IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 March 2024, accepted 2 May 2024, date of publication 8 May 2024, date of current version 15 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3397725

RESEARCH ARTICLE

Elevating Driver Behavior Understanding With RKnD: A Novel Probabilistic Feature Engineering Approach

MOHAMMAD SHARIFUL ISLAM[1], MOHAMMAD ABU TAREQ RONY[2], MEJDL SAFRAN[3], SULTAN ALFARHOOD[3], AND DUNREN CHE[4]

[1]Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali 3814, Bangladesh
[2]Department of Statistics, Noakhali Science and Technology University, Noakhali 3814, Bangladesh
[3]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[4]School of Computing, Southern Illinois University, Carbondale, IL 62901, USA

Corresponding authors: Sultan Alfarhood (sultanf@ksu.edu.sa) and Mohammad Abu Tareq Rony (rony1513@student.nstu.edu.bd)

This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSPD2024R1027.

ABSTRACT Early detection of driver behavior is a pivotal aspect in enhancing road safety, focusing on identifying and mitigating risky driving patterns before they lead to accidents. The use of smartphone sensors for data acquisition marks a significant advancement in this field. It allows for continuous, real-time monitoring of driving patterns without the need for specialized equipment. In this study, we leverage a publicly available smartphone motion sensor dataset, utilizing accelerometer and gyroscope data from a Samsung Galaxy S21 to analyze driving behaviors classified as slow, normal, and aggressive. This research introduces a novel feature engineering technique named the RKnD (Random forest, K-nearest classifier, Decision tree) probabilistic feature engineering technique, which integrates three prominent machine learning (ML) models. This blend offers a robust analysis of driver behavior, leveraging the strengths of each algorithm. This paper emphasizes the importance of data balancing in machine learning, employing the Synthetic Minority Oversampling Technique (SMOTE) to enhance the reliability of the predictions. Furthermore, k-fold cross-validation is used to ensure the model's consistency and accuracy across original features and the proposed RKnD probabilistic features of the data sets. By achieving such high accuracy, the study demonstrates the potential of smartphone-based systems to significantly improve road safety. This paper introduces a novel approach utilizing smartphone motion sensor data to detect driver behaviors with a remarkable accuracy rate of 99.63%. This research stands out for its application of machine learning techniques in a practical, accessible manner. This pioneering approach named RKnD feature engineering sets a new standard in the realm of smart transportation systems, opening avenues for further innovations in the field, and filling a gap in road safety analysis to avoid road accidents. Future research on RKnD should streamline its algorithm for real-time use, diversify datasets, integrate advanced Deep Learning for complex pattern detection, and undertake real-world testing to validate practicality and uncover challenges.

INDEX TERMS Driver behavior, smartphone sensors, deep learning, feature engineering, road safety.

I. INTRODUCTION

Driving behaviors refer to the observable actions and reactions of drivers in specific driving contexts, including their responses to traffic conditions, environmental factors, and vehicle performance. These behaviors can range from

The associate editor coordinating the review of this manuscript and approving it for publication was Juan A. Lara.

aggressive to defensive driving tactics, distinguishing them from driving styles, which encompass more consistent and characteristic patterns of driving preferences over time. Understanding and differentiating between these concepts is crucial for developing models that accurately predict behaviors leading to increased accident risks. The application of ML and Deep Learning (DL) in the early detection of driver behavior has marked a significant shift in the field. Various studies have implemented ML algorithms, achieving different levels of success. However, these methods often require large, diverse datasets for training and may struggle with the dynamic nature of data [1]. This study introduces a novel architecture for the early prediction of driver behavior, which represents an innovative integration of two ML models and a DL model. This feature engineering approach aims to harness the strengths of each model to enhance overall detection accuracy and efficiency. To address the pressing issue of road accidents caused by aggressive and irrational driving, there is an urgent need for reliable techniques to follow and recognize driver behavior. Recent developments in Internet of Things (IoT) technologies have opened avenues for the remote monitoring and identification of driving patterns. Through the examination of changes in driving-related data, scholars are investigating inventive methods to improve overall road safety [2]. According to the extensive global status report on road safety by the World Health Organization, traffic incidents are ranked as the eighth major contributor to global fatalities. Significantly, nearly one-fifth of these traffic accidents can be directly attributed to driver distractions [3]. In the realm of vehicle technologies research, the focus has been on developing smart systems to enhance vehicle efficiency and driver experience. Recent studies have leveraged smartphones for robust data collection, marking a shift towards comprehensive assessments of vehicle performance and driver behavior [4]. Hence, researchers and transportation experts strive to enhance road safety and reduce accidents. The growing availability of sensor technologies and recent advancements in ML and DL have propelled data-driven road safety research to the forefront [5]. Owing to the intricacy of conducting experiments aimed at testing driving behaviors and the significant expenses entailed in data collection, particularly for heavy-duty freight vehicles, the development of a model capable of achieving high performance in accurately discerning driving behavior patterns from a limited dataset presents a formidable challenge [6]. In contemporary times, a growing trend involves the concept of smart cities, which advocate for the integration of sensors in vehicles to enhance intercommunication among various vehicle types [7]. The transportation industry's strategic focus on performance optimization and cost reduction has propelled the integration of advanced technologies, specifically the Internet of Things (IoT) and ML. This integration has been prompted by the observed correlation between driving behavior and its impact on fuel consumption and emissions, necessitating the classification of distinct driving patterns among individuals [8]. Driving behavior profoundly affects mobility, safety, energy efficiency, and emissions. With the rise of connected vehicles equipped with high-resolution (10 Hz) data, we can now accurately classify driving styles [9]. The significant contributions of our proposed innovative research are as follows:

- A novel feature engineering approach RKnD is proposed in this study, which combines three ML models. The goal of this feature engineering approach is to leverage the strengths of each model to enhance driver behavior detection performance.
- To address data imbalance, we utilized the Synthetic Minority Oversampling Technique (SMOTE).
- We have applied eight advanced ML and DL models for evaluating RKnD techniques.
- We utilize accuracy, precision, recall, and the F1 score to assess the performance of both our ML and DL models. Moreover, this study provides comparative results analysis between the performance of our proposed RKnD probabilistic features and the original features.
- The efficacy of each model is tested by using k-fold cross-validation with further enhanced by optimizing hyperparameters.

The remaining sections of the manuscript are as follows: Section II based on the related literature analysis. Section III presents our novel proposed network for driver behavior detection. In Section IV, the results obtained from the implementation of various ML techniques are comparatively assessed. Section V summarizes the findings of this research study.

II. LITERATURE REVIEW

The demand for preventing road accidents has increased with the rise in early detection of driver behavior. This has led to the exploration of ML and DL techniques for improving the detection of driver behavior within large datasets. This study section aims to furnish a thorough review of the pertinent literature concerning the prediction of driver behavior utilizing machine learning and sensor data. This entails an examination of the employed methodologies, the attained accuracy rates, and a discussion of the limitations inherent in current approaches, as delineated in the accompanying Table 1.

In this research [10], the authors proposed an innovative approach for early detection of abnormal driving behaviors, a crucial factor in reducing traffic accidents. Their method, Serial-Feature Network (SF-Net), leverages smartphone inertial sensors and considers the continuity of driving events. The dataset included GPS data, 3-axis acceleration, and gyroscope data. SF-Net preprocesses the data, combining current sensor data with information from adjacent time frames. Deep convolutional neural networks(CNNs) extract features, enabling the recognition of ten driving behaviors based on multi-level and multi-time information fusion. Impressively, field tests yielded remarkable results, with SF-Net achieving a 97.1% accuracy rate and a 98.4% recall rate, surpassing other network models. Even with limited training samples, SF-Net remained stable and maintained high recognition rates, showcasing its potential for practical applications in enhancing road safety.

In a separate investigation conducted by [11], researchers utilized high-resolution driving behavior data sourced from 303 drivers' smartphones. Their objective was to scrutinize driver behavior at both the road segment and junction levels. This dataset was enriched with additional information encompassing traffic patterns, and road geometry characteristics, and subsequently visualized spatially using Geographical Information System (GIS) software. The primary emphasis of the study revolved around the identification and mapping of harsh driver behavior events, encompassing 8,592 instances of harsh accelerations and 3,946 occurrences of abrupt braking events, specifically pinpointing their locations within Athens, Greece. To dissect and interpret the data effectively, the research team devised and implemented two multiple-linear regression models along with two log-linear regression models. Results revealed that traffic characteristics, such as average traffic flow per lane and average occupancy in junctions, had a statistically significant impact on the frequency of harsh events, surpassing the influence of road geometry and driver behavior factors.

This study [12] presented an innovative approach for early detection of transportation modes using smartphone sensor data, a critical component in intelligent transportation systems. The study aimed to strike a balance between accuracy and earliness, crucial for real-time decision-making in systems like driver assistance. They developed a hybrid DL classifier that harnessed the power of CNNs, recurrent neural networks, and deep neural networks to uncover hidden temporal correlations within sensory time series data. Additionally, a decision policy was introduced to predict transportation modes with an acceptable trade-off. The model was evaluated using two publicly available supervised datasets and demonstrated excellent performance in terms of accuracy and earliness.

In their research conducted in [13] the authors addressed the prevalent issue of road accidents stemming from human fatigue and inattention by leveraging ML technology. They focused on identifying unsafe driver behaviors through the fusion of in-vehicle sensor signals, such as vehicle speed and engine parameters, with external sensors like gyroscope and magnetometer. Feature engineering was employed to accurately describe driver behavior, and a support vector machine (SVM) and artificial neural network were trained and tested using data from over 200 km. The reference data for evaluation was established through a methodology grounded in vehicle speed and acceleration. The results demonstrated the efficacy of the approach, achieving an average accuracy of approximately 88% with the SVM classifier and around 90% using the neural network, highlighting its potential in identifying unsafe driver behaviors.

In this article [14], the authors proposed an ML-based approach to identify safe and unsafe driving behaviors using in-vehicle sensor data. They computed descriptive features from these signals and employed SVMs and feed-forward neural networks for classification. The evaluation on a dataset with over 26 hours of driving data yields an average accuracy exceeding 90% for both classifiers. The McNemar test shows no significant performance difference between the models at the 0.05 significance level. This research demonstrates the potential of using in-vehicle sensor data to effectively identify unsafe driving behaviors.

In this investigation [15], researchers in Indonesia addressed motorcycle safety concerns by using smartphone sensors, including an accelerometer and a gyroscope, to monitor and warn motorcyclists about their driving behavior. They developed an application that categorized driver statuses into various categories, such as normal, zig-zag, sleepy, turns, braking, acceleration, and speed bumps, using an ML approach with an Artificial Neural Network (ANN) algorithm. Impressively, the system achieved a high accuracy level of 96.2% in recognizing these behaviors. This research demonstrates the potential of leveraging smartphone sensors and ANN technology to enhance motorcycle safety in regions where motorcycles are a prevalent mode of transportation.

In the work conducted by [16], a system leveraging bio-signals was designed for the real-time identification of aggressive driving behaviors within the context of the Internet of Medical Things (IoMT). The approach involved the utilization of a deep convolutional neural network (DCNN) model, seamlessly integrated with edge and cloud technologies. The system comprised three distinct modules: a vehicle-based detection module, a cloud-based training module, and an analysis module connected to a monitoring environment through a telecommunication network. Evaluation of processed bio-signal datasets yielded promising results, with the DCNN model achieving validation accuracies of 73.02% and 79.15% on two different datasets. This research demonstrates the feasibility of using DCNNs to detect aggressive driving behaviors using bio-signal data in the IoMT setting.

These [17], researchers used smartwatches to passively sense and classify driver activities, outside events, and road attributes. Analyzing data from 15 participants in a naturalistic driving study, they achieved impressive results with average F1 scores of 94.55%, 98.27%, and 97.86%, respectively, through 10-fold cross-validation. This innovative approach offers a privacy-aware and effective method for enhancing context-aware driving data collection and analysis in semi-automated and autonomous vehicles.

