IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 February 2023, accepted 15 February 2023, date of publication 1 March 2023, date of current version 8 March 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3250834

RESEARCH ARTICLE

Smart Edge-Based Driver Drowsiness Detection in Mobile Crowdsourcing

HANANE LAMAAZI[1], AISHA ALQASSAB[2], RUBA ALI FADUL[2], AND RABEB MIZOUNI[1]
[1]Center for Cyber-Physical Systems, EECS Department, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[2]Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
Corresponding author: Rabeb Mizouni (rabeb.mizouni@ku.ac.ae)

ABSTRACT Traffic accidents caused by drowsy drivers represent a crucial threat to public safety. Recent statistics show that drowsy drivers cause an estimated 15.5% of fatal accidents. With the widespread use of mobile devices and roadside units, these accidents can be significantly prevented using a drowsiness detection solution. While several solutions were proposed in the literature, they all fall short of presenting a distributed architecture that can answer the needs of these applications without breaching the driver's privacy. This paper proposes a two-stage Driver Drowsiness Detection System using smart edge computing. Mobile devices in the car are used to capture and analyze the current condition of the drivers without sharing their data. The smart edge is deployed as a decision-maker where the drowsiness is confirmed when the information about the driver status received from the mobile client and the observed car path match. Our approach relies on a) a distributed edge architecture that has two levels of hierarchy, namely the Main Edge Node (MEN) and Local Edge Node (LEN), to better manage the area of interest and b) a data fusion offloading strategy that considers: 1) local detection of driver drowsiness through facial expressions using CNN model, 2) global detection of car path through acceleration readings using YoLov5 algorithm, and finally, 3) a two-layer LSTM algorithm for drowsiness detection based on the local and the global detection. The proposed framework achieves drowsiness detection with an average accuracy of 97.7%.

INDEX TERMS Drowsiness detection, deep learning, smart edge, convolutional neural network (CNN).

I. INTRODUCTION

As Drowsy driving results in about 328,000 crashes yearly [1]. It contributes to approximately 15.5% and 13.1% of fatal accidents causing deaths and injuries [2]. It happens when the person is driving with extreme drowsiness, making him not alert enough to respond to traffic events. An accurate drowsiness detection system that runs in real-time could help reduce drowsy driving accidents. However, existing solutions are limited by the drawbacks of the methodologies they use.

In the literature, mainly three methodologies are used to detect driver status, namely *vehicle behavior monitoring* methods, *physiological-based* methodologies, and *computer vision approaches* [3]. These methods rely on different features to build detection systems. Vehicle behavior methods utilize trends such as steering activity and vehicle position variability relative to road features [4]. However, these methods suffer from the unpredictability of the driving environment and the differences in driving habits between drivers, making them less reliable than the other approaches. Physiological-based methodologies depend on electroencephalogram (EEG) and heart rate variability measures collected via wearable devices. However, EEG devices are susceptible to vibrations generated by motion and engine. Therefore they fail to perform well in real driving conditions [5]. Finally, computer vision approaches rely on eye movement, yawning, and head orientation extracted from video streams and images. They are the most user-friendly and accurate approaches as they do not require wearing any measuring devices and are relatively independent of the external environment. Nevertheless, the performance of computer vision methods is affected by varying light conditions. They also tend to be computationally expensive

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

and are not efficient to be implemented on embedded systems [6].

From an architecture point of view, many of the existing drowsiness detection solutions are deployed on centralized platforms [7], [8], [9]. However, due to the limitations of these architectures in restricting the cooperation between different entities, they are not necessarily the best for data-sensitive real-time applications, especially when it comes to safety-critical problems such as drowsiness detection, where early detection can save individuals' lives. Also, centralized platforms require high computational performances, which is expensive and increases with the amount of data collected from the real-life environment. Accordingly, distributed architectures [10] have been proposed as an alternative to centralized solutions.

This work tries to cope with the challenges mentioned above by addressing the following questions: How to build a driver drowsiness detector that (i) consistently delivers accurate detection, (ii) is efficiently deployable on embedded systems, and (iii) can preserve the data privacy of drivers.

To answer these questions, a smart edge-based drowsiness detection system is proposed. The system primarily relies on smart edge deployment and data fusion offloading.

The distributed architecture used in this work adopts the OffSEC model proposed in [11] and [12], which is a crowdsensing framework that uses mobile edge computing (MEC). OffSEC utilizes a two-layer selection mechanism to discover a more comprehensive array of workers better fitted to the tasks, achieving a higher quality of service (QoS). The selection mechanism works by delegating data collection to main edge nodes (MENs) and worker nodes discovery and assignment to local edge nodes (LENs). The MENs and LENs are selected by the edge server (ES) from a pool of heterogeneous IoT devices with WI-FI connections, as shown in Figure 4.

The drowsiness detection is a two-stage detection: local detection through the driver's facial expression monitoring, using a mobile device inside the car, and global detection through the fusion of crowdsourced data provided locally by the drivers and vehicle trajectory monitoring detected by the edge servers. This offers two complementary channels to detect drowsiness efficiently. Data fusion is used on the edge nodes that receive the mobiles' outcomes of the driver drowsiness detection and the accelerometer readings that the LENs are continuously collecting along the vehicles' path. More specifically, a driver is confirmed as drowsy only after his mobile device generates a positive drowsiness detection alongside a confirmation based on the accelerometer readings produced at the LEN using the driver behavior classification model. The accelerometer readings are collected by the LENs across the vehicle's path and shared with the MEN only after sufficient records are collected. Once the drowsiness status of a driver is confirmed, it gets reported to concerned entities.

The proposed solution relies on deep learning algorithms to build a CNN that can accurately make driver drowsiness detection and an LSTM network that detects the driving behavior to confirm the output of the CNN model on the edge level.

The major contributions can be summarized as follows:

- An edge-based architecture that is enriched by the CNN model feeding its output to the edge node. The edge-based architect allows for a fast selection of the crowdsourced drivers and data analysis for timely drowsiness detection and information propagation through the edge nodes.
- The selection of the driver's car is processed by edge nodes based on the car's location within the range of the CCTVs.
- A Local detection of drowsiness based on the driver's facial expression using video streaming recorded by the driver's device.
- A global drowsiness detection at the edge level where the driver's car is tracked using the YoLoV5 algorithm for object detection, accelerometer readings, and weather conditions. The output of this tracking is combined with local detection for the final confirmation of the driver behavior, where a two-layer LSTM algorithm is deployed.

