving Behaviors Detection and Identification with Smartphones," IEEE Transactions on Mobile Computing, vol. 16, no.8, pp. 2198-2212, 2017.

[23] J. F. J´unior, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, and G. Pessin, “Driver behavior profiling: An investigation with different smartphone sensors and machine learning”, PLOS one, vol. 12, no. 4, e0174959, 2017.

[24] P. Brombacher, J. Masino, M. Frey and F. Gauterin, "Driving event detection and driving style classification using artificial neural networks," the 2017 IEEE International Conference on Industrial Technology (ICIT), pp. 997-1002, 2017.

[25] U. Fugiglando et al., "Driving Behavior Analysis through CAN Bus Data in an Uncontrolled Environment," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 2, pp. 737-748, 2019.

[26] Breiman, L,. “Random Forests:, Machine Learning, vol. 45, pp. 5–32, 2001.

[27] Ketu, S., Mishra, P.K., “Scalable kernel-based SVM classification algorithm on imbalance air quality data for proficient healthcare”, Complex Intell. Syst. vol. 7, pp. 2597–2615, 2021.

[28] S. Ray, "A Quick Review of Machine Learning Algorithms", the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 35-39, 2019.

[29] Munaf Salim Najim Al:Din; Atef Saleh Al:Mashakbeh, “Development of a highway driving events identification and classification using smartphone”, International Journal of Nanoparticles, vol. 12, issue 1-2, pp. 152-173, 2020.

[30] M. S. Najim Al:Din, "Calibration and Pre-Processing Techniques for a Smartphone:Based Driving Events Identification and Classification System," the 2018 IEEE Electron Devices Kolkata Conference (EDKCON), pp. 396-402, 2018.

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

[32] Saiprasert, C., Pholprasit, T. and Thajchayapong, S., "Detection of Driving Events using Sensory Data on Smartphone", International Journal of Intelligent Transportation Systems Research, vol. 15, no. 1, pp. 17-28, 2015.

[33] Gurdit Singh, Divya Bansal and Sanjeev Sofat, “A smartphone based technique to monitor driving behavior using DTW and crowdsensing”, Pervasive and Mobile Computing, vol. 40, pp. 56-70, 2017.

[34] Pavlo Tkachenko, Jinwei Zhou, Davide Gagliardi and Luigidel Re, "On:line Maneuver Identification in Highway Traffic Using Elastic Template Matching”, IFAC:PapersOnLine, vol. 51, no. 15, pp. 557-562, 2018.

[35] F. Petitjean, J. Inglada, and P. Gancarski, “A global averaging method for dynamic time warping, with applications to clustering,” Pattern Recogn., vol. 44, no. 3, pp. 678–693, 2011.

[36] Yanping Chen, Bing Hu, Eamonn Keogh, and Gustavo E.A.P.A Batista., “DTW-D: time series semi-supervised learning from a single example”, In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '13), pp. 383–391, 2013.