This study by [18], presented a system designed for the automated extraction of proprietary in-vehicle information through the analysis of sensor data. The system estimates driving status and segments in-vehicle CAN frames, achieving an 84.20% accuracy in estimating driving conditions and an 82.31% accuracy in extracting in-vehicle information through real vehicle experiments in an urban environment. This research offers a viable approach for automatic proprietary in-vehicle data extraction.

In this research article [19], researchers detected driver inattention using large-scale vehicle trajectory data and identified its impact on driver behavior. They focused on common inattentive events and used a deep CNN with data augmentation techniques for detection. Additionally, an LSTM-based model predicted abnormal driving operations resulting from inattention. The study achieved a 92.27% accuracy in detecting inattentive driving and a 91.67% accuracy in predicting abnormal driving. This research has the potential to improve driving habits and road safety.

In this study [20], researchers used miniature inertial measurement units (IMUs) to monitor real-time driving behavior. They employed a deep neural network-based approach to identify different driving actions based on joint angle series, achieving recognition rates exceeding 99% in experiments. This method shows promise for driving training and guiding novice drivers. In this investigation [21], the researchers used tri-axial smartphone accelerometer signals to identify and verify drivers. Their approach included ResNet-50 and Stacked Gated Recurrent Units (SGRUs) for identification and Siamese Neural Networks and Triplet Loss Training for verification. With a dataset of 25 drivers and over 20,000 journeys, the results were impressive: 71.89% top-1 and 92.02% top-5 accuracies for identification, and a 74.09% F1 score for verification. This approach, based solely on smartphone accelerometers, shows promise for efficient driver monitoring applications.

This study [22] uses smartphone motion sensor data to analyze driver behavior efficiently. With the LR-RFC (Logistic Regression Random Forest Classifier) method, it achieves a remarkable 99% performance score, validating results through rigorous techniques like k-fold cross-validation. By generating probabilistic features from sensor data, the LR-RFC model enhances behavior prediction significantly. This innovative approach signals a crucial advancement in early driver behavior detection, promising to mitigate road accidents and associated costs.

In the paper [23] a novel approach utilizing CNNs, Fuzzy Logical Feature Selection (FLFS) and Optimized Spectral Neural Classification (OSNCA) is introduced for improving transportation safety by analyzing driver behavior. The study employs advanced techniques with a rich dataset collected from vehicle onboard diagnostics ports, encompassing vital parameters like fuel consumption and vehicle dynamics. The method involves intricate data segmentation, utilizing fuzzy logic for feature selection, and spectral neural classification for precise behavior categorization. The performance of this innovative approach is remarkable, demonstrating an accuracy of 98.9%, with precision and recall rates of 88.3% and 93.9% respectively for 20 drivers, significantly outperforming traditional methods like SVM, PCA, and GA-FCM. This highlights its potential in significantly enhance transportation safety by providing a deeper, more nuanced understanding of driver behavior.

A. RESEARCH GAP AND QUESTIONS

In the field of Early Detection of Driver Behavior, the literature review identifies a significant research gap in the realm of Behavior detection, particularly in scalability, adaptability, computational efficiency, and real-time application. Our research addresses two primary research questions we have identified from literature analysis:

- Does use a probabilistic feature engineering technique that integrates the prominent machine learning algorithms improve the accuracy of detecting drivers' behavior compared to using the original features?
- What are the most fruitful ML and DL approaches for the detection of driver's Behavior?

To bridge this gap, our paper introduces an advanced feature engineering technique, RKnD. This model aims to enhance accuracy and efficiency, catering to the Detection of Driver Behavior. By integrating diverse ML techniques, RKnD offers a novel and robust solution in the evolving landscape of road accidents.

III. PROPOSED METHODOLOGY

A comprehensive explanation of the study techniques and their associated workflow is provided in this section. We conducted an analysis of the dataset utilized in constructing the applied methods through evaluation, employing a variety of hyperparameters to assess their performance. To offer a thorough understanding of the feature engineering approach's functionality, a description of the suggested approach's architecture and its mathematical algorithm is also included. Figure 1 illustrates the analytical approach proposed for this study. To carry out the recommended experiments, we utilize a smartphone motion sensor dataset. Initially, the dataset was imbalanced, and we addressed this issue by employing the SMOTE technique to balance the dataset. Subsequently, the balanced dataset is imported and divided into two parts with ratios of 80(train) and 20(test). Through the combination of RF, KNC, and DT approaches, we devised a novel RKnD strategy. The proposed novel approach incorporates a fully optimized set of hyperparameters, showcasing effective efficiency for predicting driver behavior. The performance results of the suggested feature engineering techniques are evaluated and utilized for identifying driver behavior.

A. DATASET DESCRIPTION

In this research, we used a publicly available [24] smartphone-based sensor data to evaluate the experiments. The dataset focuses on analyzing driving behavior, crucial for developing advanced Driving Assistance and Intelligent Transportation Systems. It employs smartphone sensors specifically, an accelerometer and gyroscope to track a vehicle's movement across three axes: longitudinal, lateral, and vertical. These sensors are chosen for their widespread availability in smartphones and their ability to provide detailed data on sudden movements and orientation, rather than relying on GPS which only offers speed data. The dataset was collected using a Samsung Galaxy S10 and a Dacia

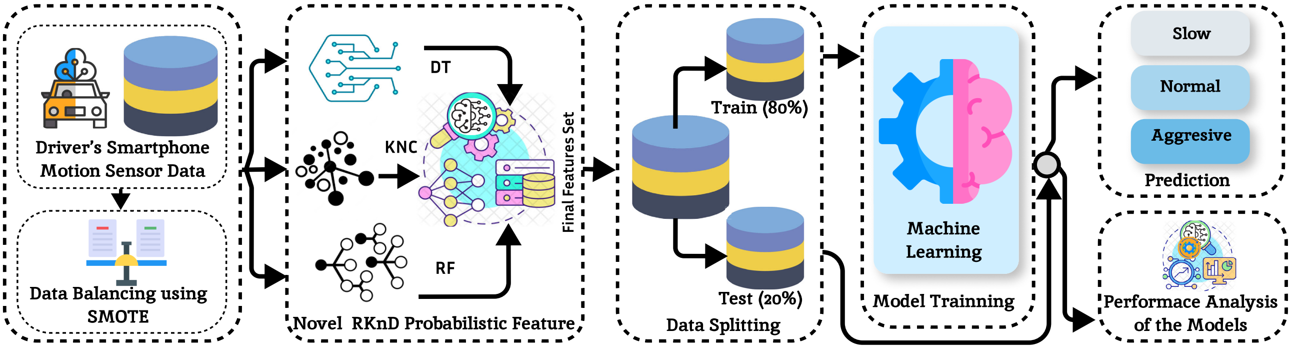

Driver's Smartphone Motion Sensor Data
Data Balancing using SMOTE
DT
KNC
RF
Final Features Set
Novel RKnD Probabilistic Feature
Train (80%)
Test (20%)
Data Splitting
Machine Learning
Model Trainning
Slow
Normal
Aggresive
Prediction
Performace Analysis of the Models

FIGURE 1. The workflow of our suggested method for early detection of driver behavior.

Sandero 1.4 MPI car, selected for its mainstream engine representative of typical ride-sharing vehicles. This choice aids in understanding driving styles in average cars, although more powerful cars could yield clearer distinctions in driving styles. The dataset includes two files, one for training and one for testing. This data aims to detect driving styles based on sudden movements, offering insights into driver comfort and experience. The dataset comprises various elements, encompassing a temporal record, and acceleration parameters in the X dimensions, Y dimensions, and Z dimensions. Additionally, the dataset is annotated with three labels: slow, normal, and aggressive [24]. These labels are defined based on specific criteria related to the driver's acceleration and deceleration patterns, as well as the vehicular orientation changes:

- Slow driving is defined by consistently lower-than-average speeds, indicating a cautious approach with gradual accelerations and decelerations.
- Normal driving encapsulates average driving events, characterized by standard adherence to traffic norms and moderate execution of driving maneuvers.
- Aggressive driving is marked by sudden lateral movements (left or right turns), and abrupt accelerations and decelerations. This category signifies a high-risk driving style characterized by rapid changes in speed and direction.

The primary data that we collected was imbalanced. The imbalance within the dataset is depicted in Figure 3, highlighting the distribution across different labels [22]. The findings indicate that there are 2,604 sensor readings classified under the "slow" category, 2,197 readings under "normal", and 1,927 readings fall into the "aggressive" category. This distribution points to a lack of balance within the dataset.

Figure 2 displays a pair plot of sensor data, depicting the relationships between acceleration (AccX, AccY, AccZ) and rotation (GyroX, GyroY, GyroZ) along three axes. Data points are color-coded to represent three different classes of motion behavior: normal (blue), slow (red), and aggressive (green), allowing for visual comparison of the variables across these categories. Table 2 indicates the five samples of data.

B. SYNTHETIC MINORITY OVER-SAMPLING(SMOTE)

To address the class imbalance in ML datasets, we applied the SMOTE technique which is very popular [25]. When one class has significantly fewer instances than another, models can become biased towards the majority class. We solve this problem by creating synthetic examples of the minority points class, which balances the dataset using the SMOTE technique. The method functions by identifying the k-nearest neighbors among instances of the minority class and subsequently generating novel, interpolated data points located in between these neighbors. This approach not only helps achieve a balanced dataset. Figure 3 shows the imbalanced dataset and balanced dataset class after applying SMOTE in this study.

C. APPLIED ARTIFICIAL INTELLIGENCE METHODS

To enhance driver behavior detection, different ML algorithms are employed to analyze data meticulously collected through motion sensors. These sensors are strategically attached to the driver, systematically recording a range of movements which is mentioned in the DATASET DESCRIPTION subsection. The collected data is then input into sophisticated ML algorithms, designed to meticulously analyze and identify distinct patterns in the driver's behavior. It provides a proactive mechanism for recognizing and alerting about potentially hazardous behaviors or environmental conditions, thereby significantly contributing to the safety and well-being of drivers.

In this novel methodology, we employ motion sensors strategically positioned on the driver data. Subsequently, this data undergoes an advanced preprocessing phase, where we implement SMOTE for balancing the class instances, alongside the RKnD probabilistic features approach to enhance the dataset's representativeness. This sophisticated integration of sensor technology, data augmentation techniques, and computational algorithms represents a comprehensive approach to understanding and analyzing driver behavior dynamics. We meticulously feed the refined data into an array of both ML and DL algorithms as follows.

1) RANDOM FOREST(RF)

RF is an ensemble method that combines many DT classifiers trained using the bagging technique. It enhances the stability

TABLE 1. Summary of recent studies on driving behavior analysis.

