IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 October 2022, accepted 20 October 2022, date of publication 26 October 2022, date of current version 4 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3217644

RESEARCH ARTICLE

Safe Deep Driving Behavior Detection (S3D)

EHSAN KHOSRAVI[1], ALI MOHAMMAD AFSHIN HEMMATYAR[1], MAHDI JAFARI SIAVOSHANI[1], AND BEHZAD MOSHIRI[2]

[1]Department of Computer Engineering, Sharif University of Technology, Tehran 11155-9517, Iran
[2]School of Electrical and Computer Engineering, College of Engineering, Tehran University, Tehran 14395-515, Iran

Corresponding author: Ali Mohammad Afshin Hemmatyar (hemmatyar@sharif.edu)

ABSTRACT The human factor is one of the most critical parameters in car accidents and even traffic occurrences. Driving style affected by human factors comprises driving events (maneuvers) and driver behaviors. Driving event detection is the fundamental step of identifying driving style and facilitates predicting potentially unsafe behaviors, preventing accidents, and imposing restrictions on high-risk drivers. This paper proposes a deep hybrid model to detect safe driver behaviors and driving events using real-time smartphone sensor signals. The ensemble of Multi-layer Perceptron, Support-Vector Machine, and Convolutional Neural Network classifiers process each driving event sample. In order to evaluate our model, we develop an Android Application to capture smartphone sensor signal data. We capture about 24000 driving data from 50 drivers. Results indicate that the fusion model performs better than each individual classifier in terms of Accuracy, False Positive Rate (FPR), and Specificity (96.75, 0.004, and 0.996). This research gives insights to Auto-mobile developers to focus on the speed and cost efficiency of smartphone driver monitoring platforms. Although some insurance and freight management companies utilize smartphones as their monitoring platforms, the market share of these use cases is meager and could improve rapidly with the promotion of new smartphones with better processing and storage.

INDEX TERMS Convolutional neural network, driver behavior, driving event, driving style, multi-layer perceptron, support-vector machine.

I. INTRODUCTION

Road accidents cause around 1.35 million fatalities and up to 50 million injuries yearly globally. Traffic-related mortality and injury cost the global economy around 518 billion dollars annually [1]. Although road quality, weather conditions, and vehicle performance all play a crucial role in accidents, human behavioral factors significantly impact the vast majority of accidents [2], [3]. Studies show that in 95% of accidents, the human factor is critical and driver behavior is recognized as the most important factor [4]. Specifically, in the United States, according to a 2016 study, it was found that only 7.27 million car accidents occurred, resulting in 37,914 deaths and 2.17 million injuries in which human factors had the most effect (about 94%) [5].

Many platforms can assist drivers in making safe trips and decrease road fatalities. One of these platforms is the Advanced Driving Assistance System (ADAS). Rather than monitoring driver driving style, ADAS focuses on assisting drivers. These systems could protect many lives after accidents but could not prevent accident from taking place [6], [7]. Due to the expensive price, ADAS implementation has been limited.

Managers in freight management companies utilized real-time, continuous, and automated driver behavior profiling to institutionalize campaigns to improve drivers' scores, decrease the accident rate, increase the resource-based economy, and extend vehicle lifetime warranties. Moreover, to prevent accidents, we could notify drivers of aggressive driving events in real-time. For example, a smartphone app could warn when the driver makes an aggressive U-turn [8].

Considering all of the aforementioned aspects, driver behavior should be regarded as one of the most vital areas for improving road safety. In addition, driver behavior could impact the decrease in fuel consumption and greenhouse gas emissions, which is the objective of Green Intelligent Transportation Systems.

The associate editor coordinating the review of this manuscript and approving it for publication was Huiyan Zhang.

Researchers used different solutions to evaluate their driving event detection and behavior models. Some of them used the Driver Behavior Questionnaire to investigate the effect of safety skills on driver behaviors [9]. These questionnaires are based on drivers' self-statements about their previous driving behaviors and experiences. However, they did not give insights into real-time driving events. In order to cover this limitation, in-vehicle black box systems were introduced to capture sensor data and driver behaviors simultaneously. Richard et al. [10] used SHRP2 in-vehicle sensor data to examine real-time driver speeding behavior which primarily comprised video images of the front and rear windshields, rate of acceleration, and some sensor data. Nevertheless, SHRP2 had its limitations, such as Gyroscope and Linear Accelerator sensors were not used in this dataset, and there was no real-time labeling mechanism for driving events and driver behaviors (except for reporting emergencies). Wu and Xu [11] also analyzed SHRP2 real-time data to detect Right Turn driving events.

Previous research [12], [13], [14] demonstrated that smartphone sensor data that had been properly preprocessed and managed for driver behavior monitoring was a valuable alternative to traditional black boxes. Thus, phones offered more precise sensor data for driver profile analysis [8]. MEMS gained popularity over OBD ports because of their small size, low weight, and low energy consumption. Therefore, it was simple to integrate various sensor units such as Magnetometers, Gyroscopes, and Accelerometers [15]. Collecting data from a car interior bus over CAN using an OBD connection was dependent on the vehicle's connection protocol and posed a hazard to human and vehicle security [15]. Therefore, MEMS was better suitable for driving event detection [7]. Although to the authors' best knowledge, there is a gap in the literature to not providing a comprehensive dataset of all possible signals of smartphones.

According to the reasons mentioned above, we used smartphone sensor signals to analyze driver behaviors because of their ubiquitousness, embedded set of comprehensive sensors, and low energy consumption. Also, other researchers of Intelligent Transportation Systems used smartphone data to evaluate driver styles [16]. White et al. [17] detected road conditions using the data generated by smartphones, and Dey et al. [18] used them for traffic monitoring. Soares et al. [19] used them for traveling mode detection; for the same purpose, Li et al. [20] used a couple of Generative Adversarial Network and Convolutional Neural Network (CNN); Ramanujam et al. [21] reviewed Deep Learning Methods for recognizing human activity such as cleaning the kitchen, washing clothes, cooking, cycling, climbing upstairs and down, jogging, running, lying, jumping, and walking.

Recent research uses Deep Learning models due to their promising performance in feature extraction which outperforms the handcraft methods and their success in identifying sensor-based behaviors. These benefits make researchers use various kinds of Neural Networks, including Recurrent Neural Networks (RNNs) and CNNs [22]. New improvements to CNN [23] and RNN models with the introduction of new large-scale data make these approaches good alternatives to traditional methods [24]. Yu et al. [25] collected data on the driving habits of 20 volunteers. They trained a Neural Network with this data and achieved more accuracy than an SVM. Also, Sarker et al. [7] utilized collected data from Magnetometer, Accelerometer, and Gyroscope installed in smartphones to detect aggressive driving events. They developed an LSTM model to classify driving events. To decrease Mean Per Class Error, the authors used specific threshold rules for each sensor data (introduced as domain knowledge).

Recent research indicates some gaps in the driving event and driver behavior recognition literature as follows (more details are given in Section II):

1. There is no comprehensive dataset of all possible smartphone sensor signals to detect driving events and driver behaviors.

2. In the data capturing phase, real-time ground truth driving events labeling is not used and researchers usually prefer to tolerate offline labeling errors.

3. The previous works did not fuse multiple machine learning classifiers to evaluate the driving style. Thus certainty level of their proposed models may be questionable.

In this manuscript, we utilized the ensemble learning method to fuse different deep learning and traditional classifiers to cover these gaps. An ensemble of classifiers processes each input sample and combines the results according to a rule. In our situation, the outputs of the fundamental classifiers are mixed by a majority vote which indicates that a sample window of the sensor signal is allocated to a specific class if at least fifty percent of the classifiers produced that class. Other combination strategies, such as the variant of weighted voting [26], have been proposed in the literature, but they were deemed inadequate for the present application. Another reason for using ensemble voting models is about certainty level. Because all Advanced Driving Assistance Systems are life-critical systems, the certainty of driving event detection is crucial. Therefore, we used three different models and a majority voting ensemble method in order to have more certainty [27], [28].

