Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(7): 19645–19669.
DOI: 10.3934/math.2024958
Received: 22 February 2024
Revised: 21 May 2024
Accepted: 29 May 2024
Published: 14 June 2024

Research article

Deep-learning-based intelligent neonatal seizure identification using spatial and spectral GNN optimized with the Aquila algorithm

Madhusundar Nelson[1], Surendran Rajendran[1,*], Osamah Ibrahim Khalaf[2] and Habib Hamam[3,4,5,6]

[1] Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, 602105, India
[2] Department of Solar, Al-Nahrain Research Center for Renewable Energy, Al-Nahrain University, Jadriya, Baghdad, Iraq
[3] Faculty of Engineering, Uni de Moncton, NB, E1A3E9, Canada
[4] Hodmas University College, Taleh Area, Mogadishu, Somalia
[5] Bridges for Academic Excellence, Tunis, Centre-Ville, Tunisia
[6] School of Electrical Engineering, University of Johannesburg, South Africa

* **Correspondence:** Email: surendran.phd.it@gmail.com; Tel: +919444013042.

Abstract: Diagnosing and treating newborn seizures accurately and promptly is crucial for providing the best possible care for these patients. For the purpose of intelligently identifying newborn seizures, this work introduced a unique method that uses spectral and spatial graph neural networks (SSGNNs) optimized with the Aquila algorithm. Using electroencephalogram (EEG) recordings, the suggested methodology takes advantage of the complex spatial and spectral characteristics of infant brain activity. Spatial and spectral GNNs were used to extract significant spatiotemporal patterns suggestive of seizure episodes by organizing the brain activity data as a graph, with nodes representing various brain regions and edges signifying functional relationships. By combining spectral and spatial data, the depiction of newborn brain dynamics was improved and made it possible to distinguish between seizure and non-seizure phases with greater accuracy. Moreover, the introduction of the Aquila algorithm improved the GNNs' performance in seizure identification tasks by streamlining the training process. A large dataset of EEG recordings from newborns with and without seizures was used to assess the effectiveness of the suggested method. Higher accuracy, sensitivity, and specificity in seizure detection were achieved in the experimental results, which showed greater performance when compared to conventional methods. This work offered an automated, data-driven method for identifying newborn seizures, which is a major development in the treatment of newborns. By combining spectral and spatial GNNs and optimizing the results using the Aquila method, it is possible to enhance seizure detection accuracy and potentially prevent neurological consequences in affected children by intervening early. This method has the potential to completely change the way neonatal care is provided by giving medical professionals a strong tool for accurate and prompt seizure monitoring in neonatal intensive care units (NICU).

Keywords: deep learning; neonatal; spatial and spectral graph neural networks; seizures; EEG; Aquila algorithm; NICU
Mathematics Subject Classification: 68Q32, 68T07, 68T40, 92D30

1. Introduction

Neonatal seizures represent a critical challenge in neonatal care. Neonatal seizures refer to seizures that occur in newborn infants, typically within the first 28 days of life. Seizures in neonates are considered a medical emergency and can manifest as abnormal electrical activity in the brain, leading to various motor, sensory, or autonomic manifestations. There are several types of neonatal seizures. Subtle seizures may be difficult to detect as they present with subtle clinical signs such as eye deviations, lip smacking, or repetitive chewing movements. Clonic seizures are characterized by rhythmic jerking movements of the limbs or face. Tonic seizures are marked by sustained muscle contractions, resulting in stiffening of the body or limbs. Myoclonic seizures involve sudden, brief muscle contractions, often appearing as brief jerks or twitches. Multifocal seizures involve multiple areas of the brain demanding swift and accurate detection to mitigate potential neurological complications. Conventional methods reliant on visual inspection of electroencephalogram (EEG) recordings are prone to subjectivity and can be time-consuming. In recent years, deep learning techniques have shown promise in automating seizure detection, offering the potential for improved accuracy and efficiency. To identify newborn seizures, this research suggests a new method that uses spectral and spatial graph neural networks (SSGNNs) in combination with the Aquila algorithm for optimization. Our methodology treats brain activity as a graph, with nodes representing different brain regions and edges denoting functional relationships, in contrast to existing methods that analyze EEG data as sequential impulses, in order to capture both the spectral and spatial properties found in the dynamics of the developing brain.

Motivation for this research mainly focuses on newborn babies, particularly those born prematurely or with birth problems, where seizures are a reasonably common occurrence. It is important to recognize seizures as soon as possible since they may indicate serious neurological disorders and, if left untreated, may result in long-term developmental problems. Conventional approaches of identifying seizures in neonates, including clinician’s visually inspecting electroencephalogram (EEG) data, take a long time and need a high level of competence. Manual detection might be inconsistent and prone to errors since newborn seizures are mild and varied. Both image and signal processing tasks have demonstrated a great deal of potential in deep learning. Systems based on deep learning have the ability to quickly and accurately analyze vast amounts of EEG data.

By identifying subtle patterns and temporal dynamics of seizures that human observers would overlook, these algorithms might speed up diagnosis and increase diagnostic accuracy.

Figure 1 includes a set of sample images of neonatal seizures in a neonatal intensive care unit (NICU). The patterns of neural firing in various brain regions are reflected in the spatial characteristics, which capture the spatial distribution of the brain activity. The frequency patterns of these activities, however, are captured by spectral features, which shed light on the underlying brain processes. The new method improves the discriminative capacity for seizure identification by providing a more comprehensive picture of newborn brain activity by merging spatial and spectral information. The Aquila algorithm, a specialized optimization method designed for GNNs in seizure identification tasks, further optimizes the performance of our model. By efficiently using spectral and geographical information, the Aquila algorithm improves the training process and raises the model's accuracy and resilience.

Figure 1. Sample images of neonatal seizures in the NICU.

A block diagram of a system that recognizes neonatal seizures using a spatial and spectral graph neural network (SSGNN) is shown in Figure 2. ECG signals from infants are fed into the system. After 256 Hz of sampling, noise is taken out of these signals using filtering. The ECG signals are filtered to eliminate noise and down sampled to 32 Hz. This facilitates the GNN's learning of the key characteristics of the signals and lowers the volume of data that must be handled. Time intervals: The ECG signal is separated into brief segments, such as one second. Frequency band: Various frequency bands, such as delta, theta, alpha, and beta, are used to determine the signal's power for each time interval. Connectedness metric: The various frequency bands are used to calculate a connectedness measure. This measurement encapsulates the relationship between the various frequency bands. Channel feature vector: For every time interval, the features of time, frequency, and connection are combined into a single feature vector.

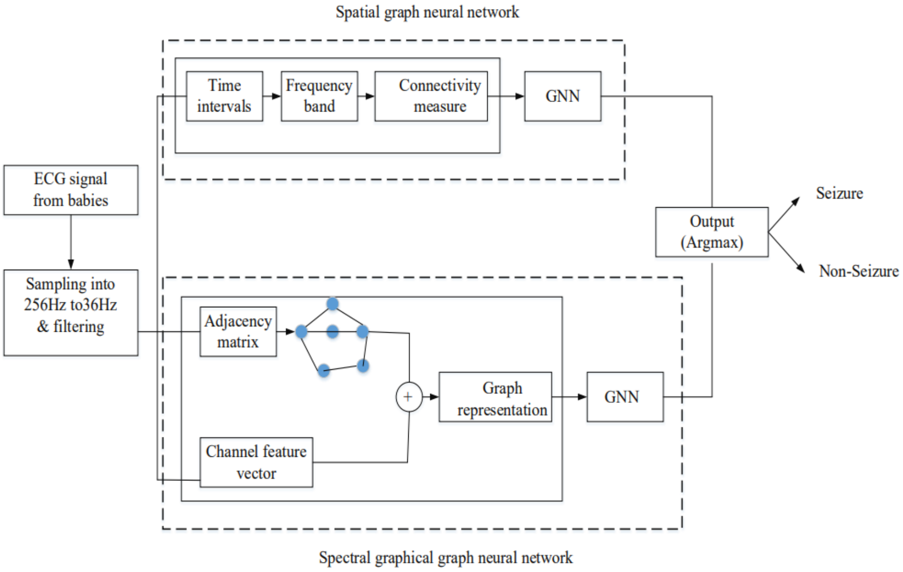


Figure 2. A detailed block diagram for neonatal seizure identification.

Display of graph: We use an adjacency matrix to show the connections between the various time intervals. The similarity between the feature vectors of the two connected time intervals determines the weight of each edge in the matrix. Graph representation: The ECG signal is represented as a graph using the adjacency matrix and feature vectors. Spectral graphical graph neural network: The links between the various time intervals in the graph are discovered using a spectral graphical graph neural network (GNN). The graph representation is fed into the GNN, which predicts whether or not a seizure is present in the ECG data. Seizures/Non-Seizures: The system's ultimate output is a binary classification that indicates whether a seizure occurred or not.

