IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 May 2022, accepted 13 June 2022, date of publication 22 June 2022, date of current version 3 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3185251

Fatigue State Detection for Tired Persons in Presence of Driving Periods

RIAD ALHARBEY[1], MOHAMED M. DESSOUKY[2,3], AHMED SEDIK[4], ALI I. SIAM[5], AND MOHAMED A. ELASKILY[6]

[1]Department of Information System and Technology, College of Computer Science and Engineering, University of Jeddah, Jeddah 22252, Saudi Arabia
[2]Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah 22252, Saudi Arabia
[3]Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf 32511, Egypt
[4]Department of the Robotics and Intelligent Machines, Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh 33516, Egypt
[5]Department of Embedded Network Systems Technology, Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh 33516, Egypt
[6]Department of Informatics, Electronics Research Institute (ERI), Cairo 71515, Egypt

Corresponding author: Mohamed A. Elaskily (mohamed_elaskily@eri.sci.eg)

This work was supported by the University of Jeddah, Saudi Arabia, under Grant UJ-21-DR-69.

ABSTRACT Due to the increasing of traffic accidents, there is an urgent need to control and reduce driving mistakes. Driver fatigue or drowsiness is one of these major mistakes. Many algorithms have been developed to address this issue by detecting fatigue and alerting the driver to this dangerous condition. The major problem of the developed algorithms is their detection accuracy, as well as the time required to detect fatigue status and alert the driver. The accuracy and the time represent a critical condition that affects the reduction of traffic accidents. Several datasets have been used in the development of fatigue or drowsy detection techniques. These data are gathered from the deriver's brain Electroencephalogram (EEG) signals or from video streaming recordings of the driver behavior. This paper develops two distinct approaches, the first based on the use of machine learning classifiers and the second depends on the use of deep learning models to produce a high-performance fatigue detection system. The machine learning approach is used to process EEG signals, whereas the deep learning approach is used to process video streams. In machine learning classifiers, Support Vector Machine (SVM) provides up to 98% of detection accuracy, which is the highest accuracy among the other five deployed classifiers. In deep learning models, Convolutional Neural Network (CNN) provides up to 99% detection accuracy, which is the highest accuracy among the other two deployed models. The experimental results demonstrate that the two proposed algorithms provide the highest detection accuracy with the shortest Testing Time (*TT*) when compared to all other recent and efficient fatigue detection algorithms.

INDEX TERMS EEG signals, fatigue detection systems, video streaming, support vector machine, convolutional neural network, testing time.

I. INTRODUCTION

Many projects are currently underway in automobile manufacturing companies to address the issue of driver fatigue by developing a fatigue detection system. According to Internet of Things (IOT) components and applications such as sensors, cloud, servers, smartphones, centralized and decentralized data processing, and so on [1], this idea is promising. Three major techniques are being used to create a robust and effective fatigue detection system. These techniques are classified as behavioral-based, vehicle-based, and physical-based techniques [2]. Figure 1 summarizes the details of the three major techniques of fatigue detection systems.

First, behavioral-based techniques analyze images and videos captured from the driver using image processing and computer vision methods. This technique is based on the analysis of some vital parameters that appear on the driver to ensure that the driver is in a state of attention, fatigue or drowsiness. These vital parameters are extracted based on monitoring features such as eye blindness, yawning by opening and closing the mouth, eye closure, facial features, and head position and nodding [3].

Second, vehicle-based techniques employ devices and sensors built into vehicle wheels to create an embedding

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

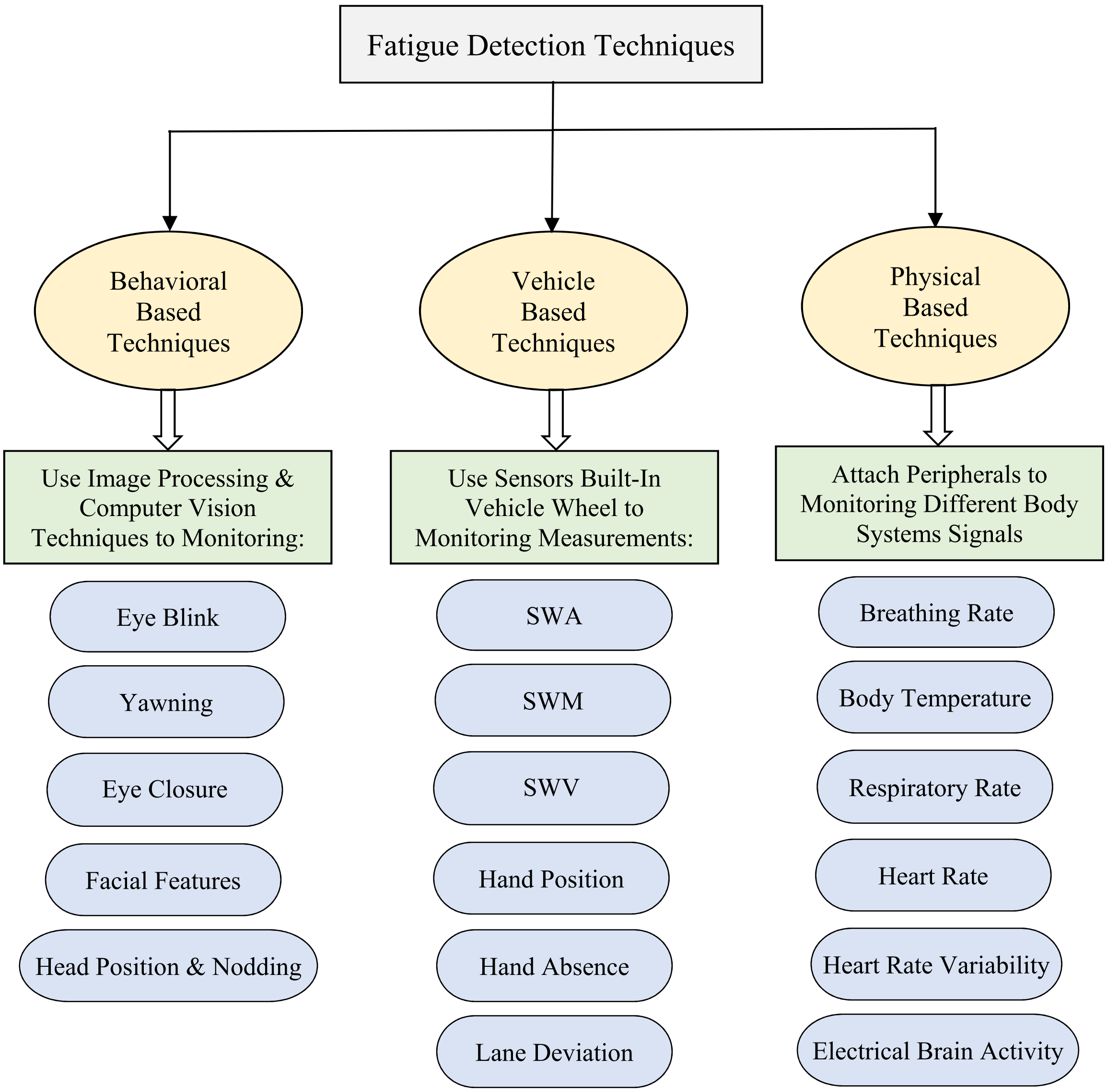


FIGURE 1. The three major techniques of fatigue detection system.

system for detecting driver fatigue. This embedded system detects driver behavior by monitoring measurements such as Steering Wheel Angle (SWA), Steering Wheel Movements (SWM), Steering Wheel Velocity (SWV), hand position, hand absence, and lane deviation [4].

Finally, the physical-based techniques use peripherals that attached to the driver's hands, head, fingers, and chest to monitor various types of body system signals. Electroencephalogram (EEG), Electrocardiogram (ECG), Electrooculogram (EOG), and Percentage of eyelid closure (PERCLOS) are the different types of attached devices. The output signals such as breathing rate, body temperature, respiratory rate, electrical brain activity, pulse rate, heart rate variability, and general heart rate are used to detect the driver's status [5].

Attempts to build a fatigue detection system, on the other hand, are divided into traditional-based algorithms and machine learning based algorithms [6]. Support Vector Machine (SVM) and Convolutional Neural Network (CNN) are the most effective and usable classifiers in machine learning algorithms [7]. SVM provides a high precision value in addition to its speed but in the small datasets but it suffers from lower speed and precision value in large datasets. CNN provides the highest precision value as well as stability in both large and small datasets, but it provides slow training with high processing cost [7].

The paper is organized as follows. Section 2 reviews the related work of fatigue and drowsiness detection algorithms. The full details and discussions of the proposed fatigue detection and prediction algorithm illustrate in section 3. Section 4 displays and discusses the experimental results obtained by the proposed algorithm with different datasets, as well as results compared to other fatigue detection algorithms. The conclusion of the paper and the future work is presented in section 5. The paper ended with acknowledge and references.

II. RELATED WORK

Developing a fatigue detection system is an ambitious project, which aims to establish driving safety rules. The algorithms in this area is divided into two main trends, image and video based techniques and signal processing techniques as highlighted in the following:

A. IMAGE AND VIDEO BASED TECHNIQUES

The earlier attempt for the behavioral-based techniques is performed by evaluating a real-time image acquisition for the driver using IR illumination, followed by monitoring driver behavior using software [8]. This proposal employs several parameters as metrics to monitor the driver behavior. These parameters are PERCLOS, blink frequency, face position, nodding frequency, and eye closure duration. A fuzzy classifier to detect the emergency status of the driver evaluates these metrics. This variety of monitoring and analyzing parameters combined with the day and night acquisition conditions, resulted in the system outperforming other algorithms at the time. Flores *et al.* [9] proposed an ADAS (Advanced Driver Assistance System) that detects and tracks the driver's face and eyes before analyzing the driver's facial emotions and eyes movement to detect drowsiness. The system had been tested in real time under different lighting conditions. Abtahi *et al.* [10] developed a straightforward image processing-based technique. Their proposal is based on detecting some signs of fatigue in the driver. These signs can be detected from: monitoring the driver face in the image and then tracking face details such as eye and mouth movements to detect yawning and eye languor. This proposal makes use of variations in the geometric features of the driver's face. Other algorithms, such as [11]–[13], attempt to improve fatigue detection performance by performing the same facial geometrical features detections as illustrated in [10].

