Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: https://www.science-gate.com/IJAAS.html

CNN–LSTM-based EEG epileptic seizure detection: A hardware–software co-design approach

Nousheen Akhtar [1], Abdul Rehman Buzdar [2,*], Jiancun Fan [1]

[1]*School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an, China*
[2]*Department of Computer Engineering, National University of Technology, Islamabad, Pakistan*

ARTICLE INFO

Article history:
Received 4 August 2025
Received in revised form
16 February 2026
Accepted 5 March 2026

Keywords:
EEG
FPGA
Epileptic seizure detection
CNN-LSTM
Zynq SoC

ABSTRACT

Epileptic seizure detection from electroencephalogram (EEG) signals is important for early clinical intervention in patients with neurological disorders. This study presents a hardware–software co-design implementation of a hybrid Convolutional Neural Network–Long Short-Term Memory (CNN–LSTM) model for automatic EEG classification on the Xilinx Zynq-7000 System-on-Chip (SoC) platform. In the proposed architecture, the CNN layers, which extract spatial features, are executed on the ARM Cortex-A9 processors, while the LSTM and fully connected layers are implemented in the FPGA fabric to enable real-time inference. The model was trained and tested on the Bonn University EEG dataset for three-class classification: normal, interictal, and seizure (ictal) states. The proposed system achieves a classification accuracy of 99.33% with an inference latency of 0.657 seconds per EEG segment. Hardware-oriented optimizations, including fixed-point quantization and efficient approximations of activation functions, reduce power consumption and increase processing speed. Compared with a full software implementation, the proposed co-design provides a 12.5× reduction in execution time. These results demonstrate that deep learning–based seizure detection can be effectively deployed on resource-limited embedded platforms for real-time medical applications.

1. Introduction

Epilepsy is among the most common neurological diseases after migraines, with a prevalence of about 50 million people around the world. According to the WHO, up to 10% of people will have at least one seizure in their lifetime (WHO, 2019), which ultimately brings an enormous burden to health treatment systems worldwide. Electroencephalogram (EEG) signals are also very important as they describe the electrical activities of the brain and offer a primary detection tool for diagnosing epilepsy. Manual signal reading from EEG by neurologists is very slow (Ein Shoka et al., 2023), subjective, and prone to human errors, especially when dealing with long-term monitoring data that may take hours and days to complete. On many occasions, even a whole day will not be enough for experts to visualize and analyze reports on patients (Tatum et al., 2018), hence requiring secondary neurology experts' involvement in evaluating the process.

Recent advancements in deep learning have brought forth encouraging outcomes (Dash et al., 2024) in the automatic interpretation of EEG data as well as epilepsy detection. It has been found that Convolutional Neural Networks (CNNs) perform very well at extracting spatial features from EEG signals, and RNNs perform even better when properly optimized to extract temporal dependencies within sequential data. The popular RNN architecture is known as Long Short-Term Memory (LSTM). By mixing these two structures into one hybrid CNN-LSTM architecture (Chaturvedi, 2023), both spatial and temporal features of the EEG signal can be exploited to raise the bar for classification accuracy.

Though deep learning models have made breakthroughs in EEG classification, practically applying them to a real-time medical monitoring system is not easy. In most cases, software implementations on general-purpose processors do not provide computational efficiency for real-time processing. As an alternative, cloud-based solutions

* Corresponding Author.
Email Address: abdul.rehman@nutech.edu.pk (A. R. Buzdar)
https://doi.org/10.21833/ijaas.2026.03.008
Corresponding author's ORCID profile:
https://orcid.org/0009-0005-2460-5797

bring up the issue of data privacy and concerns about network connectivity. Such raw signals sent through smartphones or wearable devices via the internet to the hospitals for diagnosis require high power consumption and reliance on internet connectivity, thereby defeating the very purpose of low-power automation (Dabbaghian and Kassiri, 2025). Hardware acceleration is highly attractive by using Field-Programmable Gate Arrays (FPGAs) since it opens the possibility for energy-efficient as well as high-performance computing customizable platforms (Muneeb and Kassiri, 2024) for medical applications.

Machine Learning (ML) represents a major advance over traditional visual inspection of EEG recordings for detecting epileptic seizures. The installation of ML models allows speedy and very precise treatment of EEG information through automation in extracting features, recognizing patterns, and classification. Unlike handcrafted rules and manual interpretation, ML models can specialize in fitting the complicated statistical properties of EEG signals across different patients and generalize so that even subtle changes in EEG patterns related to impending seizures can be detected for timely patient intervention. Changes in ML technology have served to shift clinical practice significantly through the creation of predictive modeling plus real-time monitoring, taking further steps than retrospective diagnosis (Dash et al., 2024).

A wide range of ML approaches has been applied to EEG data for epilepsy research. Some of these are supervised learning classifiers like SVM, RF, k-NN, and gradient boosting models. Other deep learning models include CNNs and LSTMs. The goal of supervised learning is to build a model that can learn from labeled EEG data to correctly differentiate between seizure and non-seizure states-that is essentially the main task for most algorithms for detecting seizures. Specifically, deep learning models expand on this by permitting automatic extraction of hierarchical feature representations directly from raw EEG input data, reducing dependency on manual feature engineering, hence attaining even better performance for classification tasks (Chaturvedi, 2023) in several studies.

The distinction between seizure and non-seizure EEG segments lies at the very heart of classification for detecting epileptic seizures. The performance of ML models relies heavily on both the quality of input features and support for temporal dynamics related to manifestations of seizures provided by the model architecture itself. Many works have highlighted the importance of this binary classification within a continuous monitoring system whose main goal would be to provide real-time alarms about detected seizures so that clinical intervention could take place in time. Just making an explicit distinction between seizure and non-seizure states serves in most epilepsy detection algorithms as the benchmark task on which more advanced tasks (Goel, 2025), such as type classification or even prediction, are built. Feature extraction is a very important step of preprocessing in the ML classification pipeline for EEG-based seizure detection. Raw EEG signals are not only highly dimensional but carry huge levels of noise; hence, extraction of salient and discriminative features is necessary so that ML models can focus on relevant characteristics of the signal and reduce the complexity considerably. Some major types of features include time-domain, frequency-domain, and time-frequency domain features. They represent different aspects of EEG signals. The mean, standard deviation, and variance are examples of time-domain features that give statistics about fluctuations in amplitude over the signal and can highlight abnormal activities during periods of seizures. Frequency-domain features describe the spectrum composition and energy distribution for various standard bands; their difference under normal conditions and ictal conditions has been proven as significant (Sadangi et al., 2024). The combination of time domain and frequency domain feature sets mostly provides better classification results since information about temporal dynamics as well as spectral content is available (Saini and Chandel, 2024).

Time-frequency domain approaches have lately been considered very strong for the analysis of non-stationary EEG signals during epileptic seizures. Among all pre-processing techniques, the Discrete Wavelet Transform (DWT) has probably been most widely used since it allows decomposition of the EEG signal into different frequency sub-bands while maintaining time resolution. Multi-level DWT decomposition supports the endeavor to attain transient features related to seizure activities within frequency bands such as delta, theta, alpha, beta, and gamma rhythms. There are myriad works that have implemented DWT to extract features from the above, implying better performance in classification regarding seizures. They have also incorporated statistical feature extraction with DWT for enhancing robustness in modeling. For instance, advanced hybrid models improved classification accuracy using DWT-decomposed EEG signals by exploiting both time-frequency features and ML algorithms (Cao et al., 2025).