II. BACKGROUND AND RELATED WORK

The importance of this work lies in developing an integrated solution that accurately detects driver drowsiness using a multimodal approach over a distributed architecture, namely OffSEC. At the worker's level, the detections are handled using a deep learning vision-based detection model that runs on the driver's mobile phone and uses images captured for him while driving. At the edge nodes level, the data collected for the driver across several edge nodes are forwarded to the MEN node to confirm the detection output. The proposed solution at the MEN node utilizes the collected accelerometer data from the driver with a deep learning model to predict the driving behavior that emphasizes the drowsiness attitude of the driver. The output of the MEN node will present a confirmed drowsiness detection, which will allow emergency responders to be more alert and prepared to deal with the repercussions of the situation. Moreover, alerting the driver will enable him to exit the road safely before causing an accident.

Although multiple possible inputs can be used for drowsiness detection, with the focus on deep-learning computer vision techniques. Most works on drowsiness detection use the YawDD [15], NTHU [16], and DROZY [17] datasets to set a uniform baseline. More recently, a new dataset was proposed, UTA-RLDD [18]. The YawDD dataset is a yawn detection dataset. The NTHU dataset contains IR and RGB videos filmed in day and night settings with multi-ethnic participants. The participants were instructed to perform actions that are associated with drowsiness. The DROZY database contains images and a more diverse set of information regarding drowsiness, including EEG, EOG, and

TABLE 1. Literature Review Summary.

	Benchmark	Features Used	Embedded Device Deployment	Architecture	Compressed	Compression Algorithm
[7]	NTHU	RGB videos, Optical Flow Features	No	Centralized	No	-
[8]	Custom	Facial Features, Upper body Position	No	Centralized	No	-
[9]	NTHU	Region of Interest, Global Facial Features	No	Centralized	No	-
[13]	Custom	Region of Interest	Yes	Centralized	No	-
[14]	DROZY	Region of Interest	Yes	Centralized	Yes	Teacher-Student
[6]	NTHU-Expanded	Region of Interest	Yes	Centralized	Yes	Network Pruning
[10]	YawDD	Region of Interest	Yes	Distributed	No	-

ECG signals. The UTA-RLDD dataset is composed of RGB videos collected from multi-ethnic participants. The main difference between the two datasets is that the UTA-RLDD dataset captures the subjects as they fall asleep, thus capturing the micro-expressions that appear as the subjects become drowsy.

In [7], a deep drowsiness detection (DDD) network is proposed where features are extracted both as RGB videos and optical flow, the features are passed to three networks: AlexNet to extract features related to drowsiness, VGG-FaceNet to learn features related to drowsiness but is more sensitive to variations in appearance, and FlowImageNet extracts facial and head movements. This architecture achieves 73.06% detection accuracy on the NTHU dataset. A solution proposed that is more customized for buses and large vehicles is [8]. This proposal uses existing dome cameras to make predictions based on facial features and the driver's whole upper body. While considering pose variations, it uses a multi-model approach.

A [9] a Multi-granularity Convolutional Neural Network (MCNN) was proposed to extract both local and global representations and use an LSTM network to learn the relations between the models. They achieved an accuracy of 90.05% on the NTHU dataset. While the accuracy achieved by both approaches is relatively high, in general, the networks proposed are inefficient for running on embedded systems.

The study [13] uses an MCT AdaBoost classifier and an LBF regressor for face landmark detection. To achieve real-time results on an embedded device as input, they use 68 facial landmarks and classify eye states using the eye aspect ratio. It [14] presents a compressed model that can be efficiently deployed on embedded systems and reasonably accurate results. The teacher-student compression technique was used to compress the network. The best accuracy was achieved by a 2-stream model that uses the left eye and mouth as inputs achieving the test accuracy of 93.84%. The highest compression they achieved was also in a 2-stream model where the model was three times smaller than its teacher network. They used the DROZY database for benchmarking.

The authors in [6] proposed a lightweight CNN that accomplishes real-time performance of 60fps on an embedded device. They achieved an accuracy of 94.4% on an expanded NTHU dataset. The CNN is derived from the network proposed in [14] this proposal. However, a different technique was used to reduce the size of the CNN, wherein [14] they applied the teacher-student compression technique. Here the network was pruned, reducing the number of layers and neurons without adversely affecting the accuracy.

In [10], an IoT-based multimodal monitoring system is proposed where data can be fused from several sources, including user characteristics, history, and data collected from other sensors (vital sensors, vehicle sensors), to detect drowsiness. The paper focuses on visual data only. The detection from the video feed is done via a Fully Convolutional Recurrent Deep Neural Network (FCR-DNN) on edge, and the rest of the data is enriched in the cloud. They achieved the test accuracy of 99.5% for mouth detection and 99.01% for eye state detection. Our proposal goes one step further in that the drowsiness detections are made on the agents rather than the edge; this allows for less impact on the network and the edge.

To enrich the proposed drowsiness detection solution and to deliver an accurate detection, accelerometer readings collected from smart-phones sensors and deep learning computer vision-based approach are adopted. The accelerometer readings are used to classify driving behavior, which is used to confirm the detection made by the computer vision approach at the agent level. Accelerometer sensors are the most widely used sensors to address the problem of smartphone-based driving behavior classification [19]. The authors in [20] utilized 3-axis accelerometer readings collected from a light vehicle to classify safe and aggressive driving styles. The dataset consists of ten-time sequences of triaxial accelerometer readings for a total of 2.6 hours, sampled at 17 Hz. These time sequences represent the driving sessions and are labeled in a binary fashion as aggressive or normal.

