

Embedded machine learning-based road conditions and driving behavior monitoring

Bayan Mosleh¹, Joud Hamdan¹, Belal H. Sababha¹, Yazan A. Alqudah²

¹King Abdullah II School of Engineering, Princess Sumaya University for Technology, Amman, Jordan

²Electrical and Computer Engineering, University of West Florida, Florida, United States

Article Info

Article history:

Received Jan 29, 2024

Revised Feb 28, 2024

Accepted Mar 5, 2024

Keywords:

Driving behavior

Edge machine learning

Embedded systems

Machine learning

Road conditions

ABSTRACT

Car accident rates have increased in recent years, resulting in losses in human lives, properties, and other financial costs. An embedded machine learning-based system is developed to address this critical issue. The system can monitor road conditions, detect driving patterns, and identify aggressive driving behaviors. The system is based on neural networks trained on a comprehensive dataset of driving events, driving styles, and road conditions. The system effectively detects potential risks and helps mitigate the frequency and impact of accidents. The primary goal is to ensure the safety of drivers and vehicles. Collecting data involved gathering information on three key road events: normal street and normal drive, speed bumps, circular yellow speed bumps, and three aggressive driving actions: sudden start, sudden stop, and sudden entry. The gathered data is processed and analyzed using a machine learning system designed for limited power and memory devices. The developed system resulted in 91.9% accuracy, 93.6% precision, and 92% recall. The achieved inference time on an Arduino Nano 33 BLE Sense with a 32-bit CPU running at 64 MHz is 34 ms and requires 2.6 kB peak RAM and 139.9 kB program flash memory, making it suitable for resource-constrained embedded systems.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Belal H. Sababha

Computer Engineering Department, King Abdullah II School of Engineering, Princess Sumaya University for Technology

Amman 11941, Jordan

Email: b.sababha@psut.edu.jo

1. INTRODUCTION

The World Health Organization (WHO) reported in 2022 that approximately 1.3 million human lives are lost every year due to road traffic crashes. From 20 to 50 million others are injured, and among those are many who suffer from disabilities. This is in addition to significant economic losses [1]. These accidents are often caused by reckless driving, speeding, and unsafe road conditions. Artificial intelligence (AI) is becoming a primary contributor to advancements in different industries, including healthcare, education, technology, entertainment, military, and economics. It has made products and services more efficient and effective, enabling the analysis of large amounts of data quickly and accurately. As it advances, AI is expected to help protect and save human lives in many domains. One crucial domain is human safety on the roads. Many research has concentrated on utilizing embedded smartphone sensors and other methods in systems that are capable of predicting driving styles [2]–[6], driver behaviors [7]–[24], driving events [25], [26] and road conditions [27], [28]. Such systems are meant for safety or other applications that bring extra features and autonomy to vehicles [29]–[31]. The literature reports several works in machine learning for detecting and analyzing driving safety and road conditions.

The research presented by Al-Refai *et al.* [32] proposed machine learning algorithms to classify different characteristics of a car environment and driving styles using in-car data collected from the vehicle's controller area network (CAN) and Ethernet. The data was collected, labeled, and used to grade road surface conditions if it is (soft, even, or there are holes in it), Traffic levels (light, moderate, heavy), and the style of driving if it is (normal or aggressive) [32]. These systems need specialized sensors like cameras, radars, and ultrasonic sensors to gather information about the road. This study demonstrates that by utilizing machine learning techniques, it is possible to analyze and classify the data transmitted via vehicle networks and apply it to various applications. Using traditional machine learning (ML) algorithms to process in-vehicle data, they used random forests, decision trees, and support vector machine (SVM) and tested them. Random forests had the best accuracy, between 92% and 95%, and the system was able to classify data conditions and driving states with high accuracy. This research did not work on making a lightweight ML model appropriate for embedded systems or edge devices.

Boucetta *et al.* [33] proposed a system capable of monitoring the road situation and reporting the presence of cracks to help drivers avoid them and enable the responsible authorities to control and fix such roads. The system uses a convolutional neural network (CNN) to classify and detect road surface images, which has been found to be more effective because of its ability to implicitly extract features. A set of 3D pavement photos is used to train and test the CNN model. The system can classify road surfaces with an accuracy of over 95%. The system also includes a way to calculate the severity indicators for each road segment and build a weighted road graph for the road based on the riskiness indexes. The data is processed using a "Hadoop-based framework with HBase and MapReduce" [33]. The presented work was not intended nor aware of resource-limited embedded systems.

In the research by Mohammadnazar *et al.* [34] big data based on location and ML based on high-resolution data from connected vehicles are used to classify driving styles. The classification can be used to customize the driver aid system and many other things like fuel consumption, the value of mobility, and crash risks. The study uses basic safety messages (BSMs) generated by vehicles to classify driving styles using ML methods [34]. The study analyzes temporal driving volatility to measure risky driving behavior to categorize driving types based on vehicle kinematics, such as measured velocities and vertical/horizontal accelerations. The study applies K-means and K-medoid methods to group drivers into aggressive, normal, and calm. It finds that driving styles and due different roadway types have varied threshold levels of aggressive and calm driving. The study revealed that the highest rate of aggressive driving was recorded on commercial streets, as opposed to highways and residential streets [34].

The research done by Ziakopoulos *et al.* [35] presents a framework for aggregating and modeling high-resolution driving data from smartphone sensors to identify locations with harsh driving events, like harsh braking events (HBs). The framework uses locative models overall, the geographically weighted Poisson regression, Bayesian conditional autoregressive models (CAR), and variations of extreme gradient boosting (XGBoost) to look at the factors that contribute to extreme driving incidents on urban roads and to assess how well these models predict future events in a new test region for urban networks. The models are tested for accuracy and transferability for HBs predictions. According to the research, neighborhood complexity and gradient are adversely connected with HBs, while segment length and adjusted pass count positively correlate with HBs. The spatial predictions achieved more than 87% accuracy for HB frequencies per road segment when the results of all four methods were averaged. These findings are significant in traffic safety management, and there is a possibility of extending the framework to other hashed event types [35].

In the work done by Campo *et al.* [36], a driving style classification method was proposed. The method focused on attaining comfort driving. An advanced system to assist drivers and automated vehicles in saving fuel, providing driver comfort, and maintaining public safety was presented. The system is based on a hybrid method of machine learning and data obtained from a car equipped with sensors. The hybrid ML approach uses an unsupervised clustering method and an extreme learning machine (ELM).