The objective of this research describes utilizing the capabilities of graph neural networks (GNNs) to gather both spatial and spectral characteristics from EEG data in order to create a deep learning model specifically suited for neonatal seizure identification. Taking EEG signals and extracting their spectral and spatial properties, which can then be put into the SSGNN to help detect seizures more accurately, will optimize the SSGNN's performance in terms of accuracy, sensitivity, and specificity by fine-tuning its parameters using the Aquila optimization algorithm to assess how applying the suggested system in neonatal intensive care units (NICUs) will affect the overall neonatal outcomes, diagnostic accuracy, and duration of interventions.

2. Related works

The electroencephalogram (EEG) signal, originally described by Anam et al. in 2023, is a medical diagnostic tool that has gained significant attention in recent years for the treatment of epilepsy patients [1]. It measures the electrical activity received from the cerebral cortex's nerve cells. Although the focus on grid-like data has been a major limitation of current deep-learning-based automatic algorithms for the automatic detection of epileptic seizures from raw EEG signals, it is still difficult to think of physiological recordings as a matrix because of their frequently irregular and disordered structures.

Graph-based representations of EEG data have been investigated recently for seizure identification. To capture the spatial and temporal dependencies present in EEG signals, graph neural networks (GNNs) are especially well-suited. Daniele Grattarola et al. (2020) modelled the spatial connectivity between EEG electrodes to propose a graph convolutional network (GCN) for neonatal seizure detection [2]. Their method demonstrated the efficacy of graph-based representations by outperforming conventional CNN-based approaches. Seizure detection models' generalization performance and resilience have been enhanced by the use of ensemble learning approaches like boosting and bagging. Furthermore, fusion solutions that integrate various modalities—such as EEG and other physiological signals—have demonstrated the potential to improve seizure recognition systems' accuracy. Shoeibi et al. presented an ensemble technique that merged different CNN architectures and outperformed individual models in 2021 [3].

Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in particular are two deep learning techniques that have been extensively used for the diagnosis of newborn seizures. These models identify seizure occurrences by utilizing the temporal patterns in EEG signals. Gramacki et al. achieved high accuracy and low false alarm rates in real-time neonatal seizure detection using a deep learning framework based on long short-term memory (LSTM) networks in 2022 [4]. Real-time monitoring systems: Due to the growing need for real-time monitoring in clinical settings, scientists have created online seizure prediction and detection systems. Low-latency algorithms and effective hardware implementations are frequently included in these systems to allow for prompt intervention.

A real-time seizure detection system based on adaptive spectrum analysis, for instance, was created by Rezaee et al. in 2016 to enable early alerting of seizure episodes to medical professionals [5]. Seizures are often identified using classification methods like random forests, decision trees, and support vector machines (SVMs) called as Feature extraction. In [6], Temko et al. (2011) showed encouraging results by using SVM classifiers in conjunction with statistical characteristics to detect newborn seizures. In classification, a lot of research focuses on taking discriminative characteristics out of EEG data and applying classifiers to determine which states are seizure-related and which are not. According to the work of Kukker et al. (2021) [7], techniques such as time-domain, frequency-domain, and time-frequency analysis are used in fuzzy Q-learning with assistance and fuzzy Q-learning alone, with success rates of 93.71% and 96.79%, respectively. Jareda et al. (2019) described an EEG signal-based seizure classification using a wavelet transform [8].

The steps involved in our suggested methodology are building the spectral and spatial GNNs, putting the Aquila algorithm into practice, and conducting training. Next, we use an extensive dataset of EEG recordings from newborns with and without seizures to assess our model's performance. Our findings prove the effectiveness of the suggested strategy, exhibiting higher sensitivity and accuracy than traditional techniques [9]. Providing an automated, data-driven method for seizure diagnosis, this research marks a breakthrough in the treatment of newborns. Our technique, which makes use of spatial and spectral GNNs improved with the Aquila algorithm, has the potential to lead to better clinical results and less workload for neonatal intensive care unit healthcare workers.

3. Materials and methods

3.1. Acquisition of datasets

EEG recordings from neonates with and without seizures were obtained from the publicly available dataset hosted on Zenodo [10]. The diagnosis of multiple medical diseases, such as epilepsy, sleep disorders, and dementia, Seizures can be made using these voltage changes. EEG recordings of newborns with and without seizures were collected. The EEG recordings from newborns under round-the-clock observation in neonatal intensive care units (NICUs) from various hospitals make up the dataset. Clinical specialists annotate seizure occurrences, giving teaching and assessment tools ground truth labelling. Multiple electrodes are positioned on the scalp to collect EEG signals, which capture brain activity throughout time. Since each recording session usually lasts for several hours, a thorough examination of the dynamics of the neonatal brain is possible. The seizure and non-seizure portions in the dataset make it easier to create and assess seizure detection systems. Brain activity: Electrodes are used to measure the electrical activity produced by brain neurons. This activity is measured in microvolts (μV). Multichannel recording: This technique records activity from several brain regions using multiple electrodes, giving an all-encompassing picture of brain activity.

Figure 3 offers information about gathering unprocessed EEG waves. The electrical activity along the scalp is recorded by electroencephalography or EEG. Electrodes are applied to the head during a non-invasive technique to monitor voltage changes brought on by brain activity. Noise reduction: To eliminate noise and artifacts from the signal, filters are used. Neonates often exhibit muscular contractions, external electrical interference, and motions as artifacts. Using band-pass filtering, some frequency ranges are let through while others are blocked. Lower frequencies (0.5–30 Hz) are frequently the focus of neonatal EEG.

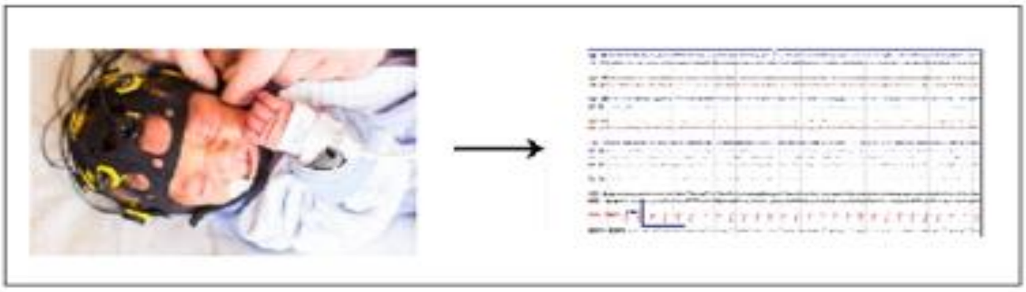


Figure 3. A sample of a raw EEG signal.

Conversion of analog to digital digitization: The process of converting analog EEG signals into digital form allows for computer-based analysis and storage. Neonatal EEG samples are usually taken at a high frequency (250–500 Hz) in order to record precise brain activity. The graph representation of non-seizure (raw EEG signal) in Figure 4 simplifies the identification of neonates.

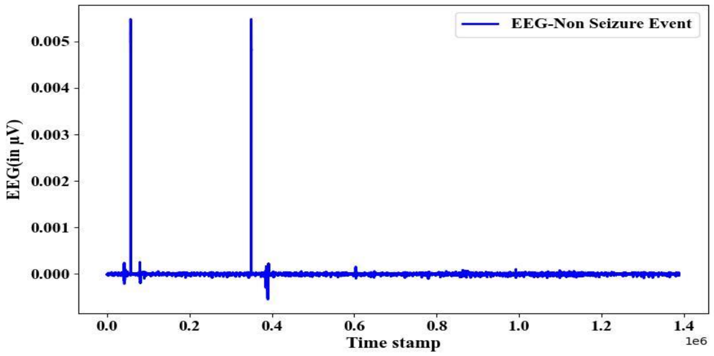


Figure 4. Non-seizure (raw EEG signal).

The visual representation of aberrant brain activity linked to seizures in Figure 5 facilitates the proof of identity in neonates.

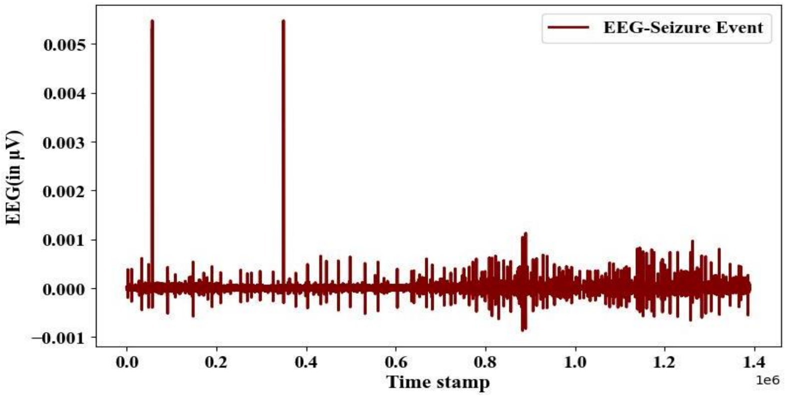


Figure 5. Seizure (raw EEG signal).

3.2. Preprocessing

To achieve optimal model performance and efficacy in neonatal seizure identification, EEG signals, while maintaining their fundamental temporal and spectral properties, produce augmented samples. The preprocessing approaches utilized to create the augmented samples uses the EEG signals as the raw data.