Following are the main contributions of the manuscript:

1) We proposed a new structure of multi-classifier fusion (hybrid CNN, SVM, and MLP model) to get better accuracy in detecting driving events.
2) We developed an aggregate model by using ensemble learning (majority voting).
3) We developed an Android application to collect smartphone sensor signals for detecting driving events.
4) Collect a comprehensive labeled dataset consisting of 24000 samples of ten different driving events from fifty drivers to distinguish between different types of safe and aggressive driving events. Data of 21 sensor signals are collected and a Police Officer simultaneously labels the data during the experiments.

Also, our research questions are as follows:

1) Is it possible to use smartphone signals to detect driving events and driver behaviors?
2) Does it make more information to divide the safe driving events into fine-grained detailed classes? Or cumulate all non-aggressive driving events as safe?
3) If other than Accelerometer and Gyroscope sensors are needed to detect driving events?
4) Is this important to have a coarse-grained set of sensors? Furthermore, what is the fine-grained set?

The remainder of this manuscript is structured as follows: Section II includes a thorough review of research about driving events and driver behavior recognition. Section III provides details of the proposed hybrid SVM-CNN-MLP model. Data collecting is explained in section IV. Results of the experimental investigation are discussed in section V. Finally, in section VI, the paper's conclusion is provided.

II. LITERATURE REVIEW

Various classification methods such as Random Forest [29], Fuzzy Modeling [30], Discrete Wavelet Transform [31], Dynamic Time Warping [32], and machine learning algorithms such as K-Means [33], K-Nearest Neighbor (KNN) [34], Hidden Markov Model (HMM) [35], Support Vector Machine (SVM) [36], and Neural Network [37] are used to classify data.

Increasingly, Deep Learning algorithms have grown popular in recent years. High accuracy and ability to process big data are the most important features of using these methods. Since the data recorded by driver behavior (via smartphone sensors) are time-series big data, the use of Deep Learning-based approaches can be very accurate. Deep and reinforcing learning methods are used in various areas of transportation systems such as predicting Macroscopic Traffic Congestion [38], [39], [40], [41], [42], Transportation System Planning [38], [39], [43], customer demand forecast for transportation [43], [44], [45], [46], traffic monitoring and congestion detection [47], [48], [49], [50], predicting driver behavior [51], [52], [53], [54], [55], [56], detecting driver, and classification of vehicle types [57]. Representing complicated nonlinear connections between related and dependent variables is the primary benefit of Deep Learning architecture over standard statistical approaches (integrating hierarchical and distributed features) [58].

Different Deep Learning architectures were provided to drive event detection using time-series smartphone sensor signals. The vanishing and exploding gradient problem of RNNs made researchers provide new architecture called LSTM to solve these problems. An unsupervised LSTM autoencoder was implemented by Sarker et al. [59] to learn the encoded feature vector. Also, the authors proposed a supervised LSTM classier of the labeled encoded feature vector to classify the driving events. Their results indicated that the supervised model outperformed other existing models that they considered. Although they provided an innovative method to analyze unlabeled data, their domain-specific knowledge was similar to traditional threshold-based techniques in signal processing. They declared some handy crafted thresholds on input signals. Also, their proposed model did not provide detailed information on the fine-grained classification of Safe driving events and the best performance model converged after 500 epochs (two hours). The authors planned to do more experiments to optimize the amount of data used to improve their model's overall accuracy.

Sarker et al. [7] also proposed another threshold-based feature extraction method that analyzed the motion sensor data changes and detected the driving events. The proposed LSTM classification model utilized the physical model of a moving vehicle in order to improve its performance. In this research, driving events were identified based on the labeler's danger perception. The authors suggested further studies on the procedure of label assignation in order to reduce potential bias.

Carvalho et al. [8] analyzed different RNN architectures (Simple RNN, GRU, and LSTM) using an Accelerometer sensor signal in order to detect the driving events. GRU model reached its best performance after 901 epochs. The authors suggested further studies on low-cost, high-performance, collaborative sensing solutions.

Researchers compared the performance of Different RNN architectures and the Random Forest model [29]. In this research, they proposed a safe driving event classifier back to the back of an event-type classifier. The best model accuracy was 95 percent. By dividing the number of positive changes by the number of time steps, the authors proposed a danger score. However, the proposed model could not distinguish between Right or Left Lane Changes, so the authors combined both events and evaluated the Lane Change driving events detection performance. Also, the used window size (between 1 and 10 seconds) might cause the occurrence of two events in window size. The authors suggested further studies incorporating events like gaze, head position, and blinking.

Recent reviews of various articles on driving event detection using driver behavior lead us to conclude that there are still gaps in the literature regarding how to improve the performance of CNNs in detecting driving events and driver behavior. Most Deep Neural Networks (i.e., CNNs, RNNs, and DBNs) used to recognize human behaviors are built on time series data from sensors with significant spatial and temporal couplings [22]. We have proposed a hybrid model to improve the performance of driving event detection. This model is explained in part C of section III. We compare evaluation criteria, limitations, advantages, model, dataset properties, device, and sensor signals of previous research in Table 1 and Table 2. These tables also provide details of our proposed model and dataset.

III. MATERIAL AND METHODS

As we mentioned above, the globally high rate of road mortalities and injuries made the researchers work on improving

TABLE 1. Related works limitations and advantages.

Reference	Evaluation Criteria	Limitation	Advantage
Proposed Model	Accuracy, FPR, Specificity, and Confusion Matrix	No camera data for further image processing techniques	• The innovative structure of the multi-classifier fusion • Better accuracy in detecting driving events • Provide an aggregate model using ensemble learning • An Android application for collecting the sensor signals • Provide a comprehensive labeled dataset • Real-time labeling of a Police Officer
[59]	Accuracy, Loss, Precision, Recall, F1-score, and Receiver Operating Characteristic	• Domain-specific knowledge idea was similar to threshold-based techniques in signal processing. • These features were handy declared on input sensor signals. • Detailed information about Safe events was not provided • Fine-grained classification of safe events • The best LSTM model reached converged after 500 epochs and two hours	• An innovative method for analyzing the unlabeled data • It could be concluded that more data samples led to more accuracy
[7]	Accuracy, Precision, Recall, F1-score, and the Mean Per Class Error	• Dependence of threshold-based methods on the type of the car and the smartphone • Generalization of the model was an issue	• Acceleration in feature extraction by using threshold-based models • Analyzing changes in the motion sensor data in detecting driving events • An LSTM classification model for fusing sensor data • Utilizing the physical model of a moving vehicle
[8]	Accuracy	• High performance, low-cost solutions, and collaborative sensing were planned for further studies • More driving events collection from different weather conditions, road types, sensors, and vehicles were planned for further study • The best result achieved after more than 901 epochs	• Different RNN architectures were evaluated • Inputting Accelerometer data to different RNN architectures • The empirical evaluation indicated that GRU performance was more accurate
[29]	Accuracy and Area Under the ROC curve	• The proposed model cannot distinguish between Right or Left Lane Changes • Window sizes between 1 and 10 seconds were used which might cause the occurrence of two events in a window size • Combining low and variable sampling rates of signals led to missing some actual values and decreased the performance • One non-expert person selected labels • Other actions such as gaze, head position, and blinking are reported in their further research	• Two laps of free driving were collected with the presence of two co-pilots inside the vehicle • A Dangerous driving classifier was proposed back to the back of an event-type classifier • Weighting the window instances in order to increase the number of samples

the detection performance of ADAS. In order to cover this need, we proposed an ensemble of classifiers to detect driving events and driver behaviors. The model was a combination of three individual classifiers, including CNN, SVM, and MLP, which also led to a high certainty level for a life-critical system such as ADAS. To evaluate our proposed model, we developed an Android application to capture smartphone sensor signals and simultaneously label the events using a Traffic Police Officer's expertise. Then the captured data is sent to a remote server for further processing and evaluation of the performance criteria. All experiments were conducted in a real-world situation using fifty drivers aged 23-45 years. Results indicated that the proposed fusion model performs better than each individual classifier in terms of Accuracy, FPR, and Specificity (96.75, 0.004, and 0.996). In this section, we describe our proposed model in detail.