Another accelerometer readings-based dataset employed by [19] and [21] to reveal aggressive driving behavior is a multi-labeled dataset. Besides the accelerometer sensor, the data contains readings from other sensors, namely linear accelerometer, magnetometer, and gyroscope sensors. The vehicle used for collecting the data experiments is a Honda Civic with two drivers using a Motorola smartphone. The data were collected at a sampling rate close to 50 Hz and ended up in a total of 104 minutes of driving divided into 69 driving sessions labeled with seven labels, six aggressive types, and one label for non-aggressive driving. This dataset is considered the most extensive set of aggressive driving types in the literature. The authors in [19] published another accelerometer-based dataset motivated by [21], consisting of 450 driving events. The data were collected at a sample rate of 50 Hz using two vehicles and Motorola smartphones. Five categories are addressed in this dataset, four of which represent abnormal driving behavior and one category for normal driving.

Smartphone-based driving behavior classification problem is addressed using threshold-based techniques and learning-based approaches. The study proposed by [22] detects events such as sudden braking, acceleration, and lane change using a threshold on accelerometer sensor readings. The authors in [23] classified the driving behavior into aggressive and safe driving using threshold-based techniques that use a g-g diagram to characterize the relationship between speed and acceleration. Although threshold-based algorithms are simple to design and implement, they suffer in defining one suitable threshold in most cases with the different accelerometer data sequences, which introduces the need for a dynamic learnable threshold.

On the other hand, machine learning algorithms achieved competitive results in addressing the problem of smartphone-based driving behavior classification. In [20], the authors conducted feature extraction, feature selection, and classification methods to distinguish aggressive driving style from safe driving behavior. They preprocessed 3-axis accelerometer sensor readings and extracted 78 driving features falling into five sets in the time and frequency domain: histogram features, correlation coefficient, data threshold violation, jerk profile features sets in the time domain, and spectral feature set in the frequency domain. Out of the 78 features, six were selected for the classification task using a random forest classifier and achieved a classification accuracy of 95.5%. The study [21] utilized data from different sensors, including the accelerometer sensor, and compared the performance of different machine learning algorithms on various combinations of the data, and concluded that the random forest classifier is the best-performing machine learning approach in addressing the problem of multi-class classification of driving behavior. This work proposes a deep learning approach based on the Long Term Short Term (LSTM) models to capture the temporal data sequences and predict the driving behavior in a binary classification fashion. The authors in [24] proposed a multifractal detrended fluctuation analysis method to detect driving fatigue. the method relies on two main subbands extracted from electroencephalogram (EEG) signals through different simulations used to calculate the Hurst exponent, symmetry of MF-DFA, and spectral width values. As a result, the proposed method shows an efficient detection against driving fatigue change when less interference and more stable data are deployed. An extensive analysis of driver drowsiness detection methods and classification techniques was deeply explained and discussed in [25]. The authors' study provides a classification of existing methods into three categories: physiological-based, behavioral-based, and vehicular-based approaches. This study proves that using a single method is not guarantee full accuracy while using a combination of techniques helps to improve drowsiness detection results. Also, in terms of supervised learning, SVM remains optimal and faster compared to other classifiers to achieve high accuracy, especially when used with a small dataset. The authors in [26] proposed a driver identification and verification approach relying on driver psychological behavior information including vehicle control operation and driver eye movement. The proposed approach deploys a squeeze-and-excitation (SE) block and a full convolutional network (FCN). The simulation was conducted using 24 participants under different driving scenarios where 99.60% of accuracy was detected out of 15 drivers. For the verification process, a Siamese neural network was added to map all the behavioral data used to compute the similarities which provided 96.91% of accurate verifications.

III. METHODOLOGY

As mentioned previously, the approach relies on two distinct steps: First, the local drowsiness detection is achieved at the driver side, and the drowsiness confirmation step is achieved at the edge side.

A. LOCAL DROWSINESS DETECTION MODEL

This section describes the proposed vision-based drowsiness detection approaches, which take input images sampled from a video feed capturing the driver's face using a front-facing camera embedded in the agent and produce a binary classification of whether the driver is drowsy or not. The first tested approach is a face-based drowsiness detector that takes as input the whole face and generates the classification. The second tested approach utilizes the eye region of interest (ROI) and mouth ROI. In this approach, two classifiers are built, and their outputs are jointly used to decide whether the driver is drowsy or not. The first classifier classifies the eye ROIs into *open* and *closed* classes, while the second classifies the mouth ROI into the *normal* or *yawing* state. When the eye is closed, and the mouth is yawning, the driver is considered drowsy and is normal otherwise.

1) FACE-BASED DROWSINESS DETECTION APPROACH

The first drowsiness detection approach uses facial features extracted from the whole face. The proposed model takes

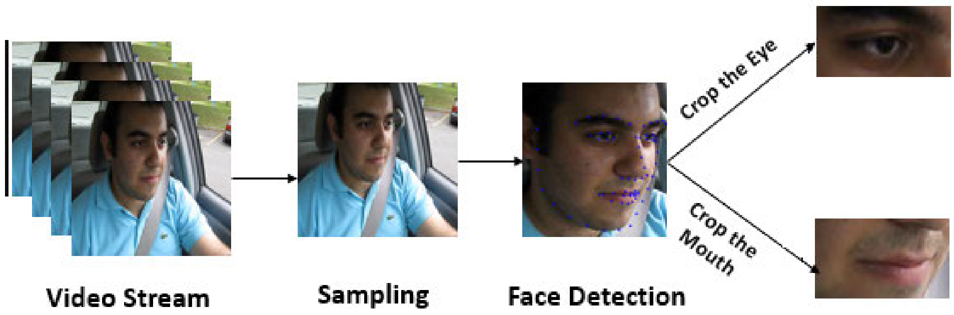


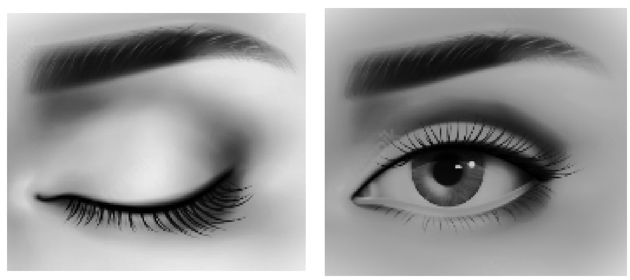
FIGURE 1. Input Data Samples for the Face-based Drowsiness Detection Model.

image frames extracted from videos at a resolution of 5 frames per second (FPS), which are fed to the model for training in two scenarios. The model is trained using the raw frame images in one scenario without applying face detection techniques to crop the face region of interest (ROI). The second scenario investigates face detection techniques to crop the face ROI from the frames before feeding the data to the model. Samples for the input data in both scenarios are shown in figure 1. The default dlib face detector is used to detect the face region, crop it from the frame and use it as an input to the model.