None of the surveyed research was appropriate for embedded systems and edge devices with limited processing power, random access memory (RAM), and program memory. Utilizing artificial intelligence (AI) to develop a system suitable to run on edge devices and detect aggressive driving behaviors and road conditions in real time would benefit humanity and save more lives on roads. In this work, data is collected from different sensors and used to train ML algorithms to predict aggressive driving patterns and detect road conditions. The trained ML system is neural network (NN) based. The dataset is collected utilizing the embedded sensors of a smartphone. The developed system is optimized for edge devices, ensuring efficient execution and real-time predictions. The system is trained through machine learning techniques and deployed on a battery-powered, resource-constrained hand-held device, specifically a smartphone. It was also deployed on a microcontroller unit (MCU) based evaluation board, namely the Arduino Nano 33 BLE Sense featuring a 32-bit ARM® Cortex®-M4 CPU running at 64 MHz. The system can learn and detect different types of driving modes. It can learn about and detect unsafe road condition anomalies. The system can be trained and

operated regardless of the device's orientation in the vehicle. And most importantly, the system running on an edge device ensures that it shall continue to operate uninterrupted with the loss of internet connectivity.

In summary, the work presented in this paper solves the problem of designing lightweight machine-learning models appropriate for deployment on embedded and edge computing systems. Embedded devices suffer from limited processing power and limited program and data memory. Such limitations make the work presented in the literature inappropriate for deployment on these resource-constrained devices. Thus, requiring high-performance devices with internet connectivity to enable communication with the edge devices. The model presented in this paper is lightweight and appropriate for deployment on embedded and edge devices. This guarantees fault-free and uninterruptible operation on edge devices in the presence of internet and network connection losses. In Section 2, the method of designing the proposed ML model is presented. Section 3 covers the implementation and experimentation work. The achieved results and a discussion are presented in Section 4. Finally, Section 5 concludes the paper.

2. METHOD

Implementing and training machine learning techniques on a battery-powered, resource-constrained hand-held device such as a smartphone or a microcontroller can be challenging. This is because training ML models typically requires a lot of data and power. In this work, the system must handle the computational demands of training a machine-learning model without consuming too much power or memory. Using edge devices such as smartphones or MCUs with limited memory and low power to collect data and train is thus not possible. So, tools that run on high-performance machines or the cloud shall be used to train the neural network based on the data collected by the embedded sensors of the edge device. Then, the trained NN is deployed back to the edge device. The edge device is a resource-constrained battery-powered hand-held device such as a smartphone or MCU. In this work, a smartphone and an embedded computer were used. Deploying the system on a smartphone or an embedded device implies that the software should be optimized to function efficiently within the limited resources available. The limitations are in terms of computational power, memory, and energy efficiency. These constraints may arise due to the device's small size or the need to maximize battery life.

The system shall be able to learn and detect different driving modes. This can allow an assessment of the driver's driving style, including whether driving is aggressive, moderate, or non-aggressive by the trained model. To this, the data from the accelerometer, speed, and gyroscope sensors can be used to train an NN that shall be able to detect the different types of driving modes. Furthermore, it shall be able to learn about and detect unsafe road condition anomalies. This could include things like a street bump, yellow street bumps, or other hazards that could cause an accident or unsafe driving conditions for the model that has been trained. The mentioned road anomalies can be detected after training an NN by utilizing the data collected from the accelerometer sensors.

The data is collected from the embedded smartphone sensors. The sensor's data is then sent to the training cloud through various methods, including file upload or directly from the edge device. The ML models of the system are developed and deployed on the edge device using a cloud-based platform called Edge Impulse [37]. This tool provides an interface for collecting data and labeling it. It also enables training the ML models and deploying them on resource-constrained devices with limited memory. Edge Impulse is designed to work with supervised models. It also supports uploading datasets to train the ML models based on NNs. After the data is collected using the embedded sensors of an edge device (smartphone), it is uploaded to the Edge Impulse cloud live through a data collection web-based API. This can be done through a file upload as well. In the Edge Impulse cloud environment, the data shall be used to train the ML model. The sensor data is split into a training dataset (70%) and a test dataset (30%). Afterward, features are extracted using a feature extraction model, as seen in the following subsections. Then, the data is classified using a classification model that is trained using the labeled data. The model learns from the provided data to recognize and classify new, unseen data instances based on the learned patterns during training. In this work, the classifier is NN-based with multiple numbers of layers.

The block diagram in Figure 1 illustrates the overall system design process. It starts by collecting data from the X, Y, and Z accelerometer sensors and labeling the data based on the desired outputs. Split the data into training and testing subsets, allocating 70% to training and the remaining to testing. Edge Impulse automates feature extraction by analyzing and extracting relevant features from the uploaded data, capturing patterns and characteristics crucial for effective neural network training. Configure the neural network using the platform's provided architectures and settings. Initiate training using labeled data, allowing Edge Impulse to handle the process based on the selected network architecture. Evaluate the model's performance using testing data, leveraging Edge Impulse's evaluation metrics and tools to ensure accurate event classification. Finally, deploy the model upon satisfaction with its performance. Following is a detailed explanation of each step.

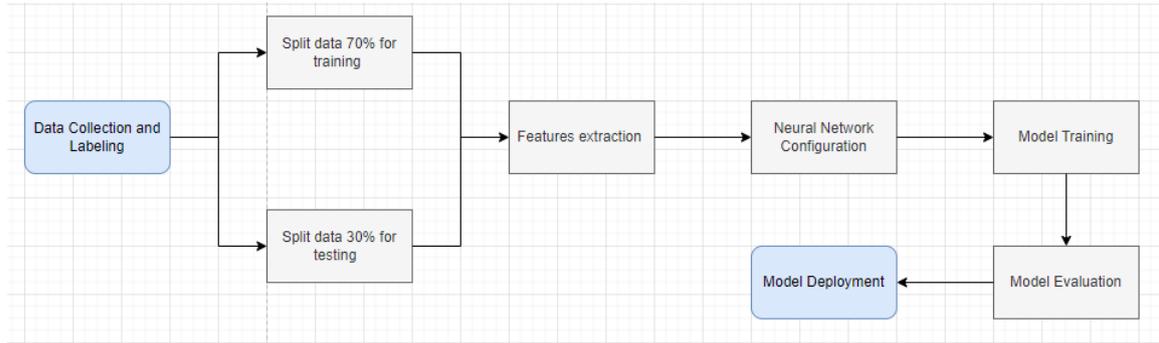


Figure 1. Block diagram for overall system design process

2.1. Data acquisition

In this work, we utilize the built-in sensors of a mobile phone to gather data for monitoring aggressive driving and road events. Specifically, we collect data from a three-axis accelerometer comprising three individual accelerometers, each measuring acceleration along a different axis: x-axis, y-axis, and z-axis. X, Y, and Z refer to the three axes of movement in the three-dimensional space. The X-axis represents the movement of the device from left to right, the Y-axis represents the movement from top to bottom, and the Z-axis represents the movement back and forth. Which can help detect sudden changes in acceleration or deceleration, sharp turns, and even road events.