Ref.	Year	Approach & Methodology	Dataset Characteristics	Classification Strategy	Performance Results
[10]	2020	SF-Net using smartphone inertial sensors	GPS, 3-axis acceleration, gyroscope data	Deep CNNs	97.1% accuracy, 98.4% recall
[11]	2020	Exploring nuanced insights from high-resolution driving behavior data analysis	Data from 303 drivers' smartphones, traffic, road geometry	Utilizing both multiple linear and log-linear regression models	Significant impact of traffic characteristics on harsh events
[12]	2021	Hybrid DL for transportation mode detection	Two publicly available supervised datasets	CNNs, RNNs, DNNs	High accuracy and earliness in mode detection
[13]	2021	Fusion of in-vehicle and external sensors for behavior identification	Over 200 km travel data	SVM and artificial neural network	Approx. 88% accuracy with SVM, 90% with neural network
[14]	2021	ML for identifying driving behaviors	Over 26 hours of driving data	SVM and feed-forward neural networks	Over 90% accuracy for both classifiers
[15]	2021	ANN for monitoring motorcyclist behavior	Smartphone sensor data	Artificial Neural Network (ANN)	96.2% accuracy in behavior recognition
[16]	2022	Bio-signal-based system for aggressive driving detection	Bio-signal datasets	DCNN	73.02% and 79.15% accuracies on different datasets
[17]	2021	Smartwatch data analysis in naturalistic driving study	Data from 15 participants	Multiple ML algorithms	F1 scores: 94.55%, 98.27%, 97.86%
[18]	2020	Automatic extraction of in-vehicle information	Real vehicle experiments in urban setting	LSTM,RF	84.20% accuracy in driving conditions, 82.31% in information extraction
[19]	2023	Analysis of vehicle trajectory data for inattention detection	Trajectory data	CNN with data augmentation, LSTM-based model	92.27% accuracy in inattention detection, 91.67% in abnormal driving prediction
[20]	2020	IMU-based real-time driving behavior monitoring	Real-time motion data	Deep neural network-based approach	Recognition rates over 99%
[21]	2020	Driver identification and verification using smartphone accelerometer	25 drivers, over 20,000 journeys	ResNet-50 and SGRUs for identification, Siamese Neural Networks for verification	71.89% top-1, 92.02% top-5 identification accuracies, 74.09% F1 score for verification
[22]	2023	The Driver Behavior detection using Sensor Data	Publicly available smartphone motion sensor data	Combined Logistic Regression Random Forest	Random forest achieving a score of 99% using the LR-RFC method.
[23]	2023	Enhancing Transportation Safety by using FLFS and OSNCA	Vehicle On-Board Diagnostics (OBD) data	FLFS and OSNCA combined with Machine Learning and Deep Learning techniques	Achieved an accuracy of 98.9%, with a precision of 88.3% and a recall of 93.9% for 20 drivers using the FLFS-OSNCA method.

TABLE 2. Snapshot of five sample data points.

AccX	AccY	AccZ	GyroX	GyroY	GyroZ	Class
-0.700552	-0.15935495	-0.0628891	0.048258353	0.002672535	0.27473664	NORMAL
0.5859189	0.43197542	0.46232033	-0.06047566	0.006948592	-0.42653665	SLOW
-0.2795186	-0.011579275	-0.036842346	0.01160644	0.04970916	-0.019700404	NORMAL
2.33095	-7.6217537	2.5290236	0.05681047	-0.18058704	-0.05207626	AGGRESSIVE
2.8472152	-6.7556214	2.22464	-0.03176499	-0.03520111	0.035277467	AGGRESSIVE

and accuracy of the model by utilizing an average model strategy. The RF classifier is essentially a group of DT classifiers, each of which has been built with a set of random vectors and can vote for the most preferred class for forecasting. RF is an ensemble learning method that constructs multiple DTs and merges their predictions to obtain more accurate and robust results [26]. RF algorithm can be expressed as:

$$\hat{y} = \text{mode}\left(\{T(\mathbf{x}; \theta_i)\}_{i=1}^{N}\right), \tag{1}$$

where:

- $\hat{y}$ referees to predicted output.
- $T(\mathbf{x}; \theta_i)$ represents the prediction of the i-th DT with parameters θ_i for the input $\mathbf{x}$. DT
- The mode function calculates the most common prediction among all the DT in the forest.

This equation showcases how the ensemble of DT in an RF collectively determines the final prediction $\hat{y}$ for a given input $\mathbf{x}$ by considering the most frequent output among the individual tree predictions.

2) DECISION TREE (DT)

DT classifier is a widely used machine learning algorithm known for its interpretability and versatility. The algorithm creates a tree-like structure where each internal node

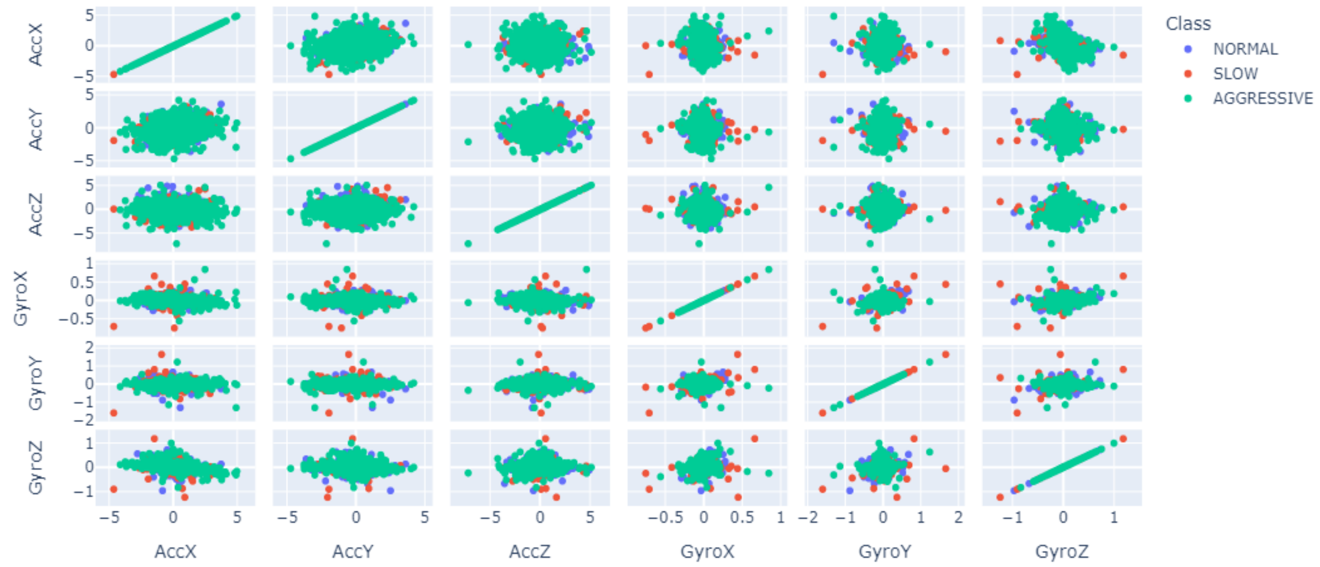


FIGURE 2. Pair plot of dataset features.

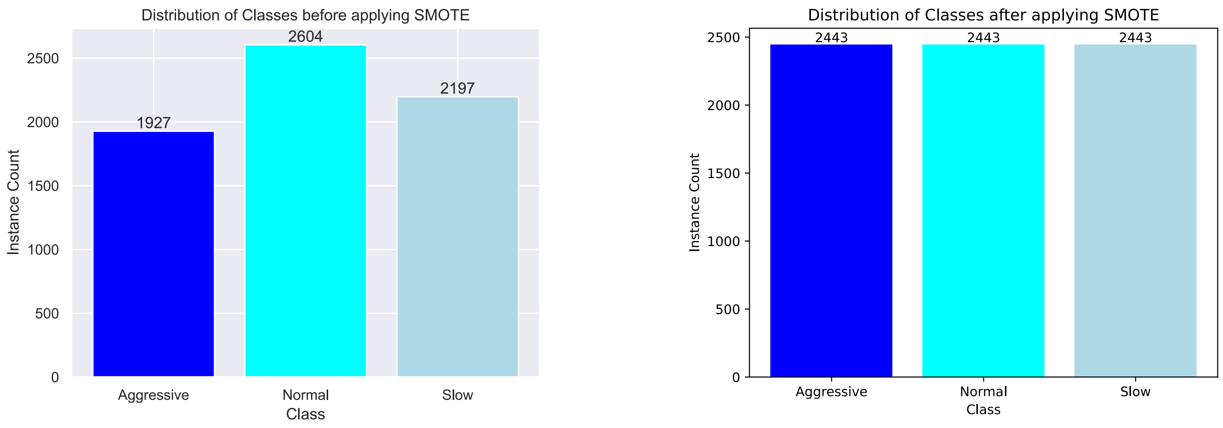


FIGURE 3. Class distribution before & after applying SMOTE approach.

represents a decision based on a particular feature, and each leaf node corresponds to the predicted class label [27].

One of the essential aspects of Decision Trees is the splitting criteria, which determines the optimal feature and threshold for partitioning the data at each decision node. Common splitting criteria include Gini impurity and entropy. Entropy, denoted by S, is a measure of disorder or impurity in a set of class labels. The entropy formula is given by:

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{2}$$

Here, c represents the number of classes, and p_i is the proportion of samples belonging to class i in the set S. The lower the entropy, the more homogeneous the class distribution.

Information Gain is another key concept used in Decision Trees, representing the reduction in entropy after a split. It is calculated as the difference between the entropy of the parent set and the weighted sum of entropies of child sets:

$$\text{Information Gain} = \text{Entropy}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v) \tag{3}$$

Here, A is the feature being considered for the split, values(A) are its possible values, and S_v is the subset of samples for which feature A takes the value v. Maximizing Information Gain leads to more effective feature selection.

Decision Trees may suffer from overfitting, especially with deep trees capturing noise in the training data. Pruning is a technique used to address this by removing unnecessary branches. Random Forests, an ensemble of Decision Trees, further enhance performance and robustness.

In conclusion, Decision Trees offer simplicity, interpretability, and versatility in handling various data types.

However, careful consideration of hyperparameters, such as tree depth, is necessary to balance model complexity and generalization.

3) K-NEAREST NEIGHBORS CLASSIFIER(KNC)

KNC is the most basic and ancient method in supervised ML. It is capable of solving both classification and regression problems, making it an adaptable approach in statistical learning [28]. It operates on the principle of the distance between data locations, and separate data are categorized with one another based on this. KNC is calculated by the Euclidean distance or Manhattan distance between two points where one is new data points $A(x_1, y_1)$ and the other is previously accessible data points $B(x_2, y_2)$ The equation 4 represents the fundamental formula for KNC:

$$d(x, y) = \sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2)} \tag{4}$$

4) RIDGE CLASSIFIER(RC)

The RC is a linear classification algorithm that addresses multicollinearity and overfitting in linear regression models. It is particularly useful when dealing with datasets with highly correlated features [29].

The Ridge Classifier introduces a regularization term to the standard linear regression objective function. The objective function for the RC is given by:

$$\text{Objective}(w, b) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} \max\left(0, 1 - y_i(w^T x_i + b)\right) \tag{5}$$

Here, w represents the weight vector, b is the bias term, x_i and y_i are the features and labels, and C means regularization strength. Moreover, the first term, $\frac{1}{2}\|w\|_2^2$, is the L2 regularization term, penalizing large values of weights to prevent overfitting.

The second term is the hinge loss, which measures the classification error. It encourages correct classification by minimizing the margin violation for each sample. The regularization parameter C controls the trade-off between fitting the training data and preventing overfitting. The Ridge Classifier is especially effective when dealing with datasets where features are correlated, as it tends to distribute the weights more evenly among correlated features.

The Ridge Classifier is a valuable tool for linear classification tasks, providing a balance between fitting the data and controlling overfitting through regularization.

5) GRADIENT BOOSTING(GB)

GB classifiers are an ensemble learning method and a group of ML algorithms that combine many weak algorithms to create a strong predictive model and work with loss functions, weak learner, and adaptive model [30]. The loss function is the main difference between GB regression and GB classification. If the target attribute is binary, the GB classifier can be applied. By mitigating the over-fitting problem and doing regularization, the performance of the GB classifier will increase. The algorithm can be defined as;

Initialize the model:

$$F_0(x) = \text{mean}(y_i) \tag{6}$$

For $t = 1$ to T (number of iterations):

1) Compute the pseudo-residuals:

$$r_i^{(t)} = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\bigg|_{F(x)=F_{t-1}(x)} \tag{7}$$

2) Fit a weak learner (e.g., DT) to predict the pseudo-residuals $r_i^{(t)}$.
3) Update the model:

$$F_t(x) = F_{t-1}(x) + \nu \cdot h_t(x) \tag{8}$$

where ν is the learning rate and $h_t(x)$ is the weak learner at iteration t.

