IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 November 2023, accepted 4 December 2023, date of publication 7 December 2023, date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3340304

RESEARCH ARTICLE

Preventing Road Accidents Through Early Detection of Driver Behavior Using Smartphone Motion Sensor Data: An Ensemble Feature Engineering Approach

ALI RAZA[1], IQRA AKHTAR[2], LAITH ABUALIGAH[3,4,5,6,7,8,9], RAED ABU ZITAR[10], MOHAMED SHARAF[11], MOHAMMAD SH. DAOUD[12], AND HEMING JIA[13], (Member, IEEE)

[1]Institute of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan
[2]Department of Electrical and Biomedical Engineering, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan 64200, Pakistan
[3]Computer Science Department, Prince Hussein Bin Abdullah Faculty for Information Technology, Al-Albayt University, Mafraq 25113, Jordan
[4]Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon
[5]Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan
[6]School of Computer Sciences, Universiti Sains Malaysia, Gelugor, Penang 11800, Malaysia
[7]Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan
[8]MEU Research Unit, Middle East University, Amman 11831, Jordan
[9]School of Engineering and Technology, Sunway University, Petaling Jaya 27500, Malaysia
[10]Sorbonne Center of Artificial Intelligence, Sorbonne University Abu Dhabi, Abu Dhabi, United Arab Emirates
[11]Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[12]College of Engineering, Al Ain University, Abu Dhabi, United Arab Emirates
[13]School of Information Engineering, Sanming University, Sanming 365004, China

Corresponding authors: Ali Raza (ali.raza.scholarly@gmail.com) and Laith Abualigah (aligah.2020@gmail.com)

The authors present their appreciation to King Saud University for funding this research through Researchers Supporting Program number (RSPD2023R704), King Saud University, Riyadh, Saudi Arabia.

ABSTRACT Driver behavior refers to the actions and attitudes of individuals behind the wheel of a vehicle. Poor driving behavior can have serious consequences, including accidents, injuries, and fatalities. One of the main disadvantages of poor driving behavior is the increased risk of road accidents, higher insurance premiums, fines, and even criminal charges. The primary aim of our study is to detect driver behavior early with high-performance scores. The publicly available smartphone motion sensor data is utilized to conduct our study experiments. A novel LR-RFC (Logistic Regression Random Forest Classifier) method is proposed for feature engineering. The proposed LR-RFC method combines the logistic regression and random forest classifier for feature engineering from the motion sensor data. The original smartphone motion sensor data is input into the LR-RFC method, generating new probabilistic features. The newly extracted probabilistic features are then input to the applied machine learning methods for predicting driver behavior. The study results show that the proposed LR-RFC approach achieves the highest performance score. Extensive study experiments demonstrate that the random forest achieved the highest performance score of 99% using the proposed LR-RFC method. The performance is validated using k-fold cross-validation and hyperparameter optimization. Our novel proposed study has the potential to revolutionize the early detection of driver behavior to avoid road accidents.

INDEX TERMS Machine learning, driver behavior, sensor data, feature engineering, ensemble learning.

I. INTRODUCTION

Driver behavior is a significant factor that influences road safety. Poor driving behavior is any action that deviates from the prescribed traffic rules and regulations. Poor driving behavior has numerous disadvantages that can lead to critical road accidents or life loss [1]. Reckless driving, speeding, and driving under the influence of drugs or alcohol are some examples of poor driving behavior. These actions can lead to road accidents resulting in injury, loss of property,

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson.

and even death. Poor driving behavior puts the driver at risk, passengers, and other road users [2]. Poor driving behavior can result in increased insurance premiums, fines, and penalties. The mortality rate caused by road accidents is alarming [3], and most of these accidents result from poor driving behavior. Therefore, it is essential to detect driver behavior using an advanced machine learning approach in order to prevent loss of life and property.

The detection of driver behavior has become increasingly critical for road safety and traffic management. This technology has several benefits, including detecting dangerous driving behaviors like distracted driving, aggressive driving, and drowsy driving before accidents occur. Early detection can result in proactive measures, such as issuing alerts or warnings to drivers, to avoid potential accidents [4]. Another advantage of driver behavior detection is its potential application in surveillance systems. These systems can use the information gathered from monitoring driving behaviors to alert law enforcement to potential criminal activity such as reckless driving, improving public safety and preventing crime. Additionally, driver behavior detection can potentially optimize traffic flow and reduce congestion in intelligent transportation systems (ITS) by identifying areas where drivers are prone to aggressive driving [5]. Overall, driver behavior detection has significant advantages and multiple applications, and with advancements in machine learning, its importance is expected to increase in the coming years.

Machine learning algorithms are increasingly used to detect driver behavior patterns by analyzing data collected through smartphone motion sensors [6], [7]. Machine learning models can identify safe or risky driving behavior patterns by monitoring speed, acceleration, and braking. Such models can help improve road safety by providing real-time feedback to drivers, encouraging them to adopt safer driving habits [8]. In addition, the data generated by these models can be used by insurance companies to personalize policies based on individual driving habits. While there are challenges associated with collecting and analyzing large amounts of data from smartphone sensors, the use of machine learning for driver behavior analysis shows great promise for improving road safety and reducing the number of accidents caused by reckless driving [8].

Our proposed study's primary contributions to driver behavior detection are followed as:

- A novel LR-RFC method that combines the logistic regression and random forest classifier for feature engineering from the smartphone motion sensor data is proposed. The original smartphone motion sensor data is input into the LR-RFC method, generating new probabilistic features. The newly extracted probabilistic features are then input to the applied machine learning methods for predicting driver behavior with high performance.
- Four advanced machine learning-based techniques and a deep learning method are applied in comparison. The applied methods include logistic regression, support vector machine, gaussian naive bayes, and random forest. Each applied is fully validated using the k-fold cross-validations and hypermeter tuning. The random forest method outperformed state-of-the-art studies using the proposed LR-RFC feature engineering.

The remaining research is structured as: Related work is described in Section II. Study methodology is analyzed in Section III. Results and discussions of conducted experiments are comparatively evaluated in Section IV. Conclusions and future work are discussed in Section V.

II. RELATED WORK

In recent years, there has been a growing interest in utilizing machine learning algorithms and sensor data for predicting driver behavior. Various studies have explored using accelerometers, gyroscopes, and magnetometers to collect motion data [9] and identify hazardous driving behaviors. Integrating machine learning algorithms has enabled the accurate prediction of driver behavior, such as sudden braking, aggressive acceleration, and speeding. However, the field faces some challenges, including privacy concerns. This study section aims to comprehensively review related work in driver behavior prediction using machine learning and sensor data, including the methods used, accuracy rates achieved, and limitations of existing approaches as analyzed in Table 1.

The article [10] discusses a research study that aims to enhance the identification of hazardous driver behavior using sensor fusion and machine learning. The study employed an Android smartphone to collect motion data using accelerometers, gyroscopes, and magnetometers. The data were then processed to obtain relevant descriptive features. The extracted features were utilized to train and test a support vector machine and an artificial neural network. The results indicate that the proposed methodology could accurately recognize unsafe driver behaviors, achieving an average accuracy rate of approximately 88% and 90% for the SVM classifier and neural network, respectively. In protecting the driver's privacy, the research team opted not to utilize GPS tracking, webcams, or microphones.

A new approach is presented in the article [11] for detecting driving behavior using feature selection based on power spectral density variance distribution [20], [21]. Data from an accelerometer and gyroscope mounted on a Samsung S5 smartphone, placed on the vehicle's steering wheel, is used. The deep learning model trained using the extracted features can accurately identify abnormal driving behavior, such as weaving and sudden braking, with 91% accuracy when only accelerometer data is used and 96.1% when accelerometer and accelerometer and gyroscope data are combined. A pattern recognition network is used to process the final feature set, with 70% of the data reserved for training, 15% for testing, and 15% for validation.

This paper [12] introduces a novel method for driving behavior recognition using deep learning techniques and smartphone sensor data. The approach utilizes a fusion of convolutional neural networks and recurrent neural networks

TABLE 1. The driver behavior detection-related literature summary analysis.

Ref.	Year	Dataset	Technique	Performance Accuracy (%)
[10]	2020	Accelerometers, gyroscopes, and magnetometer data.	Neural network	90.0
[11]	2019	203 collected driving data using accelerometer and gyroscope.	Neural network	96.1
[12]	2019	tri-axial accelerometer, orientation sensor, and tri-axial gyroscope data.	DeepConvLSTM	95.7
[13]	2022	Accelerometers, gyroscopes, and GPS as time series data	LightGBM	88.0
[14]	2023	900 samples with accelerometers and gyroscopes.	RF	95.0
[15]	2023	UAH-Driveset and FD-Driveset	CNN-LSTM	97.6
[16]	2022	Speed, acceleration, deceleration, and distance data.	CNN	96.1
[17]	2020	NGSIM dataset	Bi-LSTM network	94.2
[18]	2022	State Farm Distracted Drivers dataset.	Ensemble ResNet50 and VGG16	92.0
[19]	2021	Video image and psychological data.	dilated residual network (DRN)	93.2

with an attention unit to identify temporal and structural features while exploring correlation among sensor data. During real-world experiments in Hefei, China, the study collected data from a Xiaomi Mi 5 smartphone with various sensors positioned on the centre console. Six driving events, including straight driving, static, left turn, right turn, braking, and acceleration, were recorded with device position independence and a sampling rate of 50 Hz. The proposed model achieved a competitive f1 accuracy score of 95.7% by utilizing attention-based and L2-constrained DeepConvGRU on the smartphone sensor data.