A. CONVOLUTION NEURAL NETWORKS (CNNS)

Convolutional Neural Networks (CNNs) are one of the most significant Deep learning approaches for profoundly teaching many layers. This approach is highly efficient and is one of the most often used algorithms in computer vision applications. A CNN network comprises three primary layers and each layer is responsible for different duties: the convolutional layer, the pooling layer, and the fully connected layer. Any CNN has two training stages: feed-forward and back-propagation. Initially, in the proposed model, the sensor signals are fed into the network, and this operation is

TABLE 2. Related datasets.

Reference	Driving Events	Device	Dataset Properties	Model
Proposed Model	Stop, Safe Front Brake, Aggressive Front Brake, Rear Gear, Aggressive Rear Brake, Safe Acceleration, Aggressive Acceleration, Lane Change, Left Turn, and Right Turn	Gravity, Linear Acceleration, Gyroscope, Pressure, Magnetometer, Light meter, Inclinometer, Sound Intensity, GPS, and Speed of a Galaxy S8 Plus	• 24005 labeled driving event windows of 50 drivers • The data collection duration was about a year	Fusion of ensemble CNN, SVM, and MLP
[59]	Aggressive Acceleration, Braking, Left Lane Change, Right Lane Change, Left Turn, Right Turn, and Non-Aggressive	Accelerometer and Gyroscope of a Motorola XT1058. Also, Accelerometer and Gyroscope (MPU6050) on a Raspberry Pi	• The first dataset had 11077 labeled data from 4 car trips of approximately 13 minutes conducted by two drivers • The second dataset contained 1114 labeled data from a Raspberry Pi Model B equipped with MPU6050 • Data was collected in Ankara (capital of Turkey) • Data collection duration was two weeks by three male drivers between the ages of 27-37	Semi-supervised LSTM
[7]	Same as above driving events	Accelerometer and Gyroscope of a Motorola XT1058	Same as the above first dataset	Supervised LSTM
[8]	Aggressive Braking, Aggressive Acceleration, Aggressive Left Turn, Aggressive Right Turn, Aggressive Left Lane Change, Aggressive Right Lane Change, and Non-aggressive event.	Accelerometer of a Motorola XT1058	Same as the above dataset (the dataset was first provided by Carvalho *et al.* [8])	Supervised GRU, LSTM, and simple RNN
[29]	Same as above driving events	Same as above device addition to the CAN bus signals	Same as the above dataset (the dataset was first provided by Carvalho *et al.* [8])	Supervised GRU, LSTM, and simple RNN

nothing more than multiplying the point between the input and the parameters of each neuron, followed by convolution operations in each layer. Then driving event (network output) is computed. The overall architecture of the proposed CNN model is shown in Fig. 1.

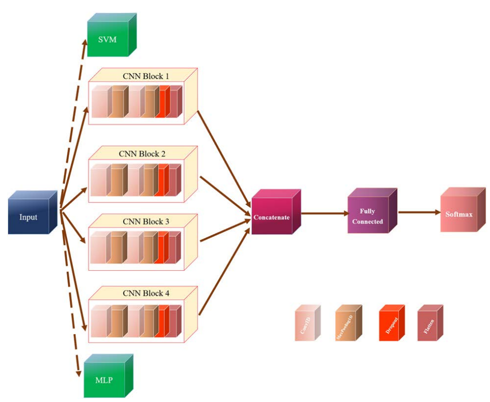

FIGURE 1. The proposed CNN architecture.

Each CNN Block has six layers: Conv1D, MaxPooling1D, Conv1D, MaxPooling1D, Dropout, and Flatten. Additionally, four blocks have different kernel sizes of three, five, seven, and eleven. Details of the architecture are described in Appendix A (Fig. 9). To establish the network parameters, the output result is utilized to compute the loss which is done by comparing the network output against the proper response using a loss function and calculating the loss rate. Following is the back-propagation step which is dependent on the estimated loss rate. A chain rule has been used to compute the derivative of each parameter which is then updated in response to the loss's influence. The next feed-forward phase begins when the parameters are changed. The network training will be completed after the accomplishment of several stages.

Each layer is described as follows:

The Convolutional Layer: comprises multiple filters that use convolution at network input to generate feature maps. The continual updating of filter components (Neural Network weights) characterizes a Deep NeuralNetwork's learning process. We must first construct several filters of the same size and with random values to mimic this layer.

The Pooling Layer: is often put after the convolutional layer and can minimize the size of feature maps and network parameters. Because of side features in computations, pooling layers (like convolutional layers) are unaffected by displacement. Commonly, to reduce feature dimensions, they are implemented in two ways: max pooling and average pooling. Feature maps taken from the preceding layer are split into window sizes of two or so. Finally, each window's average or maximum values are computed and used to populate the new feature map.

The Fully Connected Layers (FC): transform the pooled feature maps into a one-dimensional feature vector. They are similar to conventional Neural Networks and approximately account for ninety percent of a CNN network's parameters.

These layers can show grid results as a vector of a specific size which can be used for classification or additional processing. The primary disadvantage of these layers is their high training processing cost which is caused by their large number of parameters. As a result, the most often utilized approach is either eliminating these layers or minimizing the number of connections. A hybrid model based on MLP, SVM, and CNN was employed to address this limitation. Fig. 1 depicts the proposed CNN architecture.

B. SUPPORT VECTOR MACHINE (SVM)

In SVM method, we assume that there is a set of data points $(x_1, c_1), (x_2, c_2), \ldots, (x_n.c_n)\}$ which have to be divided into two separate classes $c_i = \{-1, 1\}$. Each x_i is a real number p-dimension vector that has the same variables which can express model behavior. By building a hyperplane (a linear equation), linear classification techniques attempt to segregate data. The SVM classification technique, one of the linear classification methods, determines the optimal hyperplane that divides the data into the two classes with the least distance. In this technique, input vectors are transferred to a multidimensional space. Then a hyperplane will be built to separate the input vectors by the shortest feasible distance. This hyperplane is the "hyperplane with the greatest separator border," which is not on the boundary between these two hyperplanes. "Hyperplane with Maximum Separator Boundary" is a hyperplane that optimizes the distance between two parallel hyperplanes. The greater separator boundary (distance between two parallel hyperplanes) leads to lesser classification error. The SVM classifier equation is as follows:

$$g(x) = \sum_{i=1}^{L_s} \alpha_i d_i K(x_i, x) + \alpha_0 \tag{1}$$

In equation (1), K is kernel function and x_i represents the support vector that is obtained from the training data. L_s represents the number of support vectors, d_i is the corresponding class number x_i, and finally α_i is constant training number. Support vectors are elements of training data located exactly on or within the boundaries of classification decision making. These vectors include samples that are more difficult to classify than others [60]. In Fig. 2, the schematic of an SVM is provided.

