IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 40, No. 3, June, 2023, pp. 847-856
Journal homepage: http://iieta.org/journals/ts

Real-Time Driver Drowsiness Detection and Classification on Embedded Systems Using Machine Learning Algorithms

Muhammet Emin Sahin

Department of Computer Engineering, Yozgat Bozok University, Yozgat 66900, Turkey

Corresponding Author Email: emin.sahin@bozok.edu.tr

https://doi.org/10.18280/ts.400302

Received: 25 December 2022
Accepted: 18 May 2023

Keywords:
classification, driver drowsy, embedded systems, Jetson, machine learning

ABSTRACT

Traffic accidents caused by driver drowsiness are a significant concern. An automatic, contactless device capable of early detection and identification of a driver's drowsy state could greatly enhance their safety. This study presents a real-time driver drowsiness detection and classification system implemented on Jetson Nano and Jetson TX2 embedded systems using machine learning algorithms, including Logistic Regression, Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Multi-Layer Perceptron (MLP). Feature extraction is performed on images obtained from video segments within the dataset, followed by a normalization process. The normalized features are classified using machine learning algorithms, and the results are reported. A 10-fold cross-validation model is employed during the experiment, and the grid search hyperparameter optimization (GSHPO) method is used to fine-tune the classifier algorithms' parameters for the proposed system. The MLP classifier outperformed the other classifiers, achieving an accuracy, F1-score, and AUC (the area under the receiver operating characteristic curve (ROC)) of 0.91, 0.91, and 0.90, respectively. The developed system is implemented on Jetson Nano and Jetson TX2 embedded systems, and the frames per second (FPS) results are provided for comparison. The high accuracy of this hardware-based system in detecting drowsy driving, along with its portability for in-vehicle use, is a critical aspect of this work.

1. INTRODUCTION

Most people view traffic accidents as unanticipated occurrences that cause damage and human casualties. Because of this, it has always posed a threat. A total of 39,871 traffic accidents occurred in October 2021 across Turkey. While 228 of these resulted in death, 17,971 resulted in injuries [1]. As seen in Table 1, most of these accidents in our country are driver-related accidents. Table 2 shows the total number of accidents, deaths, and injuries in 2021. Accidents do happen, but not only in Turkey; have they happened often around the world. According to the International Association for Safe Road Travel, an average of 1.3 million people die in traffic accidents each year. This means that there are 3287 deaths per day [2]. World Health Organization (WHO), traffic accidents rank among the ten leading causes of mortality worldwide [3].

The most important condition affecting the ability of drivers to drive is fatigue or drowsiness. Due to decreased focus and attention, a driver who falls asleep is more likely to be in a high-speed crash. When the studies and experiments in the literature are examined, the fact that driving performance worsens with increasing drowsiness has been confirmed [4]. Research from the Adelaide Center for Sleep Research demonstrates that drivers who remain up for 24 hours have a seven-fold increased risk of an accident and perform as poorly behind the wheel as someone with 0.1 g/100 ml of alcohol in their blood [5]. Driving in a drowsy state is one of the important causes of traffic accidents. A lot of research has been done to find ways to make transportation safer and cut down on the number of deaths caused by drivers who are too tired to drive [6].

Table 1. Causes of traffic accidents in 2021 [1]

No	Traffic Accidents	Total
1	Number of Accidents	394,827
2	Number of Fatal Accidents	1,880
3	Number of Injury Accidents	170,383
4	Number of Material Damage Accidents	222,564
5	Number of Dead*	2,245
6	Number of Injured	255,300

Table 2. Traffic accidents in 2021 [1]

Item No	Defect Elements	Year 2021
1	Driver	180,023
2	Pedestrian	16,539
3	Vehicle	5,315
4	Road	840
5	Passenger	3,545

An eye-tracking-based driver drowsiness system has been proposed by Said et al. [7]. In this operation, the system detects the driver's sleep state and sounds the alarm to warn the driver. In their study, the Viola Jones model was used to detect the face and eye regions When the tests were done inside, the accuracy was 82%, and when they were done outside, it was 72.8%.

Akalya et al. [8] introduced a fatigue detection system employing the Raspberry Pi 3 embedded system board. By analyzing facial and eye images through a Haar cascade

classifier and calculating the eye aspect ratio (EAR) utulizing the Euclidean distance between the driver's eyes, they successfully identified blink durations. Krajewski et al. [9] developed a model for finding drowsiness based on steering patterns. They utilized advanced signal processing techniques to create three distinct feature sets. Performance is evaluated using five machine learning algorithms such as Support Vector Machine (SVM) and K-Nearest Neighbor (KNN), which achieved 86% accuracy in drowsiness detection. A method has been proposed by Mittal et al. [10] that can detect driver drowsiness at an early stage and prevent problems from occurring. After pre-processing the data using various features such as mouth aspect ratio, eye aspect ratio, pupil circularity, and mouth over eye ratio, the algorithms of KNN, Naive Bayes, Logistic Regression, Decision Trees, Random Forest, XGBoost are evaluated and they obtained 75.67% accuracy [10]. In their study, Dwivedi et al. [11] proposed a model for detecting drowsiness using Convolutional Neural Networks (CNNs). They utilized CNNs to extract the underlying features from the input data, followed by a SoftMax layer for classification. The results of their experiments showed an accuracy of 78% in detecting driver drowsiness using this approach. Tadesse et al. [12] proposed a method for detecting driver drowsiness using the Hidden Markov Model (HMM) Their strategy involves several phases, including the use of the Viola Jones algorithm to extract face regions, the Gabor wavelet decomposition to extract features from the face regions, and the Adaboost learning algorithm to choose the most pertinent characteristics. After feature selection, HMM was used for the classification of drowsy or non-drowsy expressions. The proposed method leveraged the ability of HMM to model sequential data and identify patterns in time series, which is crucial for accurately detecting driver drowsiness. Jabbar et al. [13] proposed a model for detecting driver drowsiness using deep learning specifically designed for android applications. Their approach involved facial landmark point detection, where images were first extracted from video frames, and Dlib library was used to extract landmark coordinate points from the facial regions. These landmark coordinates were then used as input by a multilayer perceptron classifier to categorize the identified face landmarks as drowsy or non-drowsy. The proposed method was evaluated on the NTHU Drowsy Driver Detection Dataset, and it achieved an accuracy of more than 80%. In a separate study, Picot et al. [14] introduced a method that combined visual activity and brain activity to detect driver drowsiness. Brain activity was monitored using a single-channel EEG, while visual activity was monitored through blinking and characterization. By extracting blinking features using EOG and employing fuzzy logic to combine these features, an EOG-based detector was developed. The method underwent evaluation using a dataset comprising twenty individual drivers, yielding an accuracy of 80.6%.