The Fast Fourier Transform has been one of the classical frequency domain methods usually applied to EEG signals for deriving spectral features related to seizure activity. In essence, it transforms EEG signals from their original time representation into frequency representation so that frequency band powers and other spectral characteristics that are very useful in describing abnormal neuronal discharges can be determined. FFT-based features integrated with time-domain features make very comprehensive representations that enhance the capability of any classification algorithm toward differentiating between seizure and non-seizure states (Ahmad et al., 2024).

Also, through FFT, those frequency bands most often affected during an epileptic event can be determined; hence, interpretable information is provided to clinicians.

2. Literature review

The hardware implementation of an epileptic seizure detection system is very crucial towards achieving real-time applications (Dabbaghian and Kassiri, 2025), and FPGAs have so far gained much consideration as a platform for such systems because of their configurability, parallelism, and power efficiency. Literature provided various FPGA platforms that were used to implement the epileptic seizure detection system. In one of the implementations, a neural network was synthesized by high-level synthesis (HLS) and deployed on an FPGA, which is part of the Pynq-Z2 board having a ZYNQ SoC (Shanmugam and Dharmar, 2025; Alam et al., 2024a; Sajja and Rooban, 2024). Some platforms found frequently in the literature include high-level synthesis on Xilinx Zynq-7000 and Artix-7 chips utilizing Xilinx System Generator (XSG) for digital signal processing (Jaffino et al., 2023; Meddah et al., 2020).

Architectural designs in literature vary greatly by complexity and implementation approach. In this regard, some authors presented a hardware implementation system based on a multi-layer perception (MLP) architecture (Zairi et al., 2022) for detecting epileptic seizure activities successfully implemented on an FPGA. Two-stage architectures have been adopted by certain systems to optimize the power consumption of the system, where the proposed algorithm extracts coastline, energy, and nonlinear energy features (Razi and Schmid, 2021) from iEEG signals in a patient-specific two-stage seizure detection framework. Notably, the detection stage of the proposed system, which extracts twice as many features as the monitoring stage, is active only when the monitoring stage has detected a seizure occurrence.

Dynamic partial reconfiguration has been used at the application level to improve resource usage as well as power consumption. A design that proposed a system of reconfiguration in nature for designing an epileptic seizure detection system accommodates several feature extractors followed by a classifier. These feature extractors use DPR (Khan and da Silva, 2024) such that there will be no leakage power from different parts of the extractors when they are not in use, hence reducing resource consumption by 0.68 and consequently reducing the corresponding leakage power. The DWT has been the most popular methodology of feature extraction in FPGA-based seizure detection systems. This paper is part of efforts by several others to come up with a model for designing an FPGA that would implement DWT and principal component analysis (PCA) (Meddah et al., 2020; Feng et al., 2017) toward determining the best parameters for support vector machines (SVMs) to classify EEG data for epileptic seizures. The proposed system would include a module for feature extraction using DWT and training using MSMO. A Daubechies order 4 wavelet was used in a three-level DWT configuration since it saves on circuit area and computational time.

Also, advanced wavelet approaches have been studied by presenting an optimized medical diagnostic approach based on wavelet transform using a very small set of features (Zairi et al., 2022; Rout et al., 2022) that properly classify the relevant EEG classes. The method proposed is designed and implemented within a high-speed reconfigurable FPGA embedded processor with reduced design area, fewer hardware resources used, and low power consumed. It addresses many VHDL simulation and real problems for chipscope FPGA implementation, such that an FPGA-based standalone computer-aided design (CAD) system has been created to implement the detection of epileptic seizure epochs with improved detection accuracy.

MLP neural networks have found rich applications on FPGAs toward seizure detection (Sarić et al., 2022; 2020). Hence, MLP, ANN-based FPGA solutions are scalable and portable systems of great importance for real-time epileptic seizure diagnostics, both clinically and non-clinically. This study presents a prototype implementation of a feed-forward MLP-ANN classifier implemented on an FPGA board that is practically employed to recognize different types of epileptic seizures in real time. The architecture and training of the MLP-ANN model are developed offline in the TensorFlow Python environment, with MATLAB scripting possible, while real-time FPGA implementation would be done directly by coding in VHDL.

Classifiers based on Extreme Learning Machine were also used due to their computational efficiency (Mulla et al., 2022; Indira Priyadarshini and Reddy, 2023). In the next steps, features extracted from the modeled EEG signal are classified as normal or seizure signals using the ELM classifier. The proposed system has effectively demonstrated detection of epileptic seizures using ELM classifiers on an FPGA, hence a system that accurately detects seizures.

The SVM was implemented (Alam et al., 2024b; Rafiammal et al., 2020) in an FPGA-based seizure detection system, which proved to be very accurate. The results indicated that the SVM classifier-based model produced the highest accuracy of 99.4% with a sensitivity of 98.8%, consuming the least dynamic power of only 0.057 mW and utilizing very minimal resources of the FPGA. A binary Linear Support Vector ML algorithm-based automatic seizure detection algorithm has been developed in an FPGA for this proposed design. Experimental results give a mean detection accuracy of 86% and a sensitivity of 97%.

Deep learning, particularly CNNs, has been used to increase accuracy in detection. Average classification accuracies obtained were 93.9% for CNN models and 97.2% for Bi-LSTM models on all frequency bands from both epileptic and non-epileptic patients. Semi-supervised reduced deep CNN-based implementation of the system on FPGA indicates that it is feasible to run quite advanced deep learning models on FPGA platforms (Beeraka et al., 2022; Sahani et al., 2021).

3. Hybrid CNN-LSTM architecture

The selected hybrid CNN-LSTM architecture, shown in Fig. 1, is intended for the joint spatial and temporal feature extraction of EEG signals toward the precise detection of epileptic seizures (Pandey et al., 2023). This multi-stage architecture provides hierarchical feature extraction from raw EEG inputs progressively. The network starts with an input layer that takes raw EEG signals having dimensions (4097, 1), a single-channel EEG segment. The size of the input is selected to ensure enough temporal information and, at the same time, maintain computational efficiency. Spatial characteristics are extracted through a series of 1D convolutional layers, followed by time pattern recognition through LSTM layers, and finally classification over fully connected layers. The input layer gets the EEG samples in digital, segmented form; this segmentation strategy ensures that every input carries relevant neural activity patterns with consistent input dimensions across different patients and recording conditions. The total network depth and width are aimed at balancing the power of feature extraction against computational efficiency, particularly within its hardware implementation constraints. The progressive spatial dimension reduction achieved through CNN layers, followed by temporal processing thereafter in LSTM layers, ensures an efficient information processing pipeline that will be able to pick not only minor morphological details but also major temporal trends, which are quintessential for accurate EEG classification.

3.1. CNN feature extraction architecture

The CNN part of this hybrid model has four one-dimensional convolutional layers built to pull out hierarchical spatial features from EEG signals, as shown in Fig. 1. The first convolutional layer (Conv1D Layer 1) runs the input through three filters with a kernel size of 27, changing the input shape from (4097, 1) to (4071, 3). A rather big kernel here is used to get those characteristic waveform patterns found in EEG signals, which often extend over several samples. These three filters capture different types of neural oscillations typically observed in EEG recordings. After the initial convolutional layer, max- pooling with a stride of 2 reduces dimensions to (2035,3), hence maintaining translation invariance and reducing the computation for the following layers. The second convolutional layer increases its number of filters to 10 by using a kernel size equal to 14, outputting (2022,10). This layer further elaborates on what was detected in the previous layer by creating more compound feature mixes that may be applied towards discrimination between different types of neural activity patterns. The third and fourth convolutional layers continue to build the hierarchical feature extraction. Conv1D Layer 3 also has 10 filters with a smaller kernel size of 3, focusing on even more local patterns; Conv1D Layer 4 reduces filters to just 4, with its kernel size at 4. Every convolutional layer is followed by max-pooling that reduces spatial dimensions further but keeps the most important features. This results in the final output from the CNN component having dimensions of (250,4), which offers a compact yet very informative representation of spatial characteristics contained in the original EEG signal.