Sigari et. al [14] developed a technique that compares real-time driver's head position and prepared face template. In addition to this matching process, the top half of the driver's face image had been horizontally projected to track eye closure and eyelid distance change. The algorithm had been automatically initialized using a fuzzy expert system that combines the two previous parameters. This algorithm's performance had been generally accepted but it suffers during the day and fails when the driver wears glasses. In [15], a deep neural technique had been proposed as one of the earliest attempts to use deep learning techniques to solve the drowsiness detection problem. This method extracts facial features from the driver's RGB video. This method combined three CNN models, Visual Geometry Group16 (VGG16), InceptionV3, and Residual Network (ResNet50) to create feature fused architecture (FFA). The main disadvantages of this technique is its low accuracy of 78%. Galarza et. al [16] developed an Android-powered smartphone to develop an interaction system that detected the driver's drowsiness. This system also employed behavior metrics such as eyes position, head position, and yawning frequency. The benefits of this system include consistent results under different conditions such as lighting and driver accessories such as glasses, cap, or hearing aids. The system had a high drowsiness detection accuracy of 93.37%.

Bassi et. al [17] created an efficient fatigue detection system using machine learning techniques such as local binary pattern, SVM, and Principal Component Analysis (PCA). The proposed system compares linear kernel, polynomial kernel, and quadratic kernel for SVM and finds the most efficient one.

The accuracy of that model varies, but the highest accuracy value is achieved by using polynomial kernel of SVM to achieve approximately 99% accuracy. This method provides good performance, but it suffers from a high computational cost and a long testing time. Ouabida et. al [18] proposed an optical correlator detection system for driver's eyes. The authors used Vander Lugt Correlator (VLC) to estimate the eye center in addition to filter the eye scene in the noisy and abnormal environment. This method provides good performance through eye and closure tracking achieving 95% accuracy. The main problem of that method is its vulnerability to light reflections from other cars or street lights. Maior et. al [19] to proposed another drowsiness detection algorithm based on eye patterns extracted from a video stream. This proposal employed machine learning and computer vision algorithms to monitor eye behavior and created a history of the driver's blink motion. This method used SVM, random forest and multi-layer perceptron. This algorithm's best accuracy is about 94%, but it takes long execution time. Saurav et. al [20] proposed another algorithm based also on video streaming monitoring of yawning behavior. This method combined different deep learning models, including Bi-directional Long Short-Term Memory (Bi-LSTM) and one-dimensional CNN. The mouth region features had been extracted based on video steam. A yawning emotion had been detected and classified as a normal motion or a fatigue sign using the two previous combined models. This algorithm achieved approximately 96% accuracy with a medium computational cost when tested on the datasets YawDD [21] and NTHUDDD [22].

Biswal et. al [23] proposed an intelligent vehicle fatigue detection alarm system. This system is based on capturing video streams and analyzing blink behavior. This analysis is performed by two parameters: Euclidean distance between the eye and the face, as well as Eye Aspect Ratio (EAR). Another privilege provided by this algorithm is the ability to embed Internet of Things (IOT) modules to warm about road collisions. Jeon et. al [24] proposed a system which combined both vehicle-based and behavioral-base methods. The method is based on collecting signals from the steering wheel and pedal pressure, then processing these signals with CNN. This method achieved performance of 94%. The main issue with this method is the inconsistency of its accuracy as the road environment changes.

B. SIGNAL PROCESSING BASED TECHNIQUES

Some algorithms, on the other hand, developed the physical methods based on signals such as EEG, ECG, and EOG by combining machine and deep learning models. These methods function as a hybrid of physical-based and behavioral-based methods. Ko et. al [25] proposed an algorithm that used EEG samples to extract Differential Entropy (DE), followed by a classification process using CNN. A hierarchical feature and class-discriminative are extracted as a result of the classification process, and a density-connected layer is used for drowsiness decision. Zhu et. al [26] proposed an algorithm that collected signals from wearable EEG devices and process them using CNN. After collecting EEG signals through brain computer interface, the AlexNet module is deployed with CNN to classify these signals. This algorithm has a 94% accuracy rate. The main problem is the time lag between collecting EEG signals and processing them through CNN.

III. THE PROPOSED ALGORITHM

Because of the importance of traffic safety and preservation of road users' lives, this paper proposed a new automatic drossiness detection system. There are numerous techniques used in other researches related to drive fatigue detection process. The proposed algorithm employs a variety of techniques to achieve the highest performance in the shortest amount of time and with the least amount of processing complexity. To accomplish this goal, the proposed algorithm employs an Artificial Intelligence (AI) approach that includes both machine learning and deep learning models. Actually, this proposed method is divided into two approaches the machine learning approach and deep learning approach. In machine learning approach, the proposed algorithm deployed models such as SVM, Random Forest (RF), Naive Bayes (NB), Logistic Regression (LR), Decision tree (DT), Multilayer Perceptron (MLP), K-Nearest Neighbor (KNN) and Quadratic discriminant analysis (QDA) models. Furthermore, in deep learning approach, several models are deployed including CNN, Convolutional Long Short-Term Memory (ConvLSTM) and hybrid models that combine CNN and ConvLSTM. The goal of using all of these models is to determine which techniques or hybrid of them achieve the best performance, as well as to conduct a detailed analysis of the results of different techniques. The overall structure of the two different approaches are displayed in figure 2.

A. MACHINE LEARNING APPROACH

In machine learning approach, the proposed algorithm uses the EEG signals as a data input to perform the drowsiness detection process. First, the EEG signals are transformed using Discrete Wavelet Transform (DWT) to select the coefficients of high energy signals and discard unwanted signals. Second, discriminant features extracted from the EEG signals are presented. During the pre-processing step, PCA is used to select the important features. In order to facilitate the classification task, a standard scaling is performed in a pre-processing step to examine the differences between features. The extracted and pre-processed features are fed into classifiers, which detect drossiness in the given signal. For the classification step, DT, KNN, a multiclass SVM [27], Gaussian NB, MLP with backpropagation, QDA, RF, and LR classifiers are used. The grid search technique [28] automates the trial-and-error process in order to identify the optimal structure and hyper-parameters of classifiers. Table 1 displays the resulted appropriate hyper-parameters of classifiers used in the current study.

B. DEEP LEARNING APPROACH

Unlike the previous approach, deep learning approach handles the driver's visual expressions rather than collecting physical signals. In deep learning approach, the proposed algorithm uses video streaming produced by monitoring the driver through a camera during the driving process as a data input to perform the drowsiness detection process. First, in the video segmentation step, the video streaming for the driver is divided into frames. These segments are ready to be fed into one of the three models described in this section. In deep learning approach, three deep learning models are proposed to achieve the highest performance drowsiness detection algorithm.

The first model is based on CNN, which consists of one input layer, four convolutional layers, four pooling layers followed by one global average pooling layer, and one dense layer. The input layer is considered to batch the input frame. The convolutional layers are designed with input image size 224×224. The number of filters are implemented as 16, 32, 64, and 128 respectively for the 4 convolutional layers. A fully connected network is also performed using a global average pooling layer. A dense layer with a soft-max activation function performs the classification process. Figure 3 displays the processing steps with different input size and filters number for each layer structured of this model. In addition to number of parameters produced in each convolutional layer to tune the results of that model. The second model is based on ConvLSTM, which is implemented as a 2D recurrent network. This model consists of a one input layer considered to batch the input frame, one ConvLSTM layer with 3×3 kernel size and employ 32 filters followed by one global average pooling layer, and one dense layer. This model is used as a benchmark to demonstrate the performance of the proposed deep learning model. Figure 4 presents the processing steps for input frame with the filter size structured in addition to number of parameters tuned in the ConvLSTM approach to reach best results for that model.

Finally, the third model is a hybrid model that employs both CNN and ConvLSTM modalities. This model consists of one ConvLSTM layer with input frame size 224×224 as height and width respectively and followed by three channels. On the other hand, the size of the feature map in the sequence of layers is represented by (height, width, number of filters) such as (111, 111, 16) which refers a height of 111, width of 111, and number of filters of 16. After ConvLSTM is followed by three convolutional layers with frame size 222×222 after normalization process to reach frame size 52×52 for the third convolutional layer. Three max pooling layers is following the three convolutional layers. Finally, one global average