In this study, a system is proposed to detect driver drowsiness status and prevent possible accidents. A dataset of drowsy driver states is obtained. Basic features are extracted using the video dataset, and then the extracted features are normalized. Then, these data are classified on Jetson Nano and Jetson TX2 embedded system boards by applying different machine learning algorithms such as Logistic Regression, Naïve Bayes, KNN, Decision Tree, Rondom Forest, and MLP and the results are compared. An overview of the proposed system is given in Figure 1.

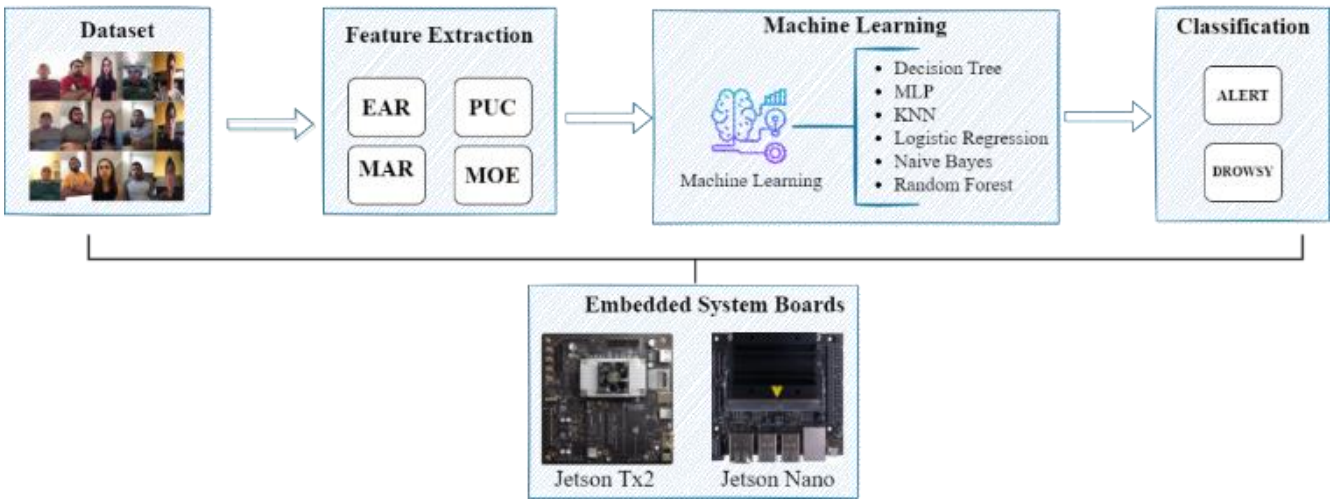


Figure 1. An overview of the proposed system

2. MATERIAL AND METHODS

2.1 Dataset

UTA Real-Life Drowsiness dataset is used in this study [15]. This dataset was created by a research team from the University of Texas specifically to detect multistage dormancy. It is presented to assist in the identification of multistage sleepiness, including not only cases of excessive sleepiness and the obvious, but also subtle cases of small micro-expressions.

The dataset consists of approximately 30 hours of RGB channelled videos of 60 healthy participants. A total of 180 videos are accessible, with each participant having a video representation across three distinct classes: wakefulness, low alertness, and drowsiness. The dataset comprises 51 male and 9 female participants, encompassing individuals from diverse ethnic backgrounds, including 10 Caucasians, 5 non-white Hispanics, 30 of Indian Aryan and Dravidian descent, 8 Middle Eastern individuals, and 7 East Asians. Additionally, the dataset encompasses a wide range of ages. Data is obtained for both the alert and drowsy states. Using methods for

extracting features, the data is changed into a set of data with 10561 rows.

2.2 Feature extraction

The term "feature," which is frequently used in applications for image processing, machine learning, deep learning, data mining, and pattern recognition, is best defined as the qualitative aspect of an object. Dimensionality reduction can be achieved through two approaches: feature selection and feature extraction. While feature selection involves choosing relevant features from the original data, feature extraction involves transforming the original data into new features that possess robust pattern recognition capabilities [16].

The process of feature extraction is facilitated by the utilization of the “Dlib” library, a versatile C++ library that encompasses various machine learning techniques, including classification, clustering, and regression [17]. The appropriate features for the grouping models are created based on 68 face iterations, as seen in Figure 2. In this study, the mouth-to-aspect ratio, eye-to-aspect ratio, and mouth-eye-aspect ratio are the features that are taken into account.

$$EAR = \frac{\|P2-P6\|+\|P3-P5\|}{2\|P1-P4\|} \quad (1)$$

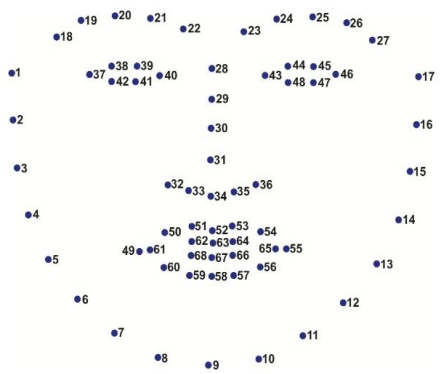


Figure 2. Iteration of 68 symbols on a human face [18]

After eliminating the bookmarks from the video frames, all of the aforementioned characteristics are computed. When the figure is examined, iterations 49-68 for the mouth and 37-48 for the eyes are used.

Eye Aspect Ratio (EAR): The Eye Aspect Ratio (EAR) was introduced by Soukupova and Cech in 2016 and serves as a measure of the proportion between the height and width of the eye [19]. It is known that blinking increases when the person is in a drowsy state. Here, if the blink frequency falls below a certain value, it is determined that the person is in a sleep state. The EAR equation is given in Eq. (1). Eye aspect ratio is shown in Figure 3.