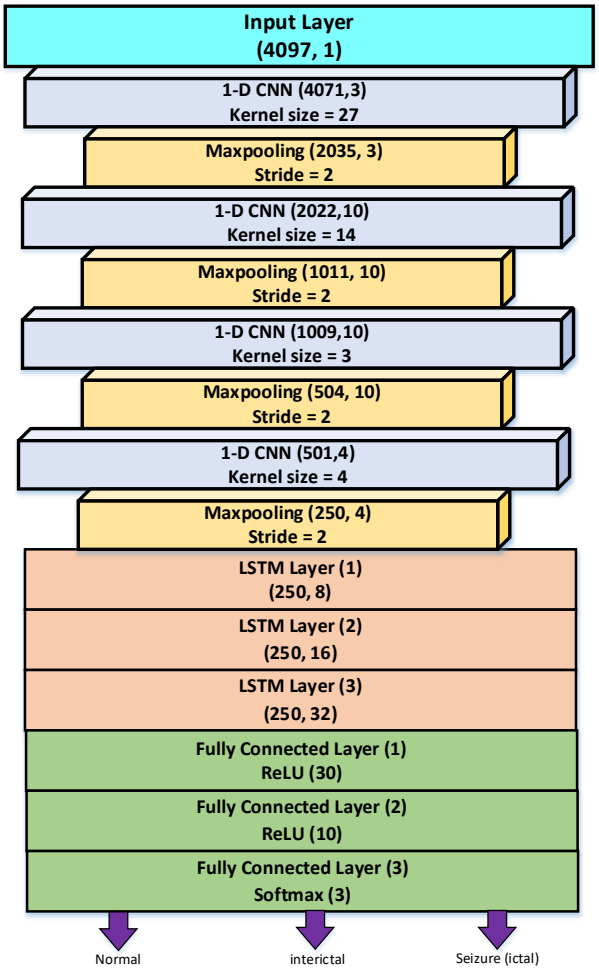

Fig. 1: Architecture of CNN-LSTM model for EEG signal classification

3.2. LSTM temporal processing

Recurrent Neural Networks (RNNs) usually learn short-term dependencies well for sequential data but do not perform well on long-term dependencies due to the vanishing gradient problem. The LSTM cell solves this by introducing memory cells and a gating mechanism, while performance can be improved using peephole LSTMs by allowing the gates direct access to the cell state. The Gated Recurrent Unit (GRU) simplifies LSTMs by merging some gates and states, which means even less complexity than a standard LSTM, having good performance on long sequences, especially in hardware-limited scenarios. The features extracted by the CNN are processed by LSTM so that temporal dependencies can be captured in those features for proper classification between baseline, interictal, and seizure (ictal) conditions in the EEG signals. Fig. 1 shows an LSTM

network with three sequential layers having increasing numbers of hidden units: 8, 16, and 32. Increasing capacity gradually allows the model to learn temporal patterns at different levels of complexity. The output of the CNN, which has a shape of (250,4), is read by the first LSTM layer and transformed into an output with a shape of (250,8); where 250 is the length of the time series, and 8 is the size of the hidden state. Mainly, it focuses on basic temporal patterns and short-term dependencies in the feature sequence. A not very large number of hidden units in this layer helps reduce overfitting and, at the same time, captures all necessary temporal patterns. The second LSTM layer increases to 16 hidden units, thereby improving the ability to recognize more complex temporal patterns. Medium-term dependencies and more detailed temporal relations between features that a CNN layer extracts can be realized by this layer. The third LSTM layer increases the capacity to 32 hidden units, therefore enabling the model to recognize long-term dependencies and complicated temporal dynamics that are found in different EEG states. Standard LSTM cell equations are used by each layer; remember the concept of forget gates, input gates, and output gates. The forget gate determines what information should be removed from the cell state, while the input gate determines what new information should be added. The output gate shares what aspects of the cell state should be output as the hidden state. Such a gating mechanism is highly important in EEG signal processing since the network must learn to keep relevant information about patterns of seizure onset and discard irrelevant background activity.

3.3. Classification layers

Three fully connected layers make up the final classification stage by processing the output from the LSTM network to class probabilities for a three-class classification problem. The first fully connected layer takes as input the output from the final LSTM layer and reduces it to 30 neurons using the ReLU activation function. This may be regarded as a feature consolidation stage by which temporal representations gained by the LSTM layers are brought into an even more compact form suitable for classification. This second fully connected layer brings the representation down to 10 neurons, again using ReLU. It allows more non-linear tricks that can make different EEG classes even easier to separate. The last fully connected layer gives an output of 3 neurons, one for each class: Normal, interictal, and Seizure (ictal) states. Here, the softmax activation function is used; it will output probabilities across the three classes so that confident decisions can be made about which state is present.

4. Hardware-software co-design implementation

The hardware-software co-design methodology refers to the concept of embedded and digital system development, such that hardware and software are not designed sequentially but rather in parallel. By functional partitioning of the system between the hardware component, for example, FPGA, ASIC, and software running on a processor, performance, cost, power consumption, and flexibility of design may be optimized (Buzdar et al., 2017a; 2017b; 2016). The hardware-software co-design is done using the Xilinx ZYNQ-7000 System on Chip that provides an ARM Cortex-A9-based Processing System (PS) and Programmable Logic (PL) inside the same chip. Our partitioning method assigns CNN operations to run on the ARM processors while LSTM and fully connected layers are implemented in FPGA hardware. This decision is made based on an analysis of the computational characteristics of all parts of the network versus what type of processing platform they best map onto. The HW-SW co-design hybrid CNN-LSTM inference setup using a SoC-FPGA platform is shown in Fig. 2.

The ARM Cortex-A9 processors found inside the PS are highly suitable for running CNNs because these types of workloads are rich in control, involve complex memory access patterns, and software libraries optimized for convolution operations are readily available. Even though CNN layers require a lot of computation, they have regular memory access patterns that can be effectively exploited by using the cache hierarchy as well as the advanced branch prediction capabilities provided by the ARM processors.