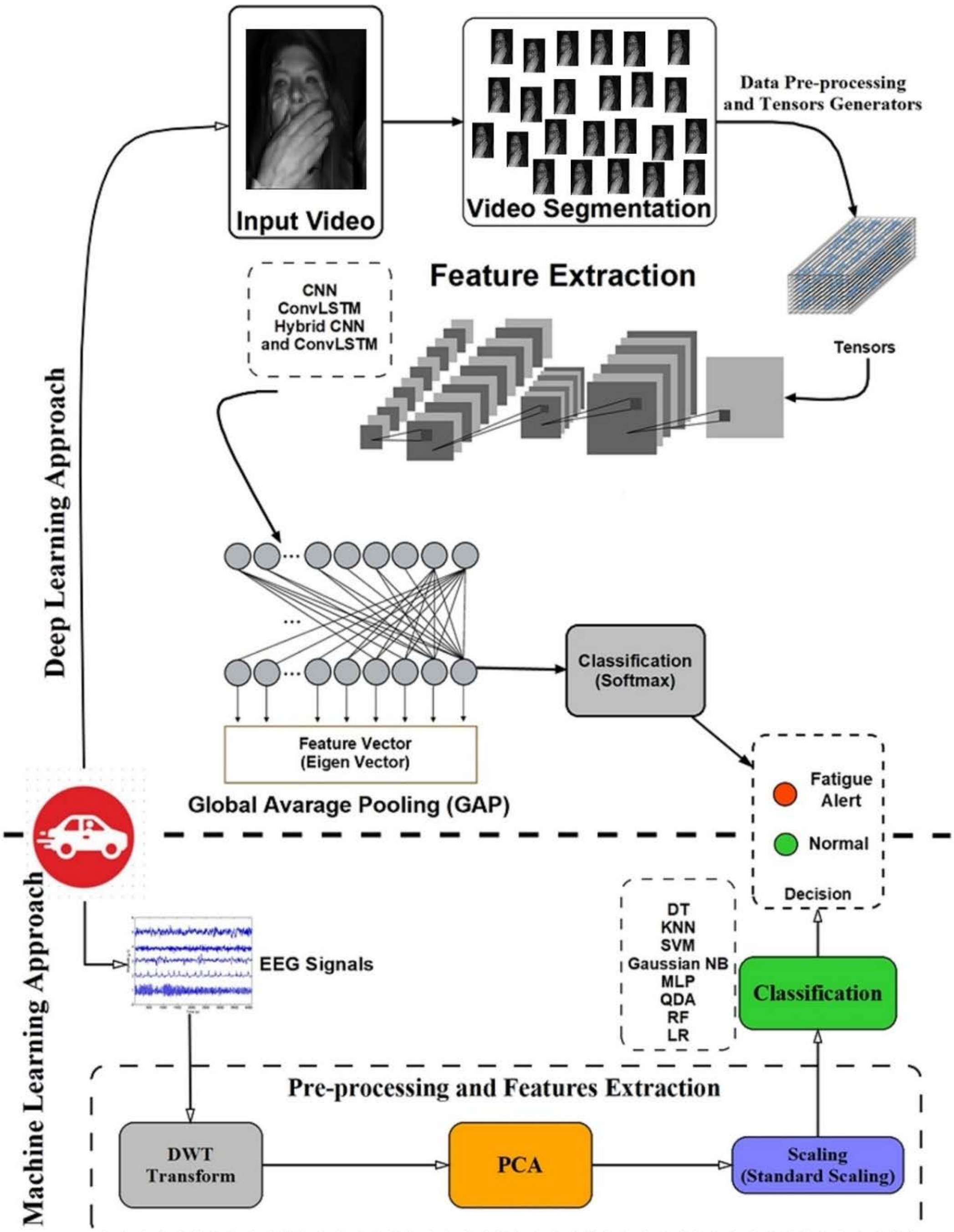


FIGURE 2. The overall structure of machine learning and deep learning approaches.

pooling layer and one dense layer is duilt. The processing steps with different input size and filters number for each layer structured of the hybrid approach (ConvLSTM and CNN) is displayed in figure 5. In addition to number of parameters produced in each layer to tune the results of that model

Generally, figures 3, 4, and 5 display the structure of each model. Each figure displays three important data. The first column, displays the type and number of layers in each model in addition to the sequence of these layers. The second column, displays the input size for each layer where each output of a layer acts as input for the next layer in his structure. The third column, displays the number of parameters or features used to tune the model in different iterations to reach the best available results for each model.

There are criteria of video segmentation by takes 51 frames for each subject or scene as shown in [29]. These 51 frames is representing the number of frames per second. In the three proposed models, all of the 51 frames are used as input images into the input layer where each frame represents as input

TABLE 1. List of hyper-parameters of adopted classifiers.

Classifier	Hyper-parameters	Classifier	Hyper-parameters
DT	criterion: 'gini' min_samples_leaf: 1 min_samples_split: 2 ccp_alpha: 0	Gaussian NB	var_smoothing: 1e-09
		MLP	Num_hidden_layers:2 hidden_layer_sizes:[28,28] activation: 'relu' max_iter: 200 solver: 'adam'
KNN	n_neighbors:1 leaf_size: 30 metric: 'minkowski' p: 2 weights': 'uniform'	QDA	tol=0.0001
		RF	n_estimators:79 Criterion: 'entropy'
SVM	C: 275 gamma: 'scale' kernel: 'rbf'	LR	solver: 'lbfgs' C: 1.0 fit_intercept: True

```
Layer (type)                 Output Shape              Param #
=================================================================
batch_normalization (BatchNo (None, 224, 224, 3)       12
conv2d (Conv2D)              (None, 222, 222, 16)      448
max_pooling2d (MaxPooling2D) (None, 111, 111, 16)      0
batch_normalization_1 (Batch (None, 111, 111, 16)      64
conv2d_1 (Conv2D)            (None, 109, 109, 32)      4640
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 32)        0
batch_normalization_2 (Batch (None, 54, 54, 32)        128
conv2d_2 (Conv2D)            (None, 52, 52, 64)        18496
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 64)        0
batch_normalization_3 (Batch (None, 26, 26, 64)        256
conv2d_3 (Conv2D)            (None, 24, 24, 128)       73856
max_pooling2d_3 (MaxPooling2 (None, 12, 12, 128)       0
batch_normalization_4 (Batch (None, 12, 12, 128)       512
global_average_pooling2d (Gl (None, 128)               0
dense (Dense)                (None, 2)                 258
=================================================================
Total params: 98,670
Trainable params: 98,184
Non-trainable params: 486
```

FIGURE 3. The processing structure of CNN approach.

```
Layer (type)                 Output Shape              Param #
=================================================================
batch_normalization_11 (Batc (None, 224, 224, 3)       12
reshapeconvtolstm (Reshape)  (None, 1, 224, 224, 3)    0
conv_lst_m2d_2 (ConvLSTM2D)  (None, 1, 224, 224, 32)   40448
reshapelstmtoconv (Reshape)  (None, 224, 224, 32)      0
global_average_pooling2d_3 ( (None, 32)                0
dense_3 (Dense)              (None, 2)                 66
=================================================================
Total params: 40,526
Trainable params: 40,520
Non-trainable params: 6
```

FIGURE 4. The processing structure of ConvLSTM approach.

```
Layer (type)                 Output Shape              Param #
=================================================================
batch_normalization_5 (Batch (None, 224, 224, 3)       12
reshapeconvtolstm (Reshape)  (None, 1, 224, 224, 3)    0
conv_lst_m2d (ConvLSTM2D)    (None, 1, 224, 224, 32)   40448
reshapelstmtoconv (Reshape)  (None, 224, 224, 32)      0
conv2d_4 (Conv2D)            (None, 222, 222, 64)      18496
max_pooling2d_4 (MaxPooling2 (None, 111, 111, 64)      0
batch_normalization_6 (Batch (None, 111, 111, 64)      256
conv2d_5 (Conv2D)            (None, 109, 109, 128)     73856
max_pooling2d_5 (MaxPooling2 (None, 54, 54, 128)       0
batch_normalization_7 (Batch (None, 54, 54, 128)       512
conv2d_6 (Conv2D)            (None, 52, 52, 128)       147584
max_pooling2d_6 (MaxPooling2 (None, 26, 26, 128)       0
batch_normalization_8 (Batch (None, 26, 26, 128)       512
global_average_pooling2d_1 ( (None, 128)               0
dense_1 (Dense)              (None, 2)                 258
=================================================================
Total params: 281,934
Trainable params: 281,288
Non-trainable params: 646
```

FIGURE 5. The processing structure of hybrid (ConvLSTM and CNN) approach.

image. For each model, different layers are processing over each input image to extract this image features. All features extracted from processing all input images are used to build a feature map. This map is used for comparing with features extracted from tested images or frames to decide if this scene or subject is a fatigue status or a normal status.

IV. EQUIPMENTS AND DATASETS

A. EQUIPMENTS

The proposed approaches are implemented on a PC with Intel core i7 CPU, a 4 GB CUDA GPU driver, an 8 GB RAM size, and Windows 10 operating system. The proposed approaches are implemented in python 3.5 and use Keras and TensorFlow toolkits. In addition, the details of the proposed approaches layers and learning parameters are listed in table 1 as well as figures 3,4, and 5.

B. DATASETS

This paper is used a well-tested dataset that is divided into two consists of two major data categories. This dataset is the "ULg Multimodality Drowsiness Database", also known as DROZY [29]. This dataset is divided into two parts. The first part is to collect data from video streaming monitoring for 14 healthy young persons, three males and eleven females. The data in this section is collected using Kinect technology and video sensors with Near-Infrared (NIR) intensity, which produced 512×424 pixels in MP4 format. Figure 6 presents examples of the produced NIR intensity scenes generated from video frames are.

The second part of the DROZY dataset is collected from drowsiness measurements such as Karolinska Sleepiness Scale (KSS), and Polysomnography (PSG) signals. Polysomnography signals include five EEG channels, two EOG channels, Electromyography (EMG) channels, and

ECG channels, all at 512 Hz. Figure 7 shows the distribution of polysomnography signals through the driver's head. In the future, the driver can wear new equipment called NeuroSky. NeuroSky is EEG and ECG biosensors, which are breaking the boundaries of body and mind monitoring and analysis for consumer-facing, wearable technology products.

FIGURE 6. Examples of NIR intensity scenes produced from drozy video frames.

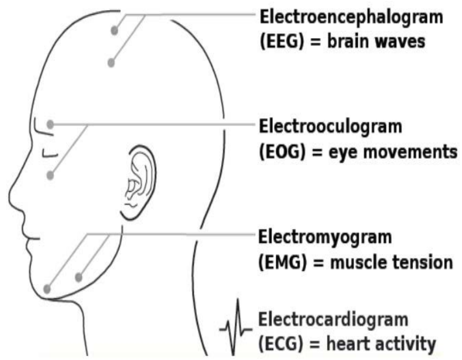


FIGURE 7. The physical distribution of drozy polysomnography signals on the driver's head.

V. EXPERIMENTAL RESULTS

This section contains a comprehensive evaluation of the proposed approaches. First, the dataset is described with its full details. Second, the evaluation metrics used to evaluate the approaches results are displayed. Next, the results as well as the discussions and comments on those results are listed. Finally, well-test comparisons are performed.