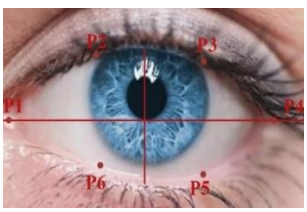


Figure 3. Eye aspect ratio

Pupil Circularity (PUC): The EAR is a complementary measurement. A drowsy person has a lower PUC than a waking person [19]. The equations of PUC are given in Eqns. (2) and (3).

$$Circularity = \frac{4*\pi*Area}{perimeter^2} \quad (2)$$

where, $perimeter = |p_1 - p_2| + |p_2 - p_3| + |p_3 - p_4| + |p_4 - p_5| + |p_5 - p_6| + |p_6 - p_1|$

$$Area = \left(\frac{Distance(p2,p5)}{2}\right)^2 * \pi \quad (3)$$

Mouth Aspect Ratio (MAR): The MAR is the ratio between the height and width of the mouth [19]. A person's drowsy state is also proportional to yawning. In case the detected yawning is above a certain value, sleep status will be detected. The MAR equation is given in Eq. (4). Mouth aspect ratio, where EF and AB are the distance between the points, is shown in Figure 4.

$$MAR = \frac{|EF|}{|AB|} \quad (4)$$

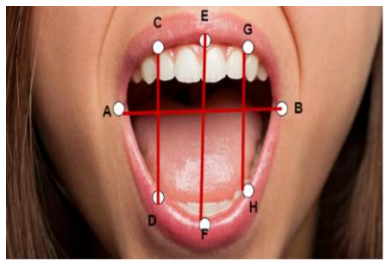


Figure 4. Mouth aspect ratio

Mouth aspect ratio over Eye aspect ratio: MOE is the ratio of MAR and EAR. The MOE change is important. It is the opposite in drowsy and alert individuals [19]. The MOE equation is given in Eq. (5).

$$MOE = \frac{MAR}{EAR} \quad (5)$$

2.3 Feature normalization

Feature normalization is a crucial preprocessing step that scales and standardizes the range of features, ensuring that they are on a comparable scale and avoiding biases caused by differing magnitudes. The default alert state of individuals is characterized by unique distinguishing features. For instance, person X may naturally have smaller or larger eyes compared to person Y, highlighting the inherent variations in physical attributes across different individuals. Since a classifier is trained on person Y, it will detect a decrease in EAR and PUC even though the person is awake, the drowsy state will be estimated each time it is tested on person X. Based on this, more accurate results will be produced by standardizing the characteristics for each individual. Therefore, the normalization process is required. The normalization formula is mathematically shown in Eq. (6) [20].

$$Normalized\ Feature_{n,m} = \frac{Feature_{n,m} - \mu_{n,m}}{\sigma_{n,m}} \quad (6)$$

where, mean and standard deviation for feature n in the first

third of blinks in the alert state video for subject m is denoted by μ(n,m) and σ(n,m), respectively [20]. After the feature extraction process, the average and standard deviation are taken to normalize the four feature values obtained, and after each value, the normalization process takes place.

The distribution of values after normalization is shown in Figure 5, and it appears that MOE_N is the most important characteristic for identifying driver drowsiness. This feature has the highest mean value and the smallest standard deviation among all the features. Thus, it indicates that MOE_N is a crucial factor in detecting driver drowsiness, and it can provide the most discriminative information for the classification model. Additionally, PUC change also appears to be an essential feature in the detection of driver drowsiness, with a mean value and standard deviation comparable to some of the other important features. This suggests that PUC change can provide meaningful information in the classification process and help improve the overall accuracy of the system.

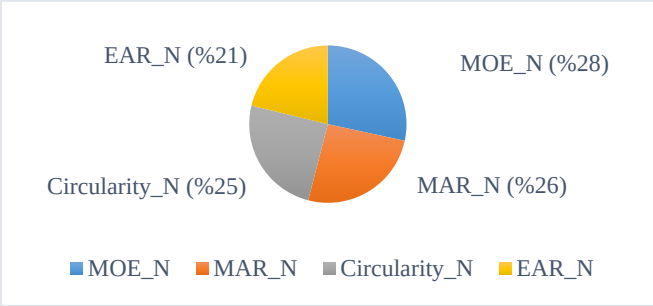


Figure 5. Diagram of distribution of features

2.4 Machine learning

Machine learning, a subfield within the broad domain of artificial intelligence, is centered around the utilization of sophisticated algorithms to meticulously examine data, extract valuable knowledge, and facilitate well-informed predictions or decisions. This powerful discipline empowers systems to autonomously learn from data patterns, adapt to new information, and continuously enhance their performance. By leveraging problem-specific training data, machine learning systems possess the ability to automate the creation of analytical models and solve various associated tasks [21]. In this study, six different algorithms (Random Forest, Decision Tree, Naive Bayes, KNN, Logistic Regression, and MLP) are used for detecting and classifying the driver's drowsiness. These algorithms are being put to the test to see which one performs the best at detecting and categorizing driver drowsiness. They then used the grid search hyperparameter optimization (GSHPO) method to fine-tune the parameters of the classifiers for optimal performance.

2.4.1 K-Nearest Neighbours (KNN)

Instead of using the probability theorem, the K-nearest neighbor (K-NN) algorithm performs classification based on the distance between data points [22]. The basic idea in this model is to determine the nearest neighbor. Euclidean calculation is made between the test sample and the training data to determine the class. Here, a loop is established for the *k*-value value and the *k*-value value that gives the best result is taken. In our study, the *k*-value is selected as 25.

2.4.2 Logistic regression

It is used to solve binary classification problems and estimate the probability of a binary result [23]. Using logistic regression is an appropriate regression strategy, especially when the dependent variable is binary. This method is a binary classifier ideal for detecting drowsy or alert states.

2.4.3 Naive Bayes

Naive Bayes is a classification method based on Bayes' Theorem. Basically, according to a Naive Bayes classifier, the presence of a particular feature in a class is unrelated to the existence of another feature. The Naive Bayesian model is easy to construct and extremely useful when dealing with large datasets [23]. In this study, Bernoulli Naive Bayes is determined as a classifier.

2.4.4 Decision tree

The decision tree is a versatile supervised learning technique utilized for both classification and regression tasks [23]. Unlike parametric methods, the decision tree does not impose any assumptions about the underlying data distribution. Instead, it learns from labelled data to construct a classification model. Decision tree algorithms are popularly employed in creating classification models due to their similarity to human reasoning processes [24]. In this study, the maximum depth is examined with a loop, and the best result is obtained by using two depths.

2.4.5 Random Forest

This classifier is based on ensemble learning and uses multiple decision trees to interpret the data [22, 25]. In the random forest, rather than using a deterministic method, each node's decision is chosen through a random process. The classifier's accuracy is determined by averaging each tree's accuracy.

2.4.6 Multi-Layer Perceptron (MLP)

MLP has a structure in which many neurons with non-linear activation functions in architectural terms are hierarchically connected [10]. In this model, the optimizer is selected as a sigmoid. Adam is an activation function and there are hidden layers in the MLP system.