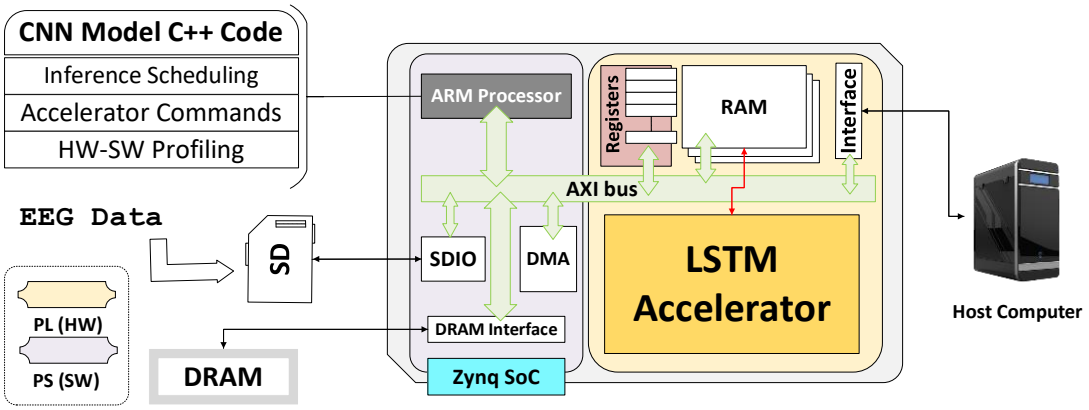

Fig. 2: HW-SW co-design hybrid CNN-LSTM inference setup using a SoC-FPGA platform

In addition, since the data types and arithmetic operations involved in CNN computations are relatively simple, they can be efficiently executed using the SIMD instructions provided on ARM Cortex-A9. LSTMs and fully connected layers have their implementation inside the FPGA fabric so that they can take advantage of deep parallelism and custom data paths typically available in programmable logic. LSTM computation requires many matrix-vector (multiply-accumulate) operations across several dimension-wise operations, which can be parallelized efficiently in hardware. The FPGA implementation supports custom precision arithmetic, optimized memory hierarchies, and special processing elements tuned particularly for the support of operations found in LSTM and fully connected layers. The Hybrid CNN-LSTM EEG classification system using the SW-HW co-design scheme is shown in Algorithm 1.

Algorithm 1. Hybrid CNN–LSTM EEG Classification using SW-HW Co-Design

1. Given: single-channel EEG segment of size 4097 × 1
2. Return: predicted class y ∈ {normal, interictal, seizure (ictal)}
3. Preprocess: apply z-score normalization
4. [ARM] CNN feature extraction: apply 4× (Conv1D → ReLU → MaxPool); obtain feature map of size 250 × 4
5. Transfer feature map to FPGA via AXI interface
6. [FPGA] Temporal modeling: three LSTM layers with hidden units [8, 16, 32]
7. [FPGA] Classification: three fully-connected layers with neurons [30, 10, 3] and activations (ReLU, ReLU, Hardmax)
8. Decision: y ← argmax output scores (Hardmax)
9. Output predicted class y via UART and update the on-board display

High-speed AXI interfaces manage data flow between the PS and PL, ensuring seamless, efficient transfer of intermediate results from CNN processing to LSTM hardware acceleration. In this architecture, the PS handles acquisition of EEG signals and preprocessing, besides feature extraction by a CNN and overall system control. The PL focuses on temporal processing implemented through LSTM layers, followed by final classification carried out via fully connected layers. This kind of partitioning minimizes any overhead associated with data transfers while maximizing the strength that can be brought to bear by each processing platform. The illustration of CNN-LSTM network inference operations using the SW-HW co-design scheme is shown in Fig. 3.

4.1. Hardware accelerator design

4.1.1. LSTM computational core architecture

The LSTM Accelerator is the computation engine for the hybrid CNN-LSTM architecture; it manages the temporal dependencies of the EEG signals. This hardware design adopts a time-shared computational approach whereby a reduced set of arithmetic units, mainly multipliers, adders, and accumulators, are sequentially reused to compute the four gates of an LSTM. This architectural selection drastically reduces resource utilization on a Zynq SoC but keeps real-time inference capability.

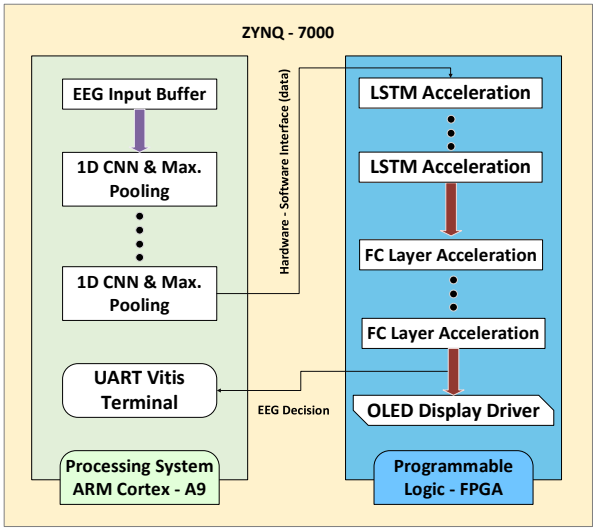

Fig. 3: Illustration of CNN-LSTM network inference operations using the SW-HW co-design scheme

The main computation for each gate (input, forget, candidate, and output) is based on matrix–vector multiplications with the use of an input vector x_t and a recurrent vector h_{t-1}. In this architecture, proposed as time-multiplexed, these operations do not take place simultaneously in parallel dedicated hardware blocks. Rather, the same MAC unit or even minimally replicated MAC units iteratively go through all required weight segments. Weight matrices for input and recurrent paths are stored in different, separate small memory banks depicted as W_x, W_h, X_t, and h_t memories shown in Fig. 4 to make sure there will be no conflict during sequential access. The architecture reuses the MAC hardware between gate computations by pipelining them under the control of a centralized FSM, weights loading orchestration, accumulation timing, bias addition, and activation function invocation.

The first MAC process computes the transformation for one gate by streaming in the appropriate weight vectors from memory and accumulating partial results. As soon as summation for a gate is completed, it immediately begins the next gate in sequence, reusing all arithmetic resources used previously. To maintain throughput, while the matrix-vector multiply of one gate is being performed by the MAC hardware, weight buffers for another gate are preloaded in the background by FSM control. Thus, memory latency can be effectively hidden, also fitting well with the time-shared philosophy of this design.

Once the forget, input, and candidate gates have been calculated, then a simple series of multiply-accumulate operations updated via time-multiplexing, as depicted in Fig. 5, can update the cell state. The new cell state c_t is obtained by reusing the same multipliers for elementwise operations

involving forget gate output, previous cell memory, input gate output, and candidate input.

The update of the hidden state requiring a tanh activation followed by an elementwise multiplication with the output gate is also performed using these shared computation units. By reusing the same hardware in sequence, the cell state c_t and hidden state h_t are computed efficiently without replicating processing units.

The proposed LSTM hardware design shares time, shares memory, and implements scheduling through hierarchy, thereby enabling an implementation compact enough to be characterized as energy efficient. The approach thereafter leads to the optimum difference between resource utilization and inference delay, hence making it highly attractive to application developers for temporal deep learning models on FPGAs.

