**TELKOMNIKA Telecommunication Computing Electronics and Control**
Vol. 24, No. 4, August 2026, pp. 1294~1306
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v24i4.27659


# Classification of P300 event-related potentials using SNN, CNN and LSTM deep learning models

**Ahlaam M. Saed[1], Ibtihal Fawzi Elshami[2], Ali I. Elgayar[3]**
[1]Department of Electrical and Electronics Engineering, Bright Star University, Brega, Libya
[2]Department of Electronics Engineering, College of Electrical and Electronics Technology, Benghazi, Libya
[3]Department of Electronics Engineering, University of Benghazi, Benghazi, Libya

**Article Info**

***Article history:***

Received Dec 11, 2025
Revised Apr 3, 2026
Accepted May 25, 2026

***Keywords:***

Brain-computer interface
Convolution neural network
Electroencephalography
Long-short-term-memory
P300
Rectified linear unit
Spiking neural network

**ABSTRACT**

Accurate classification of P300 event-related potentials remains challenging due to the complex, non-stationary, and low signal-to-noise characteristics of electroencephalography (EEG) signals in brain-computer interface (BCI) systems. P300-based devices, such as the P300 speller, enable communication for patients with severe motor impairments, including those with locked-in syndrome; however, reliable brain signal classification is still a critical limitation. This study presents a comparative evaluation of deep learning models, including convolutional neural networks (CNN), long short-term memory (LSTM) networks, and spiking neural networks (SNN), for P300 signal classification. SNNs represent a biologically inspired paradigm that models the discrete, time-dependent behavior of neural spiking activity and offers advantages in terms of energy efficiency and hardware implementability. Experimental results demonstrate that CNN achieved the highest average classification accuracy (81.04%), followed closely by SNN (80.94%) and LSTM (80.60%). Although CNN slightly outperformed the other models, SNNs showed comparable accuracy while requiring fewer training samples and offering potential benefits for low-power and real-time BCI systems. These findings highlight the trade-offs between classification performance and computational efficiency and underline the promise of SNNs as an efficient alternative for P300-based BCI applications.



***Corresponding Author:***

Ibtihal Fawzi Elshami
Department of Electronics Engineering, College of Electrical and Electronics Technology
Benghazi, Libya
Email: ifelshami@ceet.edu.ly

## 1. INTRODUCTION

Brain-computer interfaces (BCIs) enable direct communication between the human brain and external devices, offering an essential communication channel for patients with severe motor impairments, particularly those affected by locked-in syndrome. By translating neural activity into control commands, BCIs bridge the gap between cognition and action and allow users to interact with assistive technologies, as illustrated in Figure 1. Depending on the signal acquisition approach, BCIs can be classified into invasive systems, which require implanted electrodes, and non-invasive systems that rely on electroencephalography (EEG) signals recorded from the scalp. Due to their safety, accessibility, and ease of use, non-invasive EEG-based BCIs are the most widely adopted despite the inherent complexity and non-linearity of EEG signals, which makes accurate brain signal classification challenging [1], [2].

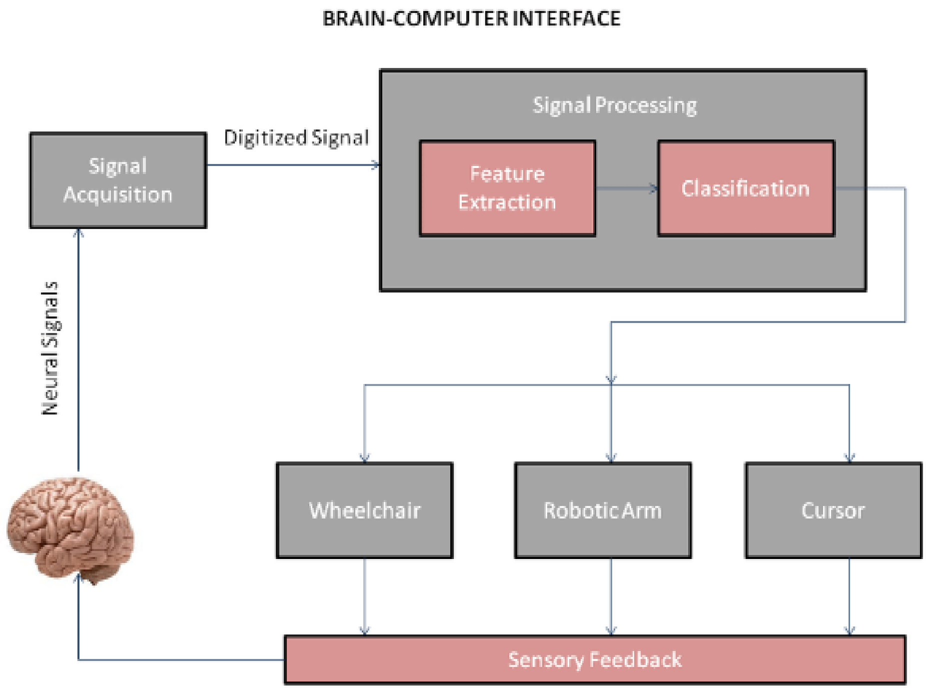



Figure 1. Working of a generic BCI [2]

The P300-based BCI, a widely used EEG paradigm, elicits a positive event-related potential (ERP) around 300 ms after task-relevant stimuli, enabling users to select symbols or commands, and is particularly useful for individuals with severe disabilities [3], [4]. Classification is challenged by low signal-to-noise ratio (SNR), non-stationarity, and inter-subject variability. Recurrent neural networks (RNNs), especially long short-term memory (LSTM) networks, capture long-term temporal dependencies in EEG, improving interpretation of user intent and overall BCI performance [5]. Spiking neural networks (SNNs) encode information via discrete spikes, offering temporal precision, energy efficiency, and suitability for low-power, real-time BCIs. Training SNNs is complicated by non-differentiable spikes, but surrogate gradient methods and artificial neural network (ANN)-to-SNN conversion techniques combine ANN training efficiency with SNN inference advantages [6], [7].