The final prediction:

$$F(x) = F_0(x) + \sum_{t=1}^{T} \nu \cdot h_t(x) \tag{9}$$

6) GAUSSIAN NAIVE BAYES (GNB)

GNB is a classification algorithm based on the Naive Bayes theorem. It is specifically designed for handling continuous data [31], such as real-valued features, and assumes that the data follows a Gaussian (normal) distribution. This algorithm is a variation of the Naive Bayes classifier, which is a probabilistic ML algorithm used for classification tasks. GNB formula is as follows:

The probability of class C_k given the features $x_1, x_2, \ldots, x_n$ can be calculated as:

$$P(C_k|x_1, x_2, \ldots, x_n) = \frac{1}{Z} \cdot P(C_k) \cdot \prod_{i=1}^{n} P(x_i|C_k) \tag{10}$$

where:

- $P(C_k|x_1, x_2, \ldots, x_n)$ is the posterior probability of class C_k given the features.
- $P(C_k)$ is the prior probability of class C_k.
- $P(x_i|C_k)$ is the probability of observing feature x_i given class C_k, typically modeled as a Gaussian distribution.
- Z known as a normalization constant.

The normalization constant Z is often not explicitly calculated since it is the same for all classes, and it's sufficient to compare the unnormalized probabilities to make a classification decision [32].

7) LONG SHORT-TERM MEMORY(LSTM)

LSTM represents a distinct type of RNN with the ability to acquire knowledge of extended temporal dependencies [33]. They were introduced to overcome the limitations of traditional RNNs, which tend to forget earlier information in the sequence over time (vanishing gradient problem). LSTMs are designed to remember information for long periods, which is crucial in many complex tasks, including driver behavior detection. The model architecture of LSTM is shown in Figure 4. Moreover, an LSTM unit can be defined as:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \tag{11}$$

where:

- f_t, i_t, o_t are the forget data, input data, and output data gates, respectively.
- W and b are the weights data and biases for each gate.
- σ is the sigmoid values function.
- tanh is the hyperbolic tangent function.

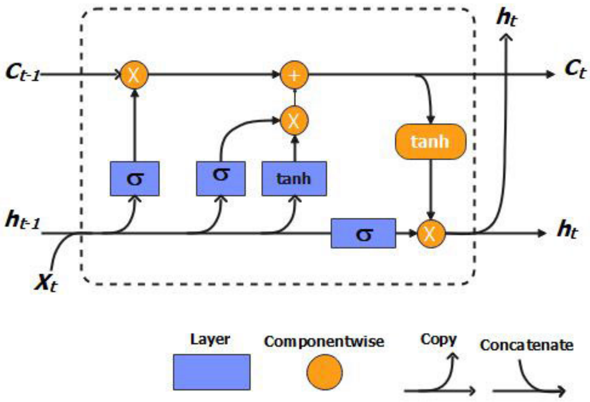


FIGURE 4. LSTM neural network model architecture.

8) CONVOLUTIONAL NEURAL NETWORK(CNN)

CNNs are DL models designed for visual data analysis, leveraging a hierarchical structure of interconnected layers to automatically learn features. Convolutional Neural Networks (CNNs) have brought about a transformation in computer vision applications such as image recognition, object detection, and segmentation. Their proficiency in learning intricate spatial hierarchies and patterns has made them indispensable across a wide range of domains, spanning from healthcare to autonomous vehicles [34]. Their efficacy stems from parameter sharing, enabling efficient feature extraction and learning representations directly from raw data.The model architecture of CNN is shown in Figure 5 and the architecture of a basic CNN [35] involves the following layers:

- **Input Layer:** The input to the CNN is typically an image represented as a matrix:

$$\text{Input} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,W} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,W} \\ \vdots & \vdots & \ddots & \vdots \\ x_{H,1} & x_{H,2} & \cdots & x_{H,W} \end{bmatrix}$$

Here, H and W denote the height and width of the input image.

- **Convolutional Layer** The convolutional layer performs convolutions on the input using learnable filters (kernels) to produce feature maps:

$$\begin{aligned} &\text{Output}(i, j, k) \\ &= \sigma\Bigg(\sum_{l=1}^{F}\sum_{m=1}^{K}\sum_{n=1}^{K}\Big(\text{Input}(i+m-1, \\ &\quad j+n-1, l) \times \text{Weight}(m, n, k, l)\Big) + \text{Bias}(k)\Bigg) \end{aligned} \tag{12}$$

where F is the number of input channels, K is the kernel size, σ is the activation function, and Bias(k) is the bias term.

- **Pooling Layer** Pooling layers downsample the feature maps to reduce spatial dimensions:

$$\text{Output}(i, j, k) = \text{Pooling\_Function}(\text{Input}(i, j, k)) \tag{13}$$

- **Fully Connected Layer** The fully connected layer connects all neurons from the previous layer to the next layer:

$$\text{Output} = \sigma\ (\text{Weight} \times \text{Input} + \text{Bias}) \tag{14}$$

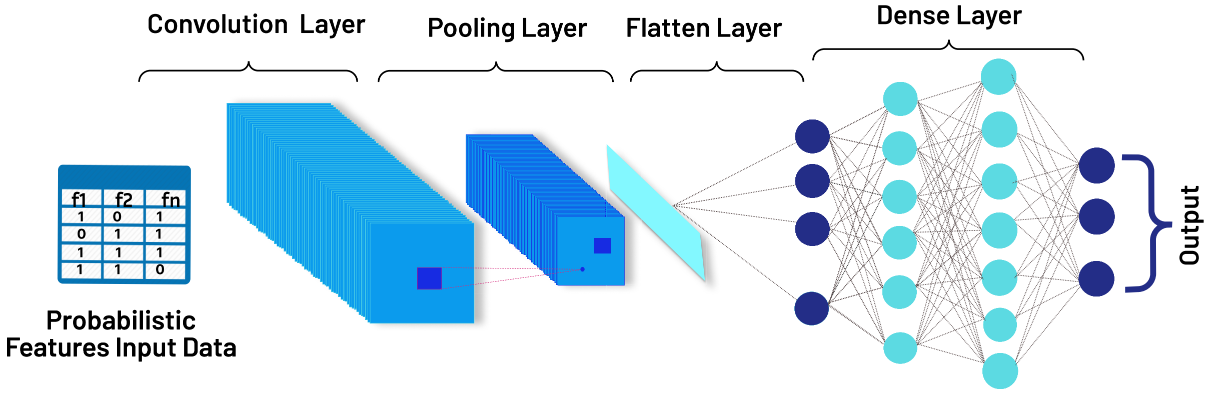
- **Output Layer** Finally, the output layer produces the final predictions or classifications.

D. HYPERPARAMETER TUNING

Finding the best possible values for model hyperparameters that have the potential to have a major influence on the performance of the model is what hyperparameter optimization is all about [36]. Through the procedure of fine-tuning, we optimized the hyperparameter to enhance the performance of the models that were applied. When it comes to increasing the performance outcomes of ML approaches for identifying botnet assaults, the findings of our study underscore the significance of hyperparameter optimization. Based on the best-fit chosen hyperparameter analysis, the results of our investigation are presented in Table 3.

TABLE 3. Model architecture and parameters tuning.

Technique	Hyperparameters
RF	n_estimators=100, criterion="gini", max_depth=20, max_features="sqrt", min_samples_split=2
DT	criterion="gini", max_depth=300, splitter="best"
KNC	n_neighbors=2, leaf_size=30, weights='uniform', metric='minkowski'
RC	alpha=1.0
GB	n_estimators=100, learning_rate=0.1, max_depth=3
GNB	priors=None, var_smoothing configured=1e-09
LSTM	optimizer = 'adam', activation='softmax', epochs=10
CNN	filter_size=(3,3), num_filters=64, activation='relu', optimizer='adam'



f1	f2	fn
1	0	1
0	1	1
1	1	1
1	1	0

FIGURE 5. Modified CNN architecture for probabilistic features input.

E. NOVEL PROPOSED FEATURES EXTRACTION METHOD

The novel RKnD probabilistic feature extraction method we propose is designed for meticulous driver behavior detection, as depicted in the architectural diagram. Figure 6 illustrates the novel proposed probabilistic feature extraction architecture. This method synergistically combines the proficiency of advanced ML approaches (RF, KNC, and DT) to form a unified features engineering system. Moreover, we applied different combinations of ML models for probabilistic feature engineering before selecting the RKnD approaches but the RKnD technique provided the best performance. For example, if we consider different data points, denoted as $I = i_1, i_2, \ldots, i_n$, along with a corresponding group of labels, denoted as $L = l_1, l_2, \ldots, l_n$. Each label l_i is a binary variable that shows whether a specific target feature is present or not.

Consider a dataset comprising a set of input data points, denoted as $I = i_1, i_2, \ldots, i_n$, along with a corresponding set of binary labels, denoted as $L = l_1, l_2, \ldots, l_n$. Each label l_i indicates the class attribute.

ML models employed to predict the probability of these labels given the input data, $P(L_i|i_i)$, resulting in an array of probabilities $[p_1, p_2, \ldots, p_n]$, where $p_i = f(i_i)$. These probability values serve as features for training and analyzing ML and deep learning (DL) models. Further, statistical properties such as the mean and variance of the probability distribution, extracted through a function $G(p)$, or direct utilization of probabilities through a function $H(L, p)$, individually or in combination with others, can contribute to the final feature set [22]:

$$F = f_1, f_2, \ldots, f_n, \text{ where } f_i = G(p) \text{ or } H(X, p). \tag{15}$$

Algorithm 1 illustrates the systematic flow of the proposed approach, detailing each step in the process.

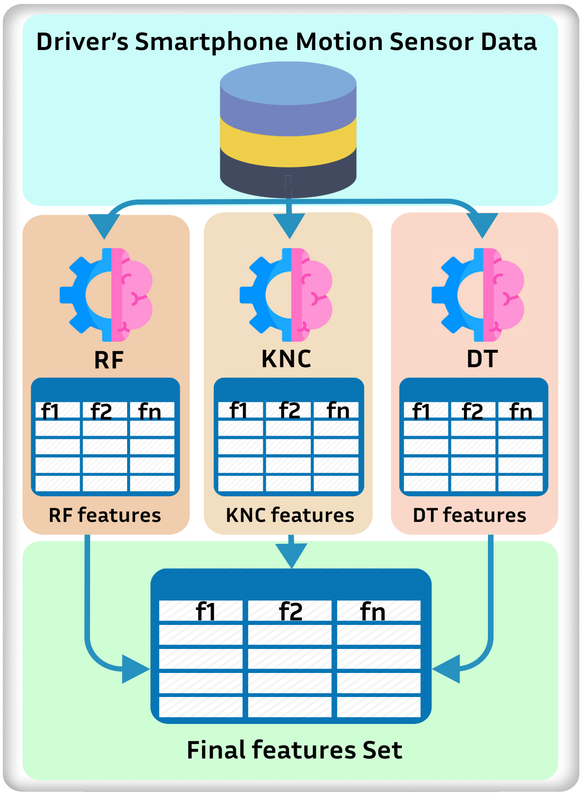


FIGURE 6. Architecture of RKnD feature extraction method.

IV. RESULTS AND DISCUSSION

A comparison and analysis of the outcomes that are achieved via the use of ML and DL strategies is presented in this section. This section gives a full overview of the approaches that are implemented by offering a detailed discussion of the performance metrics of each model.