2.2. Development of the neural network

The NN is developed in Edge Impulse and is constructed from several layers, each with a group of neurons. The layers are the Input layer with a total of 33 features for each axis: nine Inner layers and an Output layer of 6 classes. The quantity of internal layers and neurons went through stochastic tuning. Based on the results in section 4, the model selection with an acceptable size and inference time with the highest accuracy, precision, and recall was adapted (Model No. 3), shown in Figure 2.

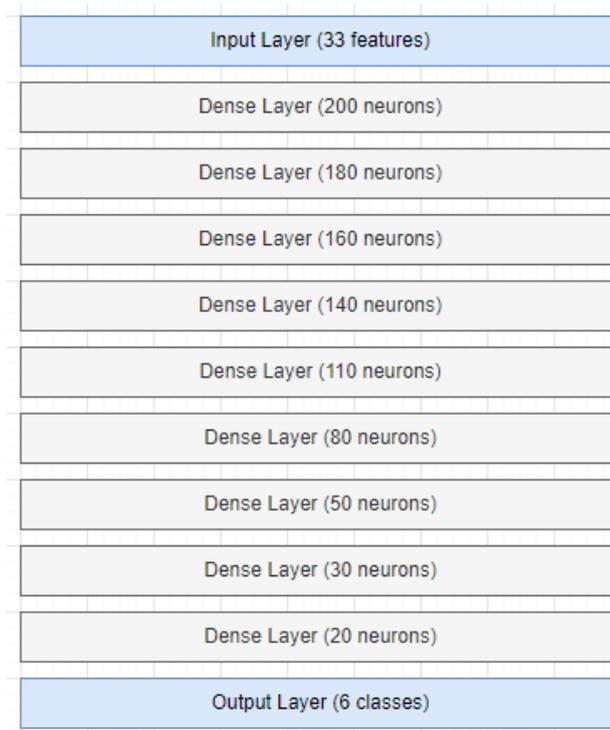


Figure 2. Developed neural network

The following features are extracted from the raw sample readings of the three accelerometer sensors:

- Root mean square (RMS) value: The RMS value represents the square root of the average of the squared accelerometer values within the window for each axis. It provides a measure of the overall magnitude or intensity of the acceleration in that axis during the specified time window.
- Three peak amplitudes from the power spectral density (PSD): The PSD is obtained by applying a Fourier transform to the accelerometer data. The three peak amplitudes refer to the highest power values observed at three distinct frequency locations within the window for each axis. These peaks indicate significant vibration or frequency components present in the signal.
- Three peak frequency locations from the power spectral density: The three peak frequency locations correspond to the frequencies at which the three peak amplitudes occur in the PSD for each axis. These frequency values represent the dominant frequencies or vibration frequencies observed in the accelerometer signal.
- Four summed bins from the PSD: The PSD is typically divided into frequency bins, representing specific ranges or segments of frequencies. The four summed bins refer to the accumulation or sum of power values across four selected frequency bins within the PSD for each axis.

A total of 33 features for each axis. These features include the RMS value, three peak amplitudes from the PSD, three peak frequency locations from the PSD, and four summed bins from the PSD. These extracted values form the input data for the NN.

3. IMPLEMENTATION & EXPERIMENTATION

The data collection process took place under dry weather conditions all around Amman, utilizing a Toyota Corolla 2019 as the vehicle of choice. Each data sample was captured at a consistent interval of 5 seconds. For a single event, 150 samples were collected, resulting in a cumulative dataset of 900 samples. The data collection process required a time investment of 32 hours, covering a distance of 492 kilometers, to gather the complete set of 900 samples. The project entailed collecting six distinct events, each corresponding to a specific label, as illustrated in Table 1.

Table 1. Events labeling representations

| Events | Label | Represent |
|--|-------|---|
| Normal Drive Normal Street Speed Bumps | A | This pertained to driving under typical conditions on a regular street, with speeds ranging from 30-60 km\hour. |
| Circular Speed Bumps | C | Circular speed bumps are employed to calm traffic near pavement markings, slow zones, traffic signals, and urban streets with frequent pedestrian and vehicle interaction. These small bumps are circular in shape, and the data is collected at multiple different speeds, low, medium, and high. |
| Sudden Start | D | Sudden start (takeoff) in drive refers to a situation where a vehicle accelerates abruptly or unexpectedly when the driver shifts the gear to the "drive" mode. The maximum speed limit of 40 was reached during data collection. |
| Sudden Stop | E | A sudden stop refers to a sudden and unexpected decrease in the speed of a vehicle, typically caused by the driver hitting the brakes abruptly or coming to an unexpected obstacle, the maximum speed limit of 40 before the stop was reached during data collection. |
| Sudden Detour | F | A sudden detour refers to a situation where a driver is forced to abruptly change direction or take a different route due to unexpected circumstances; speed was not reduced during data collection except in certain circumstances, such as when the route encountered an incline or a descent with a gradient of approximately 10 degrees during 5 sec. |

3.1. Normal drive and normal street

Assuming that "x", "y", and "z" sensors refer to the accelerometer sensors used in a vehicle, in a normal driving situation on a normal street, the sensors would typically detect little to no change in the orientation or movement of the vehicle in the x, y, and z axes. This is because the vehicle is moving at a relatively steady speed, and no sudden changes or forces are acting on the vehicle.

3.2. Speed bumps

Speed bumps are raised sections on roads or parking lots designed to slow down. They are typically made of asphalt or rubber and are used as traffic calming measures to reduce vehicle speeds in areas where safety is a concern. Speed bumps are commonly found in residential areas, school zones, and areas with

heavy pedestrian traffic. When drivers encounter speed bumps, they must slow down significantly to safely navigate over the raised surface. Speed bumps aim to increase safety by discouraging speeding and improving overall road safety for pedestrians and other road users. The sensors detect a sudden increase in acceleration as the vehicle approaches the speed bump, followed by a brief period of weightlessness as the vehicle goes over the bump, and then a sudden decrease in acceleration as the vehicle leaves the speed bump. The sensors may also detect vibrations or oscillations in the vehicle's movement as it passes over the bump, which can impact the vehicle's stability and affect the performance of its various control systems, such as the suspension, steering, and braking systems.