Motivated by this gap, the objective of this study is to conduct a comprehensive comparative evaluation of convolutional neural network (CNN), LSTM, and SNN architectures for P300 signal classification using a standardized P300 speller dataset. The main contributions of this work are threefold: (i) a fair comparison of spatial, temporal, and spiking neural models using the same preprocessing and evaluation protocol; (ii) an analysis of classification accuracy across subjects to assess model robustness; and (iii) an investigation of the potential of SNNs as an energy-efficient alternative for P300-based BCI systems.

The remainder of this paper is organized as follows. Section 2 presents the theoretical background of neural network models and the P300 potential. Section 3 describes the dataset, preprocessing procedures, experimental setup, and results. Finally, section 4 concludes the paper and outlines directions for future research.

## 2. METHOD

Neural networks are a network of interconnected processing units designed to mimic desired behavior. Because of the similarities to biological neural networks, these units are commonly referred to as neurons and their connections as synapses. An analog network's neuron is a type of cell that converts a certain input signal into an equivalent output signal [2], [5].

The activation function of the neuron is used to accomplish these modifications as show in Figure 2. The function sums the signals, or information, it gets from synapses (usually real numbers), and then outputs the result. Synapses are involved in both signal transmission and strength regulation. The activation "fires," or generates a strong output, if the total input is strong enough [3], [6]. Although there are numerous varieties of neural networks, we will talk about three here: recurrent, spiking, and convolutional CNNs are examples of neural networks.

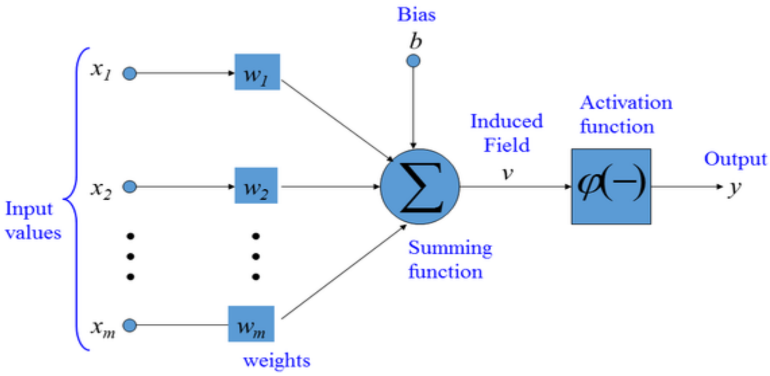



Figure 2. Model of a neuron network [5]

### 2.1. CNN

In computer vision, which deals with creating systems that evaluate visual data and recognize or segment important features or objects, CNNs are sophisticated deep learning models that are frequently utilized. Among the first deep learning models to accomplish object detection at a level comparable to humans were CNNs. CNNs have been utilized in the recent past for a variety of brain signal studies, including the detection of P300, mental task recognition, emotion recognition, and seizure detection. For instance, the four separate BCI paradigms of visual-evoked potentials (VEP), error-related negativity (ERN) responses, the model's ability to classify data was evaluated as show in Figure 3 [8].

$$y(i) = \sigma(w * x[i-k, \dots.; i+k] + b) \tag{1}$$

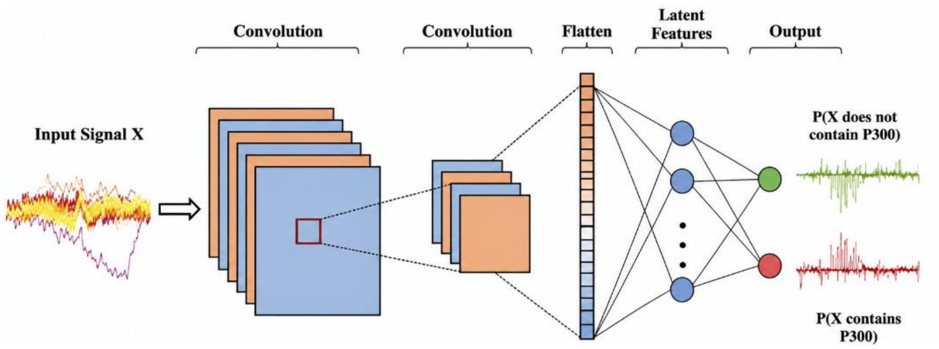



Figure 3. CNN [8]

### 2.2. RNN

RNNs can describe sequential data and create context-aware predictions by feeding previous outputs into current inputs. They perform well for time-series applications and can be enhanced with convolutional layers. But training is challenging due to vanishing and expanding gradients, which led to the development of LSTM networks for improved performance on long-term dependencies.

These networks solve issues like long-term dependencies and the vanishing gradient problem that come with conventional recurrent loops by utilizing specialized LSTM cells. The implementations of LSTM cells differ, but a simple model consists of a recurrent neuron equipped with a forget gate, which is a vector that decides what information in the cell should be kept or deleted [9], [10].

The LSTM architecture introduces gates to control data flow, allowing it to overcome challenges associated with vanishing and exploding gradients in RNNs. Each LSTM unit has two outputs: the main output and the unit's memory. The architecture uses gates to regulate data flow, facilitating the learning of long-term dependencies. A comparison between a classical RNN and an LSTM is depicted in Figure 4, emphasizing the complexity and effectiveness of LSTM with its four layers in the repeated module [11]. Inside the LSTM, the computations are defined by doors that allow or not the transmission of information [4], [10].