The investigation [13] suggests utilizing smartphone sensors to accumulate data on driver behavior and classify it into four categories normal, intermediate, aggressive, and dangerous in various external conditions such as speed limits, weather conditions, and traffic signs. The study collects data from accelerometers, gyroscopes, and GPS as time series data [22]. It uses different machine learning algorithms for time series classification to train an AI-based classifier. The research shows that smartphone sensors have dependable and low-cost sensors that are easily accessible on most Android and IOS operating systems, making data collection efficient. The outcomes exhibit an accuracy score of 88.0% achieved by the LightGBM machine learning algorithm in identifying driving profiles for a one-minute journey.

This article [14] explores using built-in smartphone sensors to classify and recognize driving manoeuvres on highways. The researchers collected raw vehicle data at 50 samples/second sampling rate using calibrated Android smartphones equipped with accelerometers and gyroscopes. The study involved drivers performing various driving manoeuvres categorized as Light, Normal, and Hard, and each manoeuvre was performed at least five times, resulting in a total of 900 samples. To identify driving manoeuvres, the authors proposed a hybrid system that combined the Dynamic Time Warping (DTW) pattern-matching technique with machine learning algorithms for classification. The hybrid system demonstrated superior performance and achieved a 95.0% accuracy score using the machine learning-based RF technique. The study suggests smartphones' built-in sensors have numerous potential applications, including driver behavior analysis, driving safety, and driver assistance systems.

The presented article [15] introduces DSDCLA, an attention-based hybrid framework that combines CNN and LSTM to identify driving style using short- and long-term spatial-temporal features. DSDCLA uses CNN and self-attention to extract local spatial features from multi-modal driving sequences, and LSTM and multi-head attention to explore long-term temporal relationships between timesteps. Additionally, three variants of DSDCLA are presented with different fusion levels to enhance interpretability. The framework is evaluated on two publicly available real-world datasets, UAH-Driveset and FD-Driveset, that contain multi-modal sensing signals gathered by six drivers and vehicles using smartphone sensors. Each driving signal includes an accelerometer, odometer, speed, and road-type data. The experimental results reveal that DSDCLA achieved f1 scores of 97.65%.

The study [16] explores deep learning methods for driver behavior recognition and profiling. Specifically, three deep learning algorithms were employed to classify driving data based on various parameters, including speed, acceleration, deceleration, distance to other vehicles, and steering. The dataset used in the study was collected through a range of sensors such as an onboard diagnostics reader, lidar, ultrasonic sensors, an inertial measurement unit, and a global positioning sensor. Results from the study showed that the Convolutional Neural Network (CNN) algorithm achieved the highest accuracy of 96.1% and an f-measure of 95.2% when evaluated on different timeframes. These findings demonstrate the potential for using deep learning techniques to accurately identify and classify driver behavior as safe or aggressive.

The study [17] presents a new method that combines time series prediction and deep learning networks to recognize vehicle behavior at intersections. The approach uses the ARIMA algorithm to predict the vehicle's lateral position, longitudinal position, speed, and acceleration, followed by using the Bi-LSTM network to detect turning behavior based on the predicted and derived parameters. The proposed method is evaluated using the NGSIM dataset provided by the FHWA NGSIM project, and it achieves an average recognition rate of 94.2% for turning behavior detection.

This study [18] presents E2DR, a novel approach for detecting driver distraction and providing in-car recommendations using deep learning ensembles. E2DR combines multiple deep learning models using stacking ensemble methods to improve accuracy and generalization. The ensemble ResNet50 and VGG16 models yielded the highest accuracy, with a test accuracy of 92% on state-of-the-art datasets. The State Farm Distracted Drivers dataset, containing 22,424 images of drivers in distracted positions, was used for the experiment. The findings demonstrate that the proposed approach can effectively detect driver distractions and provide in-car recommendations to improve safety and increase awareness.

This article [19] proposes a monitoring model for driving stress in urban areas that leverages the XGBoost algorithm. The model combines driving behavior, driving environment, and route familiarity to assess stress levels. Driving behavior is measured using vehicle speed and acceleration, and driving environment is evaluated using a dilated residual network (DRN) model that divides the video image into subregions based on the driver's attention distribution. A K-means 3D cluster analysis is applied to obtain the evaluation method of driving stress based on psychological data and driver stress inventory (DSI) results. The XGBoost model is demonstrated to outperform other mainstream machine learning algorithms and traditional models in terms of accuracy, achieving a score of 93.25%. Additionally, the article highlights the efficiency of electrocardiograph (ECG) as the most effective method for assessing a driver's stress level. The model's design and evaluation demonstrate its potential for practical application monitoring driving stress.

The research gap related to the driver behavior prediction we have determined through the literature work is followed as:

- Primarily, the researchers used classical machine learning and deep learning methods to predict driver behavior with low-performance scores. More advanced ensemble learning-based techniques can also be built to achieve high-performance accuracy scores. The classical features engineering approach was used in related literature. Smart ensemble features engineering can build to enhance the performance scores for detecting driver behavior.

III. PROPOSED METHODOLOGY

Our proposed study experiments utilize the smartphone motion sensor data based on the driver's behavior. A novel LR-RFC method is proposed that uses feature engineering from the smartphone motion sensor data. The new feature set extracted from smartphone motion sensor data is formed for further experiments. The created dataset is then divided into training and testing parts. The training part is utilized for training the applied machine learning techniques, and the testing part is utilized to evaluate the performance. The hyperparameter-tuned outperformed method is then used for predicting driver behavior with high performance, as shown in Figure 1.

A. SMARTPHONE MOTION SENSOR DATA

Our proposed study uses the publicly available benchmark smartphone motion sensor dataset [23] to evaluate the experiments. The accelerometer and gyroscope were the sensors used to collect the driver's motion. The dataset is based on driving behaviors which include slow, normal, and aggressive. Aggressive driving has sudden breaks, speeding, and sudden right or left turns. A data collector application was designed to collect motion data using a Samsung Galaxy S21. The dataset features properties are based on the sampling rate, gravitational acceleration, sensors accelerometer, and gyroscope. The dataset is based on features that include the timestamp (time in seconds), acceleration values in X, Y, and Z axis in meters per second squared, rotation values in X, Y, and Z axis in degrees per second with classification label as slow, normal, and aggressive. The dataset distribution analysis is illustrated in Figure 2. The research shows that the target label slow (2) contains a sensor value of 2,604, the label normal (1) contains a sensor value of 2,197, and the label aggressive (0) contains a sensor value of 1,927. The analysis demonstrates that the dataset is imbalanced.

B. ENSEMBLE FEATURE ENGINEERING APPROACH

The proposed study's novel feature engineering technique for predicting driver behavior is illustrated in Figure 3. The proposed LR-RFC method combines the logistic regression and random forest classifier for feature engineering from the sensor data. The original smartphone motion sensor data is input to logistic regression and random forest classifiers. Then the probabilistic features [24] are generated from both models. The extracted probabilistic features are then input to the applied machine learning methods for predicting driver behavior in this study. The study results show that the proposed LR-RFC approach achieves the highest performance score.

Detecting driver behavior is critical in the intelligent transportation system field, as it has significant implications for traffic management and road safety. To achieve high-performance accuracy scores, machine learning techniques, such as ensemble learning and feature engineering [25], are commonly employed in recent years. Ensemble learning, which involves combining multiple models to make more accurate predictions than any single model, has been demonstrated to enhance driver behavior detection model accuracy. In contrast, feature engineering involves selecting

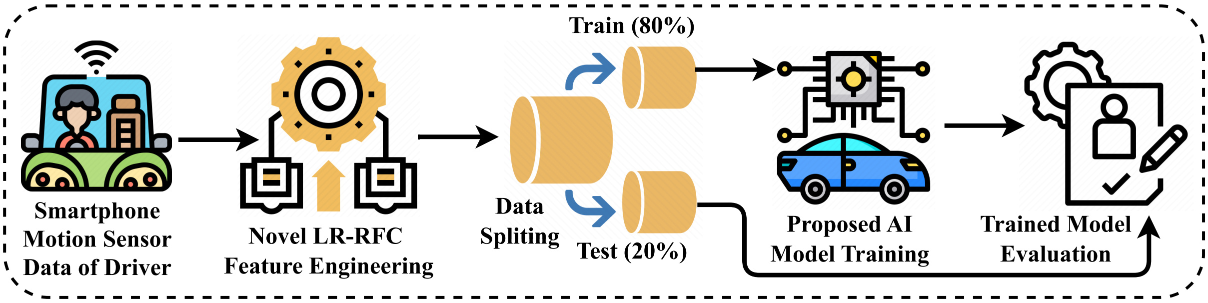


FIGURE 1. Our novel proposed study methodology analysis for predicting driver behavior.

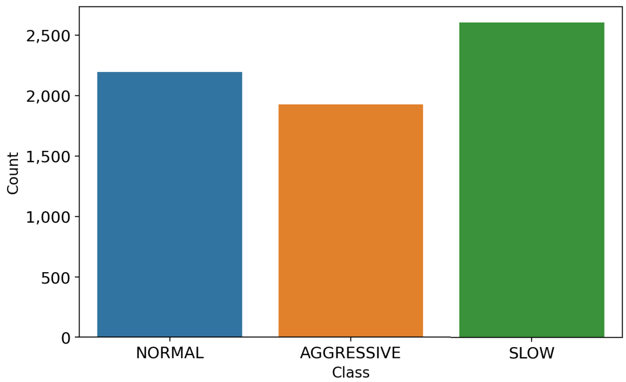


FIGURE 2. The driver behavior-based target class distribution analysis.