2.5 Embedded system

Embedded systems are microprocessor-based computer hardware systems that perform a specific task as a stand-alone system or as part of a large system and have their software. Due to its low power consumption, processing power, cost, and real-time capabilities, this study is analyzed on different embedded system boards and the results are presented. Jetson Nano and Jetson TX2 are given in Figure 6.

Jetson TX2: The Jetson TX2 is a powerful and energy-efficient artificial intelligence (AI) embedded computer system [26]. This system is based on the Nvidia Graphics Processing Unit (GPU) and contains hardware interfaces that support a wide range of goods and programs.

Jetson Nano: Image classification, object detection, and segmentation are common applications that utilize the NVidia Jetson Nano [27]. It is a tiny microprocessor board used for constructing and training models with GPU 128-core Maxwell to display AI frameworks and apps efficiently. Technical specifications are given in Table 3.

Table 3. Features of used hardware [28]

	Jetson TX2	*Jetson Nano*	*Hp pavilion*
GPU	NVIDIA Pascal, 256 CUDA cores	NVIDIA Maxwell, 128 CUDA cores	NVIDIA GeForce GTX 1050 Ti
CPU	Dual-core Denver 2 64-bit + Quad-core ARM Cortex-A57 MPCore	Quad-core ARM Cortex-A57 MPCore	Intel(R) Core(TM) i5-8300H CPU
Memory	8 GB 128-bit LPDDR4	4 GB 64-bit LPDDR4	16,0 GB Ram 64-bit
Data storage	32 GB eMMC, SDIO, SATA	MicroSD card	16 GB RAM

(a)

(b)

Figure 6. (a) Jetson Nano and (b) Jetson TX2

3. PREPROCESSING

In this section, the data obtained after the feature extraction process is classified with the help of the algorithms given below, and the results are obtained. The "significance of acquired features" refers to techniques that calculate a score for all input features for a given model. A higher score means that a particular feature will have a greater impact on the model used to predict a particular variable. The feature distribution graph of the data with the feature extraction process is given in Figure 7. In this distribution, it is shown which feature occurs and how often.

The correlation matrix is a graphical representation of the data to show which features are most correlated with the target feature [29]. Each feature in a dataset is represented as colors. This means that colours provide information to researchers about the relationships between traits. Considering the values alone may not be sufficient with big data to understand it. It shows the correlation between each square and the variables in the correlation map.

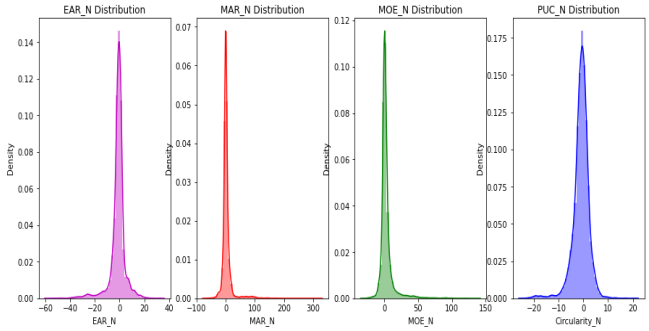


Figure 7. The feature distribution graph of the features

Correlation is often used to determine whether there is a cause-and-effect relationship between two variables. This value varies between -1 and 1. The closer it is to 1, the more positive it is. If the value is close to -1, it is negatively correlated. In addition, if the value is 0, it means that there is no relationship between the variables. The reason the diagonals are 1 is that each variable is associated with itself there. The larger the number and the darker the color, the more successful the correlation. There is a strong positive correlation between Circularity_N & EAR_N, Circularity_N & MAR_N. There is a strong negative correlation between MOE_N & EAR_N, Circularity_N & MAR_N. The feature correlation heat map is shown in Figure 8.

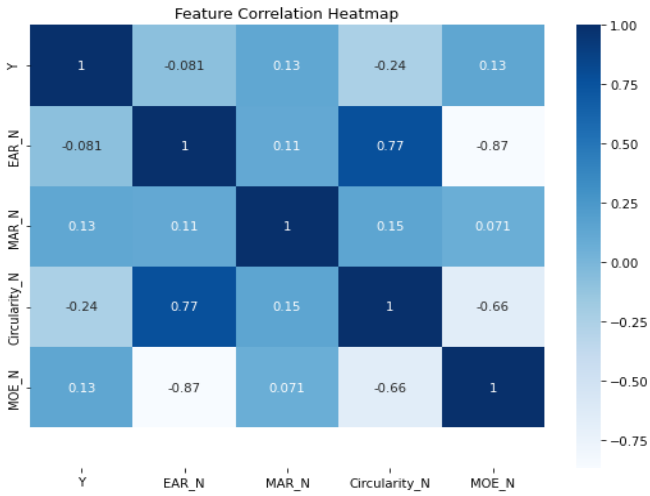


Figure 8. Feature correlation heat map

Grid Search

The most popular technique for studying the hyperparameter configuration space is grid search (GS) [29, 30]. Grid search is a thorough investigation or a brute force approach that evaluates all possible combinations of hyperparameters given a grid configuration. The way GS operates is by measuring the cartesian product of a user-defined, finite set of values [30, 31]. The grid search method is used to select the most suitable parameter for the classifiers chosen for this research and to receive classifier results.

Cross-validation

The system's ability to generalize, or how well it performs on data that has not been seen before, is typically the primary concern in regression and classification operations. The model building typically does not use the data from the testing section to obtain an accurate estimate of performance [32].

As a result, many issues arise, and it is usual practice to use

k-fold cross-validation in classification and regression, where all available data is randomly divided into k sets to deal with these issues [32, 33]. Each set is used once as a test set and k times total during the training or model fitting process, with the additional sets being used for model building. Since all the data is used for testing as well as training, the method eventually obtains *k*-independent realizations of the error measure. By taking the average of the k-error measures, an overall measure of error is made that is usually more accurate than a single measure.