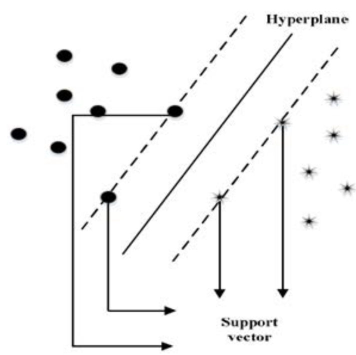

FIGURE 2. The SVM schematic of Safe driving event detection.

A multidimensional SVM is used since sensor multi-class type data ultimately leads to different driver behaviors. For enabling SVM to classify several classes, three methods are proposed: One-Against-All (OAA), One-Against-One (OAO), and All-At-Once (AAO) [39].

In the OAO model, classes are divided into $\frac{n\times(n-1)}{2}$ binary classification classes. The following problem is solved to build an SVM that classifies the *kth* class and *lth* class:

$$\min Q_p\left(W^{kl}, b^{kl}, \xi^{kl}\right) = \frac{1}{2}\left(W^{kl}\right)^T\left(W^{kl}\right) + C\sum_{i=1}^{m} \xi_i^{kl} \tag{2}$$

$$\textit{subject to } \left(W^{kl}\right)^T.\phi(x_i) + b^{kl} \geq 1 - \xi_i^{kl}, \quad \textit{if } y_i = k, \tag{3}$$

$$\left(W^{kl}\right)^T.\phi(x_i) + b^{kl} \geq 1 + \xi_i^{kl}, \quad \textit{if } y_i = 1, \tag{4}$$

$$\xi_i^{kl} \geq 0, \quad i = 1, \ldots, m \textit{ and } k, l = 1, \ldots, n \tag{5}$$

According to the above equations, W is equal to the n-dimension weights matrix, b is bias. Superplane's shape and position are determined by W and b. $\xi = \{\xi_1, \ldots, \xi_m$ is an auxiliary variable and C is the penalty coefficient that balances the model's complexity with classification error (a high value of C will cause overfitting). Also $\phi(x_i)$ is a nonlinear transformation that transforms the sample into a point space with larger dimensions called the property space. $i = 1, \ldots, m$ and n are the numbers of test data records and data dimensions, respectively. y_i is considered as a training data tag.

C. PROPOSED MODEL

Training data are entered one by one in the hybrid system. Each algorithm produces a model separately according to its logic. Then experimental samples are fed into each model to measure the accuracy of the proposed model. Each algorithm issues an answer that indicates whether the entered sample is correctly classified. Finally, all answers are entered into the maximum voting mechanism, which finds the most candidates and sends them as final answers. Since the paper's primary aim is to improve the accuracy of driving event detection, majority voting is utilized in the voting process. Fig. 3 shows the block diagram of the proposed model. The trained models of SVM, CNN, and MLP are used in voting. The output is mapped as a multilabel problem; each output neuron is linked to an event type, as shown in Table 3.

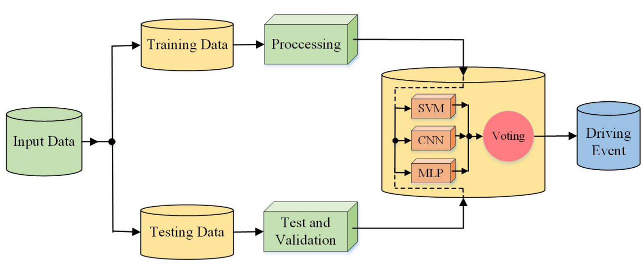

FIGURE 3. Block diagram of the proposed model.

TABLE 3. Details of the collected data.

Dataset	Description
Raw dataset size	24005 driving event windows of size 128×21
Labels	Range: 1-10 Stop (S), Safe Front Brake with Gear Two (SFB), Aggressive Front Brake with Gear Four (AFB), Rear Gear (RG), Aggressive Rear Brake (ARB), Safe Acceleration (SA), Aggressive Acceleration (AA), Lane Change (LC), Left Turn (LT), and Right Turn (RT)
Test size	7201 driving event windows
Validation size	20 percent of the training dataset 10-fold cross-validation
Training size	16804 driving event windows
Sensor signals	g-Force (Gravity), Linear Acceleration, Gyroscope, Barometer, Magnetometer, Light Meter, Inclinometer (Azimuth, Pitch, and Roll), Sound Meter, GPS (Latitude and Longitude), and Speed
Drivers	50 drivers
Age	23-45

Following stages describe the implementation process of the suggested model.

Step 1: In this step, training data are fed into the algorithm to create a model through the algorithm's training procedure. The generated model has rules and structure based on the entered sample data. First, the training data are entered into the SVM classification algorithm.

Step 2: Each model gets the experimental data as input. The models are fully structured at this stage and can apply the detection process to the data. Therefore, each model implements the detection process on each sample of experimental data.

Step 3: Step 2 applies to SVM, CNN, and MLP algorithms. As a result, we have three different or identical answers from algorithms. They are placed into the maximum voting system as three numbers. The proposed system uses majority voting to decide between the provided answers. The answer with the most candidates is the main answer for the proposed system.

Finally, steps 1 to 3 continues until all experimental data have been analyzed.

IV. DATA COLLECTION

The smartphone sensor signals (vehicle traffic data) were collected by an Android application instantaneously. The application collected data on its SD card and sent them to the remote server. Table 3 provides details of our dataset, including event window size, event types, test size, validation size, train size, captured sensor signals, number of drivers in the experiments, and the drivers' age. The Y-axis was across the vehicle's front direction, the X-axis was the vehicle's lateral direction, and the Z-axis was perpendicular to the horizontal direction.

Real-time data were collected with about 50 Hz frequency (with a 2.56-second window length). These data had various information such as event label, start timestamp, and finish timestamp. Signals of a Galaxy S8 Plus smartphone were collected, including Gravity, Linear Acceleration, Gyroscope, Pressure, Magnetometer, Light Meter, Inclinometer (Azimuth angles, Pitch, and Roll), Sound Intensity, GPS (Latitude and Longitude), and speed. For example, the Accelerometer sensor provided a 3-dimensional (x, y, and z) temporal series with nanosecond precision in the standard sensor coordinate system (relative to the device). In attempting to achieve device independence for positions, we used a rotation matrix to translate sensor data from the device's coordinate system. The Galaxy S8 Plus was chosen because its sensor set, processing, and storage capacity were adequate for our data collecting task. Fig. 4 shows a schematic of the proposed system, from data collection to driving event detection. As you can see, signals from ten sensors were captured during the experiments; then, they were sent to a remote server for further processing.

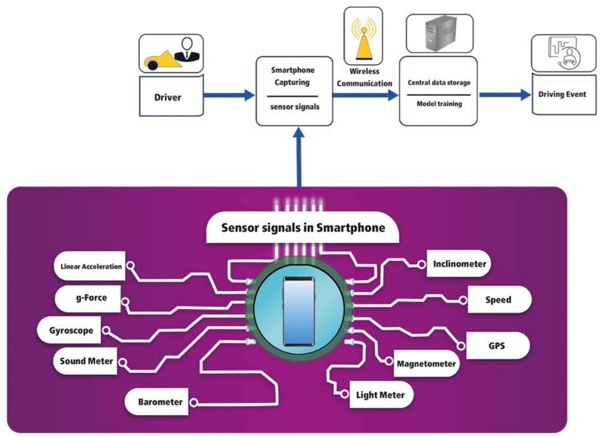

FIGURE 4. Schematic of the proposed system.

Finally, we had about 24000 samples from fifty drivers representing ten distinct types of driving events. These driving events included the following: Stop (S), Safe Front Brake with Gear Two (SFB), Aggressive Front Brake with Gear Four (AFB), Rear Gear (RG), Aggressive Rear Brake (ARB), Safe Acceleration (SA), Aggressive Acceleration (AA), Lane Change (LC), Left Turn (LT), and Right Turn (RT).