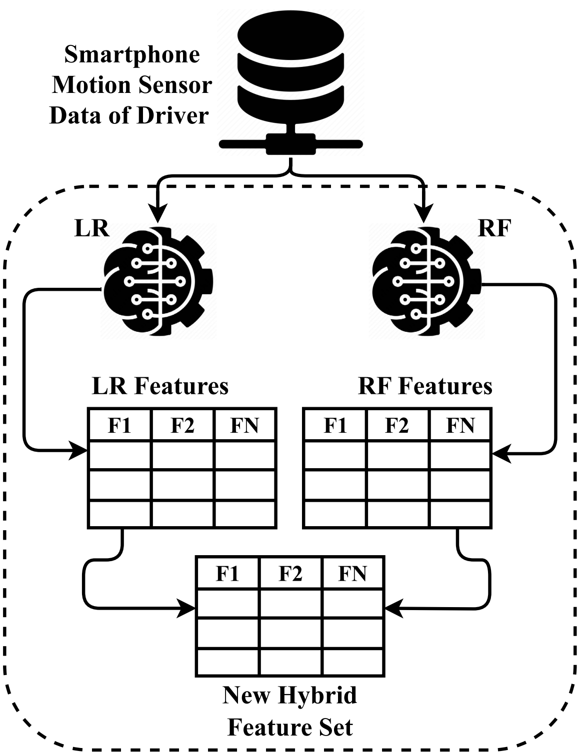


FIGURE 3. The architecture analysis of our novel proposed feature engineering for predicting driver behavior.

and transforming the most informative features from raw data, leading to improved interpretability and generalization of driver behavior detection models. Combining these two techniques has enabled the development of precise driver behavior detection models capable of classifying various driving behaviors with high precision and recall scores. The use of ensemble learning and feature engineering is critical for developing effective driver behavior detection models, with significant implications for road safety and traffic management.

Algorithm 1 shows the step-by-step flow of the proposed feature engineering approach.

Algorithm 1 LR-RFC Algorithm

Input: Smartphone motion sensor data.

Output: Hybrid features for detecting driver behavior as Slow, Normal, or Aggressive.

initiate;

1- $P_{lr} \longleftarrow LR_{probability\ features}(Sd)$ // here Sd belong to the smartphone motion sensor data and P_{lr} are predicted features.

2- $P_{rfc} \longleftarrow RFC_{probability\ features}(Sd)$ // here Sd belong to the smartphone motion sensor data and P_{rfc} are predicted features.

3- $H_{features} \longleftarrow \sum\{P_{lr}+P_{rfc}\}$ // $H_{features} \epsilon$ Hybrid features set used for driver behavior prediction as Slow, Normal, or Aggressive.

end;

1) PROBABILISTIC FEATURES EXTRACTION MECHANISM

Suppose we have a set of input data points, X = x1, x2, …, xn, and a corresponding set of labels, Y = y1, y2, …, yn, where each label yi is a binary variable indicating the presence or absence of a certain target feature. We can use a machine learning model to predict the probability of each label given the input data, P(Yilxi), which can be represented as a vector of probabilities, p = [p1, p2, …, pn], where pi = P(Yilxi).

The extracted probabilities can be used as features for further analysis or modeling. For example, we can extract the probability distribution's mean, variance, or other statistical properties as features. Alternatively, we can use the probabilities directly as features, individually or in combination with other features. Mathematically, the process of predicting and using the probabilities as features are represented below.

Predicting the Probabilities: Given a set of input data points X and corresponding labels Y. Train a machine learning model to predict the probability of each label given the input data: P(Yi|xi) represents the probabilities as a vector of probabilities p:

$$p = [p1, p2, \ldots, pn], \text{ where } pi = P(Yi|xi) \quad (1)$$

Using the Probabilities as Features We can use the extracted probabilities distribution as feature inputs to another machine-learning model. Use The probabilities can be used directly as features, individually, or in combination with other features. Mathematically:

Let X = x1, x2, …, xn be a set of input data points and Y = y1, y2, …, yn be the corresponding labels Let f(X) be a machine learning model that predicts the probability of each label given the input data: P(Yi|xi) Let p = [p1, p2, …, pn] be a vector of probabilities, where pi = f(xi) Using the probabilities as features. Let g(p) be a function that extracts statistical properties of the probability distribution as features. Let h(X,p) be a function that uses the probabilities directly as features, individually or in combination with other features. The final output of the feature extraction process is a set of features:

$$F = f1, f2, \ldots, fm, \text{ where } fi = g(p) \text{ or } h(X, p). \quad (2)$$

C. DATASET SPLITTING

The dataset-splitting approach in machine learning involves partitioning a dataset into two subsets for training and testing purposes. Our study used a dataset splitting ratio of 80:20 [26], where 80% of the data is used for training the machine learning model, and the remaining 20% is reserved for testing the model's performance. The training dataset is used to train the model's parameters using various algorithms, and the testing dataset is used to evaluate the model's performance to test its generalization ability. The dataset splitting ensures the applied model's robustness and reliability in real-world applications.

Let D be the original dataset, consisting of n samples and their corresponding labels, such that:

$$D = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \quad (3)$$

where x_i is the i-th sample and y_i is its corresponding label.

The first step in dataset splitting is to divide the original dataset into two parts: the training set D_{train} and the testing set D_{test}. This is typically done randomly, with a specified ratio of samples allocated to each set.

Let m be the number of samples in the training set such that:

$$D_{train} = (x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m) \quad (4)$$

Similarly, let k be the number of samples in the testing set, such that:

$$D_{test} = (x_{m+1}, y_{m+1}), (x_{m+2}, y_{m+2}), \ldots, (x_{m+k}, y_{m+k}) \quad (5)$$

D. APPLIED MACHINE AND DEEP LEARNING TECHNIQUES

Machine learning techniques have been increasingly applied to analyze complex data and predict driver behavior using motion sensor data [27]. This approach involves using motion sensors placed on the driver to collect data on their movements and then feeding that data into a machine learning algorithm. The algorithm can then analyze the data to identify patterns and predict the driver's behavior, such as their swimming speed, direction, and depth they are diving [28], [29], [30]. This technology has significant potential for enhancing safety in diving, as it could help identify potentially dangerous behaviors or conditions and alert drivers.

E. LOGISTIC REGRESSION

Logistic Regression (LR) is a statistical method for predicting categorical outcomes [31]. LR is a popular method for predicting driver behavior because it is easy to interpret and can handle continuous and categorical input variables. In predicting driver behavior using smartphone motion sensor data, logistic regression can be used to model the relationship between the sensor data and the driver's behavior. The model estimates the probability of a particular behavior given a set of sensor data. The sensor data could include variables such as the driver's acceleration, orientation, and position. The LR model can be trained using a dataset of labeled examples, where the outcome variable is the driver's behavior, and the input variables are the sensor data. The model then uses this training data to estimate the coefficients of the logistic regression equation, which can be used to predict the behavior of new drivers based on their sensor data. The mathematical notations of the LR model are expressed as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (6)$$

where z is the linear combination of the input features $x_1, x_2, \ldots, x_n$ and their corresponding coefficients $\beta_1, \beta_2, \ldots, \beta_n$, plus the intercept term b:

$$z = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + b \quad (7)$$

Then, we can express the probability of the binary outcome y (0 or 1) given the input features as:

$$P(y = 1|x_1, x_2, \ldots, x_n) = \sigma(z) = \frac{1}{1 + e^{-z}} \quad (8)$$

$$P(y = 0|x_1, x_2, \ldots, x_n) = 1 - \sigma(z) = \frac{e^{-z}}{1 + e^{-z}} \quad (9)$$

We can then use Maximum Likelihood Estimation (MLE) to estimate the coefficients $\beta_1, \beta_2, \ldots, \beta_n$ and the intercept b that maximize the likelihood of the observed data:

$$\ell(\beta_1, \beta_2, \ldots, \beta_n, b) = \prod_{i=1}^{N} P(y_i|x_{i1}, x_{i2}, \ldots, x_{in})^{y_i} \times (1 - P(y_i|x_{i1}, x_{i2}, \ldots, x_{in}))^{1-y_i} \quad (10)$$

where N is the number of observations in the dataset, y_i is the binary outcome for the i-th observation, and $x_{i1}, x_{i2}, \ldots, x_{in}$ are the corresponding input features.

Finally, we can use various optimization algorithms such as Gradient Descent, Newton-Raphson, or Conjugate Gradient to iteratively update the coefficients and intercept until convergence, based on the gradient of the log-likelihood function:

$$\nabla_{\beta_j}\ell(\beta_1, \beta_2, \ldots, \beta_n, b) = \sum_{i=1}^{N}(y_i - \sigma(z_i))x_{ij} \quad (11)$$

$$\nabla_{b}\ell(\beta_1, \beta_2, \ldots, \beta_n, b) = \sum_{i=1}^{N}(y_i - \sigma(z_i)) \quad (12)$$

where $z_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in} + b$ is the linear combination of input features for the i-th observation.