Performance Evaluation

Before the prediction model is developed, all models must be assessed using various evaluation criteria [30, 34]. To date, we have evaluated our prediction models using accuracy scores. However, in some cases, a model's accuracy score alone isn't sufficient to assess it properly because, in the case of a low accuracy score, it doesn't specify which class (positive or negative) our models are incorrectly predicting. We calculate the accuracy, precision score, recall score, and F1-score for both models to make this clearer. Parameters equations are given in Eqns. (7)-(10) [35, 36].

$$Accuracy = \frac{tn+tp}{tn+tp+fn+fp} \tag{7}$$

$$Recall = \frac{tp}{tp+fn} \tag{8}$$

$$F1_{Score} = 2 * \frac{Precision\ x\ Recall}{Precision+Recall} \tag{9}$$

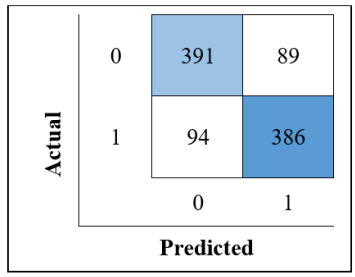
$$Precision = \frac{tp}{tp+fp} \tag{10}$$

4. EXPERIMENTAL RESULTS

This section displays the results of the 10-fold cross-validation after optimization, the performance metrics results for each model, the confusion metric results, and a comparison of each prediction model before and after the application of the grid search-based hyperparameter technique. The performance of the disease prediction test result prediction model is compared for performance evaluation in terms of goodness-of-fit. As the model is compared in this proposed method, machine learning algorithms such as Logistic Regression, Naive Bayes, KNN, Decision Tree, Random Forest, and MLP are used.

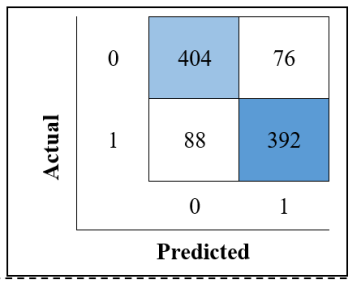
When a model makes predictions for each class, the confusion matrix displays the number of correct classifications as well as misclassifications (positive or negative). The confusion matrix, which displays correctly and incorrectly predicted samples in the test results, is used to validate our work. Based on the training data employed in this study's model, Figures 9-14 visually depict the accurate prediction of the confusion matrix and classification report for each individual model. These figures offer comprehensive insights into the model's performance, showcasing its ability to make precise predictions based on the provided training data. In the confusion matrix, "0" is the label of the alert class, and "1" is the label of the drowsy class.

The driver drowsiness detection model training has been carried out successfully using machine learning and computer vision. In this study, six different machine learning algorithms which are Random Forest, Decision Tree, Naive Bayes, KNN, Logistic Regression, MLP algorithms. These models are evaluated by comparing each other according to the success of accuracy and F1-score. The ROC curve, accuracy graph, and F1-score of all algorithms used are shown in Figure 15.



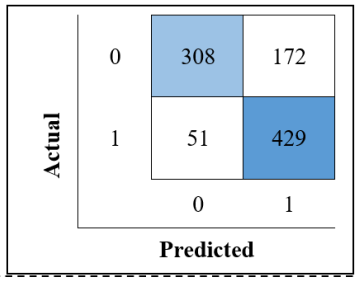
	precision	recall	f1-score	support
0	0.81	0.81	0.81	480
1	0.81	0.80	0.81	480
Accuracy			0.81	960
Macro avg	0.81	0.81	0.81	960
Weighted avg	0.81	0.81	0.81	960

Figure 9. Confusion matrix and classification report of KNN



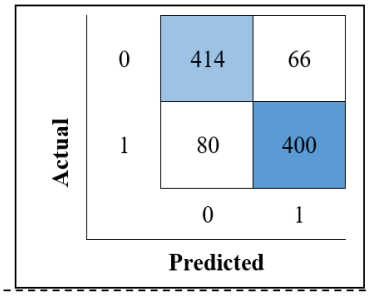
	precision	recall	f1-score	support
0	0.82	0.84	0.83	480
1	0.84	0.82	0.83	480
Accuracy			0.83	960
Macro avg	0.83	0.83	0.83	960
Weighted avg	0. 83	0.83	0.83	960

Figure 10. Confusion matrix and classification report of Logistic Regression



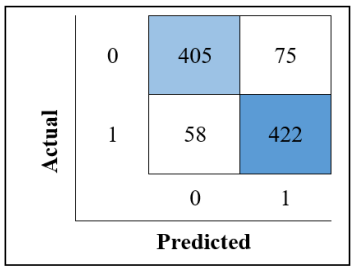
	precision	recall	f1-score	support
0	0.86	0.64	0.73	480
1	0.71	0.89	0.79	480
Accuracy			0.77	960
Macro avg	0.79	0.77	0.76	960
Weighted avg	0.79	0.77	0.76	960

Figure 11. Confusion matrix and classification report of Naive Bayes



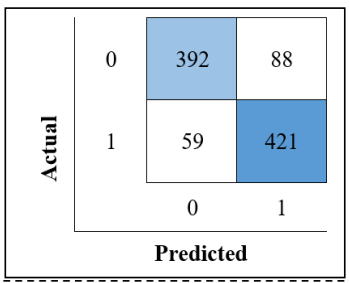
	precision	recall	f1-score	support
0	0.84	0.86	0.85	480
1	0.86	0.83	0.85	480
Accuracy			0.85	960
Macro avg	0.85	0.85	0.85	960
Weighted avg	0.85	0.85	0.85	960

Figure 12. Confusion matrix and classification report of Decision Tree



	precision	recall	f1-score	support
0	0.87	0.84	0.86	480
1	0.85	0.88	0.86	480
Accuracy			0.86	960
Macro avg	0.86	0.86	0.86	960
Weighted avg	0.86	0.86	0.86	960

Figure 13. Confusion matrix and classification report of Random Forest



	precision	recall	f1-score	support
0	0.94	0.89	0.92	480
1	0.90	0.94	0.92	480
Accuracy			0.92	960
Macro avg	0.92	0.92	0.92	960
Weighted avg	0.92	0.92	0.92	960

Figure 14. Confusion matrix and classification report of MLP

Table 4 shows the result of GSHPO for the six models used in this paper. As in Table 5, in the testing part, MLP is fitted and executed with the best parameters, after GSHPO is applied and found to be 91.77%, 91.97%, 90.04%, accuracy, F1, and AUC-ROC value, respectively. The MLP algorithm is followed by Random Forest, Decision Tree, Logistic Regression, KNN, and Naive Bayes, algorithms with F1-score of 86.38%, 84.56%, 82.70%, 80.83%, and 79.37%, respectively.