The face-based drowsiness detector is built using a pre-trained VGG16 model, previously trained on the ImageNet dataset. First, the pre-trained convolutional layers of the VGG16 model are frozed while the last fully connected (FC) layer is fine-tuned using the extracted frames from the NTHU dataset videos. To avoid model over-fitting, a dropout layer was added after the fully connected layer and L2 regularizers, which were added to the FC and output layers. The number of trainable parameters has been reduced from 15,894,849 to only 1,180,161 parameters due to the frozen pre-trained VGG16 network layers, which effectively reduced the training time and made use of the deep VGG16 architecture as well.

2) EYE AND MOUTH ROI-BASED DROWSINESS DETECTION APPROACH

To further improve the results of the driver drowsiness detection deployed on the worker nodes, a second approach that uses the facial features related to the eye and mouth region of interest is proposed. A preprocessing script is first applied to the videos captured for the driver, where image frames are extracted at 5 FPS. Then the eye ROI and face ROI are cropped from the frame and fed in parallel to two classifiers: eye and mouth. Only the eye located closer to the camera is considered to reduce the computational detection time without affecting the model efficiency [10]. The eye classifier takes the eye ROI as an input and outputs one of two classes, either eye *closed* or *open*. The input to the mouth classifier is the face ROI, and the output is *yawn* and *normal* classes. The driver's drowsiness state is determined from the frame output of the two classifiers, where the driver is considered drowsy if his eye is closed and he is yawning and is considered normal otherwise. The eye classifier is implemented using the CNN model and a sample of the input data used to train the classifier is shown in figure 2. Different variations of the network were examined to come up with the proposed architecture.



(a) Closed Eye **(b)** Open Eye

FIGURE 2. Input Data Samples for the Eye ROI Classification Model.

B. DISTRIBUTED ARCHITECTURE

This section describes the framework architecture establishment steps as presented in algorithm 1. First, the edge nodes are defined as CCTVs installed in the street, crossroads, and traffic lights, as illustrated in figure 3. They can be uni or multi-directional CCTVs, and they are considered high-performance roadside units due to the need to continuously fetch data from drowsy drivers. The distance between every edge node is 50 meters which is the detection

Algorithm 1 LEN and MEN Selection, Clustering, and Drowsiness Detection

```
Input : , Participant's coordinate, Connectivity_type = Wireless, Bluetooth,
        or CCTV, Accelerometers reading
Output: LENs, MEN, Drowsiness
1  initialization
2  for every connected participant do
3      Check the shared participant's connectivity type
4      if W_i connectivity_type = CCTV then
5          Add W_i to the list of preselected EN else
6              W_i is driver
7          end
8      end
9      end
10
11     for Every EN in predefined list do
12         calculate EN's objective function
13         Objective_function = w_1 × RE_i^W + w_2 × U_i^W + w_3 × Accuracy
14
15         for every time units do
16             Calculate the number of optimal clusters as
17             best_ size = list of LENs - 1
18             if LEN_ij = Maxt_objective function then
19                 LEN_ij is MEN_ij
20             FinalLENs = drop MEN_ij from list of LEN_ij
21             Construct the clusters as:
22             Cr_id = kmeans(L_i^W , best_size)
23             Centroids = best_size LEN_ij s
24             for Every Cluster do
25                 Select W_g^Cr where driver coordinate are within CCTV
                   range
26                 Detect driver car
27                 Calculate car accelerators
28                 Fuse local detection with accelerometers readings
29                 Apply LSTM Algorithm
30                 Confirm Drowsiness
31             end
32         end
33
34
35
```

range of CCTV (**lines 2-9**). Once the list of LENs (playing the role of CCTVs) is pre-defined, an objective function is calculated using LENs computing capabilities in terms of energy consumption RE_i^W and computing unit U_i^W and devices accuracy (**lines 11-13**) to define a final list of LENs and then the MEN. The MEN is LEN with the highest objective function, responsible for reporting the overall detection to authorities to stop the drowsy driver (**lines 16-20**). To cope with the mobility of cars and the change in the CCTV range, the clustering process is dynamic and relies on LENs and car locations (X and Y Coordinates).

Each re-clustering is the start of a new cycle. The cycle ends after the MENs' confirmation of drowsiness detection. The length of each cycle is set to five-time units. A relatively small process is selected to track the drivers as they are moving continuously.

After each cycle, LENs share records of drivers with sufficient data to the MEN, and the rest of the data is forwarded to the other LENs as carry-over. At the beginning of the next cycle, each LEN keeps only the carry-over records of its detected drivers and continues accumulating data. If a driver is detected as drowsy, his data collection continues until he gets a MEN drowsiness confirmation. This feature enables the solution to keep track of malicious drivers over time (**lines 25-31**).

C. DRIVING BEHAVIOUR CLASSIFICATION - DROWSINESS CONFIRMATION

The process of drowsiness detection is triggered by detecting driver drowsiness using a video stream captured by the mobile client. The detection result is then forwarded to the LENs, which continue to track the drowsy drivers to collect extra drowsiness detection records and accelerometer readings. Once the LENs have sufficient detection records per drowsy driver, it get forwarded to the MEN to confirm the drowsiness detection made by the LENs using the generated acceleration metrics. A list of confirmed drowsy participants is produced as output. On the MEN side, a driver behavior classification model is implemented. It uses the drivers' acceleration records collected over a while to detect driver behavior over a sliding window of records. To confirm the similarities between classes, five classes are considered as listed in [19] namely: *sudden right*, *left swerving*, *sudden acceleration* and *breaking*. A set of different sequence lengths are used to evaluate the classification accuracy as shown in table 5.