Data augmentation increases the model's capacity to generalize to previously undiscovered neonatal EEG recordings and reduces overfitting. EEG data undergoes multiple preprocessing processes before training the spatial and spectral graph neural network (GNN) supplemented with the Aquila algorithm.

3.2.1. Filtering and segmentation

Using common signal processing methods including bandpass filtering, objects like muscle activity, electrode drift, and ambient noise are eliminated. High-pass filtering is used to reduce low-frequency drifts while maintaining high-frequency neural impulses that are important for seizure activity. High-frequency noise and interference to improve signal clarity. Extended EEG data is divided into several smaller segments. This is accomplished by employing a sliding window technique with a one-second period time and a 0.5 overlap rate to divide the data into multiple periods. For instance, a 50-s EEG signal split into 100 epochs would be represented as a 50-s vector with a sampling frequency of 1 Hz. Following data segmentation, a proposed approach by Jibon et al. [11] classifies and linearly analyses the labels of various time periods.

A block schematic of a system that processes a signal using segmentation and filtering may be seen in Figure 6. A bandpass filter is used after the signal has been down sampled from 256 Hz to 32 Hz. Following the filtering process, the signal is divided into time intervals, and features are taken out of each one. Following the creation of a graph representation of the signal using these attributes, the deep learning model uses the representation as input to forecast the signal.

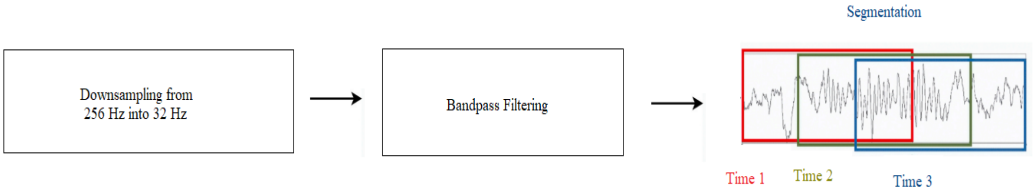


Figure 6. Filtering and segmentation.

3.2.2. Spatial feature extraction

To create the graph representation of brain activity, spatial information is taken out of electrode placements. The mapping of electrode coordinates onto a standard head model (such as the 10–20 system) ensures spatial consistency between recordings. Spatial adjacency matrices, which depict the spatial relationships and connectivity patterns between different brain regions, are defined by calculating the distances between electrodes. The input features for the spatial GNN part of the model are spatial adjacency matrices [12]. Based on the functional connectivity (e.g., coherence, correlation) between electrode signals, nodes and edges are defined as EEG electrodes. Adjacency matrix: To illustrate relationships, we create an adjacency matrix. Indicating the strength of the connection, this matrix might be weighted or binary. Binary signals the presence of a connection.

Node features: Gather information about each node (EEG electrode), including power spectral density (PSD), statistical metrics, and raw EEG readings. Convolutional layers in graphs: Convolutional layers on graphs can be used to combine data from nearby nodes. The method for extracting spatial features from newborn EEG signals in a system intended to detect seizures is depicted in Figure 7. Time intervals are the data input for this procedure, which receives segmented periods from the EEG signal. Each interval most likely denotes a brief period of the signal. The signal's power is computed within various frequency bands, such as delta, theta, alpha, and beta, for every time period. This encapsulates the different frequencies of rhythmic activity that were present throughout that time frame. Connectivity measures: The degree of connectedness among the various frequency bands is computed. This makes it easier to comprehend how activity in several frequency bands connects over time.

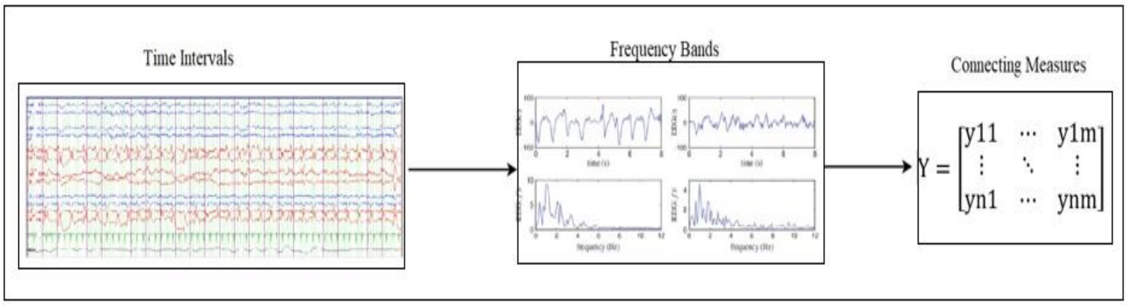


Figure 7. Spatial feature extraction.

3.2.3. Spectral feature extraction

To extract the frequency characteristics of EEG signals, spectral features are generated. Applying the Fourier or wavelet transform breaks down EEG signals into representations in the frequency domain. Estimates of power spectral density are calculated to measure the signal power distribution over various frequency bands. To identify unique spectral signatures connected to seizure activity, spectral parameters such as peak frequencies, spectral power ratios, and spectral entropy are retrieved. The spectral GNN component of the model receives spectral information as input features [13].

Figure 8 shows a more sophisticated feature extraction stage in a system that uses a spectral graph neural network (GNN) to identify seizures in neonatal EEG signals. Convolution of spectral graphs (SGC): The Laplacian matrix of the graph provides the eigenvalues and eigenvectors that are used by the SGC layer. These numbers represent the graph’s structural information, indicating how well-connected various nodes (time intervals) are to one another. Convolution process: By effectively convolving each node's feature vectors with a function of both its own and its neighbors’ eigenvalues, an SGC operation is carried out. This takes into account the context given by its connected intervals as well as the features of each node, incorporating the effect of nearby nodes into each node’s feature representation. Features extracted: A set of improved feature vectors, one for each node in the graph (which represents each time interval), is the stage’s final result. These features offer a more comprehensive representation for seizure detection by combining not only the unique properties of each interval but also the contextual data from its connected neighbors. Creating graphs from EEG data, using spectral and graph convolutional layers, and combining these features for reliable seizure diagnosis are the steps involved in implementing spatial and spectral feature extraction in deep learning for neonatal seizure identification. This all-inclusive method makes use of SSGNNs’ strengths to capture both spectral and spatial properties, allowing for the accurate and thoughtful diagnosis of newborn seizures.

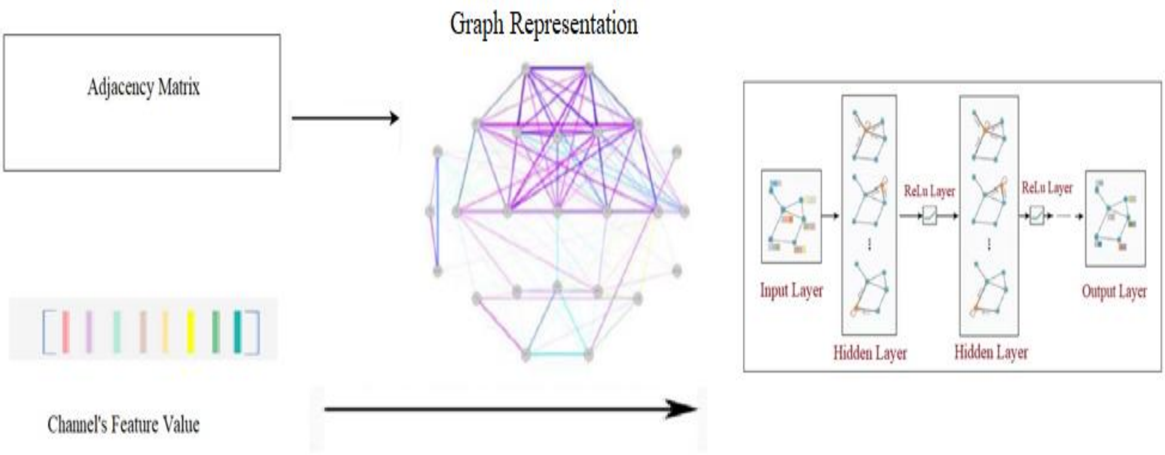


Figure 8. Spectral feature extraction.

3.3. Graph construction

A graph-based representation is created to efficiently capture the spectral and spatial dependencies present in neonatal EEG data. To create a network structure, this needs to map EEG electrodes to nodes and define functional relationships between them.

3.3.1. Definition of nodes

Every EEG electrode is represented by a graph node. To maintain consistency among recordings, electrode placements are usually standardized using well-established head models, such as the 10–20 system. Each electrode's coordinates (x, y, z) represent the spatial characteristics of the associated node.

$$D_{x,y} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}. \tag{1}$$

Equation (1) defines the Pearson correlation (D), a measure of the correlation between two time series. x and y are taken as the time series. σ_x and σ_y are the respective standard deviations σ_{xy} as the covariance of the time series.