3.3. Circular yellow speed bumps

Yellow circular speed bumps are a type of traffic-calming methods designed to slow down vehicles and enhance road safety. They are circular in shape and painted in a bright yellow color for increased visibility and attention. The main difference between yellow circular speed bumps and regular speed bumps is their appearance and purpose. While both types of speed bumps reduce vehicle speeds, yellow circular speed bumps are specifically designed to draw attention and provide a clear visual indication to drivers. As the vehicle approaches a circular speed bump, the sensors will detect a slight increase in acceleration in the x and y axes as the vehicle starts to climb the bump. As the vehicle reaches the top of the bump, there will be a brief moment of weightlessness, during which the sensors may detect changes in acceleration in all three axes. As the vehicle starts to descend the bump, the sensors will detect a decrease in acceleration in the x and y axes, followed by an increase in acceleration as the vehicle returns to the level road surface. The circular speed bump may also affect the vehicle's suspension and other control systems, causing vibrations and oscillations that can impact the vehicle's stability and performance.

3.4. Sudden start

A sudden start describes a specific driving behavior where a vehicle abruptly accelerates from a stationary position. This behavior is often characterized by a sudden and forceful movement of the vehicle, causing passengers to lurch backward and potentially destabilize the vehicle. If a vehicle experiences a sudden start, such as when the driver rapidly accelerates the vehicle from a stop, the x, y, and z sensors in the vehicle will detect changes in the vehicle's movement and orientation. Specifically, the sensors will detect a sudden increase in acceleration in the x and y axes as the vehicle starts to move forward. Depending on the severity of the sudden start, the sensors may also detect changes in the z-axis, such as if the vehicle's front end lifts up due to the force of the acceleration.

3.5. Sudden stop

A sudden stop refers to an abrupt and unexpected cessation of vehicle motion, where the vehicle suddenly stops from its previous speed or movement. This type of driving behavior is characterized by rapid deceleration, often accompanied by the screeching of tires, passengers being jerked forward, or the vehicle abruptly coming to rest. If a vehicle experiences a sudden stop, such as when the driver suddenly applies the brakes, the x, y, and z sensors in the vehicle will detect changes in the vehicle's movement and orientation. Specifically, the sensors will detect a sudden decrease in acceleration in the x and y axes as the vehicle decelerates. Depending on the severity of the sudden stop, the sensors may also detect changes in the z-axis, such as if the vehicle's front-end dips down due to the force of the braking. The sudden stop may also cause vibrations and oscillations in the vehicle's movement, which can impact the vehicle's stability and performance.

3.6. Sudden entry

A sudden entry refers to a sharp and unexpected change in the direction of a vehicle's movement. It may occur when a driver needs to take immediate action to avoid an obstacle or make a quick turn. During a sudden detour, the vehicle's speed may change abruptly, and the driver needs to maneuver the vehicle quickly to avoid any collisions or accidents. When a vehicle makes a sudden detour, it can affect the x, y, and z sensors in different ways depending on the direction and magnitude of the turn. The x-sensor, which measures acceleration in the lateral direction, will detect a sudden change in acceleration when the vehicle turns, causing a spike in the x-axis readings. The y sensor, which measures acceleration in the vertical direction, may also detect a sudden change in acceleration if the turn involves going up or down a slope or over a bump. The z sensor, which measures acceleration in the longitudinal direction, may detect a sudden deceleration if the vehicle has to slow down or stop abruptly to detour. Figure 3 shows the accelerometer data for different driving conditions.

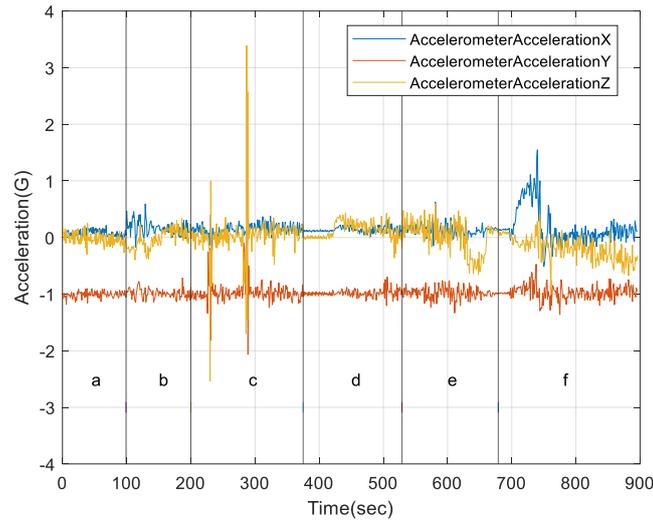


Figure 3. Accelerometer data for the different driving events, (a) normal drive, (b) speed bumps, (c) circular speed bumps, (d) sudden start, (e) sudden stop, and (f) sudden detour

4. RESULTS AND DISCUSSION

A data collection application called SensorLog [38] was installed on a smartphone and used to collect data from the accelerometer sensors of the phone while driving. The app records the relevant accelerometer data in files manually labeled afterward. The labeled data is then imported to the Edge Impulse machine learning platform to perform tasks such as data preprocessing, feature extraction, model training, and deployment. Edge Impulse provides the necessary tools and infrastructure for training and deploying machine learning models in the cloud.

In this work, we used smartphones of type iPhone 8 and iPhone 13 Pro for data collection and then deployment of the trained model within the Edge Impulse platform. The phones were securely mounted in a car using a phone holder. We collected various data types, particularly emphasizing accelerometer readings through the devices. We trained a machine-learning model using the gathered information after extracting 33 features from the X, Y, and Z accelerometers. Subsequently, the trained model was deployed back onto the same smartphones, enabling the seamless implementation of machine learning capabilities directly on the devices. The trained model was also deployed on an Arduino Nano 33 BLE Sense with a 32-bit ARM® Cortex®-M4 CPU running at 64 MHz. The evaluation board weighs 5 grams, is 18 mm wide and 45 mm long, operates at 3.3 V, and has 1 MB program flash memory and 256 kB SRAM. The board has the accelerometer sensors built in.

The developed ML model can anticipate six driving events, encompassing normal driving on normal streets, driving over speed bumps, and circular yellow speed bumps. Additionally, the model detects aggressive driving patterns characterized by sudden starts, sudden stops, and sharp turns. The model has achieved excellent performance results, as shown next.

The evaluation metrics that are used to evaluate the performance of the ML model are Accuracy, precision, recall, F1-score, loss, inference time, peak RAM, and Flash usage. These metrics serve as widely adopted benchmarks for assessing the performance of machine learning models. Following is a brief overview of each of the metrics.