IV. SIMULATION PARAMETERS

In this work, four datasets are considered: Kaggle drowsiness detection dataset [27], NTHU dataset [16], Sarwat Foursquare datasets [11]. The Kaggle dataset contains two subsets; the first is for the eye ROI, which has two classes, *closed_eye*, and *open_eye*, while the second is for classifying the mouth as a *yawn* or *normal*. The whole dataset contains 2900 images with a size of 640 × 480 pixels. However, the NTHU dataset includes videos of drivers driving in different situations in day and night illuminance settings. The videos were collected with a resolution of 640 × 480 pixels in AVI format. An infrared (IR) illumination is employed to collect the low-light (night setting) videos. It contains videos of 18 subjects for training and evaluation. Data augmentation techniques are applied to both datasets, such as shear effects, re-scaling, and horizontal flipping techniques. To reduce the algorithm's complexity, the face ROI images are down-sampled to 112 × 112 pixels while the eye ROIs are resized to 64 × 64 pixels.

The acceleration dataset used in this model is provided in [19]. It contains acceleration readings of events across the x, y, and z axes. The length of the action sequences ranges from 38 to 200 time steps. The sequences are classified into *normal behavior*, *sudden breaking*, *sudden acceleration*, and *sudden swerving* which are used to make an active decision to break or accelerate. At the same time, the vehicle can swerve unexpectedly if the driver loses grip on the steering wheel. *sudden swerving* is considered as drowsy behavior, while *normal*, *sudden acceleration* and *sudden breaking* as normal behavior.

Sarwat Foursquare dataset [11], which is a dataset for social networking applications, is used in this model to get information about the drivers' devices such as energy, sensor availability, and device accuracy. More parameters used to

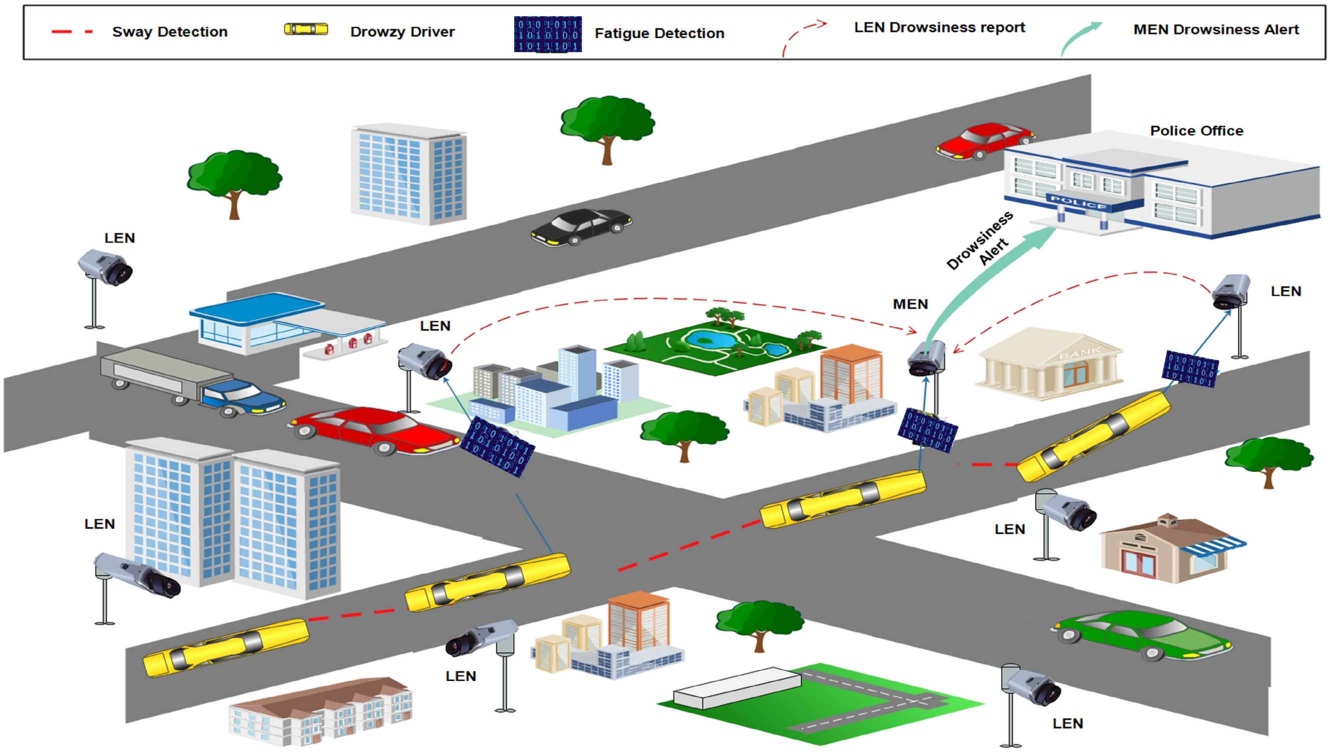


FIGURE 3. Edge-based distributed architecture for Drowsiness Detection.

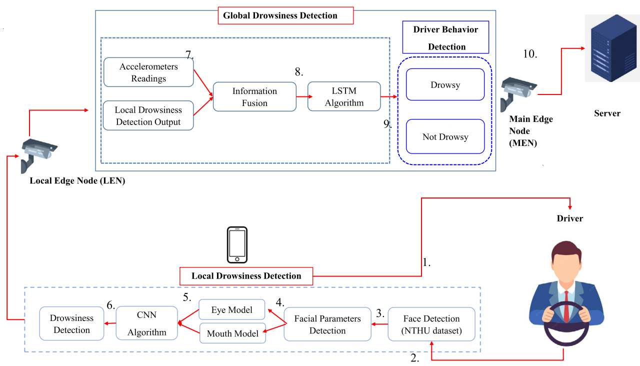


FIGURE 4. Conceptual Architecture.

simulate the drowsiness detection model are presented in table 2.

A. EVALUATION PARAMETERS

The main performance metrics are described as follows [28], [29], [30]:

- Precision rate:

$$Precision = \frac{TP}{TP + FP} * 100\% \tag{1}$$

where TP is the true positive and FP is the false positive. The precision rate presents the proportion of samples predicted to be true positive in the sample of positive cases.

- Recall rate:

$$Recall = \frac{TP}{TP + FN} * 100\% \tag{2}$$

where FN is the false negatives, and recall rate presents the proportion of samples predicted as negative cases to all positive samples.

- F1-score:

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

- Accuracy: It measures how accurate the model is in providing correct predictions.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{4}$$

V. RESULTS AND DISCUSSION

This section presents and discusses the results of the different implemented models.