3.3.2. Edge definition

Functional connections between EEG electrodes are used to define the edges in the graph. Functional connectivity measures, such as coherence, correlation, or phase synchronization, quantify the strength of the connections between the electrode pairs [14,15]. These measures reflect the degree of synchronization or correlation in neural activity between different brain regions. Edge weights are computed based on functional connectivity values, representing the strength of the interaction between nodes. By measuring the lag between two time series, normalized cross-correlation (*DD*) calculates how similar the two are in Eq (2):

$$DD_{x,y}(m) = \frac{T_{xy}}{\sqrt{T_{xy}(0)T_{yy}(0)}}. \tag{2}$$

The cross-correlation between the two signals, *x* and *y*, is represented by the term T_{xy}. The signals autocorrelations are then T_{yy} and T_{xx}.

$$DDD_{x,y}(m) = DD_{x,y}(m) - DD_{x,y}(-m). \tag{3}$$

This time, the time series are *x* and *y*, their cross-correlation is *DD*, and the latency is *m*. The corrected cross-correlation is *DDD*.

$$VD_{x,y}(f) = \frac{|H_{x,y}(I)|^2}{H_{x,x}(I)H_{y,y}(I)}. \tag{4}$$

The cross-spectral density (H_{xy}) between time series *x* and *y* is indicated in the equation above, while the power spectral densities (H_{xx} and H_{yy}) are indicated. In the frequency spectrum, *VD* gives a number between 0 and 1, where 0 denotes no correlation and 1 represents complete correlation.

While it is less susceptible to outside influences, imaginary coherence (*JD*) quantifies the relationship between the spectrum of two time series, much like squared coherence. It is described by the following Eq (5).

$$JD_{x,y}(f) = \frac{\mathrm{Imag}(H_{xy}(I))}{\sqrt{H_{xx}(I)H_{yy}(I)}}. \tag{5}$$

This equation uses H_{yy} to represent the power spectral density for both time series and H_{xy} to represent the cross-spectral density between the time series and H_{xx}.

The changes in the phase difference between two time series are measured by the phase locking value (PLV). This measure is described in Eq (6).

$$PLV = |G[\exp(i\Delta\phi_{\mathrm{rel}}\,(t))]|. \tag{6}$$

The phase difference is indicated in the above equation by $\Delta\varphi\mathrm{rel}(t)$ of *x* and *y* at time *t*, belonging to an interval from 0 to 2π, and the function *G* is averaging values from the time series data.

Similar to the preceding one, the phase lag index (PLI) takes phase sign changes into account. Eq (7) provides a description of this metric.

$$PLI = |G[\mathrm{sign}(\Delta\phi_{rel}(t))]|. \tag{7}$$

Similar to the PLI, the weighted phase lag index (WPLI) is less susceptible to outside influences. It is indicated by Eq (8).

$$WPLI = \frac{|G[|\mathrm{Imag}(H_{xy}(I))| * \mathrm{sign}\,(\mathrm{Imag}(H_{xy}(I)))]|}{G[|\mathrm{Imag}(H_{xy}(I))|]}. \tag{8}$$

Using the cross-spectral density between signals, represented by H_{xy}, this equation is comparable to the preceding one. The random impact of channel *j* on channel *i* for every frequency is described by the directed transfer function (DTF) using Eq (9).

$$DTF^2_{j\to i}(f) = \frac{|K_{i,j}(I)|^2}{\sum_{m=1}^{k} |K_{i,j}(I)|^2}. \tag{9}$$

An MVAR (multivariate autoregressive model) with distinct elements is represented by the transfer matrix *K*(*I*). The equation represents a normalized DTF with a range of 0 to 1. For all inflows to channel i, it shows the inflow ratio from channel *j* to channel i.

The difference between K and spectral entropy (SE) is in the frequency domain. Equation (10) explains it.

$$SE(Z) = \frac{\sum_{f=0}^{f^{S}/_{2}} p(I)\log_2(p(\mathrm{I}))}{\log_2(\text{psd.size})}. \tag{10}$$

The sampling frequency is f_s. $p(I)$ is the normalized power spectrum density. psd.size defines the size of the power spectrum density.

3.4. Spatial and spectral GNN architecture

To efficiently utilize both spatial and spectral characteristics taken from newborn EEG data, the spatial and spectral graph neural network (GNN) architecture was created. Convolutional layers in graphs: The graph representation of EEG data is processed by the GNN, which is made up of several graph convolutional layers. To update each node’s feature representation, each graph convolutional layer gathers data from nearby nodes to produce coarser representations of spectrum information, spatial convolutional layers collect spatial dependencies across EEG electrodes, and spectral convolutional layers capture spectral dependencies within each electrode within each node.

Pooling layers: To lower computing costs and down sample the graph representation, pooling layers are used. To produce coarser representations of spatial attributes, spatial pooling layers combine data from nearby nodes. To provide rough representations of spectral data, spectral pooling layers combine spectral information within each node. Each convolutional layer’s output is subjected to non-linear activation functions, such as rectified linear units (ReLU), which provide non-linearity and help the model understand intricate patterns. Activation functions allow the GNN to capture non-linear relationships in the data, hence improving its expressive capacity. Feature fusion: To efficiently incorporate spatial and spectral information, spatial and spectral features are fused at various points in the network design. Spatial and spectral representations are integrated using feature fusion procedures like element-wise addition or concatenation. Combining spectral and spatial variables improves the GNN’s ability to discriminate and allows it to extract complementing data from both modalities [16,17].

3.4.1. Spatial adjacency matrix

The spatial adjacency matrix, denoted as $A_{spatial}$, captures the spatial relationships between EEG electrodes. Each element A_{ij} of the matrix represents the strength of the connection between nodes i and j. The spatial adjacency matrix encodes the spatial topology of the graph and serves as the basis for spatial feature extraction. In the spatial domain, convolution can be stated as follows:

$$B_{16} = \sum_{i=1}^{16} w_{i,16} x_i, \tag{11}$$

where weight and feature are represented, respectively, by *w* and *x*. Convolution on the node *v* and *n*-channel convolution are thus represented by the following formulas:

$$B_v = \sum_{u\in M[v]} w_{u,v} x_u,$$
$$B_{v,n} = h\left(\sum_{i=1}^{f_{k-1}} \sum_{u\in M[v]} w_{i,n,u,v} x_{u,i}\right)(j = 1 \dots f_k). \tag{12}$$

The following formula can be used to compute the multilayer spatial convolution. The convolutional operation's input to the k-th layer $\{N_{k,i}; i = 1 \dots d_{k-1}\}$ depicts the collection of neighboring nodes or neighboring values for a feature map node.

$$B_{k,v,n} = h\left(\sum_{i=1}^{f_{k-1}} \sum_{u\in M[v]} w_{k,i,n,u,v} x_{k,u,i}\right) \atop (j = 1 \dots f_k, k = 1 \dots K). \tag{13}$$

Maximum pooling and average pooling are used to lessen the feature map after convolution.

3.4.2. Spectral representation

In addition to spatial connections, spectral features are incorporated into the graph representation. Spectral features capture the frequency characteristics of EEG signals and are integrated into the graph structure to augment spatial information. Spectral attributes associated with each node are computed from EEG signals using the Fourier transform or wavelet transform. These spectral attributes provide additional information about the dynamics of neural activity at different frequency bands [18–22]. To convert a picture from the spatial domain to the spectral domain, a sequence of bases should be constructed using the Laplacian matrix. After obtaining matrix P, the adjacent matrix M and degree matrix L can be computed. The following is an expression for the degree matrix and adjacency matrix.

$$M_{i,j} = \begin{cases} 1 & \text{if } i \in N_k[j], \\ 0 & \text{otherwise,} \end{cases}$$
$$L_{i,i} = \sum_j A_{i,j}, \tag{14}$$

where $V = [v_1, v_2, \dots, v_n]$ and V is the basis. V can be used to transform input from the spatial domain to the spectral domain. Convolution on a graph can therefore be computed as follows:

$$\begin{aligned} x_1 * x_2 &= \text{IGFT}(\text{GFT}(x_1) \odot \text{GFT}(x_2)), \\ &= V\left((V^T x_1) \odot (V^T x_2)\right), \\ &= V\left(\widetilde{x_1} \odot (V^T x_2)\right), \\ &= V(\text{diag}(\widetilde{x_1})(V^T x_2)), \\ &= (V\text{diag}(\widetilde{x_1})V^T)x_2, \end{aligned} \tag{15}$$

where IGFT stands for the inverse graph Fourier transform and GFT for the graph Fourier transform.

$$\begin{aligned} y = x_1 * x &= V\Lambda V^T x = Lx \\ &= V\left(\sum_{j=1}^{J} m_j \Lambda^j\right) V^T x_j. \end{aligned} \tag{16}$$

Lastly, the following expression can be used to represent the convolution operation on the k-th channel.

$$C_{k,j} = h\left(V \sum_{j=1}^{f_{k-1}} m_{k,j}^{\theta} \Lambda^{k,j} V^T x_{k,j}\right). \tag{17}$$

The rectified linear unit (ReLU) activation function is denoted by h. Examples of y_{spe} are as follows.

$$y_{\text{spe}} = \text{argmax}\left(\text{sigmoid}\left(\text{Relu}\left(O_{\text{spe}}F_{\text{spe}} + G_{\text{spe}}\right)\right)\right). \tag{18}$$