- *Accuracy*: Refers to the proportion of correct predictions made by the model out of all predictions. It is used to measure how well the model can classify input data into the correct categories. A high accuracy score indicates that the model can correctly predict the output categories with a high degree of accuracy. Accuracy (Acc) can be presented as:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where TP is the number of true positive predictions, TN is the number of true negative predictions, FP is the number of false positive predictions, and FN is the number of false negative predictions.

- *Precision*: The ratio of true positive predictions to the total number of positive predictions made by the model. It represents the model's ability to correctly identify positive samples out of all the samples it

predicted as positive. A high precision indicates a low number of false positives, meaning that the model is accurate when predicting a positive sample. Precision (P) can be presented as:

$$P = \frac{TP}{TP+FP} \quad (2)$$

- *Recall*: Also known as sensitivity or true positive rate, it is the ratio of true positive predictions to the total number of actual positive samples in the dataset. It represents the model's ability to identify all positive samples correctly without missing any. A high recall indicates that the model effectively captures positive samples and minimizes false negatives. Recall (R) can be presented as:

$$R = \frac{TP}{TP+FN} \quad (3)$$

- *F1-score*: Measure of a model's accuracy that combines both precision and recall into a single value. It is the harmonic mean of precision and recall. The *F1-score* provides a balanced assessment of a model's performance by considering both the ability to correctly identify positive samples (recall) and the ability to minimize false positives (precision). It ranges from 0 to 1, 1 being the best possible score. Mathematically, the *F1-score* can be defined as:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

- *Loss*: Measure of how well the model is able to fit the training data. It represents the difference between the predicted output and the actual output. A lower loss value indicates that the model is able to fit the training data more closely and is, therefore, more accurate in its predictions.
- *Inference time*: Refers to the time taken by the machine learning model to make a prediction or inference after receiving input data. This time is an important metric for evaluating the efficiency and real-time performance of the model. It is particularly relevant for use cases where quick responses are necessary, such as in autonomous vehicles or real-time anomaly detection.
- *Peak RAM usage*: Refers to the maximum amount of random-access memory (RAM) consumed by the machine learning model during its operation. This metric is important for evaluating the memory efficiency of the model and ensuring that it does not exceed the memory limitations of the device on which it will be deployed. It is particularly relevant for use cases where resources are limited, such as in embedded systems or internet of things (IoT) devices.
- *Flash usage*: Refers to the amount of non-volatile program memory (usually flash memory) consumed by the machine learning model. This metric is important for evaluating the storage efficiency of the model and ensuring that it does not exceed the storage limitations of the device on which it will be deployed. It is particularly relevant for use cases with limited storage space, such as in microcontrollers or embedded systems.

Several NN models were implemented and evaluated according to the mentioned performance evaluation metrics. All models had an Input layer with a total of 33 features for each accelerometer axis. A number of Inner layers (from 7 to 12) and an Output layer of 6 classes. The quantity of internal layers and neurons went through stochastic tuning. Eight different models of acceptable performance are presented in Table 2.

Based on the results presented in Table 2, it is evident that the number of layers, number of neurons, inference time, peak RAM usage, flash usage, accuracy, and loss are all significant factors in determining the final model. The selection of the best model is contingent upon the user's specific requirements. In our particular case, we opted for the model with the highest accuracy, precision, recall, and F1-score scores, which is model no. 3. The Precision, Recall, and F1-Score illustrated in Table 2 are the averages for all six driving event predictions. The details for each of the events for the deployed model (model 3) are shown in Figure 4. As illustrated in Figure 4, it is observed that F (Sudden Detour) has the highest precision, indicating accurate positive predictions and a low false positive rate. Also, A (Normal Drive, Normal Street) demonstrated the lowest precision, suggesting a higher number of false positives in the predictions made for this event. Regarding recall, D (Sudden Start) stands out with the highest value, implying a successful capture of a larger proportion of actual positive instances. In contrast, C (Circular Speed Bumps) displays the lowest recall, indicating that a significant number of positive instances for this event are being missed or classified incorrectly. Examining the F1-scores, D (Sudden Start) shows the highest value, indicating a balanced performance with high precision and recall. On the other hand, A (Normal Drive, Normal Street) and C (Circular Speed Bumps) exhibit the lowest F1-score, suggesting an imbalance between precision and recall, potentially leading to a large number of false positives or false negatives.

Table 2. Different results for different neural networks

| Model Number | Number of layers | Total number of Neurons | Inference time (ms)* | Peak RAM usage (KBytes) | Flash usage (KBytes) | Accuracy (%) | Loss (%) | F1-score (%) | Precision (%) | Recall (%) |
|--------------|------------------|-------------------------|----------------------|-------------------------|----------------------|--------------|-----------|--------------|---------------|-------------|
| 1 | 12 | 825 | 27 | 2.7 | 112.6 | 89 | 71 | 88.7 | 89 | 88.7 |
| 2 | 12 | 980 | 35 | 2.7 | 140.3 | 90.1 | 60 | 90 | 90.5 | 89.8 |
| 3 | 11 | 970 | 34 | 2.6 | 139.9 | 91.9 | 63 | 91.7 | 92 | 91.8 |
| 4 | 10 | 650 | 17 | 2.3 | 76.5 | 89.8 | 60 | 89.8 | 89.7 | 89.7 |
| 5 | 11 | 660 | 17 | 2.5 | 76.9 | 89.7 | 70 | 89.3 | 89.7 | 89.5 |
| 6 | 11 | 630 | 16 | 2.5 | 70.6 | 88.1 | 56 | 88 | 88.2 | 87.8 |
| 7 | 9 | 570 | 15 | 2.2 | 66.6 | 89.0 | 47 | 88.8 | 89.2 | 88.8 |
| 8 | 7 | 250 | 6 | 1.6 | 30.6 | 88.1 | 47 | 87.8 | 88 | 88.2 |

* The achieved inference time of the system is on the Arduino Nano 33 BLE Sense with a 32-bit ARM® Cortex®-M4 CPU running at 64 MHz.