A. FACE-BASED DROWSINESS DETECTION APPROACH

The training dataset used to train the proposed face-based drowsiness detection model is extracted from 16 subjects on the NTHU dataset. The model is validated on a randomly selected 20% of the training set and tested on a testing subset of the NTHU dataset. The best detection results are achieved using 256 neurons in the FC layer, with RMSprop and Adam optimizers using a learning rate of 0.0005 for the raw face and pre-processed face frames scenarios, respectively. Both models were trained using a batch size of 64 and 10 epochs.

TABLE 2. Implementation Parameters.

Parameters	Description
Datasets	NTHU, DoTA, OffSEC datasets
Driver Parameters	
Number of Participants	500
Number of edge nodes	4
Drivers Location (Lat, Long)	([31,...43], [129, ...144])
Drivers Information	Video Streaming, Accelerometers readings
Connectivity Type	Wireless, Bluetooth, CCTV
Car Information	ID, Name
Sensors type	Residual Energy, CPU, Accuracy
Weather Conditions	Sunny, Snowy, Rainy
Implementation Parameters	
Algorithms	CNN, LSTM
Train_Val_Test Split	80%, 20%
Number of Epochs	30 (ROI-based), 10 (Face-based)
Optimizers	Adam (Eye Model & Face-based), RMSprop (Mouth Model)
Learning Rate	0.0005 (ROI-based & Face-based)

TABLE 3. Results of Face-based/ROI-based Drowsiness Detection Model.

Model		Face-based Drowsiness Detection	ROI-based Drowsiness Detection
NTHU Validation Set	Loss	53.39%	**51.87%**
	Accuracy	75.45 %	**82.44 %**
NTHU Test Set	Loss	76.38%	**67.25%**
	Accuracy	59.08 %	**76.18 %**

The results of the face-based model are demonstrated in table 3 for the two input data scenarios, namely face-based and ROI-based detection. As seen in table 3, the ROI-based detection outperforms the Face-based detection due to the noise introduced from the area surrounding the face in the case of the Face-based scenario. However, the ROI-based frames are not performing well enough to employ this drowsiness detector in the final architecture due to the size of the dataset.

B. EYE AND MOUTH ROI-BASED DROWSINESS DETECTION APPROACH

1) EYE AND MOUTH CLASSIFIER

The training set of the Kaggle eye and mouth dataset [27] is randomly split into training and validation subsets to train and validate the model, which is then tested on Kaggle and NTHU datasets. The best classification results are achieved by training the model over 30 epochs with 16 batches with a learning rate of 0.0005 using Adam optimizer for eye training and RMSprop optimizer for mouth training. It is clear that the best classification accuracy is achieved for both eye and mouth using Kaggle dataset than the NTHU dataset. As previously explained, the Kaggle dataset contains two subsets: eye ROI and Mouth ROI which facilitate the learning process of the algorithm and reduce the loss function. However, for the NTHU dataset, the learning process is built from scratch, and processing the videos require more time and high computing performances which may lead to more loss.

TABLE 4. Results of the Eye and Mouth Classifiers.

Model	Eye Classifier	Mouth Classifier
Kaggle Validation Set	**98.79 %**	97.34 %
Kaggle Test Set	97.25 %	**98.18 %**
NTHU Test Set	91.25 %	76.56 %

TABLE 5. Results of LSTM Training on Different Sequence Lengths.

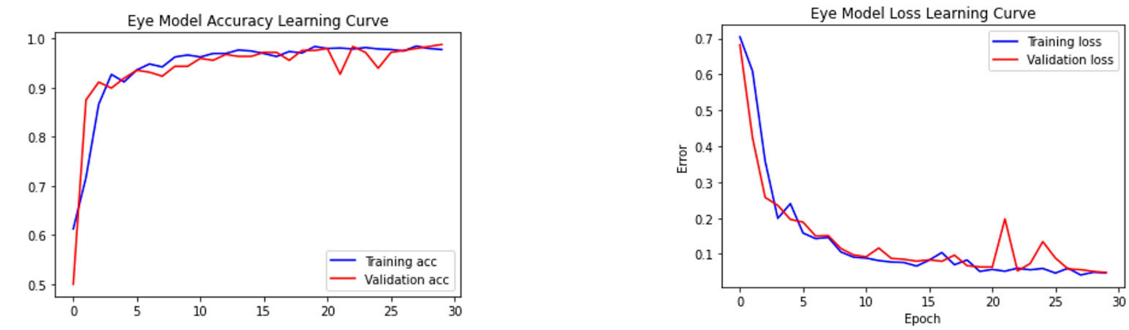
Length of Sequence	Precision	Recall	F1-score	Accuracy
8	88%	86%	87%	90%
12	88%	88%	88%	91%
16	93%	89%	91%	93%
20	90%	91%	91%	92%
24	92%	88%	89%	92%

Figures 5 (a) and 6 (a) illustrates the accuracy performances in the training and validation phases. As shown, CNN increased the training and validation accuracy rapidly over the number of epochs. This demonstrates that the model reached an optimal accuracy of 98 % and 97 % for eye and mouth detection, respectively, which proves that the model is well-trained. However, loss function performance is presented in figures 5 (b) and 6 (b). While training and validating the CNN model, the loss function decreased to achieve the lowest possible value, arriving at 5 % and 10 % for eye and mouth detection, respectively, where they converge at 30 epochs.