A. EVALUATION METRICS

Various evaluation metrics are used to assess proposed approaches. Accuracy, Sensitivity (equal to True Positive Rate (*TPR*)), False Positive Rate (*FPR*), False Negative Rate (*FNR*), False Discovery Rate (*FDR*), Specificity that is equal to True Negative Rate (*TNR*), Precision, F1 score, and Matthews Correlation Coefficient (*MCC*). Equations (1), (2), (3), (4), (5), (6), (7), (8), and (9), as shown at the bottom of the page, define all of these metrics.

Where:

1) False Negative (F_N) represents the number of drowsy statuses which had been incorrectly classified as a normal status.
2) True Positive (T_P) represents the quantity of drowsy statuses which had been correctly classified as a drowsy status.
3) True Negative (T_N) represents the quantity of normal statuses which had been correctly classified as a normal status.
4) False Positive (F_P) represents the quantity of normal statuses which had been incorrectly classified as a drowsy status.

On the other hand, another important evaluation metric is involved in the proposed approaches evaluation. This parameter is Testing Time (*TT*), and it represents the average time that each approach takes to test the driver's status over the number of (K) rounds of testing. Furthermore, the learning time (training time) is not taken into account because this time is spent offline for each approach only once to build the classification behavior.

B. RESULTS AND DISCUSSION

1) RESULTS

First, a k-fold cross validation process is used to test the performances of the proposed approaches on the DROZY

$$
\begin{aligned}
Accuracy &= \frac{No.\ of\ correctly\ detected\ images}{Total\ No.\ of\ images} \times 100 \\
&= \frac{(T_N + T_P)}{(T_P + F_P + T_N + F_N)} \times 100 && (1)\\
Sensitivity &= TPR = T_P/(T_P + F_N) = (1 - FNR) && (2)\\
FPR &= F_P/(F_P + T_N) = (1 - TNR) && (3)\\
FNR &= F_N/(T_P + F_N) && (4)\\
FDR &= F_P/(F_P + T_P) && (5)\\
Specificity &= TNR = T_N/(T_N + F_P) && (6)\\
precision &= T_P/(T_P + F_P) && (7)\\
F1 &= 2 * ((precision * recall)/(precision + recall)) && (8)\\
MCC &= \frac{(T_P \times T_N) - (F_P \times F_N)}{\sqrt{((T_P + F_P) \times (T_P + F_N) \times (T_N + F_P) \times (T_N + F_N))}} \times 100 && (9)
\end{aligned}
$$

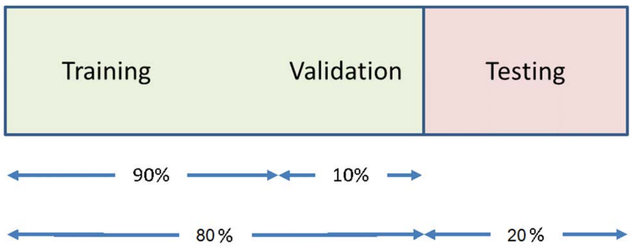


FIGURE 8. Pipeline for fragmentation of dataset.

dataset. This k-fold process helps in estimating the optimal hyper-parameters combinations to avoid overfitting. The proposed approaches used 10-fold cross-validation. This paper proposed two main approaches; machine learning approach and deep learning approach.

a: MACHINE LEARNING APPROACH

Several classifiers (SVM, RF, LR, MLP, KNN, QDA) are used in the machine learning approach to achieve the best performance. The EEG signals in the DROZY dataset are divided into train and test signals, which are separated from each other. The training component is used to train and validate the classifier. The performed technique in the data splitting process is based on two main segmentations. Firstly, the data is automatically and randomly shuffled into 80 and 20 % for training and testing, respectively. On the other hand, the training data is proceeded using k-fold cross validation technique with k of 10. Therefore, the data is training, validated with 80 % of the whole data, and tested with a blind data of 20 % to ensure the reliability of the proposed methods. The dataset had been divided into 80/20 as shown in figure 8.

The classifier had been trained using nine folds of cross-validation and the learning had been validated using the tenth fold. This method is repeated until each of the ten folds has been validated exactly once, at which point the process completed. It is now the time to put those configurations to the test.

Table 1 shows the hyper-parameters of each machine learning classifier used in this study. Table 2 displays the results of each classifier under different evaluation metrics. In addition, confusion matrices for all classifiers are shown in figure 9. The x-axis presents the predicted classes generated by the proposed machine learning algorithm, while the y-axis displays the true values. The confusion matrix is essential for samples representation.

Figure 10 shows the learning curve for each classifier. The learning curve, which is commonly used in machine learning for algorithms that learn incrementally over time such as deep learning neural networks, had been also used. The learning curve is a good representation of the actual difference between learning process behavior and the actual performance of each classifier that shown in the testing process. It compares the model's performance on training and testing data over a variable number of training instances. Learning curve presents when a model has learning as much as it can about the data. The learning curve can be used to determine whether the model is underfitting or overfitting. For the training curve if the cost is high and doesn't decrease with the number of iterations, it's underfitting. The learning curve detects overfitting when the loss decreases until a turning point is found, and then begins to rise again, this point represents the start of overfitting. No proposed algorithms suffered from underfitting or overfitting, which is a significant advantage of these algorithms.

Finally, the Receiver Operating Characteristic (ROC) curves of each classifier had been plotted. It is a graphical representation of a binary classification evaluation metric. On a ROC curve, the TPR (sensitivity) is displayed as a function of the FPR (specificity) at different cutoff points. Each point on the ROC curve represents a sensitivity/specificity pair that associated with a specific decision threshold. A successful discriminating test (no crossing between the two distributions) indicates that the ROC curve (blue curve) exceeds in the upper left corner (100 % sensitivity, 100 % specificity). As a result, the closer the ROC curve is to the upper left corner, the better the overall accuracy of the test. Figure 11 shows the ROC curve of each classifier used in the machine learning approach.

b: DEEP LEARNING APPROACH

For the deep learning approach, three models, (ConvLSTM, CNN, and hybrid CNN ConvLSTM), had been employed to reach the best performance. The video streaming samples in the DROZY dataset are segmented into frames and these frames are subjected to image processing and computer vision analysis. In the deep learning approach, the three different models used to predict the best behavior of the data training, validating, and testing using the same strategy displayed before in figure 8. Table 3 shows the results of each model under different evaluation metrics. Figure 12 depicts the confusion matrices for all models. Figure 13 shows the learning curve for each model containing model accuracy and model loss behaviors. Figure 14 shows the ROC curve of each classifier used in the deep learning approach.

C. DISCUSSION

1) MACHINE LEARNING APPROACH

From table 2, it is shown that the accuracy of classifiers ranged from approximately 91% to 98 %. It is observed that the SVM classifier has a higher accuracy (98.01%) than the others, whereas the QDA classifier has a lower accuracy (91.32%). *TPR*, *FPR*, *FDR*, sensitivity, specificity, precision, *F1* score, and *MCC* are some of the evaluation metrics used, and their values are displayed in table 2 for all classifiers. The SVM gives higher performance than the other classifiers in the problem of multi-class classifications. It is based on the probability function to perform classification between two classes. Although the SVM classifier gives relatively high *TT* equal to 0.187 seconds compared to, for example, the LR

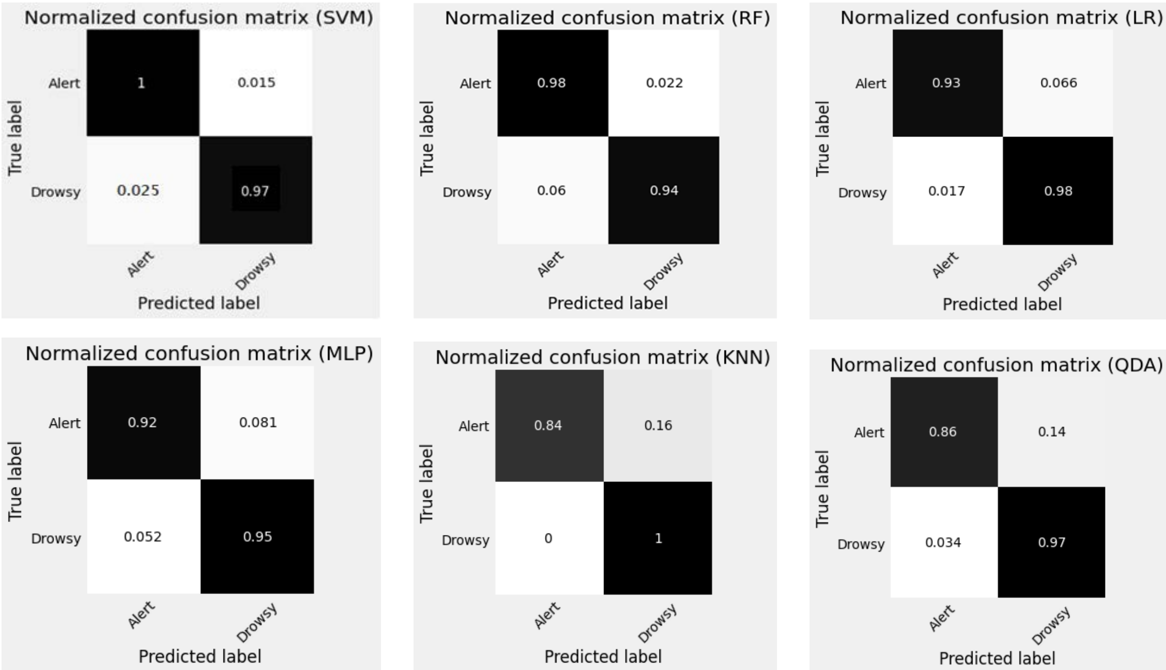


FIGURE 9. Confusion matrix for each classifier (SVM, RF, LR, MLP, KNN, and QDA) respectively.