A. SETUP OF EXPERIMENT

The experimental setup used to develop the applied ML and DL techniques is discussed here. Data manipulation and preprocessing tasks were adeptly handled using the Pandas library, version 1.4.0, enabling efficient data operations essential for preparing the Smartphone Motion Sensor dataset for analysis. For the ML model training and evaluation, we utilized the Scikit-learn library (version 1.0.2), which

Algorithm 1 Proposed RKnD Feature Engineering

```
Input: Smartphone Motion Sensor Dataset
Output: Driver Behavior Prediction | Normal, Aggressive, and Slow

Initialize training and testing sets:
    • $T_{RF} \leftarrow RF_{\text{training}}(TrS)$   // $T_{rS}$ in Smartphone Motion Sensor Dataset, here $TrS$ is the training set of the dataset.
    • $T_{KNC} \leftarrow KNC_{\text{Training}}(TrS)$
    • $T_{DT} \leftarrow DT_{\text{Training}}(TrS)$
for $i$ in testing set $TeS$ do   // $TeS$ in Smartphone Motion Sensor Dataset
    • $RF_p \leftarrow \text{Predict}_{RF}(i)$   // $RF_p$ is the RF prediction for instance $i$.
    • $KNC_p \leftarrow \text{Predict}_{KNC}(i)$   // $KNC_p$ is the DT prediction for instance $i$.
    • $DT_p \leftarrow \text{Predict}_{DT}(i)$   // $DT_p$ is the DT prediction for instance $i$.
end for
Combine predictions to form the final prediction:
    • $Final_{\text{Pred}} \leftarrow \text{MajorityVote}(RF_p, KNC_p, DT_p)$   // $Final_{\text{Pred}}$ is either Normal, Aggressive or Slow based on majority voting.
```

is acclaimed for its comprehensive suite of algorithms and tools designed for machine learning applications. The development and training of DL models were facilitated through TensorFlow, version 2.9.0, utilizing its expansive module API for the construction of sophisticated neural network architectures. Our computational experiments capitalized on the robust capabilities of Google Colab and Kaggle Kernel platforms, which provided access to advanced computing resources, including a GPU backend. The GPU utilized for our experiments was the Nvidia Tesla T4 GPU, a choice motivated by its enhanced performance characteristics suitable for the demands of high-volume data processing and complex model computations. This experimental setup was supported by 16 gigabytes of Random Access Memory (RAM) and 100 gigabytes of disk space, ensuring ample resources for the seamless execution of our research activities.

B. EVALUATION MATRICS

In assessing the efficacy of these methodologies, we closely examine metrics such as accuracy, precision, recall, and the F1 score. These pivotal metrics enable us to gauge the performance of the chosen models effectively. In classification, True Positive (TP) represents instances correctly identified as positive, while False Positive (FP) signifies incorrect positive predictions. True Negative (TN) denotes instances correctly identified as negative, and False Negative (FN) indicates instances incorrectly predicted as negative. These metrics play a crucial role in assessing a model's performance, aiding in the computation of accuracy, precision, recall, f1 score, and other evaluation measures [37].

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{16}$$

In this equation, the "Number of Correct Predictions" represents the count of correctly classified instances, and the "Total Number of Predictions" is the total number of instances in the dataset that were classified. The accuracy is a measure of the model's ability to correctly classify instances and is typically expressed as a percentage [38].

To calculate the precision metric, the total number of classified positive samples is divided by the number of correctly classified positive instances. The formula to compute this metric is expressed as follows:

$$\text{Precision} = \frac{TP}{TP\ + FP} \tag{17}$$

Recall is a metric in classification that measures true positives correctly identified from all actual positives. It shows the model's ability to detect all positive instances.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{18}$$

The F1 score is a combination of precision and recall to measure a model's overall performance. It is calculated using the harmonic mean, which gives more weight to low values.

$$\text{F1-score} = \frac{2 \times Recall\ times\ Precision}{Recall\ \ + Precision} \tag{19}$$

C. PROPOSED MODELS RESULTS WITH ORIGINAL FEATURES

The section provides an in-depth performance analysis of both ML and DL models, utilizing the dataset's original features as input. This comprehensive evaluation assesses key performance metrics, including accuracy, precision, recall, and F1-score, to gauge the effectiveness of each model. To offer a more comprehensive view, the section presents matrix plots that visually depict the performance results. Furthermore, it delves into Cross-Validation performance and computational runtime aspects, enhancing the overall understanding of model capabilities and efficiency.

1) RESULTS WITH ML MODELS

Table 4 provides a comprehensive assessment of ML models, employing the original dataset features. The evaluation centers on key performance metrics: f1-score, recall, precision, and accuracy, crucial for assessing the effectiveness of ML models. A detailed analysis indicates diverse performance levels among models, including RF, DT, KNC, RC, GB, and GNB. In this analysis, the RF model stands out, attaining a commendable accuracy score of 0.91. This score, although significant, is not the apex of what can be achieved, suggesting room for further refinement to optimize performance in the context of the study's objectives. The RF model's superiority is further underscored by its high recall and precision scores across all classes, indicating its

TABLE 4. Evaluating the performance of different ML approaches across diverse metrics.

Model	Accuracy	Class	Precision	Recall	F1-score
RF	0.91	0	0.85	0.96	0.90
		1	0.96	0.85	0.91
		2	0.90	0.90	0.92
		mean	0.91	0.91	0.91
DT	0.42	0	0.40	0.39	0.42
		1	0.44	0.43	0.43
		2	0.42	0.40	0.42
		mean	0.42	0.41	0.42
KNC	0.46	0	0.44	0.43	0.46
		1	0.48	0.45	0.45
		2	0.46	0.47	0.47
		mean	0.46	0.45	0.46
RC	0.48	0	0.46	0.45	0.48
		1	0.51	0.49	0.49
		2	0.51	0.49	0.49
		mean	0.48	0.48	0.49
GB	0.92	0	0.86	0.97	0.92
		1	0.97	0.89	0.92
		2	0.97	0.89	0.92
		mean	0.92	0.93	0.92
GNB	0.72	0	0.66	0.77	0.72
		1	0.77	0.69	0.72
		2	0.77	0.69	0.72
		mean	0.72	0.73	0.72

robustness in identifying a substantial number of positive instances accurately. Conversely, models like DT and KNC show subpar performance, with DT scoring as low as 0.42 in accuracy, and KNC at 0.46. These scores are indicative of a need for substantial improvements or a reconsideration of the model selection. RC and GNB models exhibit moderate performance, with GNB scoring an average accuracy of 0.72, highlighting the potential for enhancements in these models. On the higher end of the spectrum, the GB model closely follows RF in effectiveness, achieving an impressive accuracy of 0.92. This indicates that while RF is a strong performer, GB also holds substantial potential in this domain. The analysis illustrates that while some models like RF and GB show promising results, others like DT and KNC significantly lag in effectiveness. This disparity accentuates the importance of careful model selection and the potential need for feature engineering to achieve optimal results in ML applications. The overall findings from this study point towards an ongoing necessity to refine and enhance the performance of ML models in the pursuit of achieving the highest level of accuracy and efficiency. Figure 7 shows this performance more intuitively.

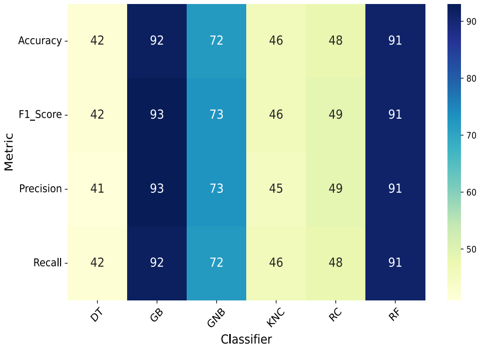


FIGURE 7. Performance analysis of the ML models with original features.

TABLE 5. Evaluating the performance of different DL approaches across diverse metrics.

Model	Accuracy	Class	Precision	Recall	F1-score
LSTM	0.94	0	0.94	0.94	0.94
		1	0.93	0.95	0.93
		2	0.95	0.93	0.95
		mean	0.94	0.94	0.94
CNN	0.93	0	0.93	0.93	0.92
		1	0.92	0.92	0.92
		2	0.94	0.94	0.95
		mean	0.93	0.93	0.93

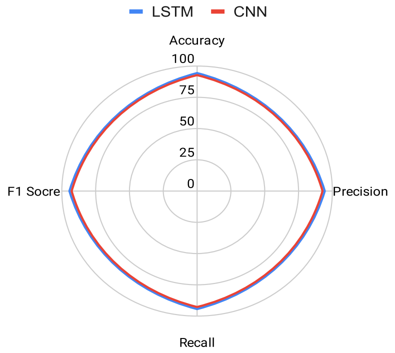


FIGURE 8. Performance analysis of the DL models with original features.

2) RESULTS WITH DL MEDELS

In Table 5 the performance analysis of the DL-based LSTM and CNN models for unseen testing data is presented. The LSTM model achieved an accuracy of 94%, with consistent precision, recall, and F1-score values of 0.94 across all three classes (0, 1, and 2). In contrast, the CNN model secured a 93% accuracy, with precision and recall scores hovering around 0.93 and an F1-score of 0.92 for class 0. Despite the commendable average accuracy, both models exhibited variability in class-specific metrics, indicating potential challenges in certain driver behavior detections. From Figure 8 the radar chart shows clearly which model performs better in certain areas.

3) K-FOLD CROSS-VALIDATION AND COMPUTATIONAL COMPLEXITY BY RUNTIME WITH ORIGINAL FEATURES

K-fold cross-validation is crucial to assess model generalization and robustness by providing a more reliable estimate of performance across different data splits, reducing the risk of overfitting compared to a single evaluation in the previous subsection. Table 6 displays the results of a 10-fold cross-validation for each method. The RF method stands out with an impressive accuracy of 0.9095 and a low standard deviation of 0.0057, indicating consistent performance across different subsets. The DT model shows a moderate accuracy of 0.4159 with a standard deviation of 0.0038. The KNC and RC exhibit accuracies of 0.4521 and 0.4927 with standard deviations of 0.0070 and 0.0026, respectively, suggesting a reasonable level of consistency. The GB model has an

TABLE 6. Cross-Validation of performance analysis of proposed methods with original features.

Method	K-fold	Accuracy	Standard deviation
RF	10	0.9095	0.0057
DT	10	0.4159	0.0038
KNC	10	0.4521	0.0070
RC	10	0.4897	0.0066
GB	10	0.9221	0.0023
GNB	10	0.7167	0.0031
LSTM	10	0.9413	0.0019
CNN	10	0.9324	0.0021

TABLE 7. Runtime Computations of proposed models with original features.

Method	Runtime Computations (Seconds)
RF	50.9312
DT	0.3443
KNC	0.1052
RC	2.2721
GB	12.8105
GNB	0.1564
LSTM	157.2025
CNN	125.4331

accuracy of 0.7167 and a higher standard deviation of 0.0031, which might indicate variability in performance across the folds. The DL methods, LSTM and CNN, show high accuracies of 0.9413 and 0.9324 with very low standard deviations of 0.0019 and 0.0021, respectively, which highlights their robustness in cross-validation.

In Table 7 the computational complexity is represented by the runtime in seconds required to complete the computations. The RF method is relatively fast, taking only 0.3443 seconds. The DT method is even quicker, requiring a mere 0.0512 seconds. The KNC and RC are also efficient, with runtimes of 0.1052 and 2.2721 seconds, respectively. The GB method takes a bit longer, with a runtime of 12.8105 seconds. In contrast, the LSTM and CNN models, which are computationally more intensive due to their DL architecture, take significantly longer with runtimes of 157.6025 and 125.4331 seconds, respectively. Combining the findings from both tables, it is evident that while DL methods like LSTM and CNN offer higher accuracy, they come at the cost of increased computational complexity and runtime. The data suggests that there is a trade-off between accuracy and runtime, and the choice of method may depend on the specific requirements of the application in terms of speed and performance.

D. RESULTS WITH NOVEL RKnD FEATURE ENGINEERING

This section introduces a method for analyzing the performance of ML and DL models, similar to the previous subsection. The technique involves using derived features as input, with novel RKnD feature engineering techniques. To provide a more comprehensive view of the results, the section presents matrix plots that visually depict the performance. Additionally, it explores Cross-Validation performance and computational runtime aspects, which improve the overall understanding of model capabilities and efficiency. Moreover, the results are compared with those of the previous section.