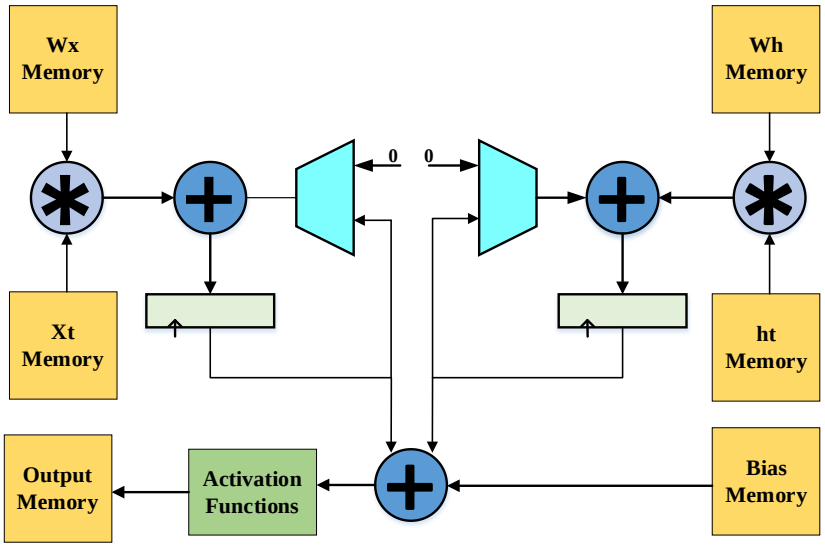

Fig. 4: The main hardware module that implements the LSTM gates

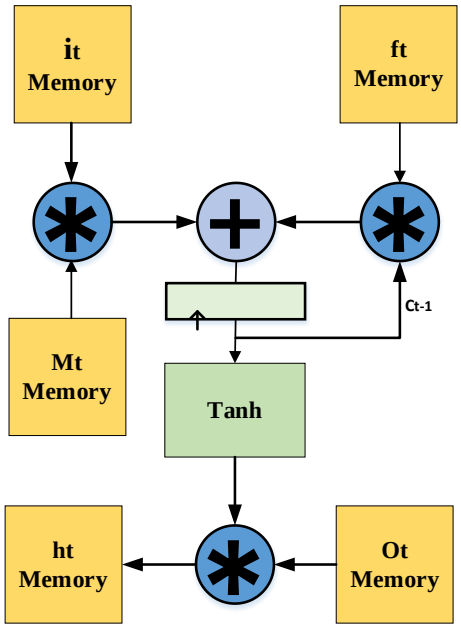

Fig. 5: The module that computes the c_t and h_t

4.1.2. Fully connected layer hardware implementation

The time-shared hardware architecture implements the fully connected layer (FC) layer in the proposed hybrid CNN-LSTM system and achieves high resource efficiency with real-time inference capability preserved. It does not adopt a classic parallel or systolic design with the replication of many processing elements, but shares a small computational core among neurons over time. The hardware implementation of the FC layer is shown in Fig. 6. LUTs, DSP slices, and BRAM consumption will also remain lightweight for this implementation on an FPGA, thereby making it the best fit for embedded low-power systems as well as mid-range FPGA devices.

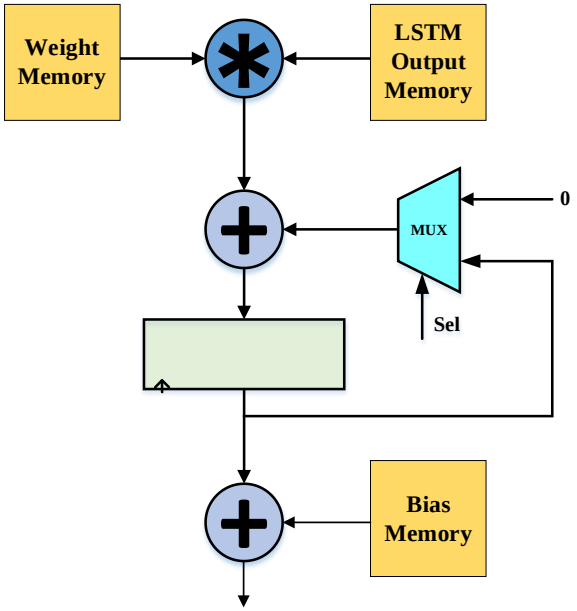

Fig. 6: The module that implements the FC Layer

In this design, the core computation matrix-vector multiplication is done by a single Multiply-Accumulate (MAC) unit or minimally replicated MAC

group in sequence with weights and input activations per neuron. For every neuron, the same MAC hardware iterates over the corresponding weight vector, accumulating partial sums until the neuron's output is fully computed. As soon as the accumulation for one neuron is completed, the hardware can be reused immediately for the next neuron. This kind of temporal reuse of computational units results in much less silicon overhead compared to spatially parallel implementations.

A deterministic control FSM is used for managing the scheduling of operations, weight indexing, partial sum initialization, and neuron-level transitions. Weights are read from a small weight buffer and streamed to the MAC unit, while the input feature vector resides in a local activation buffer so that repeated access does not create any memory bottleneck. Intermediate accumulation registers keep precision during MAC iterations, thus ensuring numerical stability throughout the computation flow.

4.1.3. Activation function hardware implementation

Activation functions play an important role in the performance and convergence characteristics of LSTM networks. The traditional activation functions used are the sigmoid function and the hyperbolic tangent (tanh) function, which require a lot of computation because they are nonlinear and transcendental by nature; hence, it is expensive to implement them directly in hardware. To overcome this problem, hardware-efficient approximations can be used. Hard sigmoid and hard tanh functions can be used as simplified piecewise linear alternatives, which reduce computational complexity but keep competitive accuracy possible.

These comprise simple linear segments approximating exponentials inside sigmoid and tanh functions, avoiding the heavy computational cost normally associated with exponentials, thus resulting in savings of FPGA resources and latency. The hard sigmoid results in a linear transform with a scale and bias added, whereas the hard tanh is saturation-based clipping for stable output value activation.

4.1.4. Top control state machine design

The hardware accelerator control unit uses a hierarchical FSM-based approach to control and manage complex interactions among CNN software processing, LSTM hardware acceleration, and fully connected layer operations. It is this top-level FSM that sits at the helm of affairs of the entire system operation and data flow management between the ARM processors and FPGA hardware. Thus, this hierarchy enables modular design and facilitates debugging as well as verification during development.

The high-level FSM carries out the major flow of operations, which starts from an idle state waiting for new EEG data from ARM processors. When input data is received, it moves to a WAIT-CNN state where its internal mechanism waits until the CNN processing on the ARM side is complete. As soon as CNN features are available, it goes into LSTM-PROCESS and manages the running of three LSTM layers in sequence. After that, LSTM processing will be followed by a fully connected layer execution FC-PROCESS, and then the OUTPUT result formatting and sending.

Every important processing step has a separate FSM that takes care of the detailed activities inside that step. The LSTM FSM manages the order in which the three LSTM layers work and handles data movement between layers, making sure everything is well coordinated. The FC FSM oversees the running of the three fully connected layers by organizing their matrix-vector multiplications and activation function uses. This kind of setup lets each processing step be made better on its own while keeping the whole system working together.

5. Experimental setting and dataset

The publicly available Bonn University EEG dataset was used for both training and testing the hybrid CNN-LSTM model. The data consists of five sets, A-E, that include 100 single-channel EEG clips per set. Each clip lasts 23.6 seconds and was sampled at 173.6 Hz. Sets A and B were taken from healthy people with the use of surface wires in two states: eyes open and eyes closed, in order. Sets C, D, and E were taken from sick patients using wires placed inside the head; set E matches EEG readings during seizure (ictal) times, while sets C and D match no seizures (interictal) times. Subset A is the normal class; it has EEG signals from healthy people. Subset D is the interictal class; it has EEG signals from patients with epilepsy in a state of non-seizure. Subset E is the ictal class; it has EEG signals recorded during periods of epileptic seizure activity. These subsets give artifact-free segments of EEG since visual inspection has confirmed them to be free from noise connected with muscle activities and eye movements. This choice makes it easy to compare brain work during normal, between seizures, and seizure times, helping in making and checking methods for good seizure finding and sorting.