Randomly seventy percent of the collected data are utilized for training, while the rest are used for testing. In addition, twenty percent of the training data were used for validation. At the start of training, initial random weights and biases were generated. The intended output value for each model is determined to avoid the risk of excessive connections while maintaining the maximum accuracy possible. The data are split into batches of size 256 to optimize speed and performance. Different batch sizes are evaluated in the hyperparameter tuning phase, indicating that the optimal size is 256.

V. RESULTS AND DISCUSSION

As mentioned before, we utilize the multi-classifier fusion technique in this article. An ensemble of classifiers processes each input sample and combines the results according to a rule. In our situation, the outputs of the fundamental classifiers (CNN, SVM, and MLP) are mixed by a majority vote, indicating that a sample window of the sensor signal is allocated to a specific class if at least fifty percent of the classifiers produced that class. Other combination strategies, such as the variant of weighted voting [26], have been proposed in the literature, but they were deemed inadequate for the present application. Another advantage of using ensemble voting models is the certainty level. Because all Advanced Driving Assistance Systems are life-critical systems, the certainty of driving event detection is crucial. Therefore we used three different models and a majority voting ensemble method in order to have a more certainty level [27], [28].

Following, we discussed the experiment procedure. Quantitative criteria are used to evaluate the simulation results, such as Accuracy, False Positive Rate, and Specificity. Also, we provide details information about the proposed model performance by drawing the confusion matrix. A confusion matrix is a C × C dimension square matrix in which C is the number of classes (C equals 10 in our experiments). Non-diagonal elements indicate incorrect samples predicted in a class different from their actual class, whereas the matrix diameter contains appropriately classified samples. Classification accuracy is computed once the interference matrix has been calculated. The following relationships define the other two quantitative requirements.

$$FPR = FalsePositiveRate = \frac{FP}{FP + TN} \tag{6}$$

$$Specificity = \frac{TN}{FP + TN} \tag{7}$$

where:

False Positive (FP):	A claimed correct outcome that was, in fact, wrong.
False Negative (FN):	An incorrect outcome that has been misidentified.
True Positive (TP):	A claimed correct outcome that was, in fact, correct.
True Negative (TN):	A claimed incorrect outcome that was, in fact, incorrect.

In four distinct circumstances, we analyze the outcomes of driving event detection to determine the performance of the proposed model. First, we use the Multi-Layer Perceptron (MLP) model. Second, a simple CNN is used. In the third scenario, we analyze data with the SVM model and in the fourth scenario, the proposed hybrid model is utilized. All simulations have been done in the Windows operating system with 32 GB of DDR4 RAM hardware, an Intel Cori7-6700K CPU, and a VGA GTX 1080.

Following, we provide the details of MLP and CNN architecture and their hyperparameter tuning. Table 4 comparison results show that the Adaptive Moment Estimation (Adam) algorithm is chosen to train the CNN (Learning Rate is set to 0.0025). The Adam Optimizer is an adaptive estimation of the moment form of the Stochastic Gradient Descent method, known as Adam in Tensorflow, which has the best correlation value among the above functions. According to Table 4, Adadelta and RMSprop have the second and third performances among optimizers.

TABLE 4. Performance comparison of different optimizers.

Row	Function	Description	Correlation
1	SGD	Gradient Descent with Momentum	0.8346
2	Adagrad	Adaptive Learning Rate Gradient Descent	0.8474
3	RMSprop	Root Mean Square Propagation	0.8563
4	Adadelta	Adaptive Delta	0.8979
5	Adam	Adaptive Moment Estimation	**0.9356**

TABLE 5. Comparison of several mlp architectures.

Row	Architecture		Correlation
	No. Neuron	No. Layer	
1	10	10	0.9647
2	20	10	**0.9860**
3	10	20	0.9746
4	15	15	0.4890
5	5	5	0.9612
6	8	8	0.8820
7	5	10	0.9786
8	10	5	0.9080
9	8	12	0.8388
10	15	10	0.9583
11	25	9	0.9686

The various MLP architectures are evaluated to find the best performance architecture. Table 5 indicates that increasing and decreasing the number of layers from ten decreases the correlation between data (as the number of neurons rises).

As a result, it can be concluded that the optimal architecture for an MLP Neural Network is ten layers and twenty neurons in order to predict driving events. The structure of the proposed MLP architecture is provided in Fig. 5. First, sensor sample data is flattened to feed the ten layers of MLP. After these hidden layers, a softmax layer of ten neurons is utilized in order to normalize the output to a distribution over predicted driving event labels.

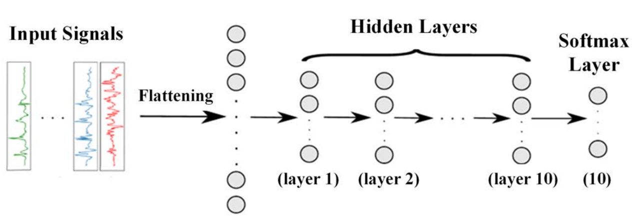

FIGURE 5. MLP structure for driving event detection.

The performances of different CNN architectures with varying numbers of convolutional layers are compared in Table 6. CNN with two convolutional layers has the best performance overall. Also, the results imply that concatenate layer's output is sufficient for describing driving behavior. Consequently, the detection process may be accomplished

TABLE 6. Performance comparison of networks with different depths.

Model	Accuracy (%)	FPR	Specificity
2 conv layers	88.70	0.088	0.911
4 conv layers	87.19	0.097	0.9321
6 conv layers	76.45	0.182	0.869

just with two convolutional layers. Once a certain depth is achieved, the accuracy begins to degrade.

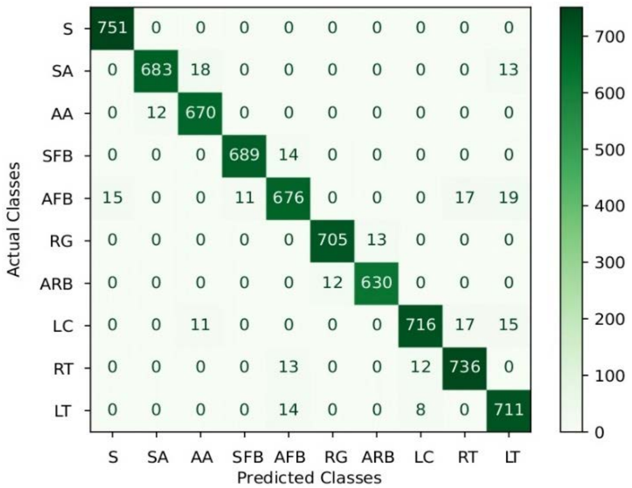

FIGURE 6. Driving event detection confusion matrix.

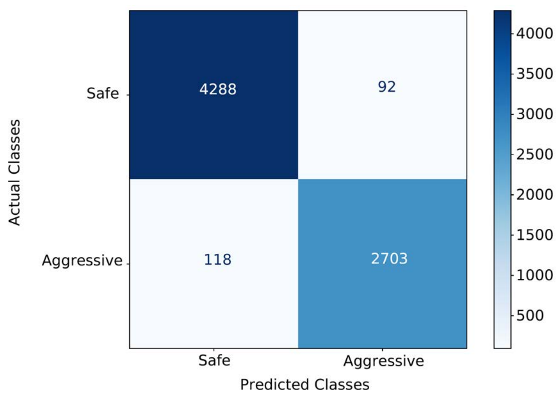

FIGURE 7. Driver behavior confusion matrix.