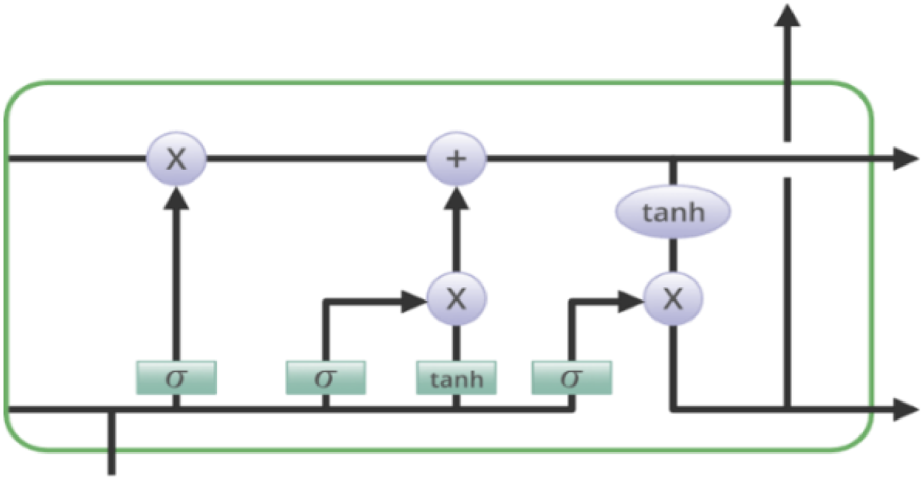



Figure 4. Diagram of an LSTM [12]

$$u_{(t)} = \sigma(w^u h(t-1) + I^u x(t) + b^u) \tag{2}$$

$$f_{(t)=}\sigma(w^f h(t-1) + I^f x(t) + b^f) \tag{3}$$

$$\hat{c}_{(t)} = tanh(w^c h(t-1) + I^c x(t) + b^c) \tag{4}$$

$$c_{(t)} = f_{(t)} \odot c_{(t-1)} + u_{(t)} \odot \hat{c}_{(t)} \tag{5}$$

$$o_t = \sigma(w^o h(t-1) + I^o x(t) + b^o) \tag{6}$$

$$h_{(t)} = o_{(t)} \odot tanh\, c(t) \tag{7}$$

$$y_{(t)} = softmax\left(W h_{(t)} + b\right) \tag{8}$$

The predicted character $\hat{c}$ in a single trial is computed by finding the character with maximal value of $f_{(t)}$ among the group of all the letters:

$$\hat{c} = argmax\{f\left(x_{(c)}\right)\}_{c\in C} \tag{9}$$

The formula for predicting the target stimuli is:

$$\hat{c} = argmax\left\{\frac{1}{R}\sum_{r\in R} f\left(x_{(c,r)}\right)\right\}_{c\in C} \tag{10}$$

In neural network, the weights are updated by deriving the loss function with respected to the network weights.

$$E = e\left(f\left(x_{(c,r)}\right), y_{r,c}\right) \tag{11}$$

Here $y_{r,c}$ refer to the true label of sample $x_{(c,r)}$, the binary log loss function:

$$e\left(f\left(x_{(c,r)}\right), y_{r,c}\right) = -\left(y_{r,c} log f\left(x_{(c,r)}\right) - \left(1 - y_{r,c}\right) log\left(1 - f\left(x_{(c,r)}\right)\right)\right) \tag{12}$$

The error in a neural network is typically calculated on multiple samples, as follows:

$$E = \frac{1}{M}\sum_i^M e\left(f(x_i, y_i)\right) \tag{13}$$

Here $M$ indicates the size of the batch of samples [10], [11].

### 2.3. SNN

A new breed of computer-simulated neural networks is represented by SNNs. In addition to trying to address some of the drawbacks of analog nets, such as their sluggish response times, high power consumption, or lack of asynchronous processing, SNNs are intended to more accurately simulate biological brain networks. Spiking networks required a modification to their structure due to the expectations that were created. As seen in Figure 5, spiking neurons, which are essentially distinct from the neurons found in ANNs, make up a spiking network. Spiking neurons lack an activation function, even if they communicate through synaptic connections as well. Rather, each neuron that fires has a membrane potential. Spike trains, or series of action potentials that change the membrane potential, are how neurons exchange and receive information. The neuron generates a spike, or stimulation, whenever a specific threshold voltage is crossed, and the membrane potential is then reset to a predetermined baseline. Another popular term for this occurrence is a neuron firing [13]–[19].

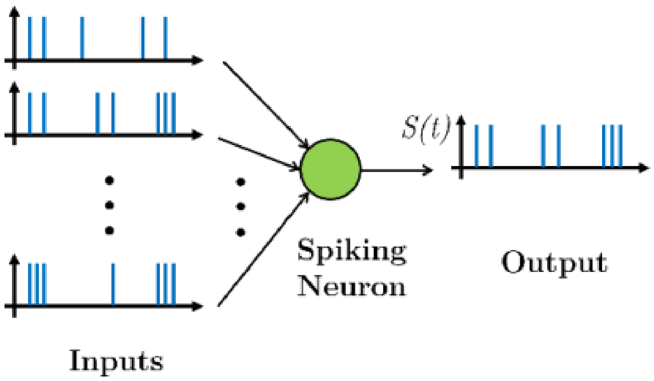



Figure 5. Visualization of a spiking neuron [14]

A leaky integrate-and-fire (LIF) neuron essentially accumulates (integrates) all of the inputs that it receives over time, discharges (leaks) exponentially constantly, and fires to release an action potential. The development of the spiking neuron model is essential to the realization of SNN. Neurons differ in their electrical and geometric properties. The spiking neuron model, as seen in Figure 5, is integrate-and-fire (IAF) model as depicted in Figure 6 [15] will be computed using the SNN and LIF method. The temporal dynamics are formulated:

$$\tau_m \frac{dv_{mem}}{dt} = -v_{mem} + w * \theta(t - t_k) \tag{14}$$

where: $v_{mem}$ = membrane potential and $\tau_m$ = time constant.
$w$ = gets modulated by the synaptic weight and $\theta(t - t_k)$ = incoming spike, a spike train can be described as:

$$S_{(t)} = \sum_f \delta_{(t-t^f)} \tag{15}$$

where $f = 1,2, \dots$
The softmax formula, sometimes called the normalized exponential [9] is given:

$$y_i = \frac{exp_{(a_i)}}{\sum_j exp_{(a_j)}} \tag{16}$$

LIF model are described by a single first-order linear differential equation:

$$\tau_m \frac{dv}{dt} = -(v - v_{rest}) + R_m I \tag{17}$$

$$\frac{dv}{dt} = -\frac{1}{\tau_m}(v - v_{rest}) + R_m I + S_{(t)}(v_{rest} - \theta) \tag{18}$$

The membrane time constant $\tau m$ satisfies [14].