F. SUPPORT VECTOR MACHINE

Support Vector Machines (SVM) have been extensively used in machine-learning applications, including predicting human behavior [32]. SVM has shown promising results in predicting driver behavior using smartphone motion sensor data. The SVM model takes the sensor data collected from the driver's smartphone as input and maps it to a higher dimensional feature space, finding the optimal hyperplane separating the behavior classes. The model then uses this hyperplane to classify new instances of driver behavior based on the sensor data. The essential advantage of using SVM in this context is its ability to handle high-dimensional data and its robustness to noisy data. First, we start with the optimization problem for the primal form of the Support Vector Machine:

$$\min_{w,b,\xi} \frac{1}{2}|w|^2 + C\sum_{i=1}^{m}\xi_i \quad (13)$$

subject to:

$$y_i(w^T x_i + b) \geq 1 - \xi_i,\ \forall i = 1, \ldots, m \quad (14)$$

and

$$\xi_i \geq 0,\ \forall i = 1, \ldots, m \quad (15)$$

where w is the weight vector, b is the bias term, ξ is the slack variable, C is the penalty parameter that controls the trade-off between maximizing the margin and minimizing the classification error, x_i and y_i are the feature vector and the corresponding label of the i-th training example, and m is the total number of training examples.

Next, we derive the dual form of the optimization problem by introducing the Lagrange multipliers α_i:

$$\max_{\alpha} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (16)$$

subject to:

$$0 \leq \alpha_i \leq C,\ \forall i = 1, \ldots, m \quad (17)$$

and

$$\sum_{i=1}^{m}\alpha_i y_i = 0 \quad (18)$$

The optimal weight vector and bias term can then be computed as:

$$w^* = \sum_{i=1}^{m}\alpha_i y_i x_i \quad (19)$$

and

$$b^* = y_k - \sum_{i=1}^{m}\alpha_i y_i x_i^T x_k \quad (20)$$

where k is any index such that $0 < \alpha_k < C$.

Finally, we can make predictions for a new input vector x using the following decision function:

$$f(x) = \text{sign}(w^T x + b) \quad (21)$$

where $\text{sign}(\cdot)$ is the sign function that returns $+1$ or -1 depending on the sign of its argument.

G. GAUSSIAN NAIVE BAYES

Gaussian Naive Bayes (GNB) is a probabilistic machine learning algorithm commonly used for classification tasks [33]. It is based on Bayes' theorem, which allows for estimating the probability of a particular class given a set of features. GNB assumes that the features are independent of each other and that they follow a Gaussian distribution. In predicting driver behavior with smartphone motion sensor data, GNB can classify different types of diving behavior, such as free diving, scuba diving, or snorkeling. The motion sensor data collected from a smartphone can provide information on the driver's movement patterns and be used as GNB algorithm features. The motion dataset can be used to estimate the conditional probabilities of each feature given a particular diving behavior class, which can then be used to predict the most likely diving behavior class for a given set of sensor data.

Assuming that we have a dataset with n observations and d features, the goal of the Gaussian Naive Bayes model is to classify a new data point x into one of the k classes.

Prior Probability: The prior probability of class c is calculated as the proportion of observations in the training set that belong to class c.

$$P(c) = \frac{\sum_{i=1}^{n}\mathbb{1}(y_i = c)}{n} \quad (22)$$

where y_i is the class label of the i-th observation and $\mathbb{1}$ is the indicator function.

Likelihood: The likelihood of observing the feature vector x given class c is assumed to follow a Gaussian distribution with mean $\mu_{c,j}$ and variance $\sigma_{c,j}^2$ for each feature j.

$$P(x_j \mid c) = \frac{1}{\sqrt{2\pi\sigma_{c,j}^2}}\exp\left(-\frac{(x_j - \mu_{c,j})^2}{2\sigma_{c,j}^2}\right) \quad (23)$$

where $\mu_{c,j}$ and $\sigma^2_{c,j}$ are the mean and variance of the j-th feature for class c, respectively.

Posterior Probability: The posterior probability of class c given the feature vector x is calculated using Bayes' theorem.

$$P(c \mid x) = \frac{P(c)\prod_{j=1}^{d} P(x_j \mid c)}{\sum_{c'} P(c')\prod_{j=1}^{d} P(x_j \mid c')} \tag{24}$$

where P(c) is the prior probability of class c, and $P(x_j \mid c)$ is the likelihood of observing the j-th feature given class c.

Classification: The new data point x is classified into the class with the highest posterior probability.

$$\hat{y} = \arg\max_{c} P(c \mid x) \tag{25}$$

where $\hat{y}$ is the predicted class label for the new data point x.

H. RANDOM FOREST

Random Forest (RF) is a popular machine-learning algorithm for classification and regression tasks [34]. RF is an ensemble learning algorithm combining multiple decision trees to make predictions. In predicting driver behavior with smartphone motion sensor data, an RF model can be trained using motion sensor data collected from smartphones worn by drivers during their drivers. The model works by dividing the data into subsets and constructing decision trees based on each subgroup. The decision trees are combined to form an RF, which can make predictions based on the input data.

The RF model is an ensemble learning method that combines multiple decision trees to improve the accuracy of the predictions. Let X be the input features and y be the target variable. Given a training set (X, y), the Random Forest algorithm works as follows:

1) For each tree t in the forest:
   a) Draw a bootstrap sample X_t of size n from the training set (X, y).
   b) Randomly select m features from X (where m is typically much smaller than the total number of features).
   c) Grow a decision tree t from the bootstrap sample X_t using only the m selected features. At each node, split the data based on the feature that maximizes the decrease in impurity (e.g., Gini index or entropy).
2) Predict the target variable y for a new input X_{new} by aggregating the predictions of all the trees in the forest. For regression problems, this is typically done by taking the average of the predicted values; for classification problems, this is typically done by taking the mode (i.e., the most common class label) of the predicted values.

The prediction of the Random Forest model can be expressed mathematically as follows:

$$\hat{y}RF(Xnew) = \frac{1}{T}\sum_{t=1}^{T} f_t(X_{new}),$$

where $\hat{y}RF(Xnew)$ is the predicted value for the new input X_{new}, T is the total number of trees in the forest, and f_t is the prediction of the tth tree.

The RF algorithm has several hyperparameters that can be tuned to optimize the performance of the model, including the number of trees, the number of features to select at each node, and the criterion used to measure the impurity. The hyperparameters can be selected using cross-validation on a validation set.

I. RECURRENT NEURAL NETWORK

Recurrent Neural Networks (RNNs) are a type of artificial neural network that can process sequential data by using feedback connections between hidden layers [35]. The feedback connections allow RNNs to use previous inputs to inform the current output, making them well-suited for tasks such as image classification [35], speech recognition, and time series prediction.

Mathematically, an RNN can be represented as a function that takes a sequence of input vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T$ and produces a sequence of output vectors $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T$. At each time step t, the RNN updates its hidden state $\boldsymbol{h}_t$ based on the current input $\boldsymbol{x}_t$ and the previous hidden state $\boldsymbol{h}_{t-1}$, using the following equation:

$$\boldsymbol{h}_t = f(\boldsymbol{W}_{hx}\boldsymbol{x}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h) \tag{26}$$

where $\boldsymbol{W}_{hx}$ and $\boldsymbol{W}_{hh}$ are weight matrices, $\boldsymbol{b}_h$ is a bias vector, and f is a non-linear activation function such as the hyperbolic tangent or the rectified linear unit (ReLU).

The output at each time step is then computed as:

$$\boldsymbol{y}_t = \boldsymbol{W}_{hy}\boldsymbol{h}_t + \boldsymbol{b}_y \tag{27}$$

where $\boldsymbol{W}_{hy}$ is another weight matrix and $\boldsymbol{b}_y$ is another bias vector.

During training, the parameters of the RNN are optimized to minimize a loss function that measures the difference between the predicted outputs and the true outputs. This is typically done using backpropagation through time (BPTT), a variant of the standard backpropagation algorithm that considers the temporal dependencies between the hidden states. The layer architecture analysis of the applied deep learning-based RNN method is discussed in Table 2.

TABLE 2. The layer architecture analysis of applied deep learning based RNN method.

Layer (type)	Output Shape	Param #
RNN layer	(None, 126)	16128
Dense layer	(None, 64)	8128
Output Dense layer	(None, 3)	195
Total Params	24,451	

J. HYPERPARAMETER TUNING

The hyperparameter tuning of each applied machine and deep learning technique is performed through a recursive training and testing process [36]. Table 3 analyzes the selected best-fit hypermeters. The hypermeter tuning helps validate the performance of applied techniques and prevent overfitting issues. High-performance results are achieved by using the fine-tuning process in our proposed study.

TABLE 3. The fine-tuning analysis of applied machine learning techniques.

Technique	Hyperparameters
LR	random_state=0, max_iter=500, solver='liblinear'
SVM	random_state=0, max_iter=50
GNB	var_smoothing=1e-9
RF	n_estimators=20, max_depth=10, random_state=0
RNN	loss = categorical_crossentropy, activation = softmax, optimizer = adam

IV. RESULTS AND DISCUSSIONS

The performance results of applied advanced machine and deep learning techniques are analyzed in this section. The scientific discussions on results and comparisons analysis are also performed to evaluate each applied machine learning technique's performance.