Different subsets of this data were used for training and validation purposes to avoid any kind of biased evaluation. The work follows a 90:10 split as an experimental setup. A tenfold cross-validation procedure was adopted to increase statistical confidence and remove random sampling biases. In each fold, nine out of the ten equally sized parts of the dataset were used for training, while the remaining part was used for testing by rotating the test set among all folds. This way, every sample gets involved in both learning and validation processes during different iterations. While training, the batch size has been set at 34 such that it achieves computational efficiency and model convergence. The network weights were updated using Adam

because this optimizer is very popular for its adaptive learning rate, as well as being insensitive to noisy gradients. Learning started off with a rate of 1×10^{-3}, and regularization was involved to control overfitting by imposing penalties on large weight updates; the model was subjected to training running across 100 epochs, while performance was tested on a reserved validation fold post every epoch to observe the trend towards convergence.

In further steps toward reducing overfitting and improving generalization, various means were taken. First, the tenfold cross-validation used in training the model is a type of regularization since it involves repeated training on different subsets of the data. Second, spatial and temporal feature dependencies present in the EEG data were well captured by the convolutional and recurrent LSTM layers, reducing redundant representations as well as lowering the degree to which hand-engineered features would be required. Lastly, categorical cross-entropy was used as a loss function that enables real-time monitoring of both training and validation accuracies and losses for stable optimization. Training convergence behavior consistently improved with no sign of divergence.

6. Results and discussion

The general performance of the CNN-LSTM model is based on the four basic classification measures that give a clear view of the model's diagnostic capabilities: accuracy, specificity, sensitivity, and positive predictive value (PPV). These measures shall emanate from the components of a confusion matrix as applied to a three-class classification problem, and for each class, such components being true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for each class. All performance measure calculations are organized under standards for formulations in medical diagnostics. Specifically, accuracy is defined in Eq. 1, sensitivity in Eq. 2, specificity in Eq. 3, and PPV in Eq. 4.

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

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (2)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (3)$$

$$\text{PPV} = \frac{TP}{TP+FP} \quad (4)$$

Table 1 gives the classification results for CNN-LSTM, averaging over tenfold cross-validation. The model has presented very high performances in all classes: Normal, Interictal, and Ictal, with accuracy, positive predictive value (PPV), sensitivity, and specificity at high values most of the time. These are average results from tenfold cross-validation with an overall accuracy of 99.33%, PPV of 99.33%, sensitivity of 99.33%, and specificity of 99.67%. This goes a long way to prove how robust and reliable such an approach can be. Fig. 7 presents the corresponding confusion matrix to visually validate how well the model can differentiate between the three states of EEG. The results show that it has almost perfectly classified all the samples with very minor misclassifications only between the Normal and Ictal classes. It predicted the Interictal class perfectly, which once again goes on to prove its generalization capability over different states of EEG.

Table 1: Classification performance of the hybrid CNN-LSTM model

Classes	Accuracy (%)	PPV (%)	Sensitivity (%)	Specificity (%)
Normal	99.00	99.00	99.00	99.50
Interictal	100.00	100.00	100.00	100.00
Seizure (ictal)	99.00	99.00	99.00	99.50

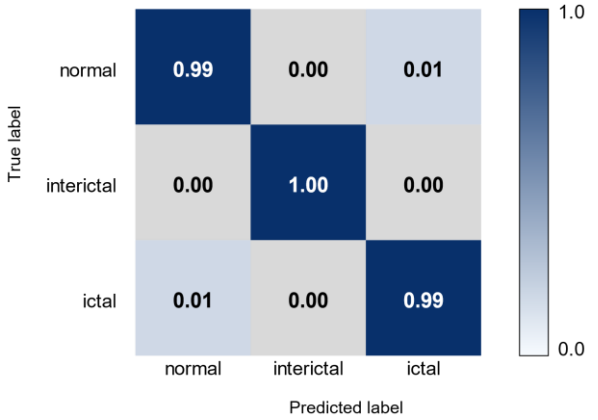

Fig. 7: Confusion matrix

The hybrid CNN-LSTM network is implemented on the AMD Zynq SoC through hardware-software codesign methodology. The whole architecture is synthesized by the AMD Vivado Design Suite and validated with multiple test cases. In the development of neural networks in software, weights and biases are normally represented using floating-point arithmetic to maintain numerical precision. But computation based on floating points is resource-intensive, hence causing a great overhead on power and latency when incorporated into embedded hardware platforms. Fixed-point quantization is adopted as a method for enabling efficient execution within a chip. A detailed precision analysis at a software level is carried out so that the minimum word length for keeping classification performance at its best while reducing complexity in arithmetic operations can be found. This evaluation determined the selection of a 3.12 signed fixed-point format comprising 1 sign bit, 3 integer bits, and 12 fractional bits. Such representation provides an optimal trade-off between quantization accuracy and hardware efficiency since it greatly brings down computational cost together with power consumption, without damaging inference reliability on Zynq SoC. Fig. 8 shows inference accuracy versus quantization bit width. There is a recovery of network accuracy as the quantization bit width increases from 2 to 12 bits. Severe quantization noise in the model at 2 bits precipitates a great drop in accuracy, which is about 60%. It falls within a range between 4 and 6 bits that will already produce

more than 90% accuracy due to a reduced quantization error. The actual floating-point baseline is about 99.33%, so further increase in bit width above eight bits yields essentially nothing towards increased accuracy. This trend has consolidated, therefore confirming mid-range fixed-point precision as optimal toward balancing hardware efficiency against attainable accuracy. Near-floating-point accuracy can be maintained with this range while reducing FPGA resource consumption and power usage, hence making it appropriate for real-time hardware deployment.

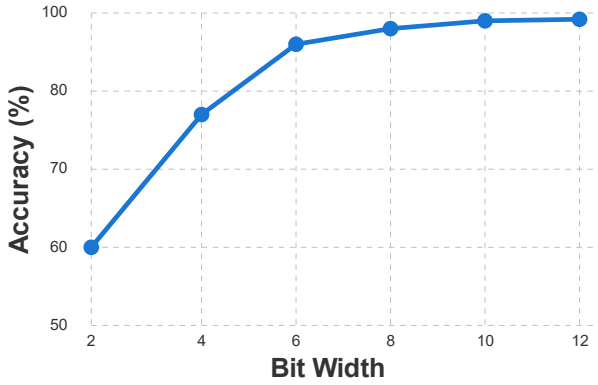

Fig. 8: Inference accuracy versus quantization bit width

Table 2 shows the FPGA resource usage on Zybo Z7-10 by the LSTM Accelerator. This proposed hardware-software co-design results in moderate usage of FPGA resources on the Zybo Z7-10 platform, consuming only 13% of Look-Up Tables (LUTs), and 5.6% Flip-Flops (FFs), whereas Digital Signal Processing (DSP) blocks consumption stood at 32.5%, and Block Random Access Memory (BRAMs) used up to 26.4%. This implies that the design has more than enough space for running real-time applications to fit into the device constraints, hence proving its feasibility. Though FPGA technologies can provide huge degrees of flexibility for prototyping and validation-on-chip, design in terms of power and speed can be optimized further when translated to a full-custom Application-Specific Integrated Circuit (ASIC) implementation. Custom hardware optimization at a very detailed level would be allowed by translating the architecture from the existing FPGA-based system into an ASIC. This transformation would strip away much of the overhead and increase energy efficiency, thus making it appropriate to use on a commercial scale or in battery-powered applications.

Table 3 presents a comparison between the proposed hardware-software co-design and software-only implementation. The results show that both configurations achieved the same classification accuracy, 99.33%. The model on FPGA hardware does not reduce or compromise its detection performance. However, there is a much reduction in inference time to 657 ms from a co-designed system when compared with 8212 ms with software running alone, representing a speed up of more than 12 times. This therefore validates that achieving such real-time low latency accurately equivalent to that achievable in a software model can be achieved with an FPGA accelerator.