$$\tau_m = C_m R_m \tag{19}$$

The loss function during training, which can be calculated as:

$$Loss\left(\left(x_{(i)}\right), y_{(i)}\right) = -\frac{1}{m}\left(y_i log\left(O_{(i)}\right) - (1 - y_i)\log\left(1 - \left(O_{(i)}\right)\right)\right) \tag{20}$$

The EEG can obtain the best recovery from the encoded spike trains by minimizing the error which is defined as (21).

$$error = \frac{\sum|eeg - eeg_r|}{\sum eeg} \tag{21}$$

Spike trains of the output layer are filtered by a double exponential synaptic filter $r_j$ for each neuron j as shown in (22), (23).

$$\dot{r_j} = -\frac{r_j}{\tau_d} + h_j \tag{22}$$

$$\dot{h_j} = -\frac{h_j}{\tau_r} + \frac{1}{\tau_d \tau_r}\sum_f \delta_{(t - t^f)} \tag{23}$$

As shown in (24), for a single channel *c*, where * indicates dot product multiplication.

$$EEG_c = \sum_j W_{cj} * r_j \tag{24}$$

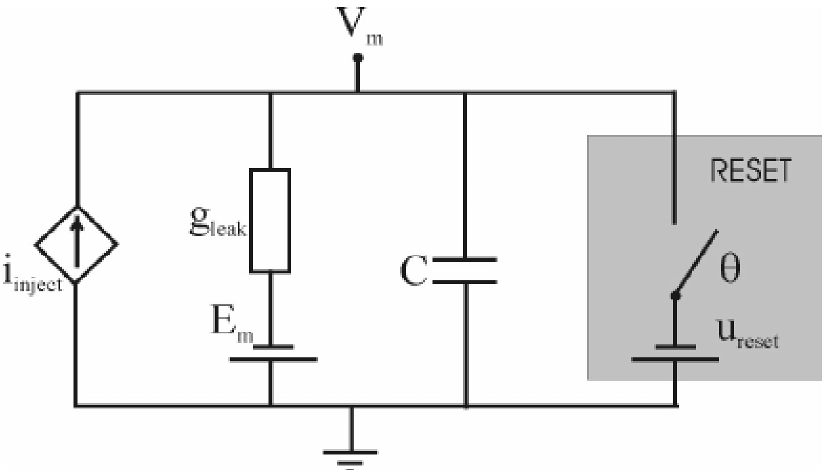



Figure 6. Equivalent circuit of constant-leak IAF model [19]

### 2.4. P300 potential in BCI

Allows users to utilize their brain signals to choose objects on a screen or spell words. The term “P300” designates a particular kind of ERP that happens about 300 ms after an individual notice a noteworthy stimulus, like a target letter or symbol [15], [16]. The general operation of a P300-based BCI speller task is as follows: the P300 speller system begins with EEG setup and calibration, where scalp electrodes record brain activity. A 6×6 matrix (36 characters) is displayed, and the user focuses on the desired character [19]. When the target flashes, a P300 response is generated and detected.

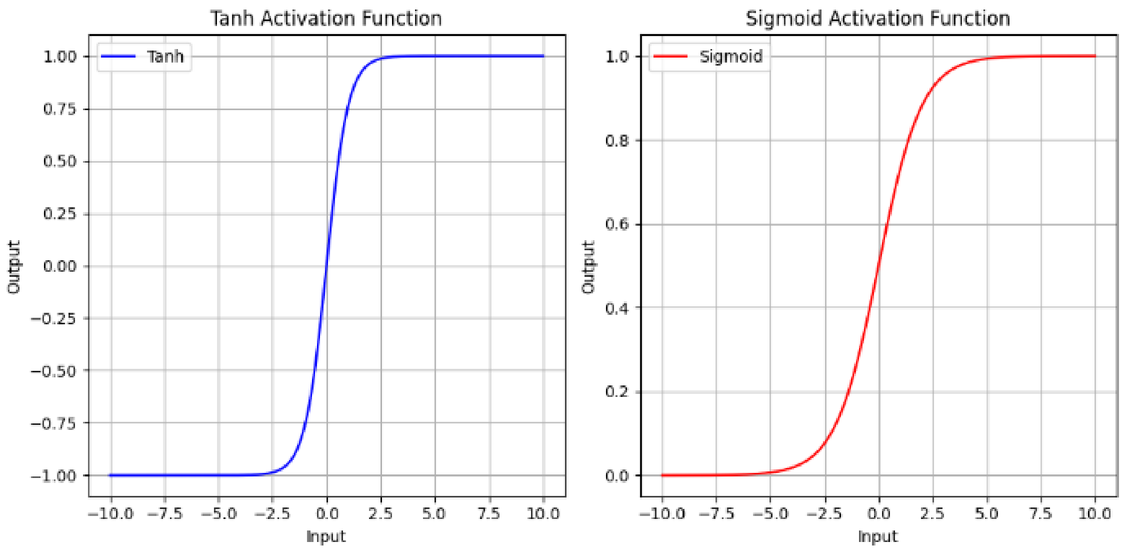
A classification algorithm identifies the selected character based on P300 responses, providing real-time feedback and error correction if needed [4], [13], [15], [16]. Electrodes are typically placed according to the International 10–20 system, and specific channels (F3, F4, C3, C4, P3, P4, O1, O2) have been shown to improve efficiency and reduce cost [10], [11], [13]–[19].

## 3. RESULTS AND DISCUSSION
### 3.1. Classification models

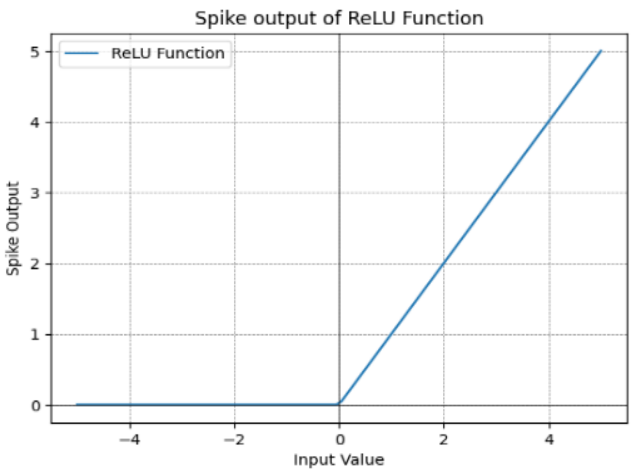
Our suggested architecture combines various ANN layer types, as follows: fully connected layer (FC), a layer is commonly referred to as such when every neuron in the input is coupled to every neuron in the output [16]. The following (25) yields the output in an FC layer:

$$y = \sigma (xW + b) \tag{25}$$

$\sigma (\cdot)$ denotes an element-wise non-linearity, such as hyperbolic tangent (TanH), sigomid, or rectified linear unit (ReLU) as show in Figures 7(a) and (b).