A. EXPERIMENTAL SETUP

Our proposed study experimental setup is analyzed in this section. Python programming 3.0 is utilized to build the applied machine learning techniques. The Google Colab environment [37] with a GPU backend with 13 GB RAM and 90 GB of disk space is used to conduct all our study experiments. Accuracy score, precision score, recall score, and f1 score are the performance metrics utilized for performance evaluations. The performance metrics used for evaluating our applied methods are expressed as:

Let y be the true class labels and $\hat{y}$ be the predicted class labels for a set of samples. We can define the following terms:

- True Positive (TP): the number of samples that are correctly classified as positive.
- False Positive (FP): the number of samples that are incorrectly classified as positive.
- True Negative (TN): the number of samples that are correctly classified as negative.
- False Negative (FN): the number of samples that are incorrectly classified as negative.

Based on these terms, we can define the following classification metrics:

1) ACCURACY SCORE

The Accuracy score measures the proportion of samples that are correctly classified:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

2) PRECISION SCORE

The Precision score measures the proportion of true positives among the samples that are predicted as positive:

$$\text{Precision} = \frac{TP}{TP + FP}.$$

3) RECALL SCORE

The Recall score measures the proportion of true positives among the samples that are actually positive:

$$\text{Recall} = \frac{TP}{TP + FN}.$$

4) F1 SCORE

The F1 score is a harmonic mean of the Precision and Recall scores:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

The F1 score balances Precision and Recall and is often used as a single metric to evaluate the overall performance of a classification model.

B. RESULTS WITH ORIGINAL FEATURES

Table 4 presents a comprehensive analysis of the performance metrics scores for the applied advanced machine learning technique using the original dataset features. Performance evaluation is based on the f1, recall, precision, and accuracy scores, which are essential performance metrics in machine learning. Upon analyzing the results, it is observed that the LR, SVM, and GNB machine-learning techniques scored poorly in comparison to the applied RF technique. The RF technique is the only model that achieved an acceptable score of 0.95. However, it was not the highest score obtained, indicating that further improvements are necessary to achieve optimal performance in detecting driver behavior. Moreover, the analysis of the performance metrics revealed that the applied RF technique outperformed the other techniques in terms of accuracy, recall, and precision. The recall score for the RF model is also high, which means that the model could identify a significant proportion of positive cases. In conclusion, the performance scores for detecting driver behavior still need improvement to achieve the highest.

The research conducted in this study aimed to evaluate the performance of various machine learning models based on f1, recall, precision, and accuracy scores, as presented in Figure 4. The bar chart-based comparison shows that the LR, SVM, and GNB models performed poorly across all metrics. On the other hand, the RF technique yielded highly favorable results, achieving an impressive score of 96%. It is worth noting that the original features used in this analysis did not yield satisfactory results, indicating the need for further feature engineering or selection. These results underscore the significance of careful model selection and feature engineering in achieving optimal machine learning performance.

The time series performance analysis of the used deep learning-based RNN method is visualized in Figure 5. This time series analysis is based on performance metrics evaluated during the training of the RNN method. During the twenty epochs of training, the loss scores are high, and accuracy scores are deficient in comparisons. The analysis shows that the deep learning-based RNN methods achieved

TABLE 4. Performance results analysis of applied methods with original features.

Technique	Accuracy	Target	Precision	Recall	F1-score
LR	0.38	0	0.00	0.00	0.00
		1	0.00	0.00	0.00
		2	0.39	1.00	0.56
		Average	0.15	0.39	0.22
SVM	0.38	0	0.00	0.00	0.00
		1	0.00	0.00	0.00
		2	0.39	1.00	0.56
		Average	0.15	0.39	0.22
GNB	0.38	0	0.00	0.00	0.00
		1	0.00	0.00	0.00
		2	0.39	1.00	0.56
		Average	0.15	0.39	0.22
RF	0.95	0	0.99	1.00	0.99
		1	0.89	0.99	0.94
		2	1.00	0.89	0.94
		Average	0.96	0.96	0.96

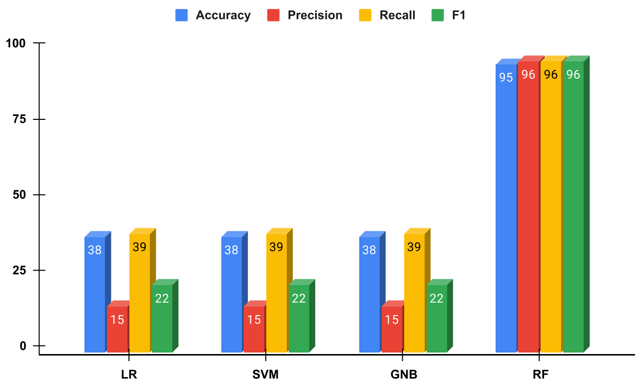


FIGURE 4. The histogram-based performance results analysis of applied methods with original features.

poor performance scores on this dataset. These findings indicate that the deep learning-based RNN method did not achieve satisfactory performance scores for detecting driver behavior.

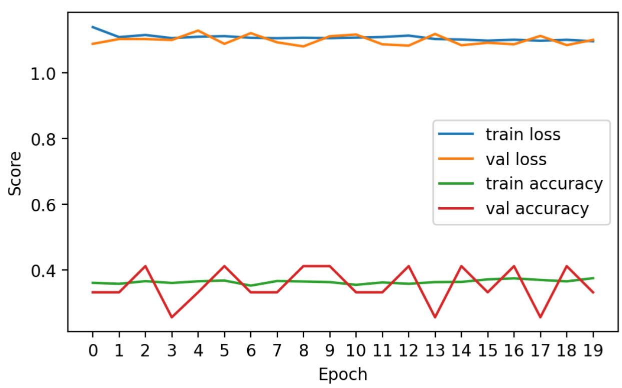


FIGURE 5. The performance results analysis of applied Recurrent Neural Network with original features.

The performance results analysis of applied deep learning-based RNN for unseen testing data is analyzed in Table 5. The unseen testing data analysis demonstrates that the deep learning model RNN achieved abysmal performance accuracy scores of 0.31 for driver behavior detection. The RNN model achieved 0.00 performance scores for f1, recall, and precision for target classes 0 and 2. This analysis concludes that the applied deep learning model shows poor performance scores for detecting driver behavior.

TABLE 5. The performance results analysis of applied Recurrent Neural Network.

Technique	Accuracy	Target	Precision	Recall	F1
RNN	0.31	0	0.00	0.00	0.00
		1	0.32	1.00	0.48
		2	0.00	0.00	0.00
		Average	0.11	0.33	0.16

The comparison results-based radar chart analysis [38] of applied machine learning techniques using original features is discussed in Figure 6. The radar charts are a powerful tool for plotting each model's performance as a point on the radar chart makes it clear which models excel in certain areas and which may need improvement. The analysis visualizes that only the applied RF technique achieved high performance by covering more performance metrics curves area under radar span. The analysis concludes that LR, SVM, and GNB achieved low-performance scores in the radar chart analysis.

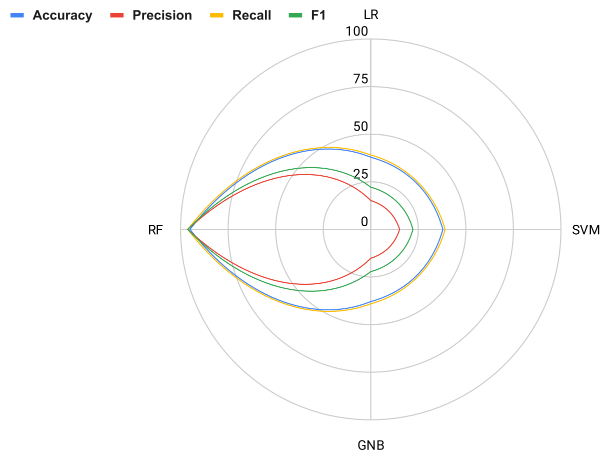


FIGURE 6. The radar chart-based performance results analysis of applied methods with original features.

The performance of each applied method is validated using the K-fold analysis as expressed in Table 6. The analysis demonstrates that using the original feature dataset, the applied machine learning technique scored inferior K-fold cross-validation scores. Only the RF technique achieved 92% of K-fold accuracy scores with fewer standard deviation scores. In conclusion, with original features, the machine learning techniques are less generalized for driver behavior detection.

The complexity computations analysis of applied machine learning techniques with original features is performed in Table 7. The computations analysis demonstrates that using the original features applied to machine learning methods, performance scores are low with high computations scores. The applied RF technique achieved the maximum runtime computations score of 0.145 seconds. The applied GNB

TABLE 6. K-fold cross-validation performance results analysis of applied method with original features.

Technique	Fold	K-fold accuracy	Standard deviations (+/-)
LR	10	0.38	0.0180
SVM	10	0.34	0.0494
GNB	10	0.38	0.0175
RF	10	0.92	0.0326

technique performed minimum runtime computations but achieved lower performance accuracy scores. The analysis concludes that with the original features, all applied techniques achieved high runtime computations in comparisons.

TABLE 7. Computation complexity analysis of applied methods with original features.

Technique	Runtime computations (Seconds)
LR	0.041
SVM	0.046
GNB	0.004
RF	0.145

A detailed confusion matrix analysis is conducted and presented in Figure 7 to assess and summarize the performance of the various methods applied in this study. The analysis revealed that the LR, SVM, and GNB techniques exhibited high target class error rates when using the original features, indicating their suboptimal performance in accurately classifying the data. In contrast, the RF method displayed considerably lower target class error rates, as illustrated in the RF confusion matrix. These findings underscore the importance of selecting appropriate feature engineering and classification techniques for optimal performance in machine learning tasks. Ensemble learning-based feature engineering is needed to further enhance the performance of machine learning models on this dataset.