3.4.3. Integration of spectral and spatial characteristics

A unified representation of the newborn brain network is created by combining spectral and spatial characteristics. For every node, a feature matrix is produced by concatenating the spectral properties with the spatial adjacency matrix. Capturing both spectral and spatial information, this feature matrix allows the graph neural network to simultaneously learn from both kinds of dependencies. In conclusion, the spatial domain convolution operation can be represented as follows.

$$y_{\text{spa}} = \text{argmax}\left(\text{sigmoid}\left(\text{Relu}\left(O_{\text{spa}}\, x_{k+1,n} + G_{\text{spa}}\right)\right)\right). \tag{19}$$

The ultimate result can be reached by combining the SPEGNN and SPAGNN results at the decision level utilizing the following procedures.

$$\begin{aligned} y &= \text{argmax}\left(y_{\text{spe}}, y_{\text{spa}}\right), \\ &= \text{argmax}\left(\text{sigmoid}\left(\text{Relu}\left(W_{\text{spe}}C_{\text{spe}} + B_{\text{spe}}\right)\right),\right. \\ &\quad \left.\text{sigmoid}\left(\text{Relu}\left(W_{\text{spa}}\, x_{k+1,n} + B_{\text{spa}}\right)\right)\right). \end{aligned} \tag{20}$$

The built graph serves as the foundation for the spatial and spectral graph neural network. Its nodes represent EEG electrodes, and its edges indicate functional connections. The graph layout effectively facilitates seizure diagnosis and feature extraction by encapsulating the spatial and spectral properties of newborn EEG data. The suggested approach improves the accuracy and efficiency of seizure event identification in neonatal EEG data by allowing the spatial and spectral graph neural network to capture complicated connections through the creation of a graph-based representation that incorporates both spectral and spatial characteristics.

3.4.4. Aquila algorithm optimization

To maximize the model performance, the Aquila algorithm is incorporated into the GNN's training procedure. The approach integrates specialized optimization methods, such as gradient clipping, adjustable learning rates, and regularization procedures, designed for GNNs. In tasks involving the diagnosis of newborn seizures, the Aquila algorithm improves model convergence, stability, and generalization performance, resulting in increased accuracy and resilience. Adaptive learning rates: To dynamically modify the learning rate during training, the Aquila algorithm makes use of adaptive learning rates. With adaptive learning rates, the model can adjust its parameters more when the gradients are steep and less when they are shallow. This adaptive modification speeds up the training process and helps avoid convergence problems like oscillations or sluggish convergence. Gradient clipping: Gradient clipping is used to reduce the risk of an "exploding gradient", which can happen when gradients get too big during training. To maintain stability, gradients are scaled down proportionately if the gradient norm is over the maximum threshold established by the Aquila algorithm. Gradient clipping enhances training stability and convergence by keeping the model from diverging or fluctuating as a result of unnecessarily large gradients [23–25].

Regularization strategies: To avoid overfitting and enhance the GNN's generalization performance, the Aquila method uses regularization strategies like L1 and L2 regularization. To encourage sparsity in the learned weights, L1 regularization adds a penalty term to the loss function that is based on the absolute values of the model parameters. By including a penalty component based on the squared values of the model parameters, L2 regularization promotes smoother decision boundaries and smaller weight magnitudes [26,27]. The Aquila algorithm lessens the possibility of overfitting to the training set and enhances the model's capacity to generalize to new data by regularizing the model parameters. Advanced optimization techniques: To speed up convergence and increase training efficiency, the Aquila algorithm uses sophisticated optimization techniques like momentum to establish the direction of updates, while momentum-based optimization algorithms aggregate previous gradients. This allows for faster convergence and improved handling of noisy gradients. Based on each parameter's historical gradient, we adaptively modify the learning rates, resulting in more reliable and effective optimization. Using variable learning rates, gradient clipping, regularization techniques, advanced optimization methods, and bespoke loss functions, the Aquila algorithm optimizes the training of the spatial and spectral GNN. The Aquila algorithm increases the accuracy and efficiency of neonatal seizure recognition, which ultimately benefits clinical decision-making and patient outcomes in neonatal care. It does this by improving the stability, convergence speed, and generalization performance of the GNN [28].

Using the learned representations, the GNN's final output layer produces predictions for seizure and non-seizure states. To calculate class probabilities, the output layer is usually subjected to a sigmoid activation function. The output layer is optimized using suitable loss functions, such as binary cross-entropy loss or cross-entropy loss during training. The suggested method effectively integrates spatial and spectral information collected from neonatal EEG data, allowing for the precise and efficient diagnosis of seizure occurrences in newborns. It does this by integrating spatial and spectral GNN architecture improved with the Aquila algorithm.

Aquila optimization (AO) was influenced by the eagle's (Aquila) hunting tactics for a dynamic search method. These techniques enhance the optimization process overall by enabling AO to move between a global and local search in an adaptable manner. Because AO is dynamic, it may be applied to a wide range of optimization problems, whether they are discrete, linear, or nonlinear. Unlike some other algorithms, its mechanisms can be adjusted to meet the needs of a particular problem, which is not always the case. In several benchmark tests, it has been demonstrated that AO converges more quickly than other metaheuristics. Its adaptive techniques provide a rapid reduction of the search space to regions that show promise, accelerating convergence without sacrificing the quality of the answer. Comparing AO to certain other complex optimization techniques, it is comparatively simple to implement. Comparing AO to certain other complex optimization techniques, it is comparatively simple to implement. It is a desirable option for practitioners and researchers due to strong performance and straightforward conceptual design. Convergence is shown in Figure 9 and compared to different optimization algorithms.

Table 1. Hyperparameter optimization.

Pseudo Code for Aquila Algorithm Optimization

```
Aquila Algorithm Optimized(signal):
    threshold = calculate Threshold (signal)
    spikes = detect Spikes (signal, threshold)
    seizures = detect Seizures (spikes)
    return seizures
function calculate Threshold(signal):
    MAD = calculate MAD (signal)
    threshold = k * MAD
    return threshold
function calculate MAD (signal):
    median = calculate Median (signal)
    deviations = []
    for each value in signal:
        deviation = abs(value - median)
        deviations.append(deviation)
    MAD = calculate Median (deviations)
    return MAD
function calculate Median (values):
    sorted_values = sort(values)
    n = length(sorted_values)
    if n mod 2 == 0:
        median = (sorted_values[n/2] + sorted_values[(n/2) + 1]) / 2
    else:
        median = sorted_values[(n+1)/2]
    return median
function detect Spikes (signal, threshold):
    spikes = []
    for i from 1 to length(signal):
        if signal[i] > threshold:
            if Spike (signal, i):
                spikes.append(i)
    return spikes
Spike (signal, index):
    if signal[index] is significantly higher than signal[index-1] and signal[index+1]
    if signal[index] > signal[index-1] and signal[index] > signal[index+1]:
        return True
    else:
        return False
function detect Seizures (spikes):
    seizures = []
    return seizures
```

Pseudo Code for Hyperparameter Optimization by Aquila Optimizer with Deep Learning

```
eeg_signals = LoadEEGSignals()
features = ExtractFeatures(eeg_signals)
labels = LoadLabels()
node_features = features
edges = DefineEdges(eeg_signals)
graph_data = CreateGraph(node_features, edges)
Function TrainModel(graph_data, labels, num_epochs=100):
    model = SeizureGNN(input_dim=graph_data.num_node_features, hidden_dim=64, output_dim=2)
    criterion = CrossEntropyLoss()
    optimizer = AquilaOptimizer(model.parameters(), learning_rate=0.001)
        For epoch from 1 to num_epochs:
        optimizer.zero_grad()
        outputs = model(graph_data)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        Print("Epoch", epoch, "Loss:", loss.item())
        Return model
trained_model = TrainModel(model,graph_data, labels)
model.fit(trained_model,test _graph _data)
```

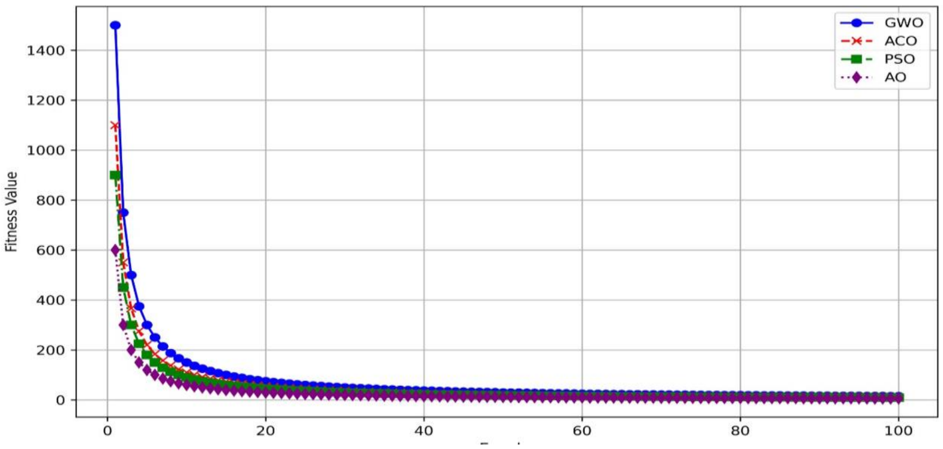


Figure 9. Convergence with different optimization algorithms.