(a)




(b)

Figure 7. Activation function: (a) TanH-sigmoid activation function and (b) ReLU function

### 3.2. Resources for simulating

The second section reviews tools for modeling and simulating various neural networks. Several powerful frameworks are available, mostly based on Python due to its simple syntax and compatibility with C/C++. Examples include TensorFlow, Keras, NEURON, Nengo, Python neural networks (PyNN), Brian, Bindings for neural networks (BindsNET), and neural simulation tool (NEST). These libraries are also used for error evaluation, graph extraction, and neural network simulation [20].

### 3.3. Evaluated models
#### 3.3.1. CNN models

A CNN layer captures local correlations between adjacent input cells, producing multi-dimensional feature maps using trainable kernels. A one-dimensional (1D) CNN example is shown in Figure 8 [9], [11], [14]. The first layer has 10 spatial filters (55×1), the second layer has 13 temporal filters (1×5) processing five consecutive time points without overlap. The third and fourth layers are fully connected, ending with a single sigmoid-activated cell that outputs a scalar.

#### 3.3.2. LSTM model

LSTM made up of 100 and 30 hidden cells, respectively, in each of its single LSTM layers. Both models culminate in a single cell that produces a scalar and has a sigmoid activation layer, as show in Figure 9 [9]–[11]: (i) each subject's data is used separately for training and testing in order to look at intra-subject differences; (ii) examining the effects of additional data by combining training and testing data from different subjects; and (iii) transfer learning: teaching all but one subject. This experiment aims to explore the advantages of adapting a model that was trained offline on several subjects to a new subject, either with or without further calibration [18], [21].

#### 3.3.3. SNN model

At the neuronal level, brain dynamics are modeled mathematically, but simulating and training SNNs with many synapses is challenging due to high event rates [19], [20]. A feed forward SNN with LIF neurons was designed to generate EEG signals. Since the true spike train for P300 is unavailable, a Poisson spike train at 10 Hz serves as input, followed by one hidden and one output layer. Layer sizes vary per P300. Output spikes are converted to rate signals via a double exponential synaptic filter and weighted to produce EEG [20]. All weight matrices, except perturbation layers, were trained, while the perturbation layer of Poisson neurons models background noise, as show in Figure 10 [19]–[23].

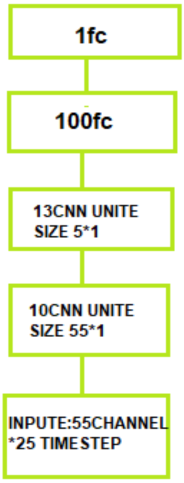



Figure 8. CNN model

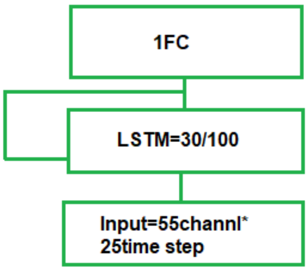



Figure 9. LSTM model

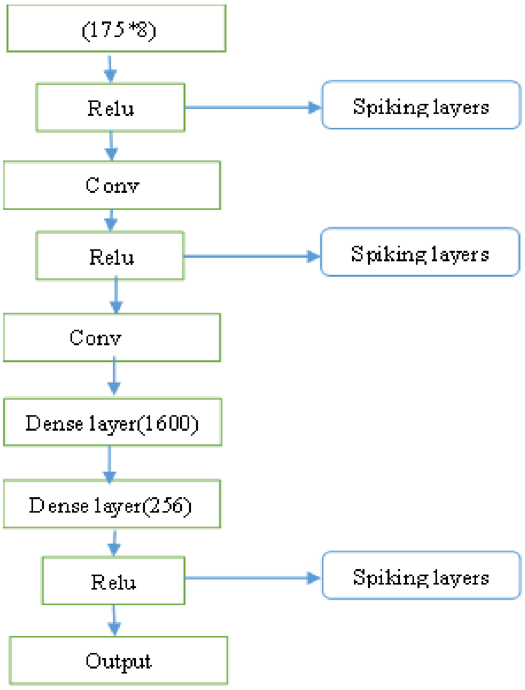



Figure 10. SNN model

### 3.4. Description of experimental data

Deep learning models, including SNN and LSTM, were used to classify P300 data obtained from Kaggle. The dataset was generated using the conventional 6×6 Donchin and Farewell P300 speller matrix. Each of the seven words contained five letters, and the matrix was stimulated 120 times per letter (10 times per row and column). The goal was to decode four words from the remaining twenty letters. A 0.1–30 Hz band-pass filter and a 50 Hz notch filter were applied, and data were collected from eight subject (Table 1) [19], [23].