Fig. 6 and Fig. 7 show the confusion matrixes of driving events and driver behaviors detection, respectively. A driver who drives with some aggressive driving events which could be caused an accident refers to an aggressive driver. Stop (S), Safe Front Brake with Gear Two (SFB),[1] Rear Gear (RG), Safe Acceleration (SA), Left Turn (LT), and Right Turn (RT) are all examples of safe driving behaviors. Aggressive behaviors include driving events such as Aggressive Front Brake with Gear Four (AFB), Aggressive Rear Brake (ARB), Aggressive Acceleration (AA), and Lane Change (LC). After investigating the confusion matrix, we identify that some of the Lane Change events are detected as Turn events.

[1] A manual car with 5 gears is used for these experiments.

TABLE 7. Comparison of various models.

Model	Specificity	FPR	Accuracy (%)
Proposed	0.996	0.004	96.75
MLP	0.781	0.145	85.43
CNN	0.911	0.088	88.70
SVM	0.823	0.101	87.5

We believe that the similarities in the first part of driving event windows for Left/Right Lane Change and Turn events are the reason for these results. Also, there is some incorrect detection of Aggressive Front Brake events which are predicted as Right or Left Turn. Reviewing the experiments' details helps us identify the root cause of these underperformances. At the start of the Right and Left Turn events, we detect that the drivers decrease their speed to perform the Turn events. Some of these decreases are similar to the Aggressive Front Brake which caused the incorrect detection.

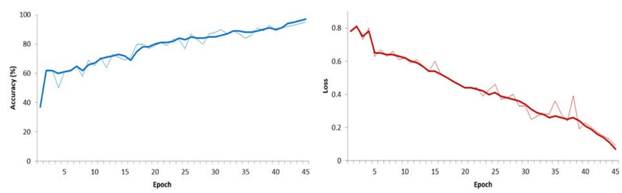

FIGURE 8. Accuracy and loss of the proposed model for driving event detection.

Fig. 8 shows the proposed model accuracy and loss (bold line indicates the training results). As you can see, the accuracy and loss of the proposed model improve gradually in each epoch and converge to their best performance after 45 epochs.

Table 7 compares the proposed and individual MLP, CNN, and SVM models. Results demonstrate that the proposed model has the best accuracy, FPR, and Specificity among all models. Also, these results indicate that the CNN model has second place in terms of performance among other models. Although the Specificity of the CNN model is near to the best result, its accuracy is far away from the proposed model. It indicates that the proposed ensemble model could achieve better accuracy by fusing different individual models.

VI. CONCLUSION

Wireless technology advancements allow for the instantaneous collection and analysis of massive amounts of high-resolution data from vehicles and drivers. These data make an opportunity for researchers to study driving events and driver behaviors in order to reduce road accidents. Researchers can use spatial and temporal characteristics of the data to introduce customized Deep Neural Networks for extracting some information. In this paper, we propose a multi-classifier fusion model in order to detect driving events and driver behaviors. An ensemble of three classifiers (CNN, SVM, and MLP) processes each sample input and combines the outputs using a majority vote. The reason behind using this technique is the certainty level. Advanced Driving

Assistance Systems are life-critical systems requiring high certainty in driving event detection. Therefore, utilizing an ensemble of three different models leads to a more certain decision on the class type of each sample window. The proposed model is trained and evaluated using data from instantaneous driving events collected via an Android application. We captured about 24000 driving data from 50 drivers and utilized a Traffic Police Officer's expertise to simultaneously label the captured sensor signal. Results indicate that the fusion model performs better than each individual classifier in terms of Accuracy, False Positive Rate (FPR), and Specificity (96.75, 0.004, and 0.996).

VII. FUTURE WORKS

For future studies, researchers may examine a broader range of driving events, conduct behavioral analysis using additional signals (e.g., EEG), or utilize driving simulators for new drivers in order to explore driving style characteristics. Also, we plan to investigate the SVM model to find if our high accuracy is at the expense of diversity. Investigating the performance effect of using an attention mechanism in driving behavior detection is another open issue in the literature. Furthermore, future works on incorporating Fuzzy Inference Systems like Adaptive neuro-fuzzy inference systems can be expected to evaluate the proposal of different models more adaptive to our problem.

APPENDIX A
DETAILED ARCHITECTURE OF THE CNN MODEL

Details of the proposed CNN architecture are shown in Fig. 9.

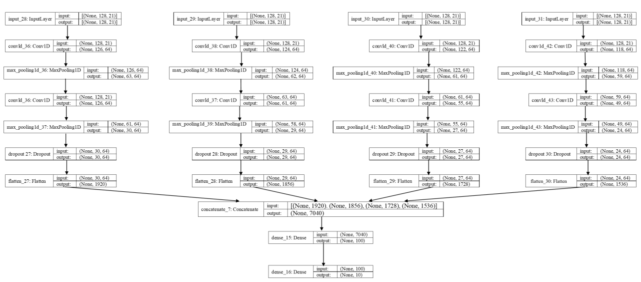

FIGURE 9. The proposed CNN tensor architecture.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers and the Editor-in-Chief for their great valuable comments, which led us to improve the article.

REFERENCES

[1] (Jun. 2018). World Health Organization. *Global Status Report on Road Safety*. WHO, Geneva, Switzerland. Accessed: Jan. 29, 2022. [Online]. Available: https://apps.who.int/iris/rest/bitstreams/1164010/retrieve

[2] M. H. Alkinani, W. Z. Khan, and Q. Arshad, "Detecting human driver inattentive and aggressive driving behavior using deep learning: Recent advances, requirements and open challenges," *IEEE Access*, vol. 8, pp. 105008–105030, 2020, doi: 10.1109/ACCESS.2020.2999829.

[3] A. Mohammed, M. R. M. Yazid, B. B. Zaidan, A. A. Zaidan, S. Garfan, R. A. Zaidan, H. A. Ameen, Z. H. Kareem, and R. Q. Malik, "A landscape of research on bus driver behavior: Taxonomy, open challenges, motivations, recommendations, limitations, and pathways solution in future," *IEEE Access*, vol. 9, pp. 139896–139927, 2021, doi: 10.1109/ACCESS.2021.3102222.

[4] D. Shinar, *Traffic Safety and Human Behavior*. Bingley, U.K.: Emerald Group Publishing, 2017, pp. 87–100.

[5] R. Arvin, A. J. Khattak, and H. Qi, "Safety critical event prediction through unified analysis of driver and vehicle volatilities: Application of deep learning methods," *Accid. Anal. Prev.*, vol. 151, pp. 1–12, Mar. 2021, doi: 10.1016/J.AAP.2020.105949.

[6] Y. Xing, C. Lv, H. Wang, H. Wang, Y. Ai, D. Cao, E. Velenis, and F. Y. Wang, "Driver lane change intention inference for intelligent vehicles: Framework, survey, and challenges," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4377–4390, May 2019, doi: 10.1109/TVT.2019.2903299.

[7] S. Sarker and M. M. Haque, "An approach towards domain knowledge-based classification of driving maneuvers with LSTM network," in *Title: Algorithms for Intelligent Systems*, M. S. Uddin and J. C. Bansal, Ed. Singapore: Springer, 2021, pp. 469–484.

[8] E. Carvalho, B. V. Ferreira, J. Ferreira, C. de Souza, H. V. Carvalho, Y. Suhara, A. S. Pentland, and G. Pessin, "Exploiting the use of recurrent neural networks for driver behavior profiling," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 3016–3021, doi: 10.1109/IJCNN.2017.7966230.

[9] Y. Üzümcüoğlu, T. Özkan, C. Wu, and H. Zhang, "Traffic climate and driver behaviors: The moderating role of driving skills in Turkey and China," *J. Saf. Res.*, vol. 75, pp. 87–98, Dec. 2020, doi: 10.1016/j.jsr.2020.08.004.