Table 2: Resource utilization summary of the proposed LSTM hardware accelerator

Resource	Utilization	Available	Usage (%)
LUT	3653	28,000	13.0%
FF	3128	56,000	5.6%
DSP	26	80	32.5%
BRAM	37	140	26.4%

Table 3: Inference accuracy and execution time comparison

Configuration	Inference accuracy	Execution time
HW-SW co-design	99.33%	657 ms
SW only	99.33%	8212 ms

Table 4 presents the comparison of the proposed CNN-LSTM hardware-software co-design with other existing FPGA-based epileptic seizure detection systems on the Bonn EEG dataset. The proposed design achieves 99.33% accuracy with moderate hardware utilization on a Zynq SoC, delivering computational efficiency together with classification performance. This is higher than those implemented previously by Rizal et al. (2022) and Jose et al. (2020), which used conventional feature extraction and machine-learning classifiers because hybrid CNN-LSTM leverages deep spatial-temporal learning and shows accuracy improvement with efficient resource usage. The results prove that this kind of co-design is applicable wherever medical applications are real-time and resources are a constraint.

Table 4: Comparison of the proposed system with existing FPGA-based EEG classification implementations

Reference	Accuracy (%)	LUT	FF	DSP	FPGA device	Classifier/model
This work	99.33	3653	3128	26	Zynq SoC	Hybrid CNN-LSTM
Rizal et al. (2022)	90.74	5202	10,196	-	Zynq SoC	Hjorth descriptor + KNN
Jose et al. (2020)	98.50	243	1084	16	Zynq SoC	Extreme learning machine
Sahani et al. (2021)	99.37	8764	-	-	Virtex-5	Reduced deep CNN + MKRVFLN
Meddah et al. (2020)	98.67	3018	1980	162	Artix-7	DWT-PCA + SVM

7. Conclusion

The work presented in this paper implemented a hardware-software co-design CNN-LSTM model using Zynq-7000 SoC for real-time epileptic seizure detection. The proposed system runs the CNN layers on an ARM processor and accelerates LSTM and fully connected layers on an FPGA to achieve 99.33% accuracy with a 12× speedup in inference time over pure software implementation. It uses a timeshared hardware architecture where LUT, DSP, and BRAM usage are significantly reduced while maintaining real-time performance compared to other hardware-based seizure detection implementations. Hybrid deep-learning architectures can be optimally run on embedded platforms for low-latency, reliable seizure

detection that is also highly amenable to further optimizations toward ASIC design as well as clinical integrations.

List of abbreviations

ANN	Artificial neural network
ARM	Advanced RISC machine
ASIC	Application-specific integrated circuit
AXI	Advanced extensible interface
BCI	Brain–computer interface
BRAM	Block random access memory
CAD	Computer-aided design
CNN	Convolutional neural network
CNN–LSTM	Convolutional neural network–long short-term memory
DSP	Digital signal processing (block)
DWT	Discrete wavelet transform
EEG	Electroencephalogram
ELM	Extreme learning machine
FC	Fully connected
FF	Flip-flop
FFT	Fast Fourier transform
FN	False negative
FP	False positive
FPGA	Field-programmable gate array
FSM	Finite state machine
GRU	Gated recurrent unit
HLS	High-level synthesis
HW–SW	Hardware–software
k-NN	k-nearest neighbors
LSTM	Long short-term memory
LUT	Look-up table
MAC	Multiply–accumulate
MKRVFLN	Multi-kernel random vector functional link network
ML	Machine learning
MLP	Multi-layer perceptron
PCA	Principal component analysis
PL	Programmable logic
PPV	Positive predictive value
PS	Processing system
QDA	Quadratic discriminant analysis
RF	Random forest
RNN	Recurrent neural network
ReLU	Rectified linear unit
SIMD	Single instruction, multiple data
SNN	Spiking neural network
STFT	Short-time Fourier transform
SVM	Support vector machine
SoC	System-on-chip
TN	True negative
TP	True positive
UART	Universal asynchronous receiver–transmitter
VHDL	Very high-speed integrated circuit hardware description language
XSG	Xilinx system generator
Zynq SoC	Xilinx Zynq system-on-chip

Compliance with ethical standards

Ethical considerations

This study uses the publicly available Bonn University EEG dataset. All data are anonymized and were collected in prior studies with appropriate ethical approval. No new experiments involving human participants or animals were conducted in this research. Therefore, ethical approval and informed consent were not required for this study.

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Ahmad I, Liu Z, Li L, Ullah I, Aboyeji ST, Wang X, Samuel OW, Li G, Tao Y, Chen Y, and Chen S (2024). Robust epileptic seizure detection based on biomedical signals using an advanced multi-view deep feature learning approach. IEEE Journal of Biomedical and Health Informatics, 28(10): 5742-5754. https://doi.org/10.1109/JBHI.2024.3396130 **PMid:38696293**

Alam MS, Hasan M, Farooq O, and Hasan M (2024a). Field programmable gate array-based energy-efficient and fast epileptic seizure detection using support vector machine and quadratic discriminant analysis classifier. Engineering Reports, 6(8): e12812. https://doi.org/10.1002/eng2.12812

Alam MS, Khan U, Hasan M, and Farooq O (2024b). Energy efficient FPGA implementation of an epileptic seizure detection system using a QDA classifier. Expert Systems with Applications, 249: 123755. https://doi.org/10.1016/j.eswa.2024.123755

Beeraka SM, Kumar A, Sameer M, Ghosh S, and Gupta B (2022). Accuracy enhancement of epileptic seizure detection: A deep learning approach with hardware realization of STFT. Circuits, Systems, and Signal Processing, 41: 461-484. https://doi.org/10.1007/s00034-021-01789-4

Buzdar AR, Latif A, Sun L, and Buzdar A (2016). FPGA prototype implementation of digital hearing aid from software to complete hardware design. International Journal of Advanced Computer Science and Applications, 7(1): 645-654. https://doi.org/10.14569/IJACSA.2016.070188

Buzdar AR, Sun L, Azhar MW, Khan MI, and Rao K (2017a). Area and energy efficient viterbi accelerator for embedded processor datapaths. International Journal of Advanced Computer Science and Applications, 8(3): 402-407. https://doi.org/10.14569/IJACSA.2017.080355

Buzdar AR, Sun L, Rao K, Azhar MW, and Khan MI (2017b). Cyclic redundancy checking (CRC) accelerator for embedded processor datapaths. International Journal of Advanced Computer Science and Applications, 8(2): 321-325. https://doi.org/10.14569/IJACSA.2017.080242

Cao X, Zheng S, Zhang J, Chen W, and Du G (2025). A hybrid CNN-Bi-LSTM model with feature fusion for accurate epilepsy seizure detection. BMC Medical Informatics and Decision Making, 25: 6. https://doi.org/10.1186/s12911-024-02845-0 **PMid:39762881 PMCid:PMC11706039**

Chaturvedi A (2023). Harnessing machine learning and deep learning for improved epileptic seizure detection: A review. In the Proceedings of the 5th International Conference on Information Management & Machine Intelligence, ACM, Jaipur, India: 1-9. https://doi.org/10.1145/3647444.3647944