In contrast to the NTHU dataset, where the eye and mouth classification proceed independently, Kaggle datasets have pre-processed ROI that is automatically used to assess driver drowsiness. As a result, training the eye and mouth separately can lead to misbehaving in the model operations and reduce its performances by providing low detection accuracy, as confirmed in table 4. However, training the whole ROI, as in the case of the Kaggle dataset, can improve the model performance and increases the detection accuracy as illustrated in table 4

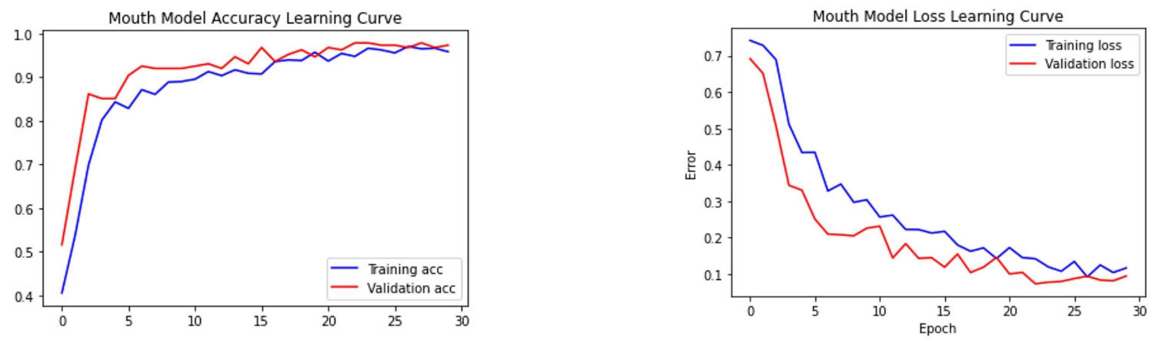
C. DRIVING BEHAVIOUR CLASSIFICATION- DROWSINESS CONFIRMATION MODEL

To select the best parameters, multiple simulations are run using input sequences of different lengths as shown in table 5. As a result, the trained model's best accuracy is obtained at the sequence of length equal to 16 and with the minimum records, where it reached 93%. For the final detection of drowsiness, LSTM achieved high performance in terms of precision, recall, and F1-score. Figure 7 shows that the model provides a detection precision of 93% and 92% for the not drowsy and drowsy classes respectively. Similarly, the not drowsy classes achieved high recall and F1-score of 98% and 95% respectively compared to the drowsy class that achieve 81% and 86% for recall and F1-score. Figure 8 presents the Macro-Average and Weighted-Average are used to calculate precision, recall, and F1-Score for each category regardless



(a) Training vs. Validation Accuracy for Eye Detection **(b)** Training vs. Validation Loss for Eye Detection

FIGURE 5. Training vs. Validation Accuracy and Loss for Eye Detection.



(a) Training vs. Validation Accuracy for Mouth Detection **(b)** Training vs. Validation Loss for Mouth Detection

FIGURE 6. Training vs. Validation Accuracy and Loss for Mouth Detection.

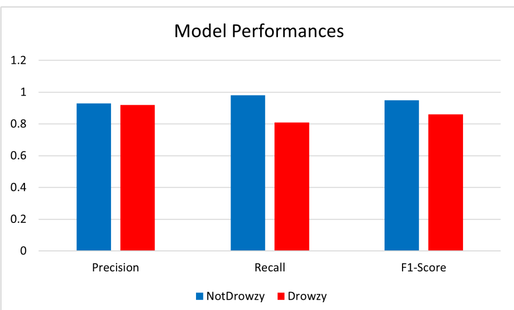


FIGURE 7. Prediction Performances of Driving Behaviour Classification.

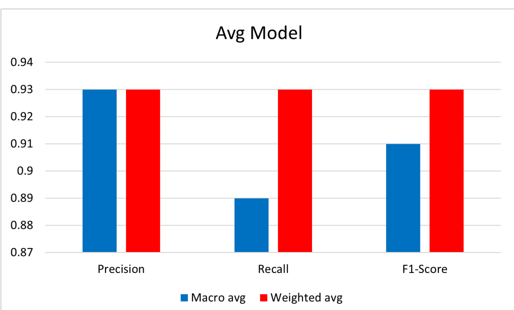


FIGURE 8. Average metrics of Driving Behaviour Classification.

of their proportion in the dataset. Both metrics achieved similar precision of 93% while they are different for recall and F1-score where macro-avg is lower than weighted-avg.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a drowsiness detection system that can provide accurate drowsiness detection. It is deployed on a distributed architecture, allowing it to overcome the drawbacks of deploying critical systems on centralized architectures. The drowsiness detection is implemented using two-stage detection: local detection through facial expression and global detection through the fusion of local and driving behavior detections. Using CNN models for eye and mouth classifiers achieve 97.3% and 98.2%, respectively. The overall drowsiness status is determined based on the output of the two classifiers and the car's accelerometer readings. the driver behavior classification model confirmed that driver drowsiness detection is processed at the edge level using LSTM algorithm, which achieved 93% accuracy. In future work, more parameters will be considered as heart rate and sensor body readings to confirm the drowsiness of the driver.

REFERENCES

[1] *Fatigued Driver National Safety Council*. Accessed: Nov. 2022. [Online]. Available: https://www.nsc.org/road/safety-topics/fatigued-driver

[2] (2010). *The Prevalence and Impact of Drowsy Driving*. [Online]. Available: https://aaafoundation.org/prevalence-impact-drowsy-driving/

[3] M. K. Hussein, T. M. Salman, A. H. Miry, and M. A. Subhi, "Driver drowsiness detection techniques: A survey," in *Proc. 1st Babylon Int. Conf. Inf. Technol. Sci. (BICITS)*, Apr. 2021, pp. 45–51.

[4] S. Lawoyin, D.-Y. Fei, and O. Bai, "Accelerometer-based steering-wheel movement monitoring for drowsy-driving detection," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 229, no. 2, pp. 163–173, 2015.