A convergence graph comparing the results of four distinct optimization algorithms—gray wolf optimization (GWO), Aquila optimization (AO), particle swarm optimization (PSO), and ant colony optimization (ACO)—is shown in Figure 9. The convergence of Aquila optimization (AO) seems to be the best and quickest because its line approaches the lowest fitness value the fastest. Aquila optimization appears to be closely followed by grey wolf optimization (GWO). Out of the four, it appears that ant colony optimization (ACO) and particle swarm optimization (PSO) have the slowest convergence rates.

4. Results and discussion

The steps involved in training and assessing the spatial and spectral graph neural network (SSGNN) are augmented by the Aquila algorithm. Preparing the dataset: EEG recordings from newborns who had seizures and those who did not are sourced from the Zenodo dataset, which can be found at https://zenodo.org/records/1280684. The dataset is partitioned using a ratio like 70-15-15 or 80-10-10 to create training, validation, and testing sets. To avoid bias, care is made to make sure that each set includes a fairly balanced proportion of seizure and non-seizure samples. Model training: Deep learning frameworks like PyTorch or TensorFlow are used to create the spatial and spectral GNN architecture, which has been improved by the Aquila algorithm. Using the backpropagation algorithm and the Aquila optimization techniques, the model is trained on the training set. The model gains the ability to map input EEG graph data to either the seizure or non-seizure class labels during training.

Table 2 represents the hyperparameters. To maximize the model performance, a validation set is used to adjust the hyperparameters such as learning rate, batch size, number of layers, and hidden units. To explore the hyperparameter space and find the ideal configuration, methods like grid search or random search can be used. Model evaluation: The trained model's performance in identifying newborn seizures is assessed using the held-out testing set.

Table 2. Hyperparameters.

Hyperparameters	
Batch Size	32
Epochs	100
Optimizer	Aquila Optimizer
Learning Rate	0.00212
Graph Pooling	Average Pooling
Dropout	0.3
Loss	Binary Cross Entropy
Activation	Relu
Output Activation	Sigmoid

Figure 10 shows the accuracy with which various machine learning and deep learning algorithms categorize ceramic samples. For accuracy, the proposed, CNN, GNN, LSTM, and SVM models are displayed on the *x*-axis. The accuracy is displayed on the *y*-axis and is expressed as a percentage. The correctness of 98.7% of the proposed model is represented. The CNN model's accuracy is shown as 93.16%. The GNN model's accuracy is 92%, and the accuracy of the LSTM model is 89.73%. The SVM model's accuracy is 90%. The results indicate that the proposed model has the highest accuracy, followed by the CNN, GNN, LSTM, and SVM models. For the F-score, the proposed, CNN, GNN, LSTM, and SVM models are displayed on the *x*-axis. The F-score is displayed on the *y*-axis and is expressed as a percentage. The F-score of 98.54% of the proposed model is represented. The CNN model displays 94.13%. The GNN model shows 92.42%. The LSTM model shows 90.17%. The SVM model's F-score is 91.68%. The results indicate that the proposed model has the highest F-score, followed by the CNN, GNN, LSTM, and SVM models. For precision, the proposed, CNN, GNN, LSTM, and SVM models are shown on the *x*-axis. The precision is presented on the *y*-axis and is expressed as a percentage. The precision of the proposed model is represented as 98.9%. The CNN model displays 94.02%. The GNN model displays 92.67%. The LSTM model shows 90.13%. The SVM model's precision is 91.87%. The results indicate that the proposed model has the highest precision, followed by the CNN, GNN, LSTM, and SVM models. For recall, the proposed, CNN, GNN, LSTM, and SVM models are shown on the *x*-axis. The recall is presented on the *y*-axis and is expressed as a percentage. A recall of 98.19% is represented for the proposed model. The CNN model displays 94.24%. The GNN model shows 92.17%. The LSTM model shows 90.21%. The SVM model's recall is 91.49%. The results designate that the proposed model has the highest recall, followed by the CNN, GNN, LSTM, and SVM models.

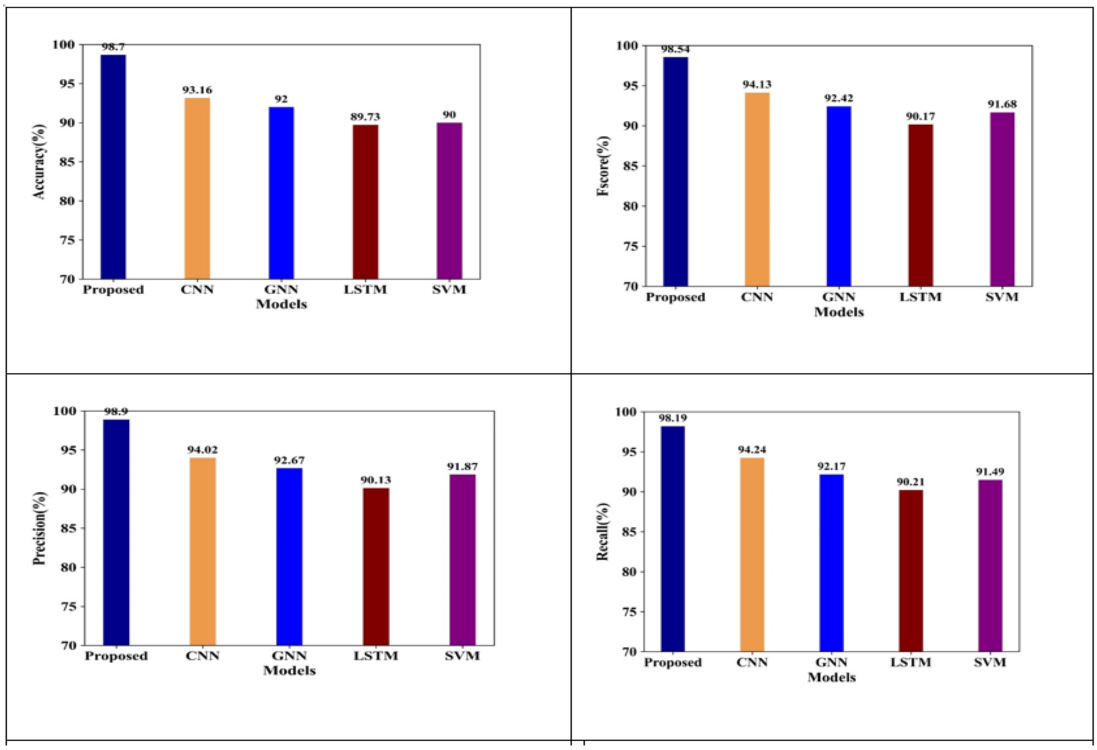


Figure 10. Accuracy, F-score, precision, and recall.

Figure 11 shows the specificity of the proposed, CNN, GNN, LSTM, and SVM models. The models are shown on the *x*-axis. The specificity is presented on the *y*-axis and is expressed as a percentage. A specificity of 98.79% is represented for the proposed model. The CNN model displays 95.02%. The GNN model shows 93%, and LSTM shows 91.32%. The SVM model's specificity is 91.95%. The results designate that the proposed model has the highest specificity, followed by the CNN, GNN, LSTM, and SVM models.

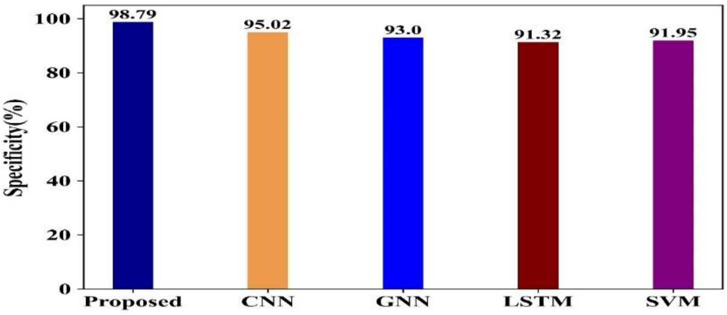


Figure 11. Specificity.

To measure the performance of the model, evaluation metrics including F-score, area under the receiver operating characteristic curve (AUC-ROC), sensitivity, specificity, and accuracy are calculated. To understand the advantages and disadvantages of the model, confusion matrices and precision-recall curves can also be examined. A comparison with baseline methods: The suggested technique was contrasted with baseline methods, such as deep learning architectures and conventional machine learning algorithms. SVM, LSTM, GNN, and CNN networks trained on the same dataset are examples of baseline techniques. Table 3 shows the comparison of the proposed model with different algorithms.

Table 3. A comparison of the proposed model with different algorithms.