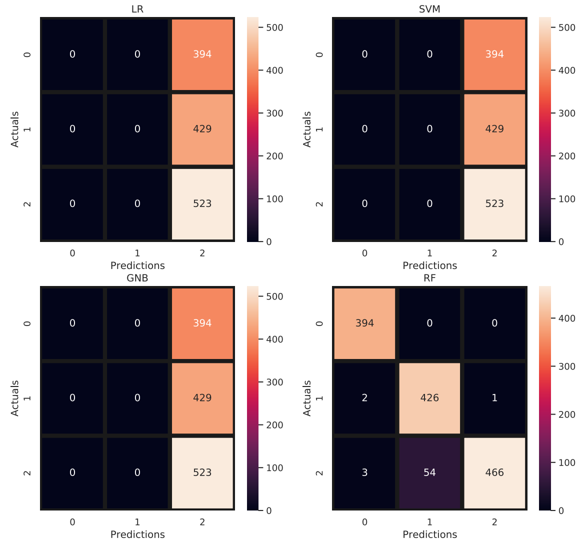


FIGURE 7. Confusion matrix results analysis of applied methods with original features.

TABLE 8. Performance results analysis of applied methods with proposed feature engineering.

Technique	Accuracy	Target	Precision	Recall	F1-score
LR	0.95	0	1.00	1.00	1.00
		1	0.92	0.93	0.93
		2	0.94	0.93	0.94
		Average	0.95	0.95	0.95
SVM	0.95	0	1.00	1.00	1.00
		1	0.92	0.93	0.93
		2	0.94	0.94	0.94
		Average	0.95	0.95	0.95
GNB	0.87	0	1.00	1.00	1.00
		1	0.73	0.98	0.83
		2	0.98	0.70	0.82
		Average	0.90	0.88	0.87
RF	**0.99**	0	1.00	1.00	1.00
		1	1.00	1.00	1.00
		2	1.00	1.00	1.00
		Average	**1.00**	**1.00**	**1.00**

C. RESULTS WITH PROPOSED FEATURE ENGINEERING

Our proposed study aimed to evaluate the performance of various methods for a given task using a novel feature engineering approach. The performance analysis results are presented in Table 8. Interestingly, the applied methods demonstrated a considerable improvement in accuracy scores using the proposed feature engineering approach. Specifically, the LR, SVM, and GNB methods exhibited impressive performance accuracy scores of 0.95, 0.95, and 0.87, respectively. However, the RF method stood out as the best-performing method, with an accuracy score of 0.99, indicating its superiority over other methods. These results clearly suggest that the proposed feature engineering approach effectively enhances the performance of all applied methods across various performance metrics. This analysis highlights the importance of incorporating feature engineering techniques to optimize the performance of machine learning algorithms for detecting driver behaviour.

The histogram-based bar chart analysis is used to evaluate the performance of various machine learning models based on f1, recall, precision, and accuracy scores, as presented in Figure 8. The bar chart-based comparison shows that the LR, SVM, and GNB models performed well across all metrics, not the highest. On the other hand, the RF technique yielded highly favourable results, achieving an impressive score of 99%. These results underscore the significance of novel feature engineering in achieving optimal machine learning performance.

The research findings presented in Figure 9 demonstrate the effectiveness of novel feature engineering approaches in enhancing machine learning techniques. The radar chart analysis reveals that all applied techniques achieved high performance, as evidenced by the significant coverage of performance metrics curves under the radar span. While the LR, SVM, and GNB techniques achieved acceptable scores, they were not the highest. Notably, the RF technique demonstrated exceptional performance, covering a vast area in the radar chart with maximal scores for all metrics. These findings suggest that the proposed feature engineering approach can significantly improve the effectiveness of machine learning techniques, with RF emerging as the most promising technique for achieving optimal performance.

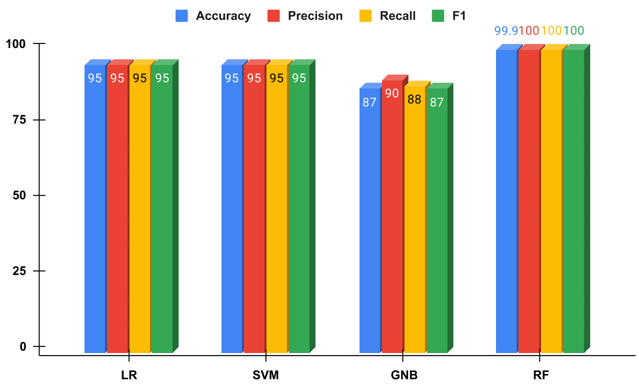


FIGURE 8. The histogram-based performance results analysis of applied methods with proposed feature engineering.

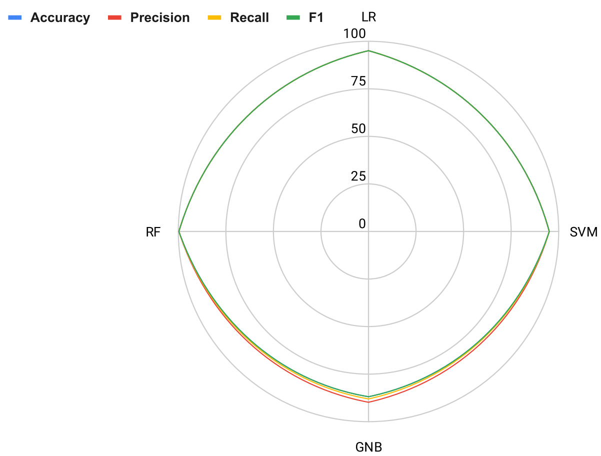


FIGURE 9. The radar chart-based performance results analysis of applied methods with proposed feature engineering.

The effectiveness of each applied method, combined with innovative feature engineering, is thoroughly evaluated using K-fold analysis, as presented in Table 9. The results indicate that the newly extracted feature dataset substantially improved the performance of the applied machine learning techniques, as evidenced by their high K-fold cross-validation scores and minimal standard deviation. The RF technique outperformed all other methods, achieving a remarkable K-fold accuracy score of 0.99, with the smallest standard deviation among all the techniques tested. These findings demonstrate the effectiveness of the proposed feature engineering method in improving the generalizability of machine learning techniques for detecting driver behavior. The results highlight the potential of a novel feature engineering approach to improve the accuracy and reliability of driver behavior detection in a wide range of applications, including transportation safety, driver assistance systems, and autonomous vehicles.

The complexity computations analysis of applied machine learning techniques with proposed feature engineering is performed in Table 10. The computations analysis demonstrates that using the proposed feature engineering approach applied machine learning methods performance very efficiently compared to original features. The applied LR technique

TABLE 9. K-fold cross-validation performance results analysis of applied method with proposed feature engineering.

Technique	Fold	K-fold accuracy	Standard deviations (+/-)
LR	10	0.95	0.0064
SVM	10	0.95	0.0066
GNB	10	0.87	0.0091
RF	**10**	**0.99**	**0.0004**

achieved the maximum runtime computations score of 0.14 seconds. The applied GNB technique performed minimum runtime computations but achieved lower performance accuracy scores. The analysis concludes that the proposed RF technique achieved less runtime computations sore with the proposed feature engineering approach.

TABLE 10. Computation complexity analysis of applied methods with proposed feature engineering.

Technique	Runtime computations (Seconds)
LR	0.140
SVM	0.025
GNB	0.005
RF	0.079

A detailed confusion matrix analysis is conducted and presented in Figure 10 to assess and summarize the performance of the various methods applied with a novel feature engineering approach. The analysis revealed that all applied reduced the target class error rates validating the high-performance scores achieved. As illustrated in the RF confusion matrix, the RF method displayed minimal target class error rates with high-performance class accuracy scores. The analysis concludes that the proposed ensemble learning-based feature engineering can potentially enhance the performance of machine learning models for detecting driver behaviour.

D. FEATURE SPACE COMPARISON ANALYSIS

The feature space representation comparisons analysis based on the original and newly created features set is illustrated in Figure 11. The feature space analysis indicates that the original motion sensors dataset features are not linearly separable, resulting in low-performance scores using machine learning techniques. However, using our proposed feature engineering, the newly created feature set is more linearly separable. In conclusion, the high linearly separability of our propped features results in a high-performance score for detecting driver behavior.

E. COMPARISON WITH OTHER STATE-OF-THE-ART STUDIES

The comparative performance analysis of our proposed approach is performed in Table 11. The previously published studies from the year 2019 to 2023 for driver behavior detection are included in this analysis. Mainly the authors presented deep learning-based hybrid models to achieve good sores. The analysis shows the superior performance of our proposed research study in comparison. Our proposed approach achieved high-performance accuracy scores for

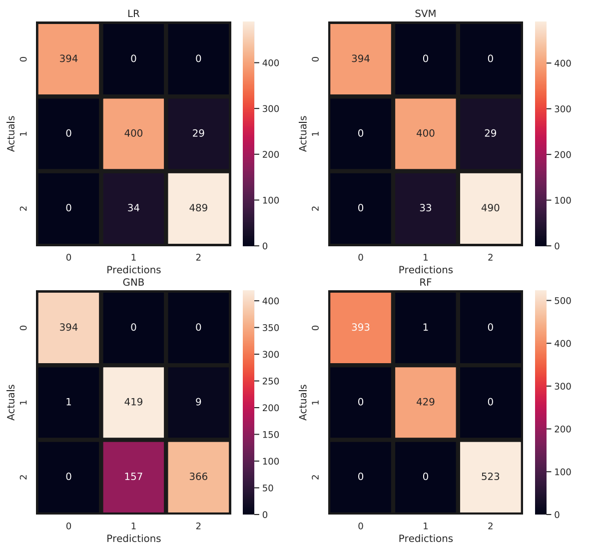


FIGURE 10. Confusion matrix results analysis of applied methods with proposed feature engineering.