TABLE 2. The results of machine learning approach with different classifier.

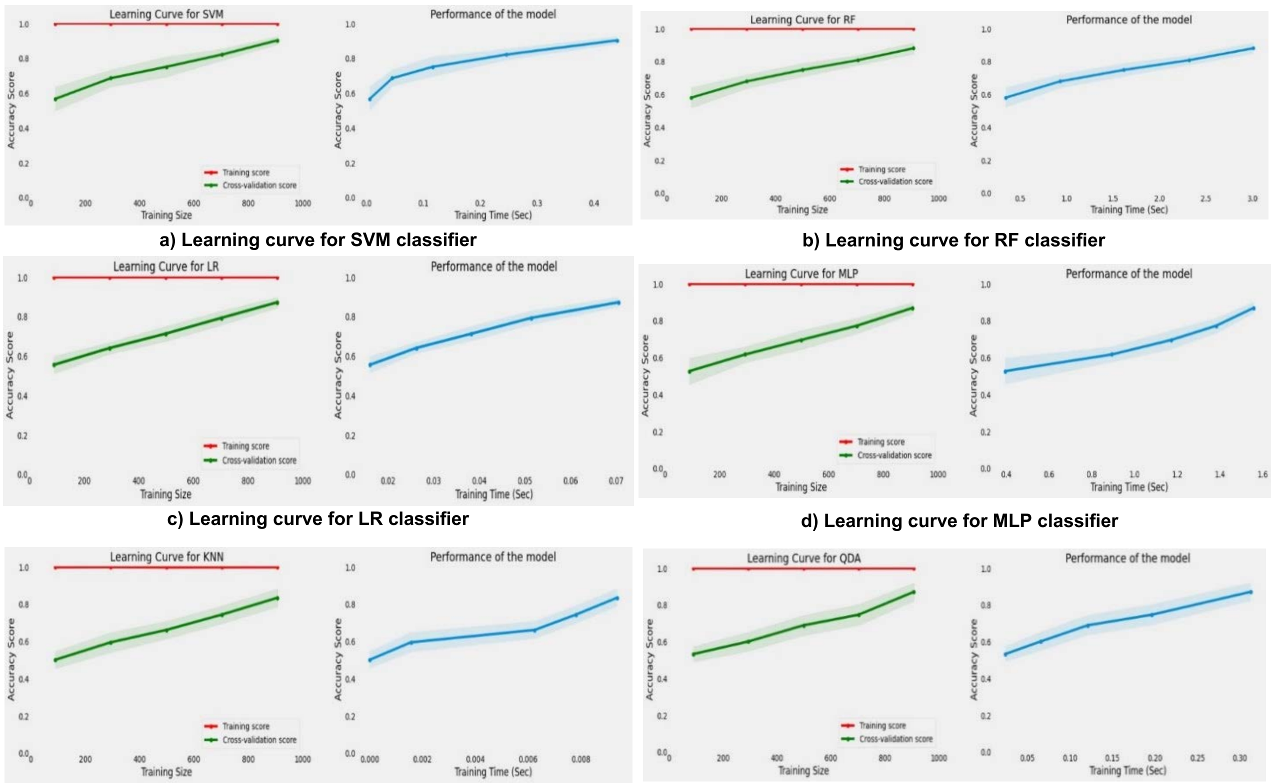
Metric / Model	*ACC. %*	*Sens. %*	*FPR %*	*FNR %*	*Spec. %*	*Prec. %*	*FDR %*	*F1 Score %*	*MCC %*	*TT (Seconds)*
SVM	98.01	97.49	1.48	2.51	98.52	98.48	1.52	97.98	96.02	0.187
RF	95.9	94	2.2	6	97.80	97.71	2.29	95.82	91.87	0.031
LR	95.84	98.29	6.63	1.71	93.37	93.69	6.31	95.94	91.78	0.002
MLP	93.36	94.81	8.09	5.19	91.91	92.14	7.86	93.46	86.76	0.016
KNN	92	100	16	0	84	86.21	13.79	92.59	85.1	0.04
QDA	91.32	96.61	14	3.39	86	87.39	12.61	91.77	83.10	0.031

classifier which gives only 0.002 seconds, but the difference is only fractional of seconds and could be negligible.

Figure 9 presents the convolution metrics of all classifiers. The x-axis presents the predicted classes by the machine learning algorithm, while the y-axis presents the true values. The confusion matrix is required for the representation of the samples. The samples that are truly predicted are presented by the two black boxes. The *TN* is shown in the upper left black box, and the *TP* is shown in the lower right black box. The samples that are incorrectly predicted are represented by the two white boxes. The *FP* is shown in the upper right white box, and the *FN* is shown in the lower left white box. The SVM classifier provides the highest *TN* (100%) and highest *TP* (97%) while having lowest *FP* (1.5%) and lowest *FN* (2.5%) compared to other algorithms.

Figure 10 shows the learning curve for each classifier. The red line presents the learning score of 100% that the classifier needs to reach. The green line presents the cross-validation score. Each classifier needs to reach to higher learning score with increasing number of iterations of training. The right blue curve represents the training time required by the algorithm with different number of training iterations. The training time is not an important metric in comparison because each algorithm is trained only once, and the difference between algorithms is fractional of a second and can be ignored. The SVM classifier achieved a 98% accuracy with 1000 training iterations with low training time equal to 0.4 seconds. The LR algorithm requires less training time (only 0.07 seconds), but it achieves an accuracy of 95.84% with the same training iterations (1000 iterations), which is lower than the SVM accuracy. Therefore, the SVM algorithm outperforms the other algorithms.

The ROC curve of each classifier used in the machine learning approach is presented in figure 11. The Larger the area under the ROC curve, the better the algorithm's performance. SVM and LR have larger area under the ROC curve than other algorithms, but the SVM algorithm has a higher area under the ROC curve equal to 98% compared to the 95.9% for the LR algorithm.



a) Learning curve for SVM classifier

b) Learning curve for RF classifier

c) Learning curve for LR classifier

d) Learning curve for MLP classifier

e) Learning curve for KNN classifier

f) Learning curve for QDA classifier

FIGURE 10. The learning curve for learning each classifier.

2) DEEP LEARNING APPROACH

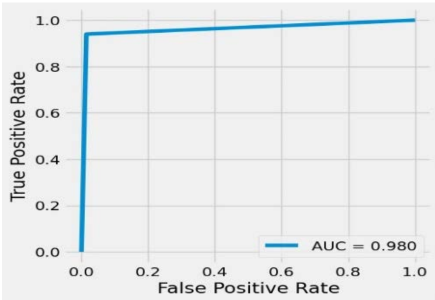
Table 3 summarizes the values of different metric parameters for three deep learning approaches (ConvLSTM, CNN, and hybrid CNN ConvLSTM). These three approaches had been applied on video streaming samples from the DROZY dataset. The accuracy ranged from approximately 70.5% for the ConvLSTM algorithm to 98.8% for the CNN algorithm. TPR, FPR, FDR, sensitivity, specificity, precision, F1 score, and MCC had been all measured as evaluation metrics parameters for all classifiers. The CNN algorithm outperforms the ConvLSTM algorithm, and it outperforms the (Hybrid CNN ConvLSTM) algorithm slightly better. Furthermore, the CNN algorithm has a lower TT of 10.61 seconds compared to 19.28 seconds for ConvLSTM algorithm and 26.60 seconds for (Hybrid CNN ConvLSTM) algorithm.

There are two different parameters are affecting on the testing time. Firstly, the size of the used dataset to evaluate an algorithm. Secondly, the number of frames taken for each second or scene. According to DROZY dataset, which consists of about 500,000 frames as mentioned in [29] and by comparing to other datasets, DROZY dataset is classified as a very large size dataset. The proposed algorithm takes 51 frames per second and these 51 frame are encapsulated as one case and tested one time to check the deriver fatigue status at this case. In the proposed algorithm, the testing time 10.61 seconds is the testing time that taken to reach final decision about 20% of the overall cases in dataset. For example, there are about 500,000 frame in DROZY dataset and the algorithm takes 51 frames per seconds so, there are a about 9804 (500,000 / 51) second. The testing part that represents 20% is equal to about 1960 cases in addition to 80% that represents about 7844 cases in training part. Therefore, the testing time equal to 10.61 seconds is for 1960 cases at the testing part not for only one case. The testing time for only one case at a time is average of $10.61/1960 = 0.005$ seconds, which represents as a very small value. Small video sizes reflects on small testing time. In addition, decreasing the number of frames taken per second is reflecting on decreasing of testing time but it effects on detection accuracy also.

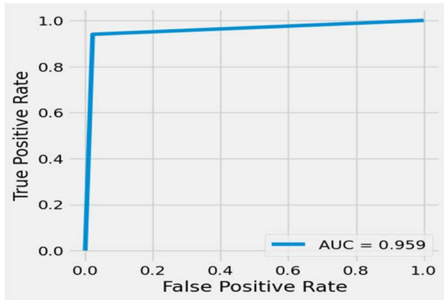
Figure 12 presents the convolution metrics of the three classifiers. The CNN algorithm has a higher *TP* and *TN* of one and 0.98, respectively, compared to one and 0.96 for the (Hybrid CNN ConvLSTM) algorithm and 0.8 and 0.61 for the ConvLSTM algorithm. The CNN algorithm also has a lower *FP* and *FN* with 0.024 and 0, respectively, compared to 0.04 and 0 for the (Hybrid CNN ConvLSTM) algorithm and 0.39 and 0.2 for the ConvLSTM algorithm.

Figure 13 shows the learning curve for each classifier. The CNN algorithm achieved near perfect (near 100%) accuracy with a smaller number of epochs equal to only 5 epochs, whereas the ConvLSTM algorithm achieved near-perfect (near 100%) accuracy with a larger number of epochs equal to 10 epochs. In addition, the CNN model has lower model loss when compared to the other algorithms.