[10] C. M. Richard, J. Lee, R. Atkins, and J. L. Brown, "Using SHRP2 naturalistic driving data to examine driver speeding behavior," *J. Saf. Res.*, vol. 73, pp. 271–281, Jun. 2020, doi: 10.1016/j.jsr.2020.03.008.

[11] J. Wu and H. Xu, "Driver behavior analysis for right-turn drivers at signalized intersections using SHRP2 naturalistic driving study data," *J. Saf. Res.*, vol. 63, pp. 177–185, Dec. 2017, doi: 10.1016/j.jsr.2017.10.010.

[12] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles, "Driving behavior analysis with smartphones: Insights from a controlled field study," in *Proc. 11th Int. Conf. Mobile Ubiquitous Multimedia (MUM)*, 2012, pp. 1–8.

[13] P. Handel, I. Skog, J. Wahlstrom, F. Bonawiede, R. Welch, J. Ohlsson, and M. Ohlsson, "Insurance telematics: Opportunities and challenges with the smartphone solution," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 4, pp. 57–70, Oct. 2014.

[14] P. Handel and J. Ohlsson, "Smartphone-based measurement systems for road vehicle traffic monitoring and usage-based insurance," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1238–1248, Dec. 2014.

[15] A. Sathyanarayana, S. O. Sadjadi, and J. H. L. Hansen, "Leveraging sensor information from portable devices towards automatic driving maneuver recognition," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 660–665, doi: 10.1109/ITSC.2012.6338717.

[16] H. Huang, Y. Cheng, and R. Weibel, "Transport mode detection based on mobile phone network data: A systematic review," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 297–312, Apr. 2019, doi: 10.1016/j.trc.2019.02.008.

[17] J. White, C. Thompson, H. Turner, H. Turner, and D. C. Schmidt, "Wreckwatch: Automatic traffic accident detection and notification with smartphones," *Mobile Netw. Appl.*, vol. 16, no. 3, pp. 285–303, Jun. 2011, doi: 10.1007/s11036-011-0304-8.

[18] M. R. Dey, U. Satapathy, P. Bhanse, B. K. Mohanta, and D. Jena, "MagTrack: Detecting road surface condition using smartphone sensors and machine learning," in *Proc. IEEE Region 10 Conf. (TENCON)*, Oct. 2019, pp. 2485–2489, doi: 10.1109/TENCON.2019.8929717.

[19] E. F. de S. Soares, K. Revoredo, F. Baiao, C. A. de M. S. Quintella, and C. A. V. Campos, "A combined solution for real-time travel mode detection and trip purpose prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4655–4664, Dec. 2019, doi: 10.1109/TITS.2019.2905601.

[20] L. Li, J. Zhu, H. Zhang, H. Tan, B. Du, and B. Ran, "Coupled application of generative adversarial networks and conventional neural networks for travel mode detection using GPS data," *Transp. Res. A, Policy Pract.*, vol. 136, pp. 282–292, Jun. 2020, doi: 10.1016/j.tra.2020.04.005.

[21] E. Ramanujam, T. Perumal, and S. Padmavathi, "Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review," *IEEE Sensors J.*, vol. 21, no. 12, pp. 13029–13040, Mar. 2021, doi: 10.1109/JSEN.2021.3069927.

[22] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019, doi: 10.1016/J.PATREC.2018.02.010.

[23] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1237–1242.

[24] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 2809–2813, doi: 10.1109/IJCNN.2011.6033589.

[25] J. Yu, Z. Chen, Y. Zhu, Y. J. Chen, L. Kong, and M. Li, "Fine-grained abnormal driving behaviors detection and identification with smartphones," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2198–2212, Oct. 2017, doi: 10.1109/TMC.2016.2618873.

[26] A. Koohestani, M. Abdar, A. Khosravi, S. Nahavandi, and M. Koohestani, "Integration of ensemble and evolutionary machine learning algorithms for monitoring diver behavior using physiological signals," *IEEE Access*, vol. 7, pp. 98971–98992, 2019, doi: 10.1109/ACCESS.2019.2926444.

[27] G. Jozanikohan, G. H. Norouzi, F. Sahabi, H. Memarian, and B. Moshiri, "The application of multilayer perceptron neural network in volume of clay estimation: Case study of Shurijeh gas reservoir, Northeastern Iran," *J. Natural Gas Sci. Eng.*, vol. 22, pp. 119–131, Jan. 2015, doi: 10.1016/j.jngse.2014.11.022.

[28] M. Kordestani, M. Dehghani B. Moshiri, and M. Saif, "A new fusion estimation method for multi-rate multi-sensor systems with missing measurements," *IEEE Access*, vol. 8, pp. 47522–47532, 2020, doi: 10.1109/ACCESS.2020.2979222.

[29] D. Alvarez-Coello, B. Klotz, D. Wilms, S. Fejji, J. M. Gomez, and R. Troncy, "Modeling dangerous driving events based on in-vehicle data using random forest and recurrent neural network," in *Proc. IEEE Intell. Veh. Symp.*, vol. 2019, pp. 165–170, doi: 10.1109/IVS.2019.8814069.

[30] H. R. Eftekhari and M. Ghatee, "A similarity-based neuro-fuzzy modeling for driving behavior recognition applying fusion of smartphone sensors," *J. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 72–83, Jan. 2019, doi: 10.1080/15472450.2018.1506338.

[31] R. Lotfi and M. Ghatee, "Smartphone based driving style classification using features made by discrete wavelet transform," 2018, *arXiv:1803.06213*.

[32] C. Saiprasert, P. Thunyasit, and P. Wasan, "Detecting driving events using smartphone," in *Proc. 20th Intell. Transp. Syst. World Congr.*, 2013, pp. 42–53. [Online]. Available: https://www.researchgate.net/profile/Wasan-Pattara-Atikom/publication/279949922_Detecting_Driving_Events_Using_Smartphone/links/55a71e2908aeb4e8e646d091/Detecting-Driving-Events-Using-Smartphone.pdf

[33] W. Cao, X. Lin, K. Zhang, Y. Dong, S. Huang, and L. Zhang, "Analysis and evaluation of driving behavior recognition based on a 3-axis accelerometer using a random forest approach," in *Proc. 16th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2017, pp. 303–304, doi: 10.1145/3055031.3055060.

[34] A. H. Ali, A. Atia, and M.-S.-M. Mostafa, "Recognizing driving behavior and road anomaly using smartphone sensors," *Int. J. Ambient Comput. Intell.*, vol. 8, no. 3, pp. 22–37, Jul. 2017. [Online]. Available: https://www.igi-global.com/article/recognizing-driving-behavior-and-road-anomaly-using-smartphone-sensors/183618

[35] B.-F. Wu, Y.-H. Chen, and C.-H. Yeh, "Fuzzy logic based driving behavior monitoring using hidden Markov models," in *Proc. 12th Int. Conf. ITS Telecommun.*, Nov. 2012, pp. 447–451, doi: 10.1109/ITST.2012.6425217.

[36] S. Ucar, B. Hoh, and K. Oguchi, "Abnormal driving behavior detection system," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–6, doi: 10.1109/VTC2021-SPRING51267.2021.9448822.

[37] T. Vesselenyi, S. Moca, A. Rus, T. Mitran, and B. Tătaru, "Driver drowsiness detection using ANN image processing," in *Proc. IOP Conf. Mater. Sci. Eng.*, vol. 252, 2017, pp. 1–8, doi: 10.1088/1757-899X/252/1/012097.

[38] Q. Cai, M. Abdel-Aty, Y. Sun, J. Lee, and J. Yuan, "Applying a deep learning approach for transportation safety planning by using high-resolution transportation and land use data," *Transp. Res. A, Policy Pract.*, vol. 127, pp. 71–85, Sep. 2019, doi: 10.1016/J.TRA.2019.07.010.