TABLE 11. State-of-the-art studies performance score comparisons for detecting driver behavior.

Ref.	Year	Technique	Performance Accuracy (%)
[11]	2019	Neural network	96.1
[12]	2019	DeepConvLSTM	95.7
[10]	2020	Neural network	90.0
[13]	2022	LightGBM	88.0
[23]	2022	ConvLSTM	79.5
[39]	2022	CNN-LSTM	91.9
[14]	2023	RF	95.0
[15]	2023	CNN-LSTM	97.6
Our	**2023**	**RF**	**99.9**

detecting the driver behavior compared to the state-of-the-art studies.

F. DISCUSSION

This study used a novel feature engineering approach for detecting the driver's behavior with high-performance efficiency. Several machine learning and deep learning techniques are applied in comparison to evaluate the performance. The performance of each applied method is validated through the k-fold cross-validations and hyperparameter training. The proposed feature engineering approach is scientifically analyzed with algorithmic mathematical notations.

Extensive results analysis shows that using the proposed novel feature engineering applied methods able to achieve high-performance scores in comparisons. The original dataset feature achieved a low-performance score, which is validated through the feature space analysis. The runtime computations cost analysis shows the efficiency of the proposed approach in this study.

In conclusion, our proposed approach can potentially revolutionize driver behaviour detection with high-performance scores. The performance comparison of our proposed technique is performed with the state-of-the-art approaches, which also shows the superiority of the proposed approach for detecting driver behavior.

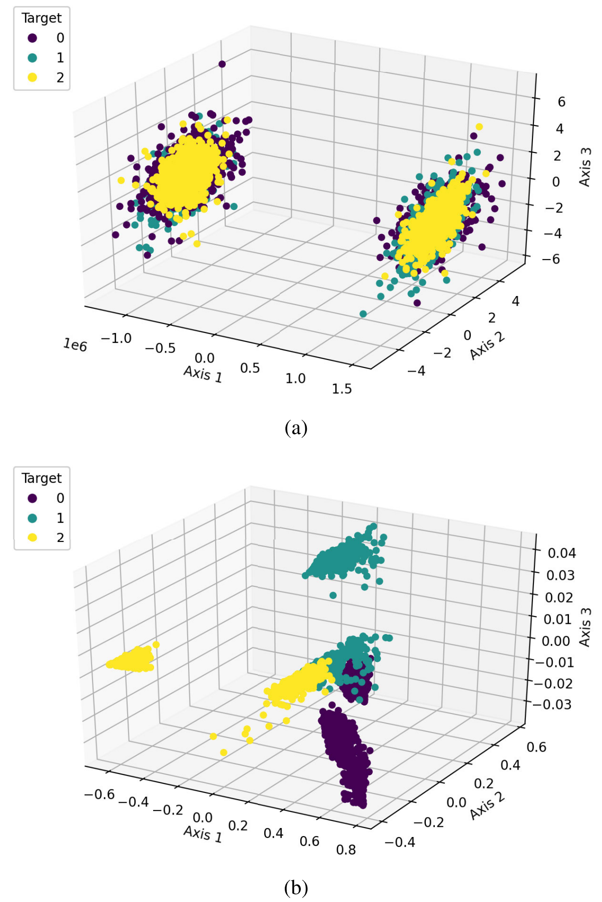


FIGURE 11. The feature space comparison analysis. The regional features representations show in Figure bf(a), and the newly created features representations shows in Figure bf(b).

V. CONCLUSION AND FUTURE WORK

The early detection of driver behavior using advanced machine learning techniques with high-performance scores is proposed in this study. The publicly available smartphone motion sensor data is utilized to conduct our study experiments. Four advanced machine learning-based techniques are applied in comparison. A novel LR-RFC method is proposed that combines the logistic regression and random forest classifier for feature engineering from the motion sensor data. The newly created feature data is used for building the applied machine learning methods for predicting driver behavior. The study results show that the proposed LR-RFC approach achieves the highest performance score. Extensive study experiments demonstrate that the random forest achieved the highest performance score of 99% using the proposed LR-RFC method. The performance is validated using k-fold cross-validation and hyperparameter optimization. The computation complexity and feature space comparison analysis are also performed to validate the high-performance efficiency of the proposed model.

A. LIMITATION AND FUTURE WORK

The current research centers on detecting driver behavior using machine-learning techniques. This is crucial for developing sophisticated neural networks and transfers learning approaches. It should be noted that the sensor data utilized in this study is imbalanced. We plan to implement advanced data balancing techniques to improve performance in future work. Additionally, we intend to construct deep learning-based and transfer learning-based neural network models to achieve higher accuracy scores in driver behavior detection. Furthermore, we aim to employ more advanced motion sensors to collect motion data.

REFERENCES

[1] P. Alrassy, A. W. Smyth, and J. Jang, "Driver behavior indices from large-scale fleet telematics data as surrogate safety measures," *Accident Anal. Prevention*, vol. 179, Jan. 2023, Art. no. 106879.

[2] C. Siebke, M. Bäumler, K. Blenz, M. Lehmann, M. Ringhand, M. Mai, and G. Prokop, "Predicting the impact on road safety of an intersection AEB at urban intersections. Using a novel virtual test field for the assessment of conflict prevention between cyclists/pedelecs and cars," *Transp. Res. Interdiscipl. Perspect.*, vol. 17, Jan. 2023, Art. no. 100728.

[3] L. Liu, F. Villavicencio, D. Yeung, J. Perin, G. Lopez, K. L. Strong, and R. E. Black, "National, regional, and global causes of mortality in 5–19-year-olds from 2000 to 2019: A systematic analysis," *Lancet Global Health*, vol. 10, no. 3, pp. e337–e347, Mar. 2022.

[4] Y. Ge, X. Luo, and W. Qu, "Impact of feedback content on dangerous driving behaviours based on the triangle model of responsibility," *Transp. Res. F, Traffic Psychol. Behaviour*, vol. 93, pp. 1–10, Feb. 2023.

[5] K. Mohammed, M. Abdelhafid, K. Kamal, N. Ismail, and A. Ilias, "Intelligent driver monitoring system: An Internet of Things-based system for tracking and identifying the driving behavior," *Comput. Standards Interfaces*, vol. 84, Mar. 2023, Art. no. 103704.

[6] A. Misra, S. Samuel, S. Cao, and K. Shariatmadari, "Detection of driver cognitive distraction using machine learning methods," *IEEE Access*, vol. 11, pp. 18000–18012, 2023.

[7] A. Mukherjee, A. K. Panja, and N. Dey, *A Beginner's Guide to Data Agglomeration and Intelligent Sensing*. New York, NY, USA: Academic Press, 2020.

[8] S. Chen, H. Yao, F. Qiao, Y. Ma, Y. Wu, and J. Lu, "Vehicles driving behavior recognition based on transfer learning," *Expert Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119254.

[9] H. Cao, Y. Wang, J. Bi, H. Qi, and M. Sun, "Robust heading estimation algorithm for Android smartphones," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

[10] E. Lattanzi, G. Castellucci, and V. Freschi, "Improving machine learning identification of unsafe driver behavior by means of sensor fusion," *Appl. Sci.*, vol. 10, no. 18, p. 6417, Sep. 2020.

[11] H. Nassuna, O. S. Eyobu, J.-H. Kim, and D. Lee, "Feature selection based on variance distribution of power spectral density for driving behavior recognition," in *Proc. 14th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2019, pp. 335–338.

[12] J. Zhang, Z. Wu, F. Li, J. Luo, T. Ren, S. Hu, W. Li, and W. Li, "Attention-based convolutional and recurrent neural networks for driving behavior recognition using smartphone sensor data," *IEEE Access*, vol. 7, pp. 148031–148046, 2019.

[13] S. B. Brahim, H. Ghazzai, H. Besbes, and Y. Massoud, "A machine learning smartphone-based sensing for driver behavior classification," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2022, pp. 610–614.

[14] M. S. N. Al-Din, "Driving maneuvers recognition and classification using a hyprid pattern matching and machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 2, pp. 247–256, 2023.

[15] J. Liu, Y. Liu, D. Li, H. Wang, X. Huang, and L. Song, "DSDCLA: Driving style detection via hybrid CNN-LSTM with multi-level attention fusion," *Appl. Intell.*, vol. 53, pp. 19237–19254, Feb. 2023.

[16] W. A. Al-Hussein, L. Y. Por, M. L. M. Kiah, and B. B. Zaidan, "Driver behavior profiling and recognition using deep-learning methods: In accordance with traffic regulations and experts guidelines," *Int. J. Environ. Res. Public Health*, vol. 19, no. 3, p. 1470, Jan. 2022.

[17] H. Zhang and R. Fu, "A hybrid approach for turning intention prediction based on time series forecasting and deep learning," *Sensors*, vol. 20, no. 17, p. 4887, Aug. 2020.

[18] M. Aljasim and R. Kashef, "E2DR: A deep learning ensemble-based driver distraction detection with recommendations model," *Sensors*, vol. 22, no. 5, p. 1858, Feb. 2022.