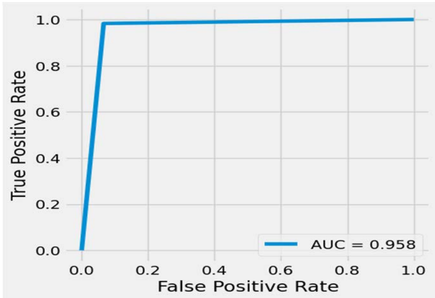
Figure 14 shows the ROC curves of all classifiers. The CNN algorithm has a larger area under the ROC curve (99%)



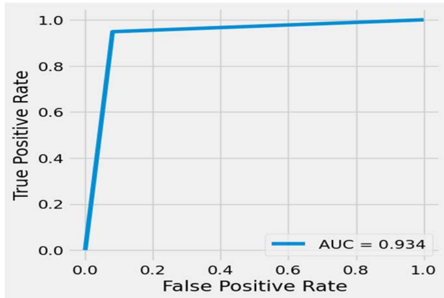
a) ROC curve for SVM classifier



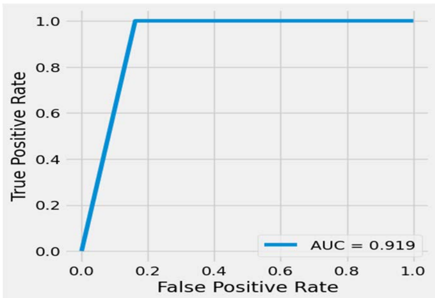
b) ROC curve for RF classifier



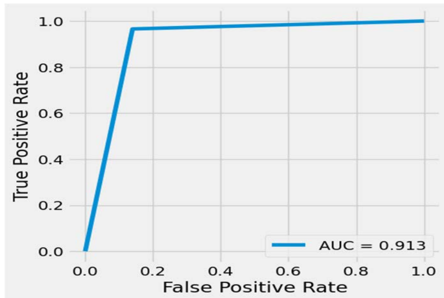
c) ROC curve for LF classifier



d) ROC curve for MLP classifier



e) ROC curve for KNN classifier



f) ROC curve for QDA classifier

FIGURE 11. The ROC curve for each classifier.

TABLE 3. The results of deep learning approach with different models.

Metric / Model	*ACC. %*	*Sens. %*	*FPR %*	*FNR %*	*Spec. %*	*Prec. %*	*FDR %*	*F1 Score %*	*MCC %*	*TT (Seconds)*
ConvLSTM	74.50	80	39	20	61	67.23	32.77	73.06	41.76	19.28
CNN	98.8	100	2.39	0	97.61	97.66	2.34	98.81	97.63	10.61
Hybrid CNN ConvLSTM	98	100	4	0	96	96.15	3.85	98.04	96.08	26.60

than the (Hybrid CNN ConvLSTM) algorithm, which has 98% and 71% for the ConvLSTM algorithm.

D. COMPARISIONS

Comparison study is presented in this section between the proposed approaches and other recent high-performance algorithms to validate their performance prior to fatigue issue. The lake of datasets unifying in other fatigue detection studies is one of the most difficult issues in this area. The objective of this study is to demonstrate the superiority of the investigated proposed approaches regardless of the dataset used in previous studies. One of the rules that supports this idea is the match between all human EEG signals and all human face expressions, which leads to the detection of fatigue status whether EEG signals or video streaming are used as a reference. The algorithms that involved in this comparison are listed as Maior et. al [19], Biswal et. al [23], Jeon et. al [24], Ko et. al [25], Zhu et. al [26], Zhang et. al [30], Gwak et. al [31], and Bakheet et. al [32] in addition to the proposed approaches. The comparison is divided into two parts, the video streaming-based algorithms and physiological signals (EEG) based algorithms. For the video streaming part, algorithms Maior et. al [19], Biswal et. al [23], Jeon et. al [24], Gwak et. al [31], and Bakheet et. al [32] have been compared with the proposed deep learning approach. For EEG signals part, algorithms Ko et. al [25],

TABLE 4. Comparison between the proposed machine learning approach and EEG based algorithms.

Algorithm / Metric	Proposed Machine Learning Approach	Ko et. al [25]	Zhu et. al [26]	Zhang et. al [30]	Gwak et. al [31]
ACC. %	98.01	96	94.68	90.70	89.80
Sens. = TPR %	97.49	----	95.32	90.20	89.50
FPR %	1.48	----	----	----	----
FNR %	2.5	----	----	----	----
Spec. = TNR %	98.52	----	----	----	----
Prec. %	98.48	97	95.57	----	88.70
F1 Score %	97.98	----	95.48	----	89.10
TT (Seconds)	0.187	----	1.00	2.20	----
Used Dataset	DROZY EEG signals	SEED-VIG [33]	Self-collected EEG signals [26]	Self-collected EEG signals [30]	Self-collected EEG signals [31]

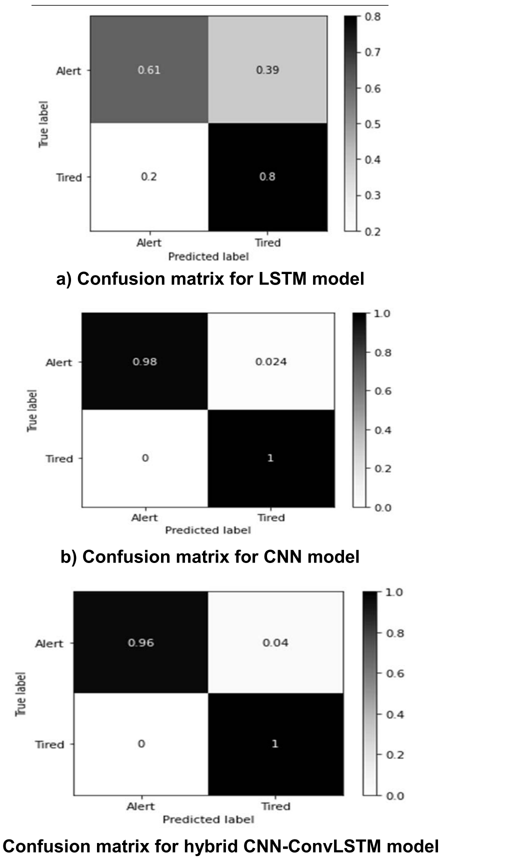
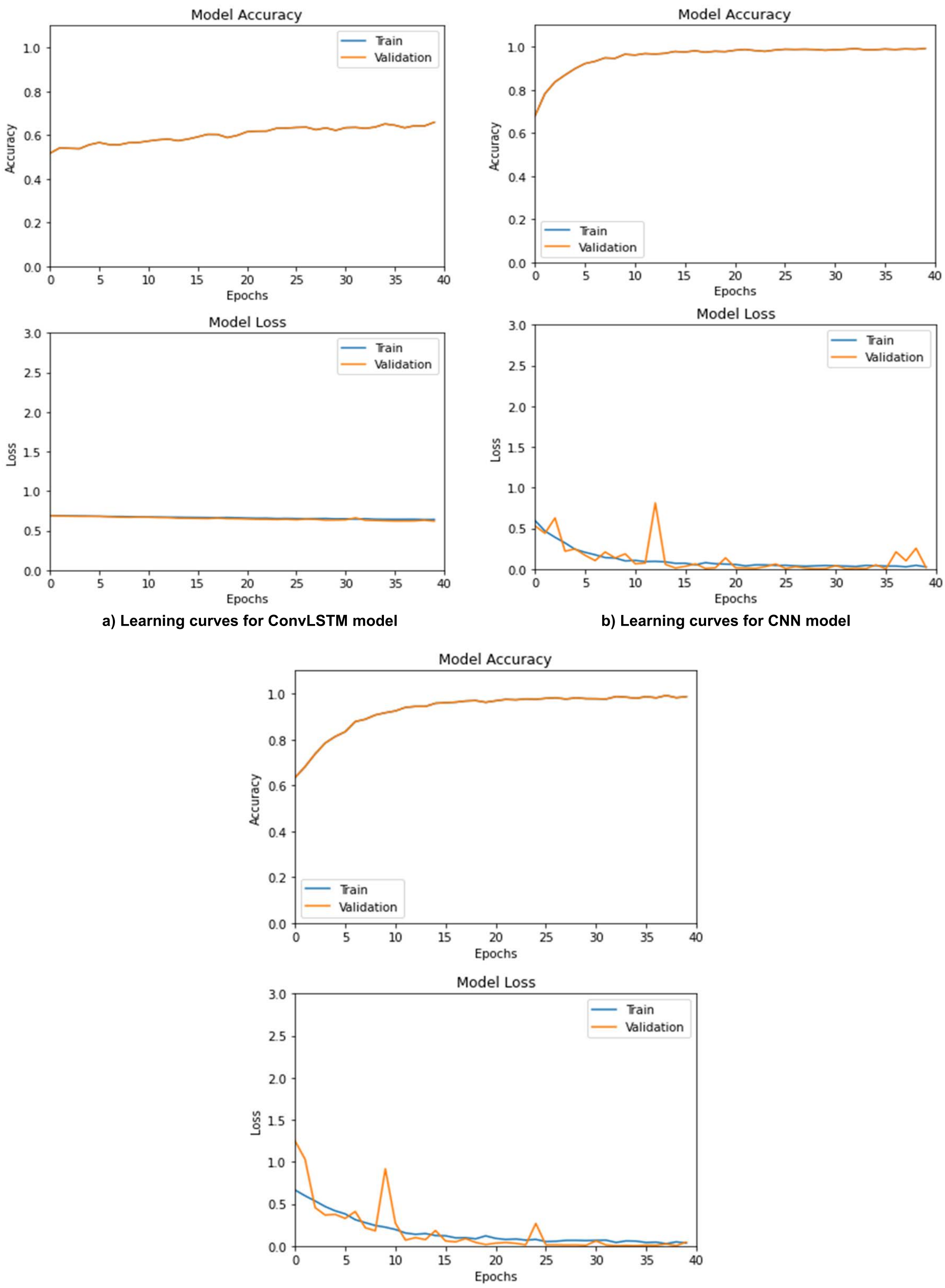


FIGURE 12. Confusion matrixes for LSTM, CNN, and hybid CNN-ConvLSTM models.