Dabbaghian A and Kassiri H (2025). Energy-efficient automated seizure detection in wearable/implantable BCIs: Motivations, methods, and example implementation. In the 2025 IEEE 55th International Symposium on Multiple-Valued Logic, IEEE Computer Society, Montreal, Canada: 57-62. https://doi.org/10.1109/ISMVL64713.2025.00020

Dash DP, Kolekar M, Chakraborty C, and Khosravi MR (2024). Review of machine and deep learning techniques in epileptic seizure detection using physiological signals and sentiment analysis. ACM Transactions on Asian and Low-Resource

Language Information Processing, 23(1): 1-29. https://doi.org/10.1145/3552512

Ein Shoka AA, Dessouky MM, El-Sayed A, and Hemdan EED (2023). EEG seizure detection: concepts, techniques, challenges, and future trends. Multimedia Tools and Applications, 82: 42021-42051. https://doi.org/10.1007/s11042-023-15052-2 **PMid:37362745 PMCid:PMC10071471**

Feng L, Li Z, and Wang Y (2017). VLSI design of SVM-based seizure detection system with on-chip learning capability. IEEE Transactions on Biomedical Circuits and Systems, 12(1): 171-181. https://doi.org/10.1109/TBCAS.2017.2762721 **PMid:29377805**

Goel A (2025). Epilepsy seizure detection based on EEG signals using machine learning. In 2025 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics, IEEE, Bengaluru, India: 1-9. https://doi.org/10.1109/IITCEE64140.2025.10915478

Indira Priyadarshini B and Reddy DK (2023). Adaptive octopus deep transfer learning based epileptic seizure classification on field programmable gate arrays. Evolving Systems, 14: 479-499. https://doi.org/10.1007/s12530-022-09474-w

Jaffino G, Jose JP, Raman R, Venkatesan RS, and Aarthi VPMB (2023). FPGA-based system for real-time epileptic seizure detection using KNN classifier. In the 2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies, IEEE, Trichirappalli, India: 1-5. https://doi.org/10.1109/ICEEICT56924.2023.10157075

Jose JP, Sundaram M, and Jaffino G (2020). FPGA implementation of epileptic seizure detection using ELM classifier. In the 2020 Sixth International Conference on Bio Signals, Images, and Instrumentation, IEEE, Chennai, India: 1-5. https://doi.org/10.1109/ICBSII49132.2020.9167598 **PMid:33655238 PMCid:PMC7903891**

Khan MI and da Silva B (2024). Harnessing FPGA technology for energy-efficient wearable medical devices. Electronics, 13(20): 4094. https://doi.org/10.3390/electronics13204094

Meddah K, Zairi H, Bessekri B, Cherrih H, and Kedir-Talha M (2020). FPGA implementation of epileptic seizure detection based on DWT, PCA and support vector machine. In the 2020 Second International Conference on Embedded and Distributed Systems, IEEE, Oran, Algeria: 141-146. https://doi.org/10.1109/EDiS49545.2020.9296466

Mulla ZA, Reddy S, Lakshmi VV, and Maturi SC (2022). Epileptic seizure detection system using single hidden layer extreme learning machine (ELM). In the 2022 9th International Conference on Computing for Sustainable Global Development, IEEE, New Delhi, India: 298-302. https://doi.org/10.23919/INDIACom54597.2022.9763268

Muneeb A and Kassiri H (2024). Customized development and hardware optimization of a fully-spiking SNN for EEG-based seizure detection. In the 2024 IEEE Biomedical Circuits and Systems Conference, IEEE, Xi'an, China: 1-5. https://doi.org/10.1109/BioCAS61083.2024.10798216

Pandey SK, Janghel RR, Mishra PK, and Ahirwal MK (2023). Automated epilepsy seizure detection from EEG signal based on hybrid CNN and LSTM model. Signal, Image and Video Processing, 17: 1113-1122. https://doi.org/10.1007/s11760-022-02318-9

Rafiammal SS, Jamal DN, and Mohideen SK (2020). Reconfigurable hardware design for automatic epilepsy seizure detection using EEG signals. Engineering, Technology and Applied Science Research, 10(3): 5803-5807. https://doi.org/10.48084/etasr.3419

Razi KF and Schmid A (2021). Two-stage hardware-friendly epileptic seizure detection method with a dynamic feature selection. In the 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), IEEE, Mexico City, Mexico: 156-159. https://doi.org/10.1109/EMBC46164.2021.9630486 **PMid:34891261**

Rizal A, Hadiyoso S, and Ramdani AZ (2022). FPGA-based implementation for real-time epileptic EEG classification using Hjorth descriptor and KNN. Electronics, 11(19): 3026. https://doi.org/10.3390/electronics11193026

Rout SK, Sahani M, Dora C, Biswal PK, and Biswal B (2022). An efficient epileptic seizure classification system using empirical wavelet transform and multi-fuse reduced deep convolutional neural network with digital implementation. Biomedical Signal Processing and Control, 72: 103281. https://doi.org/10.1016/j.bspc.2021.103281

Sadangi RK, Patra B, Gandhi TK, Sutar BC, Maity NP, and Maity R (2024). Improved EEG-based epileptic seizure classification using fusion of machine learning model with statistical feature integration. In the 2024 International Conference on Communication, Control, and Intelligent Systems, IEEE, Mathura, India: 1-4. https://doi.org/10.1109/CCIS63231.2024.10931966

Sahani M, Rout SK, and Dash PK (2021). FPGA implementation of epileptic seizure detection using semi supervised reduced deep convolutional neural network. Applied Soft Computing, 110: 107639. https://doi.org/10.1016/j.asoc.2021.107639

Saini SK and Chandel G (2024). Implementation of the frequency domain feature extraction technique for machine learning-based seizure detection. In the 2024 First International Conference for Women in Computing, IEEE, Pune, India: 1-6. https://doi.org/10.1109/InCoWoCo64194.2024.10863338

Sajja A and Rooban S (2024). FPGA implementation of automatic seizure detection in EEG signals using machine learning algorithm. Discover Applied Sciences, 6: 383. https://doi.org/10.1007/s42452-024-06060-4

Sarić R, Beganović N, Jokić D, and Čustović E (2022). Towards efficient implementation of MLP-ANN classifier on the FPGA-based embedded system. IFAC-PapersOnLine, 55(4): 207-212. https://doi.org/10.1016/j.ifacol.2022.06.034

Sarić R, Jokić D, Beganović N, Pokvić LG, and Badnjević A (2020). FPGA-based real-time epileptic seizure classification using artificial neural network. Biomedical Signal Processing and Control, 62: 102106. https://doi.org/10.1016/j.bspc.2020.102106

Shanmugam S and Dharmar S (2025). FFNN architecture implementation on FPGA for seizure detection using EEG signals. Applied Soft Computing, 181: 113533. https://doi.org/10.1016/j.asoc.2025.113533

Tatum WO, Rubboli G, Kaplan PW et al. (2018). Clinical utility of EEG in diagnosing and monitoring epilepsy in adults. Clinical Neurophysiology, 129(5): 1056-1082. https://doi.org/10.1016/j.clinph.2018.01.019 **PMid:29483017**

WHO (2019). Epilepsy: A public health imperative. World Health Organization, Geneva, Switzerland.

Zairi H, Meddah K, Cherrih H, Bessekri B, and Talha MK (2022). FPGA-based neural network system for epileptic seizure detection. In the 2022 2nd International Conference on Advanced Electrical Engineering, IEEE, Constantine, Algeria: 1-6. https://doi.org/10.1109/ICAEE53772.2022.9962009