Algorithms	**Proposed**	**CNN**	**GNN**	**LSTM**	**SVM**
Parameters					
Accuracy	98.7	93.16	92.00	89.73	90
Precision	98.9	94.03	92.67	90.13	91.87
Recall	98.19	94.25	92.17	90.21	91.48
F-score	98.54	94.13	92.45	90.17	91.67
Specificity	98.79	95.03	93.00	91.32	91.9487

Figure 12 shows the AUC-ROC of the proposed model (0.975), CNN (0.940), GNN (0.935), LSTM (0.91), and SVM (0.90).

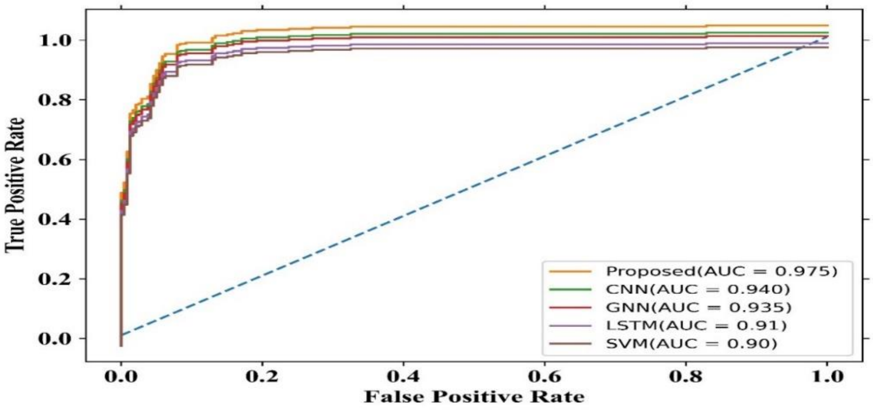
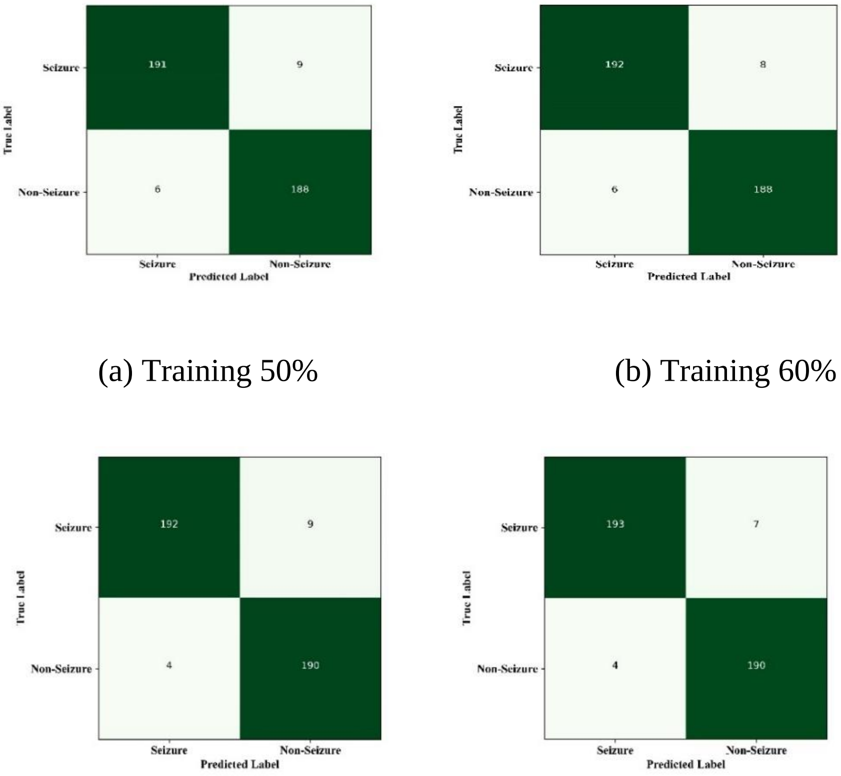


Figure 12. AUC-ROC.

Figure 13 shows the confusion matrix for training 50%, 60%, 70%, and 80% as well as testing as 50%, 40%, 30%, and 20% represents above values.

The accuracy and loss curves for the suggested SSGNN model during the training and validation stages are displayed in Figure 14. These curves reveal information about the model's ability to learn from and generalize the data. In order to make sure that the SSGNN model does not overfit or underfit the data, it is essential to assess both curves' convergence and stability. The graphic demonstrates how the model may minimize loss while achieving excellent accuracy.



(a) Training 50%

(b) Training 60%

(c) Training 70%

(d) Training 80%

Figure 13. Confusion matrix.

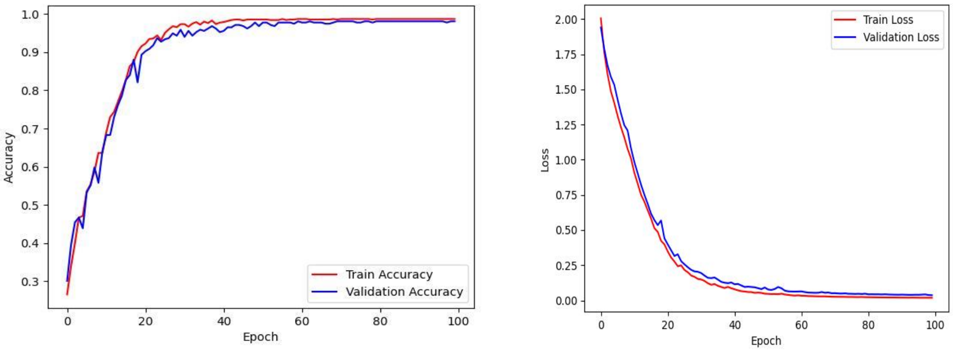


Figure 14. Accuracy-loss curve.

Figure 15 shows the K-fold with different parameters for comparing diverse deep learning and machine learning algorithms.

Cross-validation: To evaluate how resilient the trained model is, cross-validation methods like K-fold cross-validation can be used. The dataset is split up into *k* folds; the remaining folds are used for training, and each fold is used once as a validation set. An estimate of the model's generalization performance that is more accurate is obtained by averaging performance metrics over several folds. Key aspects and locations that aid in the identification of seizures can be visualized by using interpretability techniques like saliency maps or attention mechanisms. These procedures will help to improve clinical decision-making and patient care in neonatal critical care units by efficiently training and evaluating the spatial and spectral GNN supplemented with the Aquila algorithm for intelligent neonatal seizure diagnosis.

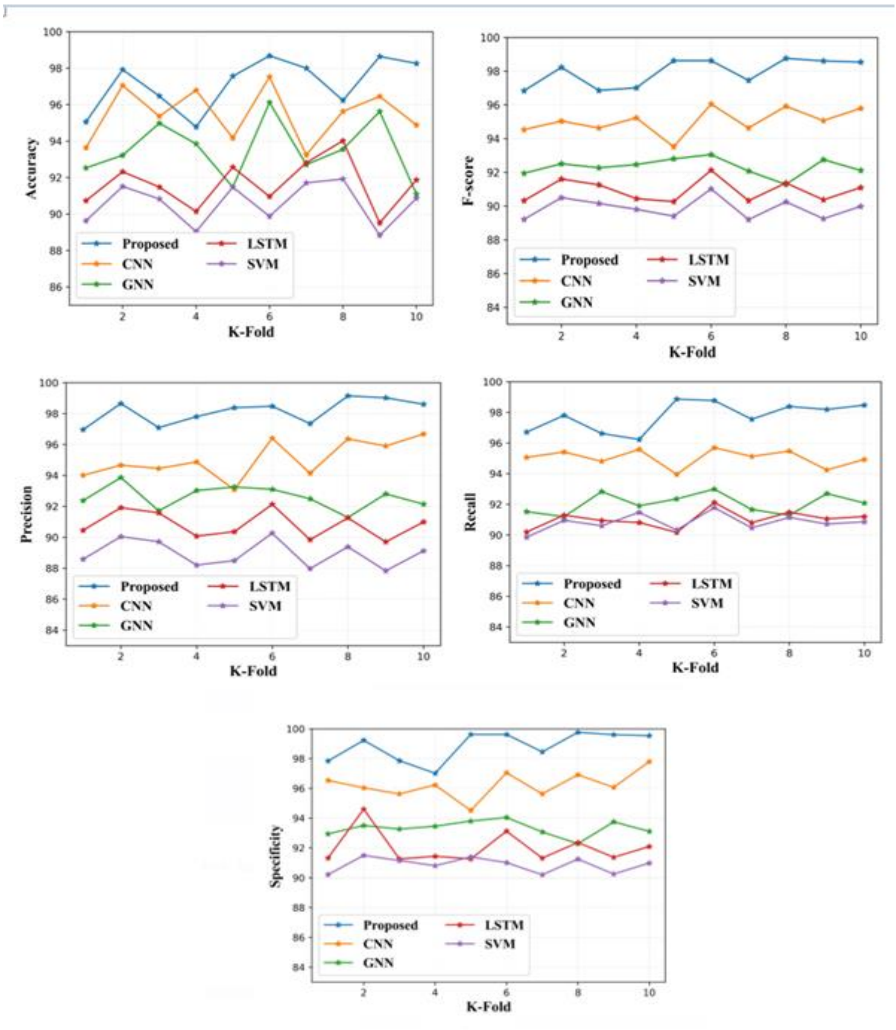


Figure 15. K-fold.