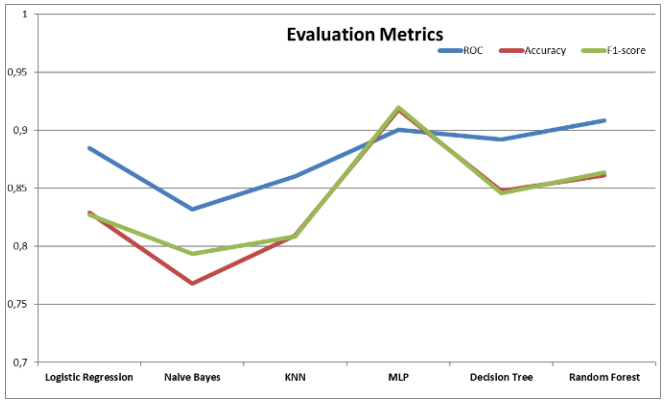


Figure 15. Graph of evaluation metrics

Table 4. Accuracy results of used algorithms

Model	Accuracy	F1-Score	ROC (AUC)
Logistic R.	0.82916	0.82700	0.88457
KNN	0.80937	0.80837	0.86021
Decision Tree	0.84791	0.84566	0.89202
Random Forest	0.86145	0.86386	0.90837
Naive Bayes	0.76770	0.79371	0.83187
MLP	0.91770	0.91979	0.90044

The results of all models in the AUC-ROC comparison are submitted. The random forest models achieve the highest ROC/AUC results, which are 0.90837, and the NB models achieve the lowest ROC/AUC results, which are 0.83187. All models' ROC results given in Figure 16 are above 80%, which indicates that GSHPO significantly contributes to improving the results.

The statistical significance of each attribute in the data is described in terms of its model effect by the importance of the feature. The model determines how to calculate each feature's importance. The significance of features for the MLP with the highest accuracy is depicted in Figure 17. All features are more significant in RF. There are no features of any value. Nearly all features help make predictions, but some are more crucial than others, such as Circularity_N and MAR_N.

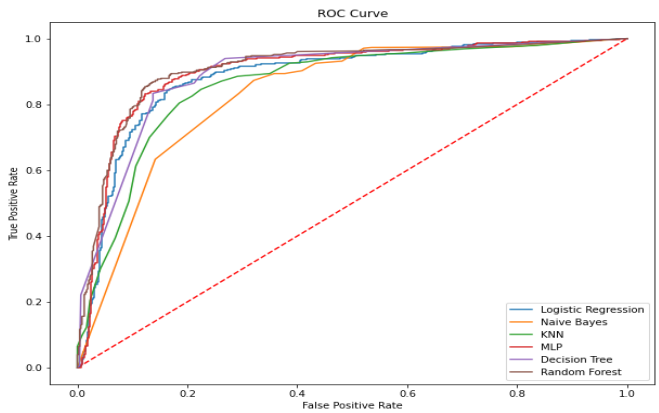


Figure 16. ROC curve of used algorithms

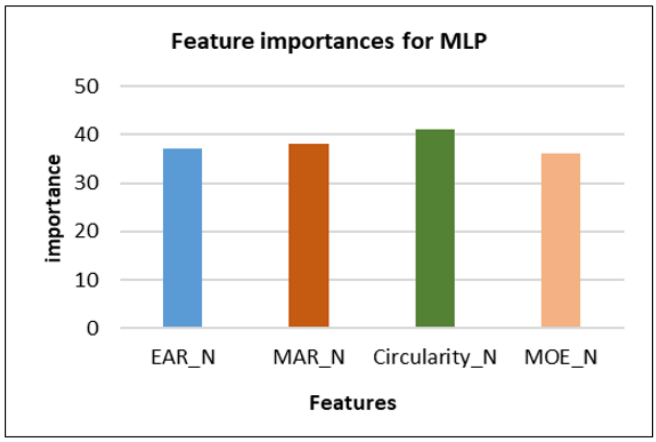


Figure 17. Feature importance for MLP algorithm

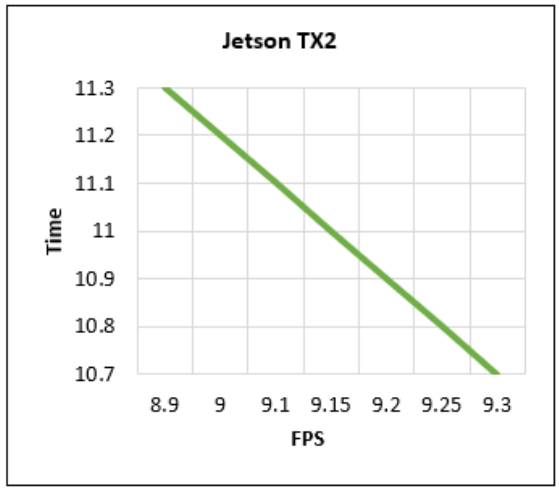
In addition to the personal computer, this study is tested on different embedded systems such as Jetson Nano and Jetson TX2, and the results are obtained, and the FPS graphics of these systems are given in Figure 18. Features that detect the driver's drowsiness state based on human facial expressions are used. Facial images are taken with the help of the camera without disturbing the driver and transferred to the Jetson Nano and Jetson TX2 cards, allowing the system to work successfully. Also, the success of this system does not depend on the age of the driver or the model of the car. With the work done here, besides taking advantage of the high speed and low power consumption features of embedded system cards, the portability of the designed model is also provided.

This study is also performed on the Jetson Nano and Jetson TX2 platforms, and the comparative results are presented in Table 5. It is seen that testing and training times are shorter in the designed system on the Jetson TX2 platform compared to the Jetson Nano.

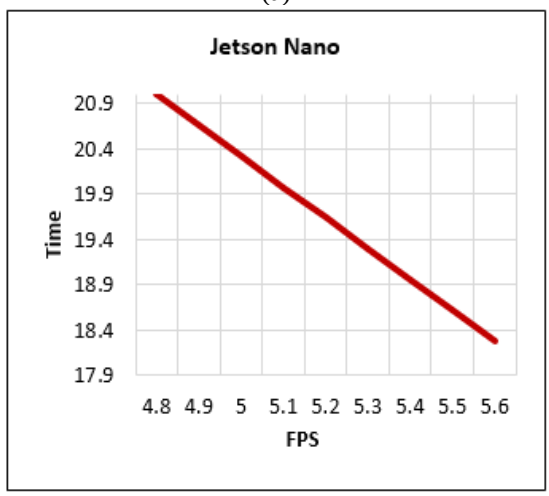
Among the classifiers used in all environments, it is seen that the random forest algorithm comes to the forefront in terms of time. The algorithm that gives the longest result is the decision tree classifier. Comparison of the score (accuracy), training, and testing times on the PC, Jetson Nano, and TX2 environments are given in Figures 19–21, respectively.

Table 6 lists and analyses related studies from the literature.

A study has compared the results of their machine learning classifiers to those of previous studies found in the literature and found that their classifiers achieved higher accuracy. This is a positive result and indicates that our machine learning classifiers may be more effective than previous methods.