TABLE 8. Evaluating the performance of different ML models of RKnD features.

Model	Accuracy	Class	Precision	Recall	F1-score
RF	0.97	0	0.97	0.97	0.96
		1	0.98	0.94	0.98
		2	0.97	0.98	0.97
		mean	0.97	0.96	0.97
DT	0.92	0	0.90	0.89	0.92
		1	0.94	0.93	0.93
		2	0.92	0.90	0.92
		mean	0.92	0.91	0.92
KNC	0.96	0	0.94	0.93	0.96
		1	0.98	0.95	0.95
		2	0.96	0.97	0.97
		mean	0.96	0.95	0.96
RC	0.98	0	0.96	0.95	0.98
		1	1.00	0.99	0.99
		2	1.00	0.99	0.99
		mean	0.98	0.98	0.99
GB	0.98	0	0.98	0.99	0.98
		1	0.97	0.99	0.98
		2	0.99	0.97	0.98
		mean	0.98	0.98	0.98
GNB	0.94	0	0.93	0.93	0.92
		1	0.92	0.95	0.92
		2	0.97	0.94	0.95
		mean	0.94	0.94	0.93

1) RESULTS WITH ML METHODS

According to Table 8, the use of RKnD probabilistic features technique has led to an overall improvement in the performance of the models. For instance, the RF model, which previously scored an accuracy of 0.91 with the original features, now exhibits a perfect accuracy score of 0.97 with RKnD features. This not only highlights the effectiveness of RKnD features in enhancing model performance but also implies that the ceiling of potential performance has been raised substantially. The DT and KNC models, previously lagging with accuracy scores of 0.42 and 0.46 respectively, have made significant strides with RKnD features, now boasting an accuracy of 0.92 for DT and 0.96 for KNC. This dramatic increase is indicative of the transformative impact of RKnD features on models that once underperformed. The RC and GNB models have also benefited from the application of RKnD features, with RC showing an exceptional mean accuracy of 0.98, and GNB improving to 0.94. This enhancement demonstrates that RKnD features have the potential to elevate moderate-performing models to levels of high accuracy. Moreover, the GB model, already effective with an accuracy of 0.92 using original features, maintains this high standard accuracy score of 0.98 with RKnD features, emphasizing the consistent performance of this model regardless of the feature set used. To deliver a more comprehensive understanding Figure 9 shows different model's performance.

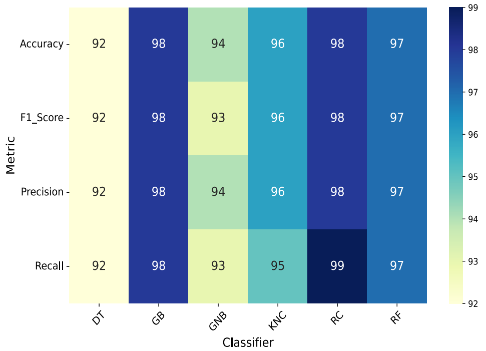


FIGURE 9. Performance analysis of the ML models after applying proposed feature engineering.

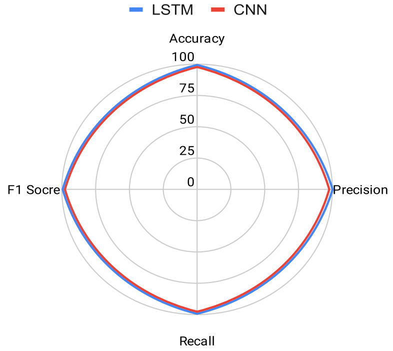


FIGURE 10. Performance analysis of the DL models after applying proposed feature engineering.

TABLE 9. Evaluating the performance of different DL approaches across diverse metrics.

Model	Accuracy	Class	Precision	Recall	F1-score
LSTM		0	1.00	0.99	0.99
	0.99	1	1.00	1.00	1.00
		2	1.00	1.00	1.00
		mean	1.00	0.99	0.99
CNN		0	0.99	0.99	0.99
	0.98	1	0.98	0.98	0.99
		2	0.97	0.98	0.97
		mean	0.98	0.98	0.98

Overall, the adoption of RKnD feature engineering has not only improved the metrics for each model across the board but also narrowed the performance gap between them. The increased accuracy, precision, recall, and F1 scores serve as robust indicators that RKnD features engineering is a critical advancement in optimizing ML models for superior predictive performance

2) RESULTS WITH DL METHODS

Table 9 presents the performance of two DL models With RKnD features, the LSTM model has achieved an impressive accuracy of 0.99, an increase from the 0.94 accuracy obtained with the original features. This perfect score is reflected across all classes for precision, recall, and F1-score, highlighting the LSTM's enhanced ability to consistently and accurately predict all classes after applying RKnD features. Similarly, the CNN model has shown an improvement in accuracy, going from 0.93 with the original features to 0.98 with RKnD features. The precision and recall for all classes have seen a uniform increase, achieving 0.99 for class 0 and maintaining high scores for classes 1 and 2. The mean values of precision, recall, and F1-score for CNN have all risen to 0.98, demonstrating the model's increased reliability and balanced performance across classes.

When compared to their performance with the original features, where both LSTM and CNN models exhibited some variability in class-specific metrics, the results with RKnD features suggest a more robust and consistent capability in detecting diverse driver behaviors. The LSTM model, in particular, has reached a level of performance that can be considered near-perfect, while the CNN has closed the gap substantially, showing that both models benefit significantly from the sophisticated feature engineering provided by RKnD. The radar chart in Figure 10 shows clearly that both models have achieved high performance as they covered a vast area in the radar chart.

3) K-FOLD CROSS-VALIDATION AND COMPUTATIONAL COMPLEXITY BY RUNTIME WITH RKnD FEATURES

Table 10 presents the results of a 10-fold cross-validation. The RF model achieves an accuracy of 0.9695 with a standard deviation of 0.0015, indicating high consistency in performance. The DT has an accuracy of 0.9242 with a standard deviation of 0.0018, while the KNC scores an accuracy of 0.9592 with a standard deviation of 0.0010. The RC achieves an accuracy of 0.9897 with a notably low standard deviation of 0.0011, suggesting highly stable performance across folds. The GB and GNB models show accuracies of 0.9821 and 0.9447 with standard deviations of 0.0013 and 0.0023, respectively. For the DL methods, the LSTM network achieves an accuracy of 0.9963 with a very low standard deviation of 0.0003, and the CNN reaches an accuracy of 0.9854 with a standard deviation of 0.0008.

Table 11 details the computational time each method takes to execute. The RF method requires 0.2943 seconds, and the DT method takes only 0.1493 seconds. KNC shows a runtime of 0.1742 seconds, and RC needs 1.2351 seconds. The GB model takes longer with a runtime of 10.8105 seconds. The DL models, LSTM and RNN, take considerably more time at 123.1025 and 112.4351 seconds, respectively. Comparing the performance and computational complexity of these models with RKnD features to those with the original features, we observe substantial improvements. Notably, the accuracies of all models with RKnD features have increased, with LSTM showing an exceptional accuracy improvement from 0.9413 to 0.9963. Additionally, the standard deviation of performance metrics across folds has decreased for all models, indicating more stable and reliable performance when using RKnD features. From a computational complexity standpoint, while the runtime for DL models remains high, there is a slight improvement in the efficiency of

TABLE 10. Cross-Validation of proposed methods with RKnD probabilistic features.

Method	K-fold	Accuracy	Standard deviation
RF	10	0.9695	0.0015
DT	10	0.9242	0.0018
KNC	10	0.9592	0.0010
RC	10	0.9897	0.0011
GB	10	0.9821	0.0013
GNB	10	0.9477	0.0021
LSTM	10	0.9963	0.0003
CNN	10	0.9854	0.0008

TABLE 11. Runtime computations of various methods.

Method	Runtime Computations (Seconds)
RF	0.4093
DT	0.2943
KNC	0.1142
RC	1.2731
GB	10.8105
GNB	0.1264
LSTM	123.1025
RNN	112.4531

LSTM, which has decreased from 157.6025 seconds to 123.1025 seconds when using RKnD features. This suggests that RKnD features not only enhance model accuracy but also can contribute to more efficient model training and validation processes.

E. COMPARISON BETWEEN ORIGINAL AND PROPOSED RKnD PROBABILISTIC FEATURES BY SCATTER PLOTS

The comparison between the original features and the RKnD probabilistic features is visually illustrated in the provided images, Figure 11. These figures depict 3D scatter plots of data points classified into three distinct target classes.

In Figure 11, which represents the original features in the first part, the data points are spread out but with considerable overlap between the classes, as evidenced by the interspersed blue, orange, and yellow points. This overlap indicates a degree of ambiguity in class separability, which could be the reason for the lower performance metrics observed with the original features in the ML models. The less distinct clustering of the data points can lead to decreased model accuracy, as the model may struggle to define clear decision boundaries between the classes. Contrastingly, In the second part of Figure 11 showcases the RKnD features and demonstrates a stark improvement in class separability. The clusters are much more defined, with each class forming a distinct group. The blue, orange, and yellow points are segregated into tighter clusters, reducing the overlap and thus potentially decreasing the misclassification rate. This enhanced separation is likely to contribute to the higher accuracy and performance metrics of the ML and DL models when RKnD features are applied. The distinct clustering observed in the RKnD features suggests that these features capture the underlying structure of the data more effectively, allowing models to learn more discriminative patterns. The RKnD features seem to provide a transformed feature space where the target classes are linearly separable to a higher degree, which is beneficial for both traditional ML algorithms and DL architectures in achieving higher precision and recall rates. In summary, the visual comparison clearly indicates the superiority of RKnD probabilistic features over original features in terms of facilitating better class discrimination, which directly correlates with the improved performance of predictive models utilizing these features.

F. COMPARISON BETWEEN ORIGINAL AND PROPOSED RKnD PROBABILISTIC FEATURES BY CONFUSION MATRIX

The comparison of confusion matrices in 12 and 13 between original features and RKnD features for proposed ML and DL models reveals substantial improvements in classification accuracy with RKnD features. The RF model shows a notable increase in TP and a decrease in misclassifications, with the TP for class 0 rising from 444 to 459. The DT and KNC models demonstrate more pronounced enhancements, particularly KNC, with TPs for class 1 soaring from 219 to 454. RC exhibits dramatic gains in predictive accuracy, and GB sees a slight uptick in performance. The GNB model's TP rate for class 2 significantly increases, indicating a better grasp of the class by the model. The DL models, LSTM, and CNN, already performing exceptionally with original features, achieve marginal yet meaningful improvements, achieving near-perfect classification with RKnD features. Overall, RKnD features have markedly advanced model performance, emphasizing the pivotal role of sophisticated feature engineering in ML.

G. EVALUATIONS AGAINST PRIOR WORKS

The comparative analysis of our findings in relation to studies that are recognized as state-of-the-art is presented in Table 12. This review considered a broad spectrum of cutting-edge methods developed in the past year. Notably, the performance scores of various current approaches display differences, with the lowest accuracy score recorded at 97.1%, suggesting room for improvement. Our proposed RKnD approaches stand out significantly, achieving the maximum accuracy score of 99.63%. This underscores its superiority over other early detection of driver behavior methods. The results highlight the substantial progress RKnD has made compared to other methodologies currently employed in the field.