[39] N. Formosa, M. Quddus, S. Ison, M. Abdel-Aty, and J. Yuan, "Predicting real-time traffic conflicts using deep learning," *Accident Anal. Prevention*, vol. 136, 2020, Art. no. 105429, doi: 10.1016/J.AAP.2019.105429.

[40] A. B. Parsa, H. Taghipour, S. Derrible, and A. Mohammadian, "Real-time accident detection: Coping with imbalanced data," *Accident Anal. Prevention*, vol. 129, pp. 202–210, Aug. 2019, doi: 10.1016/J.AAP.2019.05.014.

[41] J. Yuan, M. Abdel-Aty, Y. Gong, and Q. Cai, "Real-time crash risk prediction using long short-term memory recurrent neural network," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2673, no. 4, pp. 314–326, Apr. 2019, doi: 10.1177/0361198119840611.

[42] Q. Zeng, H. Huang, X. Pei, S. C. Wong, and M. Gao, "Rule extraction from an optimized neural network for traffic crash frequency modeling," *Accident Anal. Prevention*, vol. 97, pp. 87–95, Dec. 2016, doi: 10.1016/J.AAP.2016.08.017.

[43] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, "Age of information aware trajectory planning of UAVs in intelligent transportation systems: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12382–12395, Nov. 2020, doi: 10.1109/TVT.2020.3023861.

[44] J. Bao, H. Yu, and J. Wu, "Short-term FFBS demand prediction with multi-source data in a hybrid deep learning framework," *IET Intell. Transp. Syst.*, vol. 13, no. 9, pp. 1340–1347, Sep. 2019, doi: 10.1049/IET-ITS.2019.0008.

[45] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach," *Transp. Res. C, Emerg. Technol.*, vol. 97, pp. 258–276, Dec. 2018, doi: 10.1016/J.TRC.2018.10.011.

[46] C. Xu, J. Ji, and P. Liu, "The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 47–60, Oct. 2018, doi: 10.1016/J.TRC.2018.07.013.

[47] H. Cui, G. Yuan, N. Liu, M. Xu, and H. Song, "Convolutional neural network for recognizing highway traffic congestion," *J. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 279–289, May 2020, doi: 10.1080/15472450.2020.1742121.

[48] D. Ding, J. Tong, and L. Kong, "A deep learning approach for quality enhancement of surveillance video," *J. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 304–314, May 2020, doi: 10.1080/15472450.2019.1670659.

[49] M. K. Galab, A. Taha, and H. H. Zayed, "Adaptive technique for brightness enhancement of automated knife detection in surveillance video with deep learning," *Arabian J. Sci. Eng.*, vol. 46, no. 4, pp. 4049–4058, Apr. 2021, doi: 10.1007/S13369-021-05401-4.

[50] Y. Li, K. Khoshelham, M. Sarvi, and M. Haghani, "Direct generation of level of service maps from images using convolutional and long short-term memory networks," *J. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 300–308, May 2019, doi: 10.1080/15472450.2018.1563865.

[51] M. Liu and J. Shi, "A cellular automata traffic flow model combined with a BP neural network based microscopic lane changing decision model," *J. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 309–318, 2019, doi: 10.1080/15472450.2018.1462176.

[52] J. C. Nwadiuto, S. Yoshino, H. Okuda, and T. Suzuki, "Variable selection and modeling of drivers' decision in overtaking behavior based on logistic regression model with gazing information," *IEEE Access*, vol. 9, pp. 127672–127684, 2021, doi: 10.1109/ACCESS.2021.3111753.

[53] A. Quddus, A. S. Zandi, L. Prest, and F. J. E. Comeau, "Using long short term memory and convolutional neural networks for driver drowsiness detection," *Accident Anal. Prevention*, vol. 156, Jun. 2021, Art. no. 106107, doi: 10.1016/J.AAP.2021.106107.

[54] J. Tang, S. Yu, F. Liu, X. Chen, and H. Huang, "A hierarchical prediction model for lane-changes based on combination of fuzzy C-means and adaptive neural network," *Expert Syst. Appl.*, vol. 130, pp. 265–275, Sep. 2019, doi: 10.1016/J.ESWA.2019.04.032.

[55] S. F. Varotto, R. Jansen, F. Bijleveld, and N. van Nes, "Adaptations in driver deceleration behaviour with automatic incident detection: A naturalistic driving study," *Transp. Res. F, Traffic Psychol. Behaviour*, vol. 78, pp. 164–179, Apr. 2021, doi: 10.1016/J.TRF.2021.02.011.

[56] S. Zhang, M. Abdel-Aty, J. Yuan, and P. Li, "Prediction of pedestrian crossing intentions at intersections based on long short-term memory recurrent neural network," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2674, no. 4, pp. 57–65, Apr. 2020, doi: 10.1177/0361198120912422.

[57] R. V. Nezafat, O. Sahin, and M. Cetin, "Transfer learning using deep neural networks for classification of truck body types based on side-fire LiDAR data," *J. Big Data Anal. Transp.*, vol. 1, no. 1, pp. 71–82, Jun. 2019, doi: 10.1007/S42421-019-00005-9.

[58] W. Wang, R. D. J. Samuel, and C.-H. Hsu, "Prediction architecture of deep learning assisted short long term neural network for advanced traffic critical prediction system using remote sensing data," *Eur. J. Remote Sens.*, vol. 54, pp. 65–76, Mar. 2021, doi: 10.1080/22797254.2020.1755998.

[59] S. Sarker, M. M. Haque, and M. A. A. Dewan, "Driving maneuver classification using domain specific knowledge and transfer learning," *IEEE Access*, vol. 9, pp. 86590–86606, 2021, doi: 10.1109/ACCESS.2021.3089660.

[60] M. F. Akay, "Support vector machines combined with feature selection for breast cancer diagnosis," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3240–3247, 2009, doi: 10.1016/J.ESWA.2008.01.009.

EHSAN KHOSRAVI received the B.Sc. degree in information technology engineering from the School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran, in 2015, and the M.Sc. degree in computer engineering from the Sharif University of Technology, Tehran, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Computer Engineering. His research interests include intelligent transportation systems, autonomous driving, sensor data fusion, machine learning, cognitive radio, and wireless communication. He received the Excellency Scholarship from the Sharif University of Technology for his M.Sc. and Ph.D. degrees.

ALI MOHAMMAD AFSHIN HEMMATYAR received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 1988, 1991, and 2007, respectively. Since 1991, he has been with the Department of Computer Engineering, Sharif University of Technology, where he is currently an Assistant Professor. His research interests include smart vehicles, intelligent transportation systems, mobile ad-hoc networks, wireless sensor networks, and the Internet of Things (IoT).

MAHDI JAFARI SIAVOSHANI received the B.Sc. degree in communication systems with a minor in applied physics from the Sharif University of Technology, Tehran, Iran, in 2005, and the M.Sc. and Ph.D. degrees in computer and communication sciences from the Ecole Polytechnique Federale de Lausanne, Switzerland, in 2007 and 2012, respectively. He is currently an Assistant Professor with the Department of Computer Engineering, Sharif University of Technology, and a Leader with the Information, Network, and Learning Laboratory. His research interests include information processing, large scale networks, network coding, information theory, machine learning, graphical models, and wireless communications.

BEHZAD MOSHIRI received the B.Sc. degree in mechanical engineering from the Iran University of Science and Technology, Tehran, Iran, in 1984, and the M.Sc. and Ph.D. degrees in control systems engineering from the University of Manchester, Manchester, U.K., in 1986 and 1991, respectively. He has been a Professor with the School of Electrical and Computer Engineering, University of Tehran, Tehran, since 1992. His research interests include intelligent transportation systems, advanced process control, sensor fusion, and mechatronics.