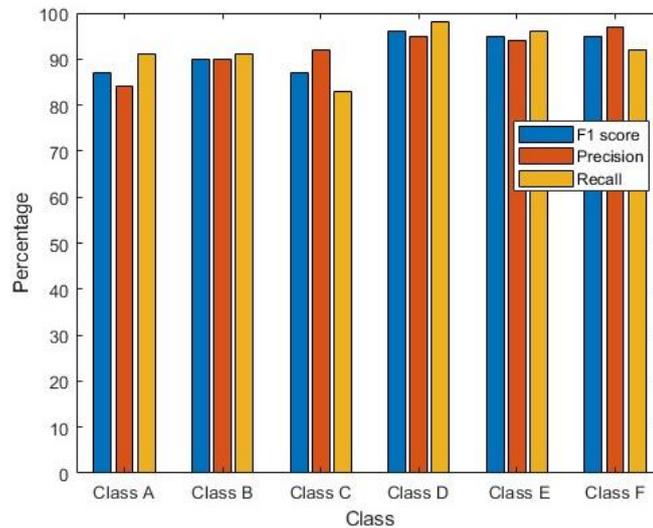


Figure 4. Model performance measurements

Overall, this graph provides valuable insights into precision, recall, and F1-score for the six events, facilitating a nuanced assessment of the classification model's performance. By considering these metrics, a better understanding of the model's strengths and weaknesses can be gained. This enables informed decisions for potential improvements or adjustments.

A confusion matrix summarizing the predictions made by the classification model and displaying the number of true positives, true negatives, false positives, and false negatives for each class is illustrated in Table 3. The highlighted cells indicate the correspondence between each group's expected and actual labels; among the groups, D achieved the highest accuracy at 98.0%, indicating a strong performance in recognizing a sudden start. E followed with an accuracy of 95.9%, demonstrating proficiency in identifying sudden stops. F achieved an accuracy of 92.4% for recognizing sudden turns. The accuracy for normal driving and street conditions, represented by group A, was 90.9%. For bumps, the accuracy was 90.5%. However, the accuracy for yellow bumps was comparatively lower at 83.3%, incorrectly identifying them as normal bumps 7.6% of the time. It is believed that the results may become better if trained on larger datasets.

In the confusion matrix, the row-wise values represent the distribution of predicted labels for each actual class. Thus, they always add up to 100% since they cover all possibilities for a given class. On the other hand, the column-wise values indicate the distribution of actual labels for each predicted class. The confusion matrix provides the actual class labels (rows) and the predicted class labels (columns), enabling evaluation from both angles and offering insights into the model's performance in classifying different classes and identifying potential biases or patterns in its predictions.

When comparing the work proposed in this paper with all surveyed research in the literature, the authors could not find any machine learning-based work appropriate for resource-constrained embedded devices or edge devices. Being able to develop machine learning-based solutions that can be deployed to edge devices and perform efficiently with an acceptable amount of accuracy is crucial. One of the most important advantages is to keep the system running on the edge device when internet connectivity is lost. A machine learning solution that is appropriate for embedded systems and edge devices shall be able to fit in

the limited program memory of the embedded device. It shall not use more RAM than that available of the edge device. Finally, given the limited processing power of the embedded or edge device, it shall be able to perform the computation efficiently and deliver results in a timely manner. In the work presented in this paper, we have developed an ML-based model that has an inference time of 34 ms on an embedded system with a 32-bit CPU running at 64 MHz. The model requires only 2.6 kB peak RAM usage and 139.9 kB program flash memory, which makes it suitable for resource-limited embedded systems. It achieved an accuracy of 91.9%, precision of 93.6%, recall of 92%, and F1-score of 91.7%.

Table 3. Confusion matrix

| Class | A | B | C | D | E | F |
|-------|---------------|---------------|---------------|---------------|---------------|---------------|
| A | 90.90% | 2.50% | 3.40% | 0.60% | 2.20% | 0.30% |
| B | 4.90% | 90.50% | 2.00% | 0.90% | 1.10% | 0.60% |
| C | 6.70% | 7.60% | 83.30% | 1.50% | 0.30% | 0.60% |
| D | 0.90% | 0.30% | 0.30% | 98.00% | 0.60% | 0% |
| E | 1.20% | 0% | 0% | 1.80% | 95.90% | 1.20% |
| F | 2.60% | 0.90% | 1.80% | 1.80% | 1.80% | 92.40% |

5. CONCLUSION

In this work, an embedded machine learning system that is capable of detecting road conditions and driving events was presented. The developed ML model was developed with attention to program and data memory needs and resource-constrained devices. Leveraging cloud-based training, the model achieved efficient training within a short time frame. The developed system features a 34 ms inference time on an embedded system with a 32-bit CPU running at 64 MHz. The model requires 2.6 kB peak RAM and 139.9 kB program flash memory, which makes it suitable for resource-constrained embedded systems. The resulting model achieved an accuracy of 91.9%, precision of 93.6%, recall of 92%, and F1-score of 91.7%. The developed system can detect several road conditions and some driving events via a resource-constrained edge device. This lays the foundation for developing models with more road conditions and driving events trained with larger datasets for potential real-world applications to enhance road safety and save more lives. The presented system brings the power of uninterrupted operation on edge devices in case of network and internet disconnection or loss.

Data Availability Statement

The dataset generated during the current study is available from the authors on request.

Executable Code Statement

The developed executable model is available and can run on a smartphone by scanning the following QR code.



REFERENCES

- [1] WHO, "Road traffic injuries," World Health Organization, <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries> (accessed Mar. 18, 2024).
- [2] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, Oct. 2011, pp. 1609–1615. doi: 10.1109/ITSC.2011.6083078.
- [3] F. Feng, S. Bao, J. R. Sayer, C. Flannagan, M. Manser, and R. Wunderlich, "Can vehicle longitudinal jerk be used to identify aggressive drivers? An examination using naturalistic driving data," *Accident Analysis and Prevention*, vol. 104, pp. 125–136, Jul. 2017, doi: 10.1016/j.aap.2017.04.012.
- [4] F. Yan, M. Liu, C. Ding, Y. Wang, and L. Yan, "Driving style recognition based on electroencephalography data from a simulated driving experiment," *Frontiers in Psychology*, vol. 10, no. MAY, May 2019, doi: 10.3389/fpsyg.2019.01254.
- [5] B. Jachimczyk, D. Dziak, J. Czaplą, P. Damps, and W. J. Kulesza, "IoT on-board system for driving style assessment," *Sensors (Switzerland)*, vol. 18, no. 4, p. 1233, Apr. 2018, doi: 10.3390/s18041233.
- [6] M. M. Rahman, M. T. Ismail, and M. K. M. Ali, "Car following and lane changing behavior using NGSIM and China data," *International Journal of Advances in Applied Sciences*, vol. 8, no. 1, pp. 14–25, Mar. 2019, doi: 10.11591/ijaas.v8.i1.pp14-25.
- [7] T. Pholprasit, W. Choochaiwattana, and C. Saiprasert, "A comparison of driving behaviour prediction algorithm using multi-sensory data on a smartphone," *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence,*