(a)



(b)

Figure 18. FPS graph of embedded systems (a) JetsonTX2; (b) Jetson Nano

Table 5. Comparison of training and testing time between different environments

Hardware	Parameters	Model					
		Random F.	Logistic R.	Naïve Bayes	KNN	Decision T.	MLP
PC	Score	0.8614	0.8291	0.7677	0.80937	0.8479	0.9177
	Training time (s)	141.121	4.2408	3.8186	31.8176	17.0075	8.4737
	Testing time (s)	0.06860	0.0068	0.0037	0.04266	0.00261	0.00546
Jetson Nano	Score	0.8614	0.8291	0.7677	0.8093	0.8479	0.8479
	Training time (s)	232.695	7.4163	0.3604	57.6301	33.2625	21.4636
	Testing time (s)	0.0700	0.0076	0.0319	0.1859	0.0076	0.0122
Jetson TX2	Score	0.8614	0.8291	0.7677	0.8093	0.8479	0.8468
	Training time (s)	179.164	4.1637	3.1086	44.7255	24.0499	10.4484
	Testing time (s)	0.0498	0.0034	0.0034	0.1453	0.0034	0.0163

Table 6. Comparison of studies in the literature

No	Study	Logistic Regression	Naive Bayes	KNN	Decision Tree	Random Forest	MLP
1	Our Study	0.8291	0.7677	0.8093	0.8479	0.8614	0.9177
2	[10]	0.7567	0.6479	0.6515	0.6541	0.6041	0.6942
3	[19]	0.64	0.70	0.75	0.74	0.77	
4	[9]			0,861			
5	[37]				0.735	0.824	

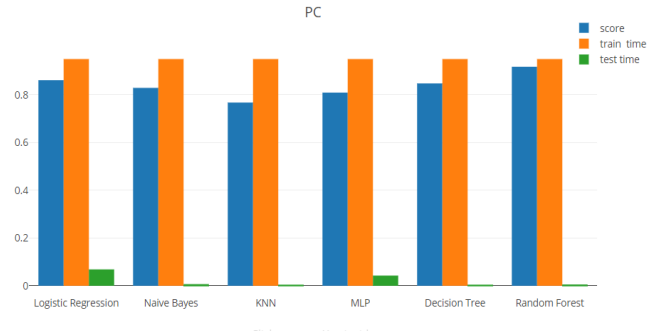


Figure 19. Comparison of the score (Accuracy), training, and testing time on the PC environment

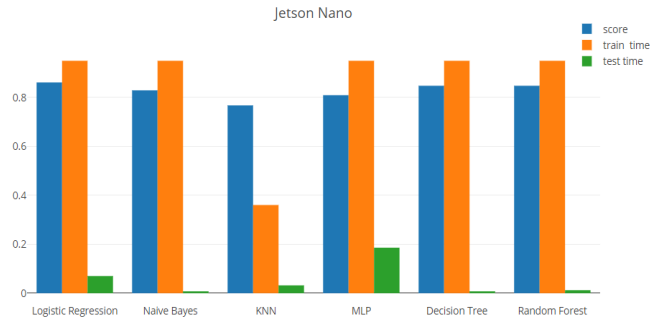


Figure 20. Comparison of the score (Accuracy), training, and testing time on the Jetson Nano environment

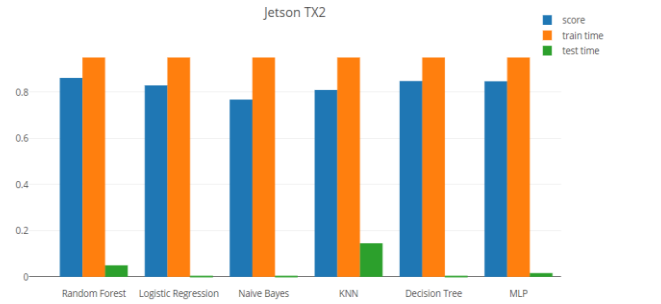


Figure 21. Comparison of the score (Accuracy), training, and testing time on the Jetson TX2 environment

5. CONCLUSIONS

This research proposes a system for real-time driver drowsiness monitoring based on ocular behavior and machine learning by using hyperparameter optimization with a grid search approach (GSHPO). Here, visual behavior features such as MOE-N, EAR-N, Circularity-N, and MAR-N are calculated from a webcam's video stream and feature extraction is carried out. Normalization is applied to the obtained features, and feature distribution and heat map graphs are provided. By using normalized features, classification is performed with the help of Logistic regression, Naive Bayes, KNN, Decision Tree, Random Forest, and MLP machine learning algorithms. The MLP classifier has been observed to outperform other classifiers. The accuracy, ROC, and F1 scores for the MLP are obtained to be 0.91, 0.90, and 0.91, respectively. In addition, the planned system has been implemented on the embedded system platforms Jetson Nano and Jetson TX2, and a comparison is shown. The fact is that this study, which is offered as hardware, can be utilized on the vehicle system in a portable manner is crucial to the work accomplished. The method put forth in this paper could be expanded upon for use in subsequent studies by using a larger number of datasets and/or more expensive algorithms to aid in the validation of the GSHPO under various circumstances.

DATA AVAILABILITY

Data in the work is available and collected in the literature which is referenced in the manuscript.

REFERENCES

[1] Emniyet Genel Müdürlüğü Trafik Başkanlığı. Trafik İstatistik Bülteni Ekim 2021, accessed on Jun. 11, 2022.

[2] Girit, A. (2014). Drowsy driver detection using image processing. Master's thesis, Middle East Technical University.

[3] World Health O. https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death, accessed on Jun. 11, 2022.

[4] Rau, P.S. (2005). Drowsy driver detection and warning system for commercial vehicle drivers: Field operational test design, data analyses, and progress. In 19th International Conference on Enhanced Safety of Vehicles, pp. 6-9.

[5] Lowrie, J., Brownlow, H. (2020). The impact of sleep deprivation and alcohol on driving: A comparative study. BMC Public Health, 20(1): 1-9. https://doi.org/10.1186/s12889-020-09095-5

[6] Renn, R.P., Cote, K.A. (2013). Performance monitoring following total sleep deprivation: Effects of task type and error rate. International Journal of Psychophysiology, 88(1): 64-73. https://doi.org/10.1016/j.ijpsycho.2013.01.013

[7] Said, S., AlKork, S., Beyrouthy, T., Hassan, M., Abdellatif, O., Abdraboo, M.F. (2018). Real-time eye tracking and detection-a driving assistance system. Advances in Science, Technology, and Engineering Systems Journal, 3(6): 446-454.