[5] H. Iwamoto, K. Hori, K. Fujiwara, and M. Kano, "Real-driving-implementable drowsy driving detection method using heart rate variability based on long short-term memory and autoencoder," *IFAC-PapersOnLine*, vol. 54, no. 15, pp. 526–531, 2021.
[6] S. Khare, S. Palakkal, T. V. Hari Krishnan, C. Seo, Y. Kim, S. Yun, and S. Parameswaran, "Real-time driver drowsiness detection using deep learning and heterogeneous computing on embedded system," in *Computer Vision and Image Processing* (Communications in Computer and Information Science), N. Nain, S. K. Vipparthi, and B. Raman, Eds. Cham, Switzerland: Springer, 2020, pp. 86–97.
[7] S. Park, F. Pan, S. Kang, and D. C. Yoo, "Driver drowsiness detection system based on feature representation learning using various deep networks," in *Proc. Asian Conf. Comput. Vis.* in Lecture Notes in Computer Science, C.-S. Chen, J. Lu, and K.-K. Ma, Eds. Cham, Switzerland: Springer, 2017, pp. 154–164.
[8] B. Mandal, L. Li, G. S. Wang, and J. Lin, "Towards detection of bus driver fatigue based on robust visual analysis of eye state," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 545–557, Mar. 2017.
[9] J. Lyu, Z. Yuan, and D. Chen, "Long-term multi-granularity deep framework for driver drowsiness detection," *CoRR*, vol. abs/1801.02325, 2018.
[10] M. Hashemi, B. Farahani, and F. Firouzi, "Towards safer roads: A deep learning-based multimodal fatigue monitoring system," in *Proc. Int. Conf. Omni-Layer Intell. Syst. (COINS)*, Aug. 2020, pp. 1–8.
[11] H. Lamaazi, R. Mizouni, S. Singh, and H. Otrok, "A mobile edge-based CrowdSensing framework for heterogeneous IoT," *IEEE Access*, vol. 8, pp. 207524–207536, 2020.
[12] H. Lamaazi, R. Mizouni, H. Otrok, S. Singh, and E. Damiani, "Smart-3DM: Data-driven decision making using smart edge computing in hetero-crowdsensing environment," *Future Gener. Comput. Syst.*, vol. 131, pp. 151–165, Jun. 2022.
[13] J. W. Baek, B.-G. Han, K.-J. Kim, Y.-S. Chung, and S.-I. Lee, "Real-time drowsiness detection algorithm for driver state monitoring systems," in *Proc. 10th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2018, pp. 73–75.
[14] B. Reddy, Y.-H. Kim, S. Yun, C. Seo, and J. Jang, "Real-time driver drowsiness detection for embedded system using model compression of deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 438–445.
[15] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: A yawning detection dataset," in *Proc. 5th ACM Multimedia Syst. Conf.*, Mar. 2014, pp. 24–28.
[16] 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.* in Lecture Notes in Computer Science, C.-S. Chen, J. Lu, and K.-K. Ma, Eds. Cham, Switzerland: Springer, 2017, pp. 117–133.
[17] Q. Massoz, T. Langohr, C. Francois, 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.
[18] R. Ghoddoosian, M. Galib, and V. Athitsos, "A realistic dataset and baseline temporal model for early drowsiness detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 178–187.
[19] M. R. Carlos, L. C. González, J. Wahlström, G. Ramírez, F. Martínez, and G. Runger, "How smartphone accelerometers reveal aggressive driving behavior?—The key is the representation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3377–3387, Aug. 2020.
[20] G. Žylius, "Investigation of route-independent aggressive and safe driving features obtained from accelerometer signals," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 2, pp. 103–113, Apr. 2017.
[21] J. Ferreira, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, and G. Pessin, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLoS ONE*, vol. 12, no. 4, Apr. 2017, Art. no. e0174959.
[22] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and M. C. González, "Safe driving using mobile phones," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1462–1468, Sep. 2012.
[23] L. Eboli, G. Mazzulla, and G. Pungillo, "How to define the accident risk level of car drivers by combining objective and subjective measures of driving style," *Transp. Res. Part F, Traffic Psychol. Behav.*, vol. 49, pp. 29–38, Aug. 2017.
[24] F. Wang, H. Wang, X. Zhou, and R. Fu, "A driving fatigue feature detection method based on multifractal theory," *IEEE Sensors J.*, vol. 22, no. 19, pp. 19046–19059, Oct. 2022.
[25] 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.
[26] J. Xu, S. Pan, P. Z. H. Sun, S. H. Park, and K. Guo, "Human-factors-in-driving-loop: Driver identification and verification via a deep learning approach using psychological behavioral data," *IEEE Trans. Intell. Transp. Syst.*, early access, Dec. 26, 2023, doi: 10.1109/TITS.2022.3225782.
[27] *Drowsiness Dataset*. Accessed: Mar. 2022. [Online]. Available: https://www.kaggle.com/dheerajperumandla/drowsiness-dataset
[28] W. Choukri, H. Lamaazi, and N. Benamar, "RPL rank attack detection using deep learning," in *Proc. Int. Conf. Innov. Intell. Informat., Comput. Technol. (3ICT)*, Dec. 2020, pp. 1–6.
[29] W. Choukri, H. Lamaazi, and N. Benamar, "Abnormal network traffic detection using deep learning models in IoT environment," in *Proc. 3rd IEEE Middle East North Afr. Commun. Conf. (MENACOMM)*, Dec. 2021, pp. 98–103.
[30] Y. Guo, H. Lamaazi, and R. Mizouni, "Smart edge-based fake news detection using pre-trained BERT model," in *Proc. 18th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2022, pp. 437–442.

HANANE LAMAAZI received the M.Sc. degree in networks and telecommunications from Chouaib Doukkali University, in 2013, and the Ph.D. degree in network and computer sciences from Moulay Ismail University, Morocco, in 2018. She was a Postdoctoral Fellow with the Center for Cyber-Physical Systems, Khalifa University, United Arab Emirates. Her research interests include the Internet of Things (IoT), RPL routing protocol, edge computing, and crowd-sensing.

AISHA ALQASSAB received the B.S. degree in computer science from the Faculty of Sciences, University of Sharjah, Sharjah, United Arab Emirates, in 2019. She is currently pursuing the M.S. degree in computer science with the Electrical Engineer and Computer Science Department, Khalifa University, Abu Dhabi, United Arab Emirates. Her research interests include machine learning applications, recommender systems, and deep learning applications, such as object tracking and audio source separation.

RUBA ALI FADUL received the B.Sc. degree in electrical and electronic engineering (software engineering of electronic systems) from the University of Khartoum, Khartoum, Sudan, in 2017. She is currently pursuing the M.Sc. degree in computer science with the Electrical Engineer and Computer Science Department, Khalifa University, Abu Dhabi, United Arab Emirates. Her research interests include deep learning, machine learning, keystroke dynamics, and mental health.

RABEB MIZOUNI received the M.Sc. and Ph.D. degrees in electrical and computer engineering from Concordia University, Montreal, Canada, in 2002 and 2007, respectively. She is currently an Associate Professor in electrical and computer engineering with the Center for Cyber-Physical Systems, Khalifa University. Her current research interests include deploying blockchain, crowd sensing, machine learning applied to behavior analysis, and cloud computing.