[19] Y. Lu, X. Fu, E. Guo, and F. Tang, "XGBoost algorithm-based monitoring model for urban driving stress: Combining driving behaviour, driving environment, and route familiarity," *IEEE Access*, vol. 9, pp. 21921–21938, 2021.

[20] M. Hamraz, N. Gul, M. Raza, D. M. Khan, U. Khalil, S. Zubair, and Z. Khan, "Robust proportional overlapping analysis for feature selection in binary classification within functional genomic experiments," *PeerJ Comput. Sci.*, vol. 7, p. e562, Jun. 2021.

[21] M. Hamraz, A. Ali, W. K. Mashwani, S. Aldahmani, and Z. Khan, "Feature selection for high dimensional microarray gene expression data via weighted signal to noise ratio," *PLoS One*, vol. 18, no. 4, Apr. 2023, Art. no. e0284619.

[22] A. Mukherjee and N. Dey, *Smart Computing With Open Source Platforms*. Boca Raton, FL, USA: CRC Press, 2019.

[23] I. Cojocaru and P. S. Popescu, "Building a driving behaviour dataset," in *Proc. Int. Conf. Human-Computer Interact. (RoCHI)*, 2022, pp. 101–107.

[24] A. Raza, H. U. R. Siddiqui, K. Munir, M. Almutairi, F. Rustam, and I. Ashraf, "Ensemble learning-based feature engineering to analyze maternal health during pregnancy and health risk prediction," *PLoS One*, vol. 17, no. 11, Nov. 2022, Art. no. e0276525.

[25] A. Raza, F. Rustam, H. U. R. Siddiqui, I. D. L. T. Diez, and I. Ashraf, "Predicting microbe organisms using data of living micro forms of life and hybrid microbes classifier," *PLoS One*, vol. 18, no. 4, Apr. 2023, Art. no. e0284522.

[26] G. Prakash, C. Perera, and V. Jhanji, "Comparison of machine learning-based algorithms using corneal asymmetry vs. single-metric parameters for keratoconus detection," *Graefe's Archive Clin. Exp. Ophthalmol.*, vol. 261, no. 8, pp. 1–8, Apr. 2023.

[27] A. Raza, M. R. Al Nasar, E. S. Hanandeh, R. A. Zitar, A. Y. Nasereddin, and L. Abualigah, "A novel methodology for human kinematics motion detection based on smartphones sensor data using artificial intelligence," *Technologies*, vol. 11, no. 2, p. 55, Apr. 2023.

[28] N. Thalji, A. Raza, M. S. Islam, N. A. Samee, and M. M. Jamjoom, "AE-Net: Novel autoencoder based deep features for SQL injection attack detection," *IEEE Access*, vol. 11, pp. 135507–135516, 2023, doi: 10.1109/ACCESS.2023.3337645.

[29] A. Raza, K. Munir, M. S. Almutairi, and R. Sehar, "Novel class probability features for optimizing network attack detection with machine learning," *IEEE Access*, vol. 11, pp. 98685–98694, 2023.

[30] A. Raza, A. M. Qadri, I. Akhtar, N. A. Samee, and M. Alabdulhafith, "LogRF: An approach to human pose estimation using skeleton landmarks for physiotherapy fitness exercise correction," *IEEE Access*, vol. 11, pp. 107930–107939, 2023.

[31] M. Gouda, A. Lugnan, J. Dambre, G. V. D. Branden, C. Posch, and P. Bienstman, "Improving the classification accuracy in label-free flow cytometry using event-based vision and simple logistic regression," *IEEE J. Sel. Topics Quantum Electron.*, vol. 29, no. 2, pp. 1–8, Mar. 2023.

[32] R. Sethuraman, S. Sellappan, J. Shunmugiah, N. Subbiah, V. Govindarajan, and S. Neelagandan, "An optimized AdaBoost multi-class support vector machine for driver behavior monitoring in the advanced driver assistance systems," *Expert Syst. Appl.*, vol. 212, Feb. 2023, Art. no. 118618.

[33] J. Li, Z. Li, C. Wen, Q. Peng, and P. Huang, "Train operation conflict detection for high-speed railways: A naïve Bayes approach," *Int. J. Rail Transp.*, vol. 11, no. 2, pp. 188–206, 2023.

[34] S. Kianoush, S. Savazzi, V. Rampa, L. Costa, and D. Tolochenko, "A random forest approach to body motion detection: Multisensory fusion and edge processing," *IEEE Sensors J.*, vol. 23, no. 4, pp. 3801–3814, Feb. 2023.

[35] A. Raza, K. Munir, M. Almutairi, F. Younas, M. M. S. Fareed, and G. Ahmed, "A novel approach to classify telescopic sensors data using bidirectional-gated recurrent neural networks," *Appl. Sci.*, vol. 12, no. 20, p. 10268, Oct. 2022.

[36] A. Raza, K. Munir, M. Almutairi, F. Younas, and M. M. S. Fareed, "Predicting employee attrition using machine learning approaches," *Appl. Sci.*, vol. 12, no. 13, p. 6424, Jun. 2022.

[37] N. Bhargava, P. S. Rathore, and A. Bhowmick, "Study of X ray detection using CNN in machine learning," in *Proc. 1st Int. Conf. Advancements Smart Comput. Inf. Secur.*, Rajkot, India. New York, NY, USA: Springer, 2023, pp. 295–303.

[38] C. Chen, X. Xu, X. Ma, L. Yang, G. Wang, and X. Ye, ''Comprehensive performance evaluation of intrusion detection model based on radar chart method,'' in *Proc. 2nd Int. Conf. Electron. Inf. Eng. Comput. Technol. (EIECT)*, Oct. 2022, pp. 44–47.

[39] I. Cojocaru, P. S. Popescu, and M. C. Mihaăescu, ''Driver behaviour analysis based on deep learning algorithms,'' in *Proc. Int. Conf. Human-Computer Interact. (RoCHI)*, 2022, pp. 108–114.

ALI RAZA received the B.Sc. degree in computer science from the Institute of Computer Science, Khwaja Fareed University of Engineering and Information Technology (KFUEIT), Rahim Yar Khan, Pakistan, in 2021, and the M.S. degree in computer science from KFUEIT. He has published several articles in reputed journals. His current research interests include data science, artificial intelligence, data mining, natural language processing, machine learning, deep learning, and image processing.

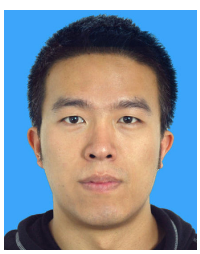
IQRA AKHTAR received the B.Sc. degree in physics from the Institute of Physics, Khwaja Fareed University of Engineering and Information Technology (KFUEIT), Rahim Yar Khan, Pakistan, where she is currently pursuing the M.S. degree in electronics with the Department of Electrical and Biomedical Engineering. Her current research interests include artificial intelligence, physics applications, and physical science.

LAITH ABUALIGAH received the degree in computer information system from Al-Albayt University, Jordan, in 2011, the master's degree in computer science from Al-Albayt University, in 2014, and the Ph.D. degree from the School of Computer Science, Universiti Sains Malaysia (USM), Malaysia, in 2018. He is an Associate Professor with the Prince Hussein Bin Abdullah College for Information Technology, Al-Albayt University. He is also a Distinguished Researcher with the School of Computer Science, USM. According to the report published by Clarivate, he was one of the highly cited researchers in 2021 and 2022, and the 1% influential researchers, which depicts the 6,938 top scientists in the world, and a first researcher in the domain of computer science in Jordan, in 2021. According to the report published by Stanford University, in 2020, he is one of the 2% influential scholars, which depicts the 100,000 top scientists in the world. He has published more than 350 journal articles and books, which collectively have been cited more than 12 500 times (H-index is 53). His main research interests include arithmetic optimization algorithm (AOA), bio-inspired computing, nature-inspired computing, swarm intelligence, artificial intelligence, meta-heuristic modeling, optimization algorithms, evolutionary computations, information retrieval, text clustering, feature selection, combinatorial problems, optimization, advanced machine learning, big data, and natural language processing. He currently serves as an Associate Editor for the *Journal of Cluster Computing* (Springer), the *Journal of Soft Computing* (Springer), and the *Journal of Engineering Applications of Artificial Intelligence* (Elsevier).

RAED ABU ZITAR is a Professor with the Sorbonne Centre for Artificial Intelligence (SCAI Abu Dhabi), Sorbonne University Abu Dhabi. His main research interests include artificial intelligence, machine learning, pattern recognition, artificial intelligence in software engineering, robotics, computer networks modeling, corpuses development, languages dictionaries, emotions modeling, emotional agents, genetic algorithms, neural networks, intelligent control, scheduling methods, speech recognition, evolutionary based applications, and fuzzy logic.

MOHAMED SHARAF received the Industrial Engineering degree from Chiba University, Japan. He is the Head of the Development and Quality Unit, College of Engineering, King Saud University. He has published more than 30 papers in the areas of spare parts control, quality management, maintenance, six sigma methodology, and academic accreditation.

MOHAMMAD SH. DAOUD received the Ph.D. degree in computer science from De Montfort University, U.K. He is currently an Associate Professor with the College of Engineering, Al Ain University, United Arab Emirates. His research interests include artificial intelligence, swarm systems, secured systems and networks, and smart applications.



HEMING JIA (Member, IEEE) received the Ph.D. degree in system engineering from Harbin Engineering University, China, in 2012. He is currently a Professor with Sanming University. His research interests include nonlinear control theory and application, image segmentation, and swarm optimization algorithm.