[8] Chellappa, A., Reddy, M.S., Ezhilarasie, R., Suguna, S. K., Umamakeswari, A. (2018). Fatigue detection using raspberry pi 3. International Journal of Engineering & Technology, 7(2.24): 29-32.

[9] Krajewski, J., Sommer, D., Trutschel, U., Edwards, D., Golz, M. (2009). Steering wheel behavior-based estimation of fatigue. In Driving Assessment Conference, 5(2009): 118-224. https://doi.org/10.17077/drivingassessment.1311

[10] Mittal, S., Gupta, S., Shamma, A., Sahni, I., Thakur, N. (2021). Driver drowsiness detection using machine learning and image processing. In 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, pp. 1-8. https://doi.org/10.1109/ICRITO51393.2021.9596358

[11] Dwivedi, K., Biswaranjan, K., Sethi, A. (2014). Drowsy driver detection using representation learning. In 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, India, pp. 995-999. https://doi.org/10.1109/IAdCC.2014.6779459

[12] Tadesse, E., Sheng, W., Liu, M. (2014). Driver

drowsiness detection through HMM-based dynamic modeling. In 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, pp. 4003-4008. https://doi.org/10.1109/ICRA.2014.6907440

[13] Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., Jiang, S. (2018). Real-time driver drowsiness detection for an android application using deep neural networks techniques. Procedia Computer Science, 130: 400-407. https://doi.org/10.1016/j.procs.2018.04.060

[14] Picot, A., Charbonnier, S., Caplier, A. (2011). Online detection of drowsiness using brain and visual information. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 42(3): 764-775. https://doi.org/10.1109/TSMCA.2011.2164242

[15] UTA-RLDD. https://sites.google.com/view/utarldd/home, accessed on Jun. 11, 2022.

[16] Cai, J., Luo, J., Wang, S., Yang, S. (2018). Feature selection in machine learning: A new perspective. Neurocomputing, 300: 70-79. https://doi.org/10.1016/j.neucom.2017.11.077

[17] dlib C++ Library. http://dlib.net/, accessed on Jun. 19, 2022.

[18] Phan, A.C., Nguyen, N.H.Q., Trieu, T.N., Phan, T.C. (2021). An efficient approach for detecting driver drowsiness based on deep learning. Applied Sciences, 11(18): 8441. https://doi.org/10.3390/app11188441

[19] Meda, H., Ganesh, J.M.P., Sahani, A. (2021). Machine learning models for drowsiness detection. In 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Glasgow, United Kingdom, pp. 1-5. https://doi.org/10.1109/I2MTC50364.2021.9460017

[20] Ghoddoosian, R., Galib, M., Athitsos, V. (2019). A realistic dataset and baseline temporal model for early drowsiness detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

[21] Janiesch, C., Zschech, P., Heinrich, K. (2021). Machine learning and deep learning. Electronic Markets, 31(3): 685-695. https://doi.org/10.1007/s12525-021-00475-2

[22] Raghu, S., Sriraam, N. (2018). Classification of focal and non-focal EEG signals using neighborhood component analysis and machine learning algorithms. Expert Systems with Applications, 113: 18-32. https://doi.org/10.1016/j.eswa.2018.06.031

[23] Rashed, B.M., Popescu, N. (2021). Machine Learning Techniques for Medical Image Processing. In 2021 International Conference on e-Health and Bioengineering (EHB), Iasi, Romania, pp. 1-4. https://doi.org/10.1109/EHB52898.2021.9657673

[24] Kotsiantis, S.B. (2013). Decision trees: a recent overview. Artificial Intelligence Review, 39: 261-283. https://doi.org/10.1007/s10462-011-9272-4

[25] Ho, T.K. (1998). The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8): 832-844. https://doi.org/10.1109/34.709601

[26] Nguyen, H.H., Tran, D.N.N., Jeon, J.W. (2020, November). Towards real-time vehicle detection on edge devices with Nvidia Jetson TX2. In 2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Korea (South), pp. 1-4. https://doi.org/10.1109/ICCE-Asia49877.2020.9277463

[27] Vijitkunsawat, W., Chantngarm, P. (2020). comparison of machine learning algorithm's on self-driving car navigation using Nvidia Jetson Nano. In 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, Thailand, pp. 201-204. https://doi.org/10.1109/ECTI-CON49241.2020.9158311

[28] Süzen, A.A., Duman, B., Şen, B. (2020). Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-CNN. In 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), pp. 1-5. https://doi.org/10.1109/HORA49412.2020.9152915

[29] Buyrukoğlu, S., Akbaş, A. (2022). Machine Learning based early prediction of type 2 diabetes: A new hybrid feature selection approach using correlation matrix with heatmap and SFS. Balkan Journal of Electrical and Computer Engineering, 10(2): 110-117. https://doi.org/10.17694/bajece.973129

[30] Belete, D.M., Huchaiah, M.D. (2021). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. International Journal of Computers and Applications, 44(9): 875-886. https://doi.org/10.1080/1206212X.2021.1974663

[31] Hutter, F., Kotthoff, L., Vanschoren, J. (2019). Automated machine learning: Methods, systems, challenges (p. 219). Springer Nature. https://doi.org/10.1007/978-3-030-05318-5

[32] Bergmeir, C., Benítez, J.M. (2012). On the use of cross-validation for time series predictor evaluation. Information Sciences, 191: 192-213. https://doi.org/10.1016/j.ins.2011.12.028

[33] Arlot, S., Celisse, A. (2010). A survey of cross-validation procedures for model selection. Statistics Surveys, 4: 40-79. https://doi.org/10.1214/09-SS054

[34] Ulutaş, H., Aslantaş, V. (2023). Design of efficient methods for the detection of tomato leaf disease utilizing proposed ensemble CNN model. Electronics, 12(4): 827. https://doi.org/10.3390/electronics12040827

[35] Sahin, M.E. (2022). Deep learning-based approach for detecting COVID-19 in chest X-rays. Biomedical Signal Processing and Control, 78: 103977. https://doi.org/10.1016/j.bspc.2022.103977

[36] Sahin, M.E. (2023). Image processing and machine learning-based bone fracture detection and classification using X-ray images. International Journal of Imaging Systems and Technology, 33(3): 853-865. https://doi.org/10.1002/ima.22849

[37] Gwak, J., Hirao, A., Shino, M. (2020). An investigation of early detection of driver drowsiness using ensemble machine learning based on hybrid sensing. Applied Sciences, 10(8): 2890. https://doi.org/10.3390/app10082890