Zhu et. al [26], Zhang et. al [30], and Gwak et. al [31] have been compared with the proposed machine learning approach. Gwak et. al [31] algorithm is involved in both comparisons because it performs experiments in both video streaming and EEG signals. Table 4 shows a comparison of the proposed machine learning approach to the other EEG based algorithms. While the comparison of the proposed deep learning approach to other video steaming-based algorithms is shown in table 5.

According to table 4, the proposed machine learning approach has an accuracy value of 98.01%, comparing to 96%, 94.68%, 90.70%, and 89.80% for the algorithms Ko et. al [25], Zhu et. al [26], Zhang et. al [30], and Gwak et. al [31] respectively. Also, the proposed approach has a sensitivity value of 97.49%, whereas algorithms Zhu et. al [26], Zhang et. al [30], and Gwak et. al [31] have values of 95.32%, 90.20%, and 89.50%, respectively. In addition, the precision value of the proposed algorithm is superior by value 98.48% comparing with 97%, 95.57, and 88.70% for algorithms in Ko et. al [25], Zhu et. al [26], and Gwak et. al [31] respectively. *F1* score for the proposed machine learning algorithm is 97.98% which it is larger than values 95.48% and 89.10% in algorithms Zhu et. al [26], and Gwak et. al [31] respectively. The *TT*, which represents the speed of the algorithm response in detecting fatigue status, is an important parameter used in this study. The proposed algorithm produces *TT* with a value representing a fraction of a second equal to 0.187 second, as opposed to one second and 2.20 seconds for algorithms Zhu et al [26] and Zhang et al [30], respectively. Based on that comparison, it is clear that the proposed machine learning algorithm outperformed all other compared algorithms across all evaluation metrics.

According to table 5, accuracy value for the proposed deep learning approach is reaching 98.01% comparing with 94%, 97.10%, 94.20%, 95.4%, and 85.62% for algorithms Maior et. al [19], Biswal et. al [23], Jeon et. al [24], Gwak et. al [31], and Bakheet et. al [32] respectively. The sensitivity value of the proposed approach is reaching 100% comparing with 97.13%, 94.74%, and 93.5% for algorithms Biswal et. al [23], Jeon et. al [24], and Gwak et. al [31] respectively. The values of *TNR* are close for all algorithms which reaching 97.61% for the proposed algorithm comparing with 97.05%, 94.74%, and 95.4% for algorithms in Biswal et. al [23], Jeon et. al [24], and Gwak et. al [31] respectively. For *F1* score the proposed deep learning approach is gives 98.81% while other algorithms



a) Learning curves for ConvLSTM model

b) Learning curves for CNN model

c) Learning curves for hybrid CNN-ConvLSTM model

FIGURE 13. The learning curves for LSTM, CNN, and hybrid CNN-ConvLSTM models.

give 97.65%, 94.18%, 94.9%, and 87.84% for algorithms Biswal et. al [23], Jeon et. al [24], Gwak et. al [31], and Bakheet et. al [32] respectively. For the values of *FPR*, the proposed approach gives 2.39%, which it is nearing 2.95% offered by algorithm in Biswal et. al [23] but the overall accuracy in favor of the proposed algorithm. The *TT* is neglected in

TABLE 5. Comparison between the proposed deep learning approach and video streaming based algorithms.

Metric \ Algorithm	Proposed Deep Learning Approach	Maior et. al [19]	Biswal et. al [23]	Jeon et. al [24]	Gwak et. al [31]	Bakheet et. al [32]
ACC. %	98.01	94	97.10	94.20	95.4	85.62
Sens. = TPR %	100	----	97.13	94.74	93.5	----
FPR %	2.39	----	2.95	----	----	----
FNR %	0	----	2.87	----	----	----
Spec. = TNR %	97.61	----	97.05	94.74	95.4	----
Prec. %	97.66	----	97.07	93.90	97.1	----
F1 Score %	98.81	----	97.65	94.18	94.9	87.84
TT (Seconds)	10.61	----	----	----	----	----
Used Dataset	DROZY video streams	DROZY	Self-recorded video streams [23]	Self-recorded video streams [24]	Self-recorded video streams [31]	NTHU-DDD [22]

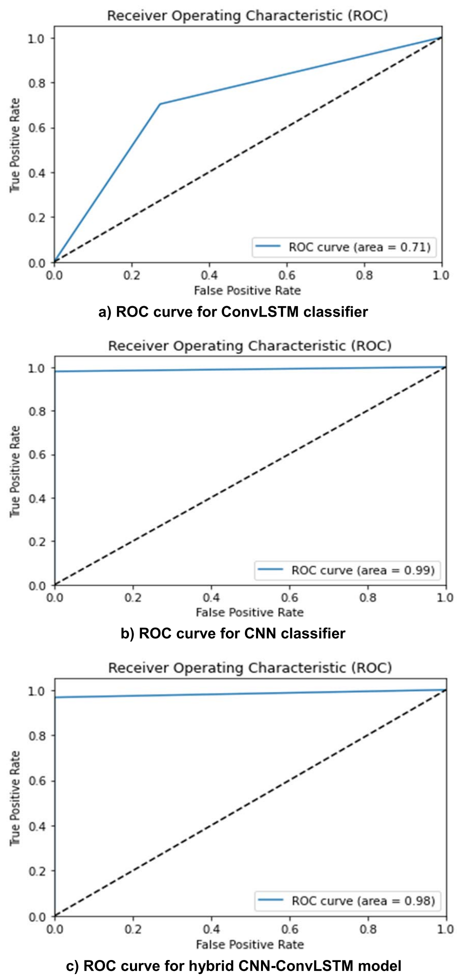


FIGURE 14. The ROC curves for LSTM, CNN, and hybrid CNN-ConvLSTM models.

other compared algorithms because the deep learning methods especially, on video streaming analysis takes large time. The proposed deep learning approach gives 10.61 seconds that represents a minor time for testing process performed over a complete video streaming analysis.

From that study, it is noticed that the processing time of machine learning classifiers is less than the processing time taken by deep learning classifier. The reason for this is due to the nature of each dataset, where machine learning classifiers process signals and deep learning classifiers process video streaming. Furthermore, deep learning classifiers provide good performance and stability in video streaming analysis. Based on previous comparisons and evolution metric values, it is clear that the two proposed approaches outperform all other compared algorithms. This progress is reflected in lower *TT* by high performance with low computational cost. Because of their high performance and low *TT*, the proposed methods more applicable in real time in the context of car systems.

By comparing the proposed algorithm testing time with other algorithms testing time it is shown that the testing time is neglected in other compared algorithms because the deep learning methods especially, on video streaming analysis takes large time. The proposed deep learning approach takes 10.61 seconds that represents a very little time for testing process performed over 20% of the complete video streaming analysis for all DROZY dataset videos streaming.

VI. CONCLUSION

This paper presents two different approaches: machine learning approach and deep learning approach. The main goal of that paper is to develop an effective fatigue detection system for high-performance cars drivers. The machine learning approach addresses the EEG signals processing by predicting driver behavior to detect fatigue status. This proposed machine learning algorithm had been developed by applying different machine learning classifiers such as SVM, RF, LR, MLP, KNN, and QDA. The proposed algorithm used this variety to achieve the best performance, as measured by the highest detection accuracy and the shortest detection time, with a high weight for the accuracy metric. According to the proposed algorithm results and using the DROZY EEG signals dataset, SVM is the best classifier deployed for solving this issue. The SVM classifier used in the proposed algorithm achieves detection accuracy of nearly 98% within a *TT* less than 0.18 seconds. These results outperform other well-known and efficient fatigue detection algorithms that use EEG signals from drivers. The deep learning approach, on the other hand, predicts the driver's behavior to detect fatigue status by processing the video streaming recordings of him. This deep learning proposed algorithm was created by combining several deep learning models including CNN, ConvLSTM, and hybrid CNN-ConvLSTM. The CNN is the best model for detecting driver fatigue status, according to the proposed algorithm results and using the DROZY video streams dataset. The CCN achieved 99% detection accuracy within *TT* of 10.61 seconds, outperforming other resent and efficient fatigue detection algorithms that process driver video streams.

In the future, the two approaches will be tested under other challenging datasets to demonstrate their efficiency against a variety of driving conditions. Furthermore, the ability to combine the two proposed approaches to improve results is being investigated. On the other hand, significant efforts are being performed to deal with video streams processing in order to integrate this detection system with IOT systems via cloud computing. In addition, we will prepare a prototype for the proposed algorithm.

REFERENCES

[1] Q. Abbas and A. Alsheddy, "Driver fatigue detection systems using multi-sensors, smartphone, and cloud-based computing platforms: A comparative analysis," *Sensors, Intell. Sens. Syst. Vehicle*, vol. 21, no. 1, p. 56, Dec. 2020, doi: 10.3390/s21010056.

[2] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas, and A. Mahmood, "A survey on state-of-the-art drowsiness detection techniques," *IEEE Access*, vol. 7, pp. 61904–61919, 2019, doi: 10.1109/ACCESS.2019.2914373.

[3] A. Rafid, A. R. Niloy, A. I. Chowdhury, and N. Sharmin, "A brief review on different driver's drowsiness detection techniques," *Int. J. Image, Graph. Signal Process.*, vol. 10, no. 3, pp. 41–50, 2020, doi: 10.5815/ijigsp.2020.03.05.

[4] P. Choudhary, R. Sharma, G. Singh, and S. Das, "A survey paper on drowsiness detection & alarm system for drivers," *Int. Res. J. Eng. Technol.*, vol. 3, no. 12, pp. 1433–1437, 2016.