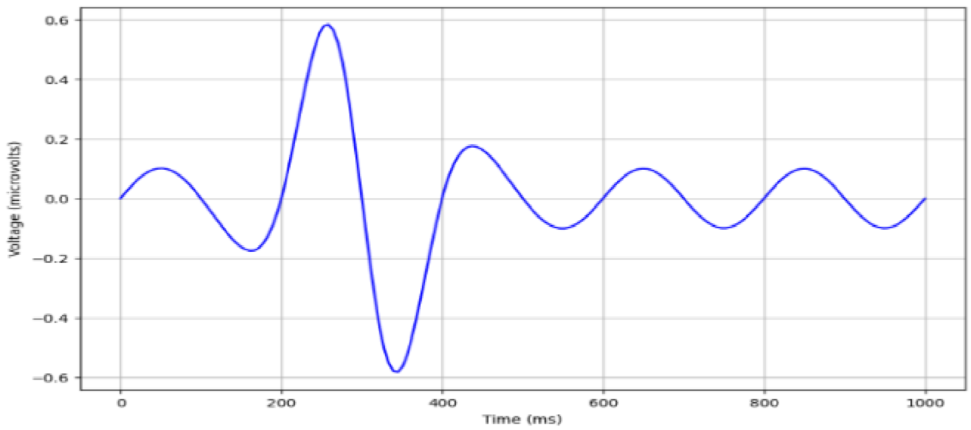
Table 1. Basic information of subjects

| Subject | Age | Gender |
|---|---|---|
| 1 | 56 | Male |
| 2 | 21 | Male |
| 3 | 23 | Female |
| 4 | 29 | Male |
| 5 | 20 | Female |
| 6 | 24 | Male |
| 7 | 29 | Male |
| 8 | 30 | Male |

### 3.5. Data preprocessing

The code was implemented using Keras. The dataset was divided into two classes: target P300 and non-target. Each sample contains 55 channels with 175 sampling points per channel. After preprocessing, the training set included 2810 non-target samples and 548 target P300 samples. To evaluate the model, fivefold cross-validation was applied. Five subsets of the training data were created. Four subsets were utilized for training and one was used for validation in each iteration. The final evaluation metric was calculated as the average accuracy across the five folds [19], [22], [24]. The P300 potential is identified by a positive peak occurring around 300 ms after stimulus onset, as shown in Figures 11(a) and (b). It is elicited when an individual detects an infrequent but expected stimulus among frequent ones. In two-class classification tasks, rare stimuli generate a P300 response in the EEG signal. This framework is known as the oddball paradigm and has been extensively studied [15], [19], [25].


(a)




(b)

Figure 11. Wave characterization: (a) P300 potential diagram and (b) postive peak

### 3.6. Results

The models' predicted actions are first displayed in these Figures 12 and 13. The models' accuracy in identifying the needed letter is then displayed in the Tables 2 to 4 that follow, along with a loss percentage calculation for each model. The membrane dynamics and deformation of the constant-leak IAF model are shown in Figure 14. Tables 2 to 4 present the subject-wise classification performance of the CNN, LSTM, and SNN models in terms of loss and accuracy. The results demonstrate variability across subjects, which is expected due to inter-subject differences in EEG patterns and the non-stationary nature of brain signals.

The CNN model achieved consistently high classification accuracy across most subjects, with subjects 1 and 8 obtaining the highest performance and low loss values, indicating stable training and effective spatial feature extraction. Table 3 shows that the LSTM model also achieved competitive results by capturing temporal dependencies in EEG signals.

The SNN model achieved strong subject-wise performance, with accuracy comparable to CNN for several subjects and relatively low loss values, confirming the effectiveness of spike-based temporal encoding for P300 signals. Among the models, CNN achieved the highest average accuracy (81.04%), followed by SNN (80.94%) and LSTM (80.60%). Although CNN slightly outperformed the others, the gap between CNN and SNN is minimal these results show that SNNs balance accuracy and efficiency, providing competitive performance for real-time, low-power BCI applications.

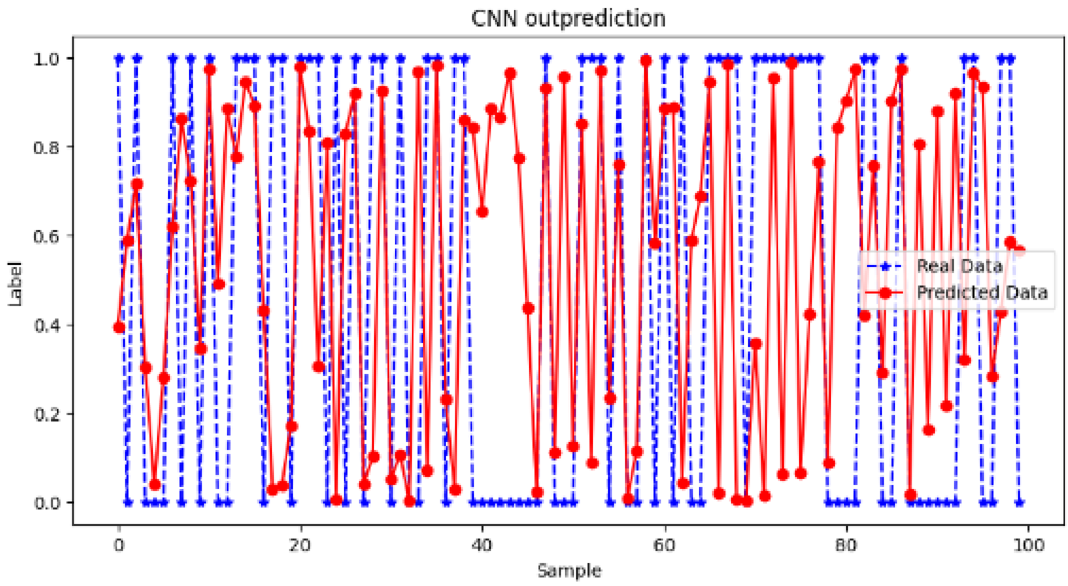
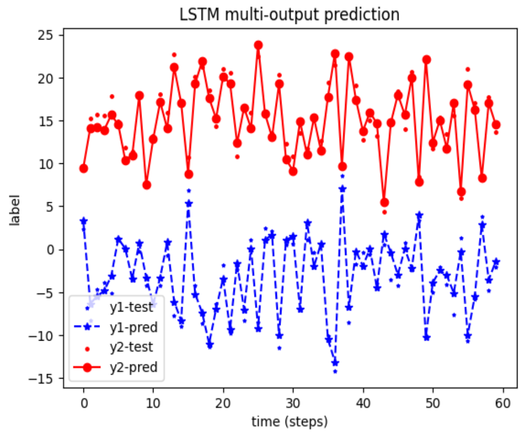