H. DISCUSSION

The paper discusses the importance of early detection of driver behavior for enhancing road safety, with a focus on utilizing smartphone sensors for data acquisition. It introduces a novel approach called RKnD for driver behavior detection, which combines RF, KNC, and DT algorithms. The key discussions in the paper encompass the methodology and

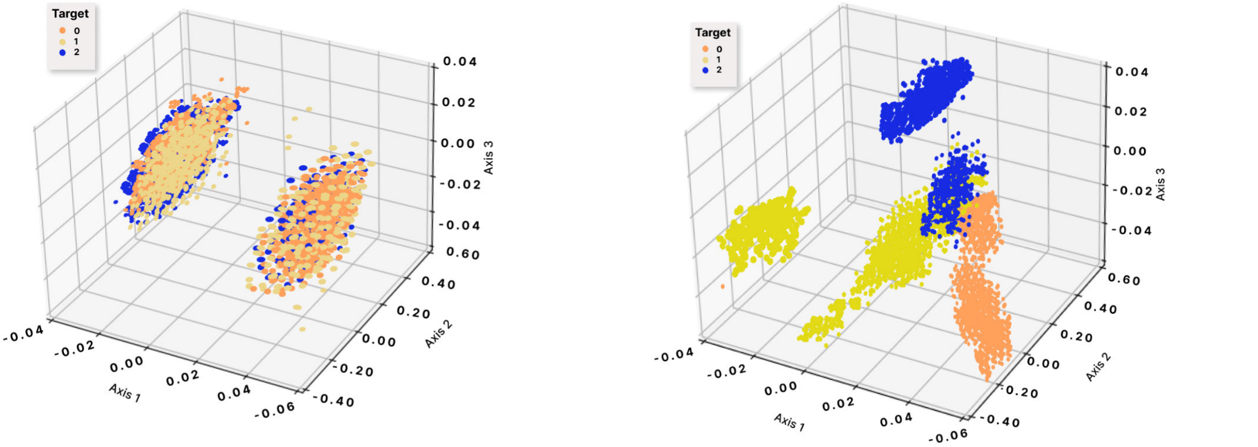


FIGURE 11. Scatter plot of original features & RKnD probabilistic features.

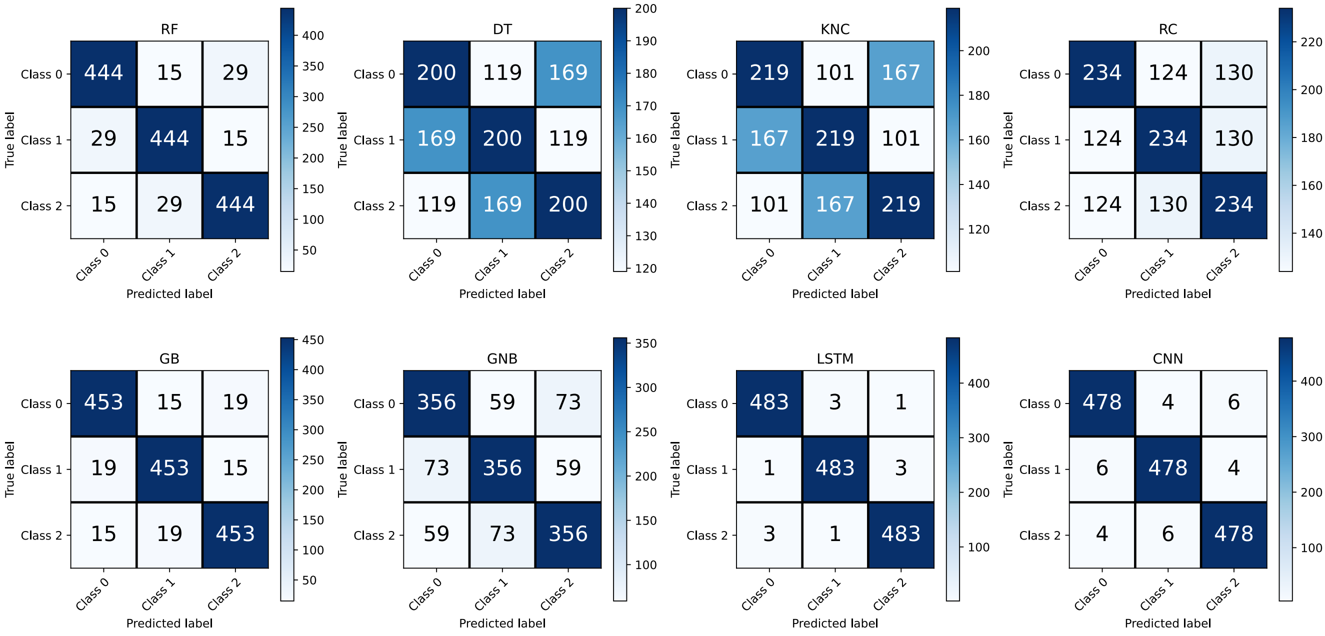


FIGURE 12. Models confusion matrix with original features.

results of the proposed RKnD model on both the original and novel RKnD feature engineering, as well as the superiority of RKnD feature engineering over the original features. The methodology section outlines the use of SMOTE to address data imbalance and k-fold cross-validation to ensure model consistency and accuracy across different datasets. The authors emphasize the need for reliable techniques to detect and recognize driver behavior, citing the significant impact of driver distractions on road accidents. The RKnD approach is highlighted as a pioneering solution in the field of smart transportation systems, leveraging ML and DL techniques in a practical and accessible manner. The paper presents the results of their model, which achieved a remarkable accuracy rate of 99.63% in detecting driver behaviors. The precision and accuracy of our model underscore its viability for real-life applications, particularly in the context of smart transportation systems. Furthermore, the paper discusses related literature, summarizing various approaches and methods used in driver behavior analysis. These studies include the use of smartphone inertial sensors, high-resolution driving behavior data analysis, hybrid deep learning models, and the integration of in-vehicle and external sensors. The paper concludes that detecting driver behavior early is crucial and introduces the RKnD strategy as a promising way to do so. By combining RF, KNC, and DT algorithms, the RKnD strategy outperforms other methods of driver behavior detection, highlighting its potential to improve road safety and prevent accidents.

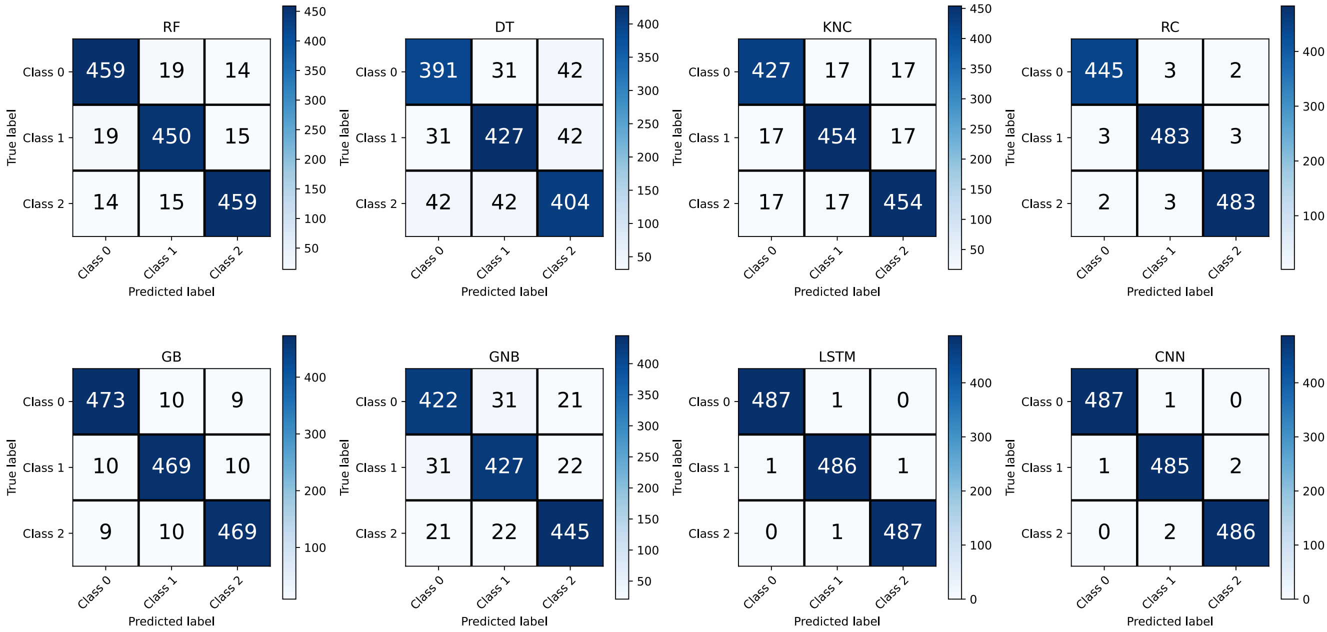


FIGURE 13. Models confusion matrix with RKnD probabilistic features.

TABLE 12. State of the art in early detection of driver behavior.

Ref.	Models	Approach	Result
[10]	Deep CNN	Serial-Feature Network (SF-Net) using smartphone inertial sensors	97.1% accuracy, 98.4% recall
[12]	CNNs, RNNs, DNNs	Hybrid DL classifier for transportation mode detection	High accuracy and earliness in mode detection
[15]	Artificial Neural Network (ANN)	ANN for monitoring motorcyclist behavior using smartphone sensors	96.2% accuracy in behavior recognition
[19]	CNN with data augmentation, LSTM-based model	Analysis of vehicle trajectory data for inattention detection	92.27% accuracy in inattention detection, 91.67% in abnormal driving prediction
[20]	Deep neural network-based approach	IMU-based real-time driving behavior monitoring	Recognition rates over 97%
[22]	RF	Ensemble feature engineering approach	Accuracy rates of 99%
Proposd model	**RKnD**	**RF, KNC, DT, Feature Engineering**	**Accuracy 99.63%**

V. CONCLUSION AND FUTURE DIRECTION

This study introduces RKnD, a groundbreaking feature engineering that amalgamates the strengths of RF, KNC, and DT networks to revolutionize early detection of driver behavior. This model stands out for its exceptional performance, showcasing a remarkable enhancement over traditional approaches. Rigorous testing indicates that RKnD significantly surpasses existing models in various performance metrics for identifying normal, aggressive, and slow behaviour. This high level of efficiency marks a pivotal advancement in the field of road accidents, suggesting that the combination of ML techniques is not only viable but also highly effective in tackling contemporary driver behavior. The success of RKnD can be attributed to its innovative design, leveraging the individual strengths of RF, KNC, and DT. RF excellence in processing image data, KNC's capability to classify data based on similarity measures, and DT's proficiency in decision-making through a tree-like model, and collectively contribute to RKnD's robustness and accuracy. This synergy enables the model to adeptly handle the complexities and nuances of smartphone motion sensor data, distinguishing between benign and malicious activities with remarkable precision. The RKnD model shows great promise in detecting and classifying driver behaviour.

There are several potential avenues for future research. Firstly, it is imperative to refine the algorithmic structure of RKnD to improve its computational efficiency and suitability for real-time applications. This includes streamlining the model to reduce computational burden while maintaining or enhancing its detection capabilities. Secondly, broadening the scope of datasets, particularly those encompassing diverse behavior types, is essential to evaluate the model's adaptability and efficacy across different contexts of road accidents. Thirdly, integrating advanced DL techniques like

Self-Organizing Maps (SOMs) could bolster the model's capacity to recognize intricate and evolving behavior patterns. Lastly, field-testing RKnD in real-world scenarios is crucial to assess its practical usefulness and identify any potential challenges or constraints not apparent in controlled settings.

ACKNOWLEDGMENT

The authors extend their appreciation to King Saud University for funding this research through Researchers Supporting Project Number (RSPD2024R1027), King Saud University, Riyadh, Saudi Arabia.

REFERENCES

[1] Z. Elamrani Abou Elassad, H. Mousannif, H. Al Moatassime, and A. Karkouch, "The application of machine learning techniques for driving behavior analysis: A conceptual framework and a systematic literature review," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103312.

[2] 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 Interface*, vol. 84, Mar. 2023, Art. no. 103704.