- Networking and Parallel/Distributed Computing (SNPD)*, Takamatsu, Japan, 2015, pp. 1-6, doi: 10.1109/SNPD.2015.7176249.
- [8] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *IEEE Intelligent Vehicles Symposium, Proceedings*, Jun. 2012, pp. 234–239. doi: 10.1109/IVS.2012.6232298.
- [9] L. Qin, Z. (Richard) Li, Z. Chen, M. S. Andi Bill, and D. A. Noyce, "Understanding driver distractions in fatal crashes: An exploratory empirical analysis," *Journal of Safety Research*, vol. 69, pp. 23–31, Jun. 2019, doi: 10.1016/j.jsr.2019.01.004.
- [10] S. F. Liang, C. T. Lin, R. C. Wu, Y. C. Chen, T. Y. Huang, and T. P. Jung, "Monitoring driver's alertness based on the driving performance estimation and the EEG power spectrum analysis," in *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 2005, vol. 7 VOLS, pp. 5738–5741. doi: 10.1109/iembs.2005.1615791.
- [11] I. H. Kim, J. H. Bong, J. Park, and S. Park, "Prediction of driver's intention of lane change by augmenting sensor information using machine learning techniques," *Sensors (Switzerland)*, vol. 17, no. 6, Art. no. 1350, Jun. 2017, doi: 10.3390/s17061350.
- [12] D. Ahn, H. Park, K. Shin, and T. Park, "Accurate driver detection exploiting invariant characteristics of smartphone sensors," *Sensors (Switzerland)*, vol. 19, no. 11, Art. no. 2643, Jun. 2019, doi: 10.3390/s19112643.
- [13] W. Kim, W. S. Jung, and H. K. Choi, "Lightweight driver monitoring system based on multi-task mobilenets," *Sensors (Switzerland)*, vol. 19, no. 14, p. 3200, Jul. 2019, doi: 10.3390/s19143200.
- [14] Y. Yao, X. Zhao, J. Li, J. Ma, and Y. Zhang, "Traffic safety analysis at interchange exits using the surrogate measure of aggressive driving behavior and speed variation," *Journal of Transportation Safety and Security*, vol. 15, no. 5, pp. 515–540, Jul. 2023, doi: 10.1080/19439962.2022.2098439.
- [15] N. O. Khanfar, M. Elhenawy, H. I. Ashqar, Q. Hussain, and W. K. M. Alhajjaseen, "Driving behavior classification at signalized intersections using vehicle kinematics: Application of unsupervised machine learning," *International Journal of Injury Control and Safety Promotion*, vol. 30, no. 1, pp. 34–44, Jul. 2023, doi: 10.1080/17457300.2022.2103573.
- [16] S. Chen, H. Yao, F. Qiao, Y. Ma, Y. Wu, and J. Lu, "Vehicles driving behavior recognition based on transfer learning," *Expert Systems with Applications*, vol. 213, p. 119254, Mar. 2023, doi: 10.1016/j.eswa.2022.119254.
- [17] K. Mohammed, M. Abdelhafid, K. Kamal, N. Ismail, and A. Ilias, "Intelligent driver monitoring system: An Internet of Things-based system for tracking and identifying the driving behavior," *Computer Standards and Interfaces*, vol. 84, Art. no. 103704, Mar. 2023, doi: 10.1016/j.csi.2022.103704.
- [18] Y. Shichkina, R. Fatkueva, and M. Kopylov, "Analysis of driving style using self-organizing maps to analyze driver behavior," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 2, pp. 2212–2225, Apr. 2024, doi: 10.11591/ijece.v14i2.pp2212-2225.
- [19] W. M. S. W. Bukhari, S. F. Toha, R. A. Hanifah, and N. A. Kamisan, "Speed analysis of motorcycle's wheel drive in various road conditions," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 13, no. 1, pp. 30–38, Mar. 2022, doi: 10.11591/ijped.v13.i1.pp30-38.
- [20] S. Liew, R. Hamidun, and A. Hamzah, "Factors that cause anger among motorcyclists: exploratory factor analysis," *International Journal of Public Health Science (IJPHS)*, vol. 11, no. 3, pp. 893–902, Sep. 2022, doi: 10.11591/ijphs.v11i3.21079.
- [21] N. Kamaruddin, A. W. A. Rahman, K. I. M. Halim, and M. H. I. M. Noh, "Driver behaviour state recognition based on speech," *Telkonnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 2, pp. 852–861, Apr. 2018, doi: 10.12928/TELKOMNIKA.v16i2.8416.
- [22] N. Kadri, A. Ellouze, M. Ksantini, and S. H. Turki, "New LSTM deep learning algorithm for driving behavior classification," *Cybernetics and Systems*, vol. 54, no. 4, pp. 387–405, Apr. 2023, doi: 10.1080/01969722.2022.2059133.
- [23] M. Malik and R. Nandal, "A framework on driving behavior and pattern using On-Board diagnostics (OBD-II) tool," *Materials Today: Proceedings*, vol. 80, pp. 3762–3768, 2023, doi: 10.1016/j.matpr.2021.07.376.
- [24] P. Ping, C. Huang, W. Ding, Y. Liu, M. Chiyomi, and T. Kazuya, "Distracted driving detection based on the fusion of deep learning and causal reasoning," *Information Fusion*, vol. 89, pp. 121–142, Jan. 2023, doi: 10.1016/j.inffus.2022.08.009.
- [25] Y. A. Alqudah and B. H. Sababha, "A statistical approach to estimating driving events by a smartphone," in *Proceedings - 2016 International Conference on Computational Science and Computational Intelligence, CSCI 2016*, Dec. 2017, pp. 1021–1025, doi: 10.1109/CSCI.2016.0195.
- [26] Y. A. Alqudah, B. Sababha, E. Qaralleh, and T. Youssef, "Machine learning to classify driving events using mobile phone sensors data," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 2, pp. 124–136, Jan. 2021, doi: 10.3991/ijim.v15i02.18303.
- [27] Y. A. Alqudah and B. H. Sababha, "On the analysis of road surface conditions using embedded smartphone sensors," in *2017 8th International Conference on Information and Communication Systems, ICICS 2017*, Apr. 2017, pp. 177–181. doi: 10.1109/IACS.2017.7921967.
- [28] K. K. M. Vamsee, K. Vimalkumar, R. E. Vinodhini, and R. Archanaa, "An early detection-warning system to identify speed breakers and bumpy roads using sensors in smartphones," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 3, pp. 1377–1384, Jun. 2017, doi: 10.11591/ijece.v7i3.pp1377-1384.
- [29] Y. A. Alqudah and B. H. Sababha, "Coded rumble strips to enhance reliability of autonomous vehicles," in *IEEE International Conference on Electro Information Technology*, May 2018, vol. 2018-May, pp. 296–301. doi: 10.1109/EIT.2018.8500114.
- [30] Y. A. Alqudah, B. H. Sababha, and T. Youssef, "Audition ability to enhance reliability of autonomous vehicles: Allowing cars to hear," in *Conference Proceedings - IEEE SOUTHEASTCON*, 2019, vol. 2019-April. doi: 10.1109/SoutheastCon42311.2019.9020496.
- [31] Y. A. Alqudah and B. H. M. Sababha, "Traffic notification system and method." U.S. Patent Number: US 10890462 B2, 2021.
- [32] G. Al-refai, H. Elmoaqt, and M. Ryalat, "In-vehicle data for predicting road conditions and driving style using machine learning," *Applied Sciences (Switzerland)*, vol. 12, no. 18, Art. no. 8928, Sep. 2022, doi: 10.3390/app12188928.
- [33] Z. Boucetta, A. El Fazziki, and M. El Adnani, "A deep-learning-based road deterioration notification and road condition monitoring framework," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, pp. 503–515, Jun. 2021, doi: 10.22266/ijies2021.0630.42.
- [34] A. Mohammadnazar, R. Arvin, and A. J. Khattak, "Classifying travelers' driving style using basic safety messages generated by connected vehicles: Application of unsupervised machine learning," *Transportation Research Part C: Emerging Technologies*, vol. 122, p. 102917, Jan. 2021, doi: 10.1016/j.trc.2020.102917.
- [35] A. Ziakopoulos, E. Vlahogianni, C. Antoniou, and G. Yannis, "Spatial predictions of harsh driving events using statistical and machine learning methods," *Safety Science*, vol. 150, Art. no. 105722, Jun. 2022, doi: 10.1016/j.ssci.2022.105722.
- [36] I. Del Campo, E. Asua, V. Martinez, O. Mata-Carballeira, and J. Echanobe, "Driving style recognition based on ride comfort using a hybrid machine learning algorithm," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, Nov. 2018, vol. 2018-Novem, pp. 3251–3258. doi: 10.1109/ITSC.2018.8569722.
- [37] Edge Impulse, "The leading edge AI platform," *Edge Impulse*. <https://edgeimpulse.com/> (accessed Oct. 10, 2023).
- [38] B. Thomas, "SensorLog v5.3," *SensorLog*, <https://sensorlog.berndthomas.net/> (accessed Oct. 22, 2023).