Figure 12. CNN predication



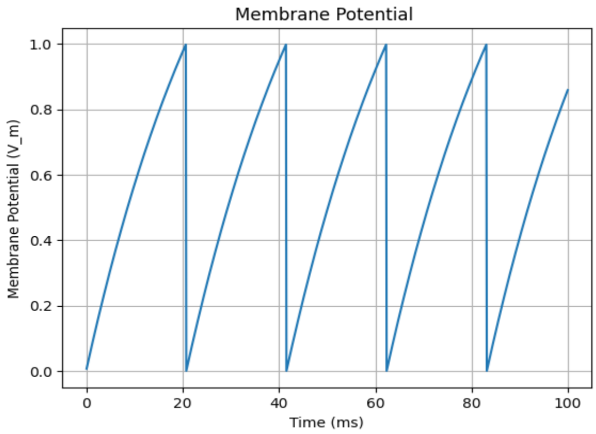




Figure 13. LSTM prediction

Figure 14. Constant-leak IAF membrane dynamics

Table 2. Average accuracy per subject (CNN model)

| Subject | Loss | Accuracy |
|---|---|---|
| 1 | 0.333 | 0.8437 |
| 2 | 0.4238 | 0.7762 |
| 3 | 0.4038 | 0.7894 |
| 4 | 0.3550 | 0.8251 |
| 5 | 0.3435 | 0.7985 |
| 6 | 0.3400 | 0.7893 |
| 7 | 0.332 | 0.8258 |
| 8 | 0.3060 | 0.8355 |

Table 3. Average accuracy per subject (LSTM model)

| Subject | Loss | Accuracy |
|---|---|---|
| 1 | 0.4038 | 0.8179 |
| 2 | 0.3885 | 0.8050 |
| 3 | 0.3722 | 0.7900 |
| 4 | 0.3595 | 0.8147 |
| 5 | 0.3456 | 0.7961 |
| 6 | 0.3442 | 0.7969 |
| 7 | 0.3459 | 0.8136 |
| 8 | 0.3459 | 0.8136 |

Table 4. Average accuracy per subject (SNN model)

| Subject | Loss | Accuracy |
|---|---|---|
| 1 | 0.3057 | 0.8524 |
| 2 | 0.4099 | 0.7635 |
| 3 | 0.3435 | 0.7826 |
| 4 | 0.3330 | 0.8311 |
| 5 | 0.4020 | 0.7846 |
| 6 | 0.3459 | 0.7912 |
| 7 | 0.3420 | 0.8302 |
| 8 | 0.3061 | 0.8391 |

## 4. CONCLUSION

This study presented a comparative evaluation of three deep learning architectures CNN, LSTM, and SNN for P300-based BCI classification. The experimental results demonstrate that all three models achieved competitive performance, with CNN attaining the highest average accuracy of 81.04%, followed closely by SNN at 80.94% and LSTM at 80.60%. These findings indicate that while CNN slightly outperforms the other models in terms of classification accuracy, the performance differences remain marginal. Despite their temporal modeling capabilities, neither LSTM nor SNN showed a decisive accuracy advantage over CNN under the investigated experimental conditions. However, the comparable performance of SNN highlights its effectiveness in P300 classification tasks and underscores its potential as an alternative deep learning paradigm. In particular, SNNs offer advantages in terms of temporal precision and energy efficiency, which are desirable for real-time and low-power BCI applications. It should be noted that the low-power benefits of SNNs cannot be fully realized under the conventional Von Neumann computing architecture, suggesting the need for neuromorphic hardware to exploit their full potential.

The results also indicate that model performance is highly dependent on data quantity and quality. As the size of the training dataset increases, the effectiveness of deep learning models improves, particularly in cross-subject learning scenarios. This observation confirms the suitability of deep neural networks for handling inter-subject variability in EEG-based BCI systems. Future work: will extend this study by exploring transfer learning for cross-subject generalization, hybrid CNN-SNN architectures for combined spatial and temporal processing, and testing on larger, diverse datasets, including clinical populations. Implementation of SNNs on neuromorphic platforms will also be evaluated for real-time performance and energy efficiency in practical BCIs.


## ACKNOWLEDGMENTS

The authors express their gratitude to everyone who provided personal assistance, insightful comments, and technical feedback that helped improve the quality of this work.

## FUNDING INFORMATION

The authors state that no funding was involved in the support of this research work.


## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ahlaam M. Saed | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | | |
| Ibtihal Fawzi Elshami | ✓ | | | | ✓ | ✓ | | | | ✓ | | ✓ | ✓ | ✓ |
| Ali I. Elgayar | ✓ | | | | | | | | | ✓ | | ✓ | ✓ | ✓ |

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

## CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

## DATA AVAILABILITY

The dataset utilized in this study is publicly available from the Kaggle repository. The underlying simulation codes and processed data supporting the findings are available from the corresponding author upon reasonable request.