5. Conclusions

Aquila algorithm-enhanced spatial and spectral graph neural networks (SSGNN) are a breakthrough in the field of intelligent newborn seizure diagnosis. Accurate and effective seizure identification is made possible by the model, which provides a thorough and nuanced understanding of neonatal brain dynamics by utilizing both spatial and spectral information collected from neonatal EEG data. The model uses spectral and spatial GNN architecture to distinguish between seizure and non-seizure states by utilizing spectrum features and spatial connectivity to capture complex correlations in neonatal EEG signals. Improving model convergence, stability, and generalization performance, the Aquila algorithm further optimizes the training procedure. The suggested method is guaranteed to be repeatable and broadly applicable by using publicly accessible datasets, such as the one stored on Zenodo. The effectiveness of the model in identifying newborn seizures is confirmed by a comparison with baseline approaches and a thorough evaluation of its performance using standard measures. For affected infants, the model may lower the chance of long-term neurological issues by facilitating prompt intervention and treatment. The future understanding of the model's decision-making process and finding putative biomarkers for newborn seizures are made easier with the help of visualization.

Author contributions

Conceptualization, M.N.; methodology, S.R.; software, M.N.; validation, S.R., and M.N.; formal analysis, O.I.K.; investigation, S.R.; resources, O.I.K.; data curation, M.N.; writing—original draft preparation, S.R.; writing—review and editing, S.R.; visualization, H.H.; supervision, M.N.; project administration, S.R.; funding acquisition, O.I.K, H.H. All authors have read and agreed to the published version of the manuscript.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgements

The authors thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the New Brunswick Innovation Foundation (NBIF) for the financial support of the global project. These granting agencies did not contribute to the design of the study and collection, analysis, and interpretation of the data.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. F. A. Jibon, M. H. Miraz, M. U. Khandaker, M. Rashdan, M. Salman, A. Tasbir, et al., Epileptic seizure detection from electroencephalogram (EEG) signals using linear graph convolutional network and DenseNet based hybrid framework, *J. Radiat. Res. Appl. Sci.*, **16** (2023), 100607. https://doi.org/10.1016/j.jrras.2023.100607

2. D. Grattarola, L. Livi, C. Alippi, R. Wennberg, T. A. Valiante, Seizure localisation with attention-based graph neural networks, *Expert Syst. Appl.*, **203** (2022), 117330. https://doi.org/10.1016/j.eswa.2022.117330
3. A. Shoeibi, M. Khodatars, N. Ghassemi, M. Jafari, P. Moridian, R. Alizadehsani, et al., Epileptic seizures detection using deep learning techniques: A review, *Int. J. Environ. Res. Public Health.*, **18** (2021), 5780. https://doi.org/10.3390/ijerph18115780
4. A. Gramacki, J. Gramacki, A deep learning framework for epileptic seizure detection based on neonatal EEG signals. *Sci. Rep.*, **12** (2022), 13010. https://doi.org/10.1038/s41598-022-15830-2
5. K. Rezaee, E. Azizi, J. Haddadnia, Optimized seizure detection algorithm: A fast approach for onset of epileptic in EEG signals using GT discriminant analysis and K-NN classifier, *J. Biomed. Phys. Eng.*, **6** (2016), 81–94.
6. A. Temko, E. Thomas, W. Marnane, G. Lightbody, G. Boylan, EEG-based neonatal seizure detection with Support Vector Machines, *Clin. Neurophysiol.*, **122** (2011), 464–473. https://doi.org/10.1016/j.clinph.2010.06.034
7. A. Kukker, R. Sharma, A Genetic Algorithm Assisted Fuzzy Q-Learning epileptic seizure classifier, *Comput. Electr. Eng.*, **92** (2021), 107154. https://doi.org/10.1016/j.compeleceng.2021.107154
8. M. K. Jareda, R. Sharma, A. Kukker, EEG signal based seizure classification using wavelet transform, In: *2019 International Conference on Computing, Power and Communication Technologies (GUCON)*, 2019, 537–539.
9. M. Nelson, S. Rajendran, Y. Alotaibi, Vision graph neural network-based neonatal identification to avoid swapping and abduction, *AIMS Mathematics*, **8** (2023), 21554–21571. https://doi.org/10.3934/math.20231098
10. N. Stevenson, K. Tapani, L. Lauronen, S. Vanhatalo, A dataset of neonatal EEG recordings with seizures annotations, 2018. Available from: https://zenodo.org/records/1280684.
11. F. A. Jibon, M. H. Miraz, M. U. Khandaker, M. Rashdan, M. Salman, A. Tasbir, et al., Epileptic seizure detection from electroencephalogram (EEG) signals using linear graph convolutional network and DenseNet based hybrid framework, *J. Radiat. Res. Appl. Sci.*, **16** (2023), 100607. https://doi.org/10.1016/j.jrras.2023.100607
12. A. Nogales, Á. J. García-Tejedor, P. Chazarra, A. Ugalde-Canitrot, Discriminating and understanding brain states in children with epileptic spasms using deep learning and graph metrics analysis of brain connectivity, *Comput. Meth. Prog. Biomed.*, **232** (2023), 107427. https://doi.org/10.1016/j.cmpb.2023.107427
13. K. Raeisi, M. Khazaei, P. Croce, G. Tamburro, S. Comani, F. Zappasodi, A graph convolutional neural network for the automated detection of seizures in the neonatal EEG, *Comput. Meth. Prog. Biomed.*, **222** (2022), 106950. https://doi.org/10.1016/j.cmpb.2022.106950
14. J. Zhang, G. Sun, K. Zheng, S. Mazhar, X. Fu, Y. Li, et al., SSGNN: A Macro and Microfacial expression recognition graph neural network combining spatial and spectral domain features, *IEEE T. Hum.-Mach. Syst.*, **52** (2022), 747–760. https://doi.org/10.1109/THMS.2022.3163211
15. S. Rajendran, O. I. Khalaf, Y. Alotaibi, S. Alghamdi, MapReduce-based big data classification model using feature subset selection and hyperparameter tuned deep belief network, *Sci. Rep.*, **11** (2021), 24138. https://doi.org/10.1038/s41598-021-03019-y
16. Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et al., Handwritten digit recognition with a back-propagation network, In: *NIPS'89: Proceedings of the 2nd International Conference on Neural Information Processing Systems*, 1989, 396–404.

17. T. Tamilvizhi, R. Surendran, K. Anbazhagan, K. Rajkumar, Quantum behaved particle swarm optimization-based deep transfer learning model for sugarcane leaf disease detection and classification, *Math. Probl. Eng.*, **2022** (2022), 3452413. https://doi.org/10.1155/2022/3452413
18. R. Surendran, O. I. Khalaf, C. A. T. Romero, Deep learning based intelligent industrial fault diagnosis model, *Comput. Mater. Con.*, **70** (2022), 6323–6338. https://doi.org/10.32604/cmc.2022.021716
19. A. M. Maitin, A. Nogales, P. Chazarra, Á. J. García-Tejedor, EEGraph: An open-source python library for modeling electroencephalograms using graphs, *Neurocomputing*, **519** (2022), 127–134. https://doi.org/10.1016/j.neucom.2022.11.050
20. A. M. Maitin, J. P. R. Muñoz, Á. J. García-Tejedor, Survey of machine learning techniques in the analysis of EEG signals for Parkinson's disease: A systematic review, *Appl. Sci*., **12** (2022), 6967. https://doi.org/10.3390/app12146967
21. L. Hug, M. Alexander, D. You, L. Alkema, National, regional, and global levels and trends in neonatal mortality between 1990 and 2017, with scenario-based projections to 2030: A systematic analysis, *Lancet Glob. Health*, **7** (2019), E710–E720. https://doi.org/10.1016/S2214-109X(19)30163-9
22. O. Meinich-Bache, S. L. Austnes, K. Engan, I. Austvoll, T. Eftestøl, H. Myklebust, et al., Activity recognition from newborn resuscitation videos, *IEEE J. Biomed. Health*, **24** (2020), 3258–3267. https://doi.org/10.1109/JBHI.2020.2978252
23. Laith Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. A. Al-qaness, A. H. Gandomi, Aquila optimizer: A novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.*, **157** (2021), 107250. https://doi.org/10.1016/j.cie.2021.107250
24. K. A. Ogudo, R. Surendran, O. I. Khalaf, Optimal artificial intelligence-based automated skin lesion detection and classification model, *Comput. Syst. Sci. Eng.*, **44** (2023), 693–707. https://doi.org/10.32604/csse.2023.024154
25. S. Deepthi, P. S. Arun, Recognition of new born babies using multi class SVM, In: *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 2017. https://doi.org/10.1109/ICCPCT.2017.8074303
26. X. Ji, Z. Dong, Y. Han, C. S. Lai, G. Zhou, D. Qi, EMSN: An energy-efficient memristive sequencer network for human emotion classification in mental health monitoring, *IEEE T. Consum