BIOGRAPHIES OF AUTHORS



Bayan Mosleh    is a dedicated computer engineer. She embarked on her academic journey at Princess Sumaya University for Technology, earning her bachelor's degree in 2023. During her academic pursuits, Bayan demonstrated a keen interest in various facets of technology, particularly artificial intelligence, machine learning, embedded systems, and embedded sensors. In 2023, Bayan applied her knowledge in a practical setting when she undertook a role as an Intern Engineer at the Royal Scientific Society. Over two months, she immersed herself in programming and quality assurance, gaining valuable hands-on experience. She can be contacted at email: bay20180678@std.psut.edu.jo.



Joud Hamdan    is a committed communications engineer who studied at the King Abdullah II School of Engineering, Princess Sumaya University for Technology, Amman, Jordan. She obtained her Bachelor's degree in communications engineering in 2023. In 2022, she worked as an intern at Orange Telecom company in the Mobile Core Department. Throughout her academic journey, she actively participated in diverse projects. Beyond her interests in communications engineering technologies, Joud displays a keen interest in edge machine learning and embedded systems. She can be contacted at email: joudhamdan@gmail.com.



Belal H. Sababha    is a professor of electrical and computer engineering and a drones and embedded systems consultant. He has been with Princess Sumaya University for Technology (PSUT) since 2012. Dr. Sababha is a US patent holder for three granted patents and served as a Chair for two international IEEE conferences. Prior to moving to Academia, Dr. Sababha worked in the automotive industry. He worked as a senior controls engineer in the Powertrain Controls Department at Chrysler Group LLC, Michigan, USA. He received his Ph.D. in electrical and computer engineering – embedded systems from Oakland University, MI, USA, in 2011. He has taught undergraduate and graduate electrical and computer engineering courses at various US and Jordan universities. Dr. Sababha has extensive experience in embedded systems design, control algorithm design, and software development with applications related to gasoline engine controls and unmanned aerial vehicles (UAVs). He is a consultant in the fields of embedded systems and UAV design and control for various governmental and commercial firms. His research concentration areas are UAV development and control, Biomedical instrumentation, embedded sensors, embedded RTOS and CAN networks, distributed embedded systems, graceful degradation in embedded systems, rapid prototyping, machine vision, and artificial intelligence. Dr. Sababha has served in several senior leadership positions as a dean for two terms, Acting Executive Dean, Associate Executive Dean, Director, and Department Chair. He is a senior member of IEEE and a member of several national and international professional organizations. He can be contacted at email: b.sababha@psut.edu.jo.



Yazan A. Alqudah    is professor of electrical and computer engineering at Hal Marcus College of Science and Engineering at the University of West Florida. He received his Ph.D. degree from Pennsylvania State University in 2003 in electrical engineering. He joined Intel Corporation, Oregon in 2003 as Senior Researcher where he worked with Logic Technology Development (LTD) and Mobile wireless groups (MWG). In his capacity, he led the development of LTD yield analysis system and the development and integration of WiMAX technology. Dr. Alqudah received three Intel's recognition awards in 2005 and 2007 for his successful efforts. In 2008, he joined the Communication Engineering Department at Princess Sumaya University for Technology. Between Fall 2014-Spring 2016, Dr. Alqudah served as chair of the Communications Engineering Department. During 2016/2017, Dr. Alqudah taught at Western Carolina University School of Engineering. In Fall 2018, Dr. Alqudah joined the Department of Electrical and Computer Engineering at the University of West Florida as a professor. Dr. Alqudah is a senior member of IEEE. In 2014, he won best The Best Researcher Award at PSUT. In 2023, he received the Excellence in Teaching Award at UWF. His current research interests include broadband optical wireless communication, next generation mobile networks deployment and performance, and mobile development. He can be contacted at email: yalqudah@uwf.edu.