[5] M. Q. Khan and S. Lee, "A comprehensive survey of driving monitoring and assistance systems," *Sensors*, vol. 19, no. 11, p. 2574, Jun. 2019, doi: 10.3390/s19112574.

[6] L. Chen, X. Zhi, H. Wang, G. Wang, Z. Zhou, A. Yazdani, and X. Zheng, "Driver fatigue detection via differential evolution extreme learning machine technique," *Electronics*, vol. 9, no. 11, p. 1850, 2020, doi: 10.3390/electronics9111850.

[7] V. E. M. Arceda, J. P. C. Nina, and K. M. F. Fabian, "A survey on drowsiness detection techniques," in *Proc. CEUR Workshop*, vol. 2724, Nov. 2020, pp. 152–161.

[8] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. E. Lopez, "Real-time system for monitoring driver vigilance," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 63–77, Mar. 2006.

[9] M. J. Flores, J. M. Armingol, and A. de la Escalera, "Real-time warning system for driver drowsiness detection using visual information," *J. Intell. Robot. Syst.*, vol. 59, no. 2, pp. 103–125, Aug. 2010, doi: 10.1007/s10846-009-9391-1.

[10] S. Abtahi, B. Hariri, and S. Shirmohammadi, "Driver drowsiness monitoring based on yawning detection," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2011, pp. 1–4, doi: 10.1109/IMTC.2011.5944101.

[11] A. M. Malla, P. R. Davidson, P. J. Bones, R. Green, and R. D. Jones, "Automated video-based measurement of eye closure for detecting behavioral microsleep," in *Proc. Annu. Int. Conf. IEEE EMBS*, Buenos Aires, Argentina, Aug./Sep. 2010, pp. 6741–6744.

[12] J. Jo, S. J. Lee, H. G. Jung, K. R. Park, and J. Kim, "Vision-based method for detecting driver drowsiness and distraction in driver monitoring system," *Opt. Eng.*, vol. 50, no. 12, Dec. 2011, Art. no. 127202.

[13] A. A. Lenskiy and J.-S. Lee, "Driver's eye blinking detection using novel color and texture segmentation algorithms," *Int. J. Control, Autom. Syst.*, vol. 10, no. 2, pp. 317–327, Apr. 2012, doi: 10.1007/s12555-012-0212-0.

[14] M.-H. Sigari, M. Fathy, and M. Soryani, "A driver face monitoring system for fatigue and distraction detection," *Int. J. Veh. Technol.*, vol. 2013, Jan. 2013, Art. no. 263983, doi: 10.1155/2013/263983.

[15] V. Vijayan and E. Sherly, "Real time detection system of driver drowsiness based on representation learning using deep neural networks," *J. Intell. Fuzzy Syst.*, vol. 36, no. 3, pp. 1977–1985, 2019.

[16] E. E. Galarza, F. D. Egas, F. M. Silva, P. M. Velasco, and E. D. Galarza, "Real time driver drowsiness detection based on driver's face image behavior using a system of human computer interaction implemented in a smartphone," in *Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)*, vol. 721. 2018, pp. 563–572, doi: 10.1007/978-3-319-73450-7_53.

[17] H. Bassi, H. A. Sulaimon, and M. A. Ahmad, "Drowsy driver detection and monitoring system using support vector machine," *ATBU J. Sci., Technol. Educ.*, vol. 8, no. 3, pp. 122–130, 2020.

[18] E. Ouabida, A. Essadike, and A. Bouzid, "Optical correlator based algorithm for driver drowsiness detection," *Optik, Int. J. Light Electron Opt.*, vol. 204, Feb. 2020, Art. no. 164102, doi: 10.1016/j.ijleo.2019.164102.

[19] C. B. S. Maior, M. J. D. C. Moura, J. M. M. Santana, and I. D. Lins, "Real-time classification for autonomous drowsiness detection using eye aspect ratio," *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113505, doi: 10.1016/j.eswa.2020.113505.

[20] S. Saurav, S. Mathur, I. Sang, S. S. Prasad, and S. Singh, "Yawn detection for driver's drowsiness prediction using bi-directional LSTM with CNN features," in *Proc. Int. Conf. Intell. Hum. Comput. Interact.*, 2019, pp. 189–200.

[21] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: A yawning detection dataset," in *Proc. 5th ACM Multimedia Syst. Conf. (MMSys)*, Singapore, 2014, pp. 24–28, doi: 10.1145/2557642.2563678.

[22] C. H. Weng, Y. H. Lai, and S. H. Lai, "Driver drowsiness detection via a hierarchical temporal deep belief network," in *Proc. Asian Conf. Comput. Vis. Workshop Driver Drowsiness Detection Video*, Taipei, Taiwan, Nov. 2016, pp. 117–133.

[23] A. K. Biswal, D. Singh, B. K. Pattanayak, D. Samanta, and M.-H. Yang, "IoT-based smart alert system for drowsy driver detection," *Wireless Commun. Mobile Comput.*, vol. 2021, Mar. 2021, Art. no. 6627217, doi: 10.1155/2021/6627217.

[24] Y. Jeon, B. Kim, and Y. Baek, "Ensemble CNN to detect drowsy driving with in-vehicle sensor data," *Sensors*, vol. 21, no. 7, p. 2372, Mar. 2021, doi: 10.3390/s21072372.

[25] W. Ko, K. Oh, E. Jeon, and H.-I. Suk, "VIGNet: A deep convolutional neural network for EEG-based driver vigilance estimation," in *Proc. 8th Int. Winter Conf. Brain-Comput. Interface (BCI)*, Feb. 2020, pp. 1–3, doi: 10.1109/BCI48061.2020.9061668.

[26] M. Zhu, J. Chen, H. Li, F. Liang, L. Han, and Z. Zhang, "Vehicle driver drowsiness detection method using wearable EEG based on convolution neural network," *Neural Comput. Appl.*, vol. 33, pp. 13965–13980, May 2021, doi: 10.1007/s00521-021-06038-y.

[27] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, Dec. 2001.

[28] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6765–6816, 2017.

[29] Q. Massoz, T. Langohr, C. François, and J. G. Verly, "The ULg multimodality drowsiness database (called DROZY) and examples of use," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–7, doi: 10.1109/WACV.2016.7477715.

[30] X. Zhang, J. Li, Y. Liu, Z. Zhang, Z. Wang, D. Luo, X. Zhou, M. Zhu, W. Salman, G. Hu, and C. Wang, "Design of a fatigue detection system for high-speed trains based on driver vigilance using a wireless wearable EEG," *Sensors*, vol. 17, no. 3, p. 486, Mar. 2017, doi: 10.3390/s17030486.

[31] J. Gwak, A. Hirao, and M. Shino, "An investigation of early detection of driver drowsiness using ensemble machine learning based on hybrid sensing," *Appl. Sci.*, vol. 10, no. 8, p. 2890, Apr. 2020, doi: 10.3390/app10082890.
[32] S. Bakheet and A. Al-Hamadi, "A framework for instantaneous driver drowsiness detection based on improved HOG features and Naïve Bayesian classification," *Brain Sci.*, vol. 11, no. 2, p. 240, Feb. 2021, doi: 10.3390/brainsci11020240.
[33] W.-L. Zheng and B.-L. Lu, "A multimodal approach to estimating vigilance using EEG and forehead EOG," *J. Neural Eng.*, vol. 14, no. 2, Apr. 2017, Art. no. 026017.

RIAD ALHARBEY received the Bachelor of Science degree in computer science from KAU, in 2002, the master's degree in information systems from the University of Maryland Baltimore County (UMBC), USA, in 2013, and the Ph.D. degree in information systems and technology from Claremont Graduate University (CGU), USA, in 2018. His master's thesis was related to the privacy of health data. His Ph.D. dissertation aimed to design an innovative assistive technology for patients with chronic diseases. He is currently an Assistant Professor with the Information Systems and Technology Department, College of Computer Science and Engineering (CCSE), University of Jeddah. He has some publications in the field of health informatics at HICSS, JMIR, AMCIS, BioMed, and DESRIST. His current research interests include the fields of health informatics and persuasive technology.

MOHAMED M. DESSOUKY received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees from the Faculty of Electronic Engineering (FEE), Menoufia University, Egypt, in 2006, 2011, and 2016, respectively. He is currently an Assistant Professor with the Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering (CCSE), University of Jeddah. He has published several indexed ISI journal articles, conference papers, and book chapters. His research interests include artificial intelligence, medical image processing, deep learning, and computer network security. He is a CISCO Academy Curriculum Lead. He is also a CISCO Certified Instructor. He had received an award from CISCO as a Best Instructor for more than ten years.

AHMED SEDIK received the B.Sc. and M.Sc. degrees in engineering from the Faculty of Engineering, Tanta University, Egypt, in 2012 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Faculty of Engineering, Minia University, Egypt. He is also an Assistant Professor with the Faculty of Artificial Intelligence, Kafrelsheikh University.

ALI I. SIAM received the B.Sc. (Hons.) and M.Sc. degrees in electronics and electrical communications engineering from the Faculty of Engineering, Tanta University, Egypt, in 2009 and 2016, respectively, and the Ph.D. degree in electronics and electrical communications engineering from the Faculty of Electronic Engineering, Menoufia University, Menouf, Egypt, in 2020. He is currently working as a Lecturer with the Faculty of Artificial Intelligence, Kafrelsheikh University, Egypt. His research interests include the IoT, computer vision, cloud computing, systems security, software engineering, and signal processing.

MOHAMED A. ELASKILY received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees from the Faculty of Electronic Engineering (FEE), Menoufia University, Egypt, in 2008, 2013, and 2019, respectively. He got the CCNA and CCNP certifications from CISCO. He is currently an Assistant Professor with the Department of Informatics, Electronics Research Institute (ERI), Egypt. He has published several indexed ISI journal articles, conference papers, and book chapters. His research interests include information security, digital forensics, multimedia security, data hiding, artificial intelligence, image processing, deep learning, and computer network security.