[3] J.-C. Chen, C.-Y. Lee, P.-Y. Huang, and C.-R. Lin, "Driver behavior analysis via two-stream deep convolutional neural network," *Appl. Sci.*, vol. 10, no. 6, p. 1908, Mar. 2020.

[4] A. E. Campos-Ferreira, J. D. J. Lozoya-Santos, J. C. Tudon-Martinez, R. A. R. Mendoza, A. Vargas-Martínez, R. Morales-Menendez, and D. Lozano, "Vehicle and driver monitoring system using on-board and remote sensors," *Sensors*, vol. 23, no. 2, p. 814, Jan. 2023.

[5] A. Sohail, M. A. Cheema, M. E. Ali, A. N. Toosi, and H. A. Rakha, "Data-driven approaches for road safety: A comprehensive systematic literature review," *Saf. Sci.*, vol. 158, Feb. 2023, Art. no. 105949.

[6] 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.

[7] A. Barodi, A. Zemmouri, A. Bajit, M. Benbrahim, and A. Tamtaoui, "Intelligent transportation system based on smart soft-sensors to analyze road traffic and assist driver behavior applicable to smart cities," *Microprocessors Microsystems*, vol. 100, Jul. 2023, Art. no. 104830.

[8] R. Kumar and A. Jain, "Driving behavior analysis and classification by vehicle OBD data using machine learning," *J. Supercomput.*, vol. 79, no. 16, pp. 18800–18819, Nov. 2023.

[9] A. Mohammadnazar, R. Arvin, and A. J. Khattak, "Classifying travelers' driving style using basic safety messages generated by connected vehicles: Application of unsupervised machine learning," *Transp. Res. Part C, Emerg. Technol.*, vol. 122, Jan. 2021, Art. no. 102917.

[10] R. Wang, F. Xie, J. Zhao, B. Zhang, R. Sun, and J. Yang, "Smartphone sensors-based abnormal driving behaviors detection: Serial-feature network," *IEEE Sensors J.*, vol. 21, no. 14, pp. 15719–15728, Jul. 2021.

[11] V. Petraki, A. Ziakopoulos, and G. Yannis, "Combined impact of road and traffic characteristic on driver behavior using smartphone sensor data," *Accident Anal. Prevention*, vol. 144, Sep. 2020, Art. no. 105657.

[12] A. Sharma, S. K. Singh, S. S. Udmale, A. K. Singh, and R. Singh, "Early transportation mode detection using smartphone sensing data," *IEEE Sensors J.*, vol. 21, no. 14, pp. 15651–15659, Jul. 2021.

[13] 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.

[14] E. Lattanzi and V. Freschi, "Machine learning techniques to identify unsafe driving behavior by means of in-vehicle sensor data," *Expert Syst. Appl.*, vol. 176, Aug. 2021, Art. no. 114818.

[15] F. M. Nuswantoro, A. Sudarsono, and T. B. Santoso, "Abnormal driving detection based on accelerometer and gyroscope sensor on smartphone using artificial neural network (ANN) algorithm," in *Proc. Int. Electron. Symp. (IES)*, Sep. 2020, pp. 356–363.

[16] A. Alamri, A. Gumaei, M. Al-Rakhami, M. M. Hassan, M. Alhussein, and G. Fortino, "An effective bio-signal-based driver behavior monitoring system using a generalized deep learning approach," *IEEE Access*, vol. 8, pp. 135037–135049, 2020.

[17] A. Tavakoli, S. Kumar, M. Boukhechba, and A. Heydarian, "Driver state and behavior detection through smart wearables," in *Proc. IEEE Intell. Vehicles Symp.*, Jul. 2021, pp. 559–565.

[18] B. Kim and Y. Baek, "Sensor-based extraction approaches of in-vehicle information for driver behavior analysis," *Sensors*, vol. 20, no. 18, p. 5197, Sep. 2020.

[19] L. Jiang, W. Xie, D. Zhang, and T. Gu, "Smart diagnosis: Deep learning boosted driver inattention detection and abnormal driving prediction," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4076–4089, Mar. 2022.

[20] L. Liu, Z. Wang, and S. Qiu, "Driving behavior tracking and recognition based on multisensors data fusion," *IEEE Sensors J.*, vol. 20, no. 18, pp. 10811–10823, Sep. 2020.

[21] S. Hernández Sánchez, R. F. Pozo, and L. A. H. Gómez, "Driver identification and verification from smartphone accelerometers using deep neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 97–109, Jan. 2022.

[22] A. Raza, I. Akhtar, L. Abualigah, R. A. Zitar, M. Sharaf, M. S. Daoud, and H. Jia, "Preventing road accidents through early detection of driver behavior using smartphone motion sensor data: An ensemble feature engineering approach," *IEEE Access*, vol. 11, pp. 138457–138471, 2023.

[23] M. Malik, P. Sharma, and C. Prabha, "Enhancing transportation safety: An integrated approach using FLFS and OSNCA for advanced driving behavior analysis," *MindVanguard, Beyond Behav.*, vol. 1, no. 1, pp. 1–10, Dec. 2023.

[24] I. Cojocaru and P. S. Popescu, "Building a driving behaviour dataset," in *Proc. Int. Conf. Hum.-Comput. Interact.*, 2022, pp. 101–107.

[25] A. D. Amirruddin, F. M. Muharam, M. H. Ismail, N. P. Tan, and M. F. Ismail, "Synthetic minority over-sampling technique (SMOTE) and logistic model tree (LMT)-adaptive boosting algorithms for classifying imbalanced datasets of nutrient and chlorophyll sufficiency levels of oil palm (Elaeis guineensis) using spectroradiometers and unmanned aerial vehicles," *Comput. Electron. Agricult.*, vol. 193, Feb. 2022, Art. no. 106646.

[26] R. Genuer, J.-M. Poggi, R. Genuer, and J.-M. Poggi, *Random Forests*. Cham, Switzerland: Springer, 2020.

[27] N. A. Priyanka and D. Kumar, "Decision tree classifier: A detailed survey," *Int. J. Inf. Decis. Sci.*, vol. 12, no. 3, p. 246, 2020.

[28] L. Xiong and Y. Yao, "Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm," *Building Environ.*, vol. 202, Sep. 2021, Art. no. 108026.

[29] C. Peng and Q. Cheng, "Discriminative ridge machine: A classifier for high-dimensional data or imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2595–2609, Jun. 2021.

[30] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021.

[31] R. Islam, M. K. Devnath, M. D. Samad, and S. M. Jaffrey Al Kadry, "GGNB: Graph-based Gaussian naive Bayes intrusion detection system for CAN bus," *Veh. Commun.*, vol. 33, Jan. 2022, Art. no. 100442.

[32] S. Naiem, A. E. Khedr, M. Marie, and A. M. Idrees, "Enhancing the efficiency of Gaussian naïve Bayes machine learning classifier in the detection of ddos in cloud computing," *IEEE Access*, vol. 11, pp. 124597–124608, 2023.

[33] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.

[34] Q. Zhang, J. Xiao, C. Tian, J. Chun-Wei Lin, and S. Zhang, "A robust deformed convolutional neural network (CNN) for image denoising," *CAAI Trans. Intell. Technol.*, vol. 8, no. 2, pp. 331–342, Jun. 2023.

[35] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," *Artif. Intell. Agricult.*, vol. 9, pp. 22–35, Sep. 2023.

[36] J. Parker-Holder, V. Nguyen, S. Desai, and S. J. Roberts, "Tuning mixed input hyperparameters on the fly for efficient population based autorl," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 15513–15528.

[37] D. Carneiro, M. Guimarães, M. Carvalho, and P. Novais, "Using meta-learning to predict performance metrics in machine learning problems," *Expert Syst.*, vol. 40, no. 1, Jan. 2023, Art. no. e12900.

[38] G. Varoquaux and O. Colliot, "Evaluating machine learning models and their diagnostic value," *Mach. Learn. Brain Disorders*, vol. 197, pp. 601–630, 2023.

MOHAMMAD SHARIFUL ISLAM received the B.Sc. degree in computer science and telecommunication engineering from Noakhali Science and Technology University, Bangladesh, in 2023, with a focus on deep passion for cutting-edge technologies to the research community. His academic journey, rooted in the confluence of computer science and telecommunications, has evolved into a fervent pursuit of specialized areas, including data science, machine learning, natural language processing, and image processing. His work in these fields is driven by a quest to uncover hidden insights within data, develop intelligent learning algorithms, bridge the communication gap between humans and machines, and artistically enhance digital imagery. As a researcher, his approach is characterized by a blend of technical proficiency and creative problem-solving, aiming to contribute significantly to the frontiers of technology and its application in understanding and improving their digital world.

MOHAMMAD ABU TAREQ RONY received the B.Sc. degree in statistics from Noakhali Science and Technology University, Noakhali, Bangladesh. In addition, he possesses expertise in devising advanced analytics strategies using data. His professional experience is diverse and includes three years of research in areas, such as artificial intelligence. He is currently a Research Data Scientist at aiQuest Intelligence, Dhaka, Bangladesh. Moreover, he actively engages in partnerships with international researchers, recognizing that research plays an indispensable role in fostering innovation. Overall, he is a hardworking individual who has taught himself various skills, such as data analysis, statistics, ML, and DL. He has published articles in refereed journals and conference proceedings, such as IEEE Access, *Data in Brief* (Elsevier), *Children* (MDPI), and International Conferences.

MEJDL SAFRAN received the M.Sc. and Ph.D. degrees in computer science from Southern Illinois University Carbondale, in 2013 and 2018, respectively. He is currently an Assistant Professor with the Department of Computer Science, King Saud University, where he has been a Faculty Member, since 2007, and the Director of the Research Chair of Online Dialogue and Cultural Communication. Since 2018, he has been providing part-time consulting services in the field of artificial intelligence to private and public organizations and firms. He has published papers in refereed journals and conference proceedings, such as *ACM Transactions on Information Systems*, *Applied Computing and Informatics*, *Biomedicines* (MDPI), *Sensors* (MDPI), IEEE International Conference on Cluster, IEEE International Conference on Computer and Information Science, International Conference on Database Systems for Advanced Applications, and International Conference on Computational Science and Computational Intelligence. His research interests include computational intelligence, artificial intelligence, deep learning, pattern recognition, natural language processing, predictive analytics, developing efficient recommendation algorithms for large-scale systems, predictive models for online human activities, machine learning algorithms for performance management, and modeling and analyzing user behavior.

SULTAN ALFARHOOD received the Ph.D. degree in computer science from the University of Arkansas. He is currently an Assistant Professor with the Department of Computer Science, King Saud University (KSU). Since joining KSU in 2007, he has made several contributions to the field of computer science through his research and publications. His research spans a variety of domains, including machine learning, recommender systems, linked open data, text mining, and ML-based IoT systems. His work includes proposing innovative approaches and techniques to enhance the accuracy and effectiveness of these systems. His recent publications have focused on using deep learning and machine learning techniques to address challenges in these domains. His research continues to make significant contributions to the field of computer science and machine learning. His work has been published in several high-impact journals and conferences.

DUNREN CHE received the B.S. degree in electronic engineering from Harbin University of Commerce, China, in 1985, the M.S. degree in computer science from the National University of Defense Technology, in 1988, and the Ph.D. degree in computer science from Beijing University of Aeronautics and Astronautics, in 1994. He was the Director of the Undergraduate Computer Science Programs, School of Computing, Southern Illinois University (SIU), Carbondale, from 2013 to 2022, where he is currently a Professor of computer science. Before joining SIU, in 2001, he was a Postdoctoral Research Fellow with Tsinghua University, German National Research Center for Information Technology, and Johns Hopkins University. He has authored/coauthored more than 120 peer-reviewed papers published in various venues, such as *VLDB Journal*, *Future Generation Computer Systems*, and various ACM/IEEE Transactions and associated conferences. His research interests include databases, data mining, machine learning (collectively data science), cloud computing, and scientific workflow.