## REFERENCES

[1] J. M. C. Delgado, D. Achanccaray, E. R. Villota, and S. Chevallier, "Riemann-Based Algorithms Assessment for Single- And Multiple-Trial P300 Classification in Non-Optimal Environments," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 2754–2761, 2020, doi: 10.1109/TNSRE.2020.3043418.
[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
[3] K. Värbu, N. Muhammad, and Y. Muhammad, "Past, Present, and Future of EEG-Based BCI Applications," *Sensors*, vol. 22, no. 9, p. 3331, Apr. 2022, doi: 10.3390/s22093331.
[4] P. W. Ferrez and J. del R. Millán, "Error-Related EEG Potentials in Brain-Computer Interfaces," in *Toward Brain-Computer Interfacing*, G. Dornhege, J. del R. Millán, T. Hinterberger, D. McFarland, and K.-R. Müller, Eds., Cambridge, MA: MIT Press, 2007, pp. 291–302. doi: 10.7551/mitpress/7493.003.0022.
[5] WikiStat, *Neural Networks and Introduction to Deep Learning*, 2017. [Online]. Available: https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-hdstat-rnn-deep-learning.pdf
[6] P. P. Nanda, S. Sethi, and A. Rout, "A study on EEG in brain computer interface," in *Proceedings National Conference on Next Generation Computing and its Applications in Computer Science and Technology (NGCAST)*, no. 1, New York, NY, USA: Foundation of Computer Science, Aug. 2017, pp. 6–10.
[7] Y. Zhou, S. He, Q. Huang, and Y. Li, "A Hybrid Asynchronous Brain-Computer Interface Combining SSVEP and EOG Signals," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 10, pp. 2881–2892, 2020, doi: 10.1109/TBME.2020.2972747.
[8] R. K. Maddula, "Going Deeper with Recurrent Convolutional Neural Networks for ClassifyingP300 BCI Signals.," 2017. [Online]. Available: http://www.escholarship.org/uc/item/30c012pr
[9] O. Tal and D. Friedman, "Recurrent Neural Networks for P300-based BCI," 2019, [Online]. Available: http://arxiv.org/abs/1901.10798
[10] Ahlaam. M. Saed, I. F. Elshami, and A. Almijbari, "Long Short-Term Memory Employed for Classifying P300-Based BCI," *2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering, MI-STA 2024 - Proceeding*, pp. 666–671, 2024, doi: 10.1109/MI-STA61267.2024.10599751.
[11] O. Tal, "Using Recurrent Neural Networks for P300-based BCI," p. 44, 2017.
[12] J. Pan, C. Shao, Y. Dai, Y. Wei, W. Chen, and Z. Lin, "Research on Fault Prediction Method of Elevator Door System Based on Transfer Learning," *Sensors*, vol. 24, no. 7, 2024, doi: 10.3390/s24072135.
[13] G. Hinton et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, Nov. 2012, doi: 10.1109/MSP.2012.2205597.
[14] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Direct Training for Spiking Neural Networks: Faster, Larger, Better," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 1311–1318, 2018,
[15] J. J. Podmore, "Classification of Frequency and Phase Encoded Steady State Visual Evoked Potentials for Brain Computer Interface Speller Applications using Convolutional Neural Networks," Durham University, 2018.
[16] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: A systematic review," *Journal of Neural Engineering*, vol. 16, no. 5, p. 051001, 2019, doi: 10.1088/1741-2552/ab260c.
[17] V. Alvarez and A. O. Rossetti, "Electroencephalography and evoked potentials: Technical background," *Clinical Neurophysiology in Disorders of Consciousness: Brain Function Monitoring in the ICU and Beyond*, pp. 7–24, 2015, doi: 10.1007/978-3-7091-1634-0_2.
[18] D. Structures, I. Preparation, and T. Practice, "Introduction to Recurrent Neural Network Start Your Coding Journey Now !," pp. 1–8, 2022.
[19] K. Wang, D. H. Zhai, and S. Xu, "A Shallow Spiking Neural Network for P300 Classification," in *Proceeding - 2021 China Automation Congress, CAC 2021*, IEEE, Oct. 2021, pp. 5963–5968. doi: 10.1109/CAC53003.2021.9728063.
[20] A. Novák, "Use of Spiking Neural Networks," University of West Bohemia, 2021.
[21] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization," 2015, doi: http://arxiv.org/abs/1409.2329.
[22] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, Mar. 2019, doi: 10.1016/j.neunet.2018.12.002.
[23] J. Tang, J. Lai, X. Xie, L. Yang, and W.-S. Zheng, "SNN2ANN: A Fast and Memory-Efficient Training Framework for Spiking Neural Networks," *Pattern Recognition*, vol. 144, p. 109826, Jun. 2022, doi: 10.1016/j.patcog.2023.109826.
[24] H. Paugam-Moisy and S. Bohte, "Computing with spiking neuron networks," in *Handbook of Natural Computing*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 335–376. doi: 10.1007/978-3-540-92910-9_10.
[25] A.-M. Brouwer and J. B. F. van Erp, "A tactile P300 brain-computer interface," *Frontiers in Neuroscience*, vol. 4, art. 19, May 2010, doi: 10.3389/fnins.2010.00019.

## BIOGRAPHIES OF AUTHORS

**Ahlaam M. Saed** received the B.Eng. degree in Electrical and Electronic Engineering (Telecommunications) from the University of Benghazi, Benghazi, Libya, in 2019, graduating among the top students in her class. She obtained the M.Sc. degree in Artificial Intelligence from the same university in 2023, where her thesis ranked first and was awarded the Innovation Prize for Best Scientific Research. She is currently a faculty member at Bright Star University and the Libyan International University. Her research interests include artificial intelligence, brain computer interfaces, and intelligent communication systems. She is also passionate about empowering women in STEM fields and advancing research in AI-driven communication technologies. She can be contacted at email: Ahlaam.saed@bsu.edu.

**Ibtihal Fawzi Elshami** received the Ph.D. degree in Wireless Telecommunication Engineering from the University of Technology Malaysia (UTM), Johor Bahru, Malaysia, in 2016, with a dissertation focused on radio wave propagation modeling in tropical environments. She is currently an Associate Professor in the Department of Electrical and Electronics Engineering at the College of Electrical and Electronics Technology (CEET), Benghazi, Libya. She supervises Master's and Ph.D. students and has served on academic and scientific committees. Her research interests include wireless communication systems, electromagnetic compatibility (EMC), brain-computer interface (BCI), and the application of deep learning techniques in biomedical signal processing. She has served on the scientific committees of international conferences such as the International Conference on Electrical Engineering and Information Technology (ICEEIT) and the IEEE Maghreb Conference in Engineering and Technology. She can be contacted at email: ifelshami@ceet.edu.ly.

**Ali I. Elgayar** is an Associate Professor in the Department of Electrical Engineering at the Faculty of Engineering, University of Benghazi, Libya. He has over 14 years of academic and research experience in the field of electrical engineering. His research interests include power systems, electrical machines, renewable energy integration, and smart grid technologies. He is a member of the Arab-German Young Academy of Sciences and Humanities (AGYA), where he actively collaborates on interdisciplinary research and science policy initiatives. He has published numerous research papers in peerreviewed journals and conferences and is committed to advancing engineering education and research in the region. He also supervises both undergraduate and postgraduate students. He can be contacted at email: ali.elgayar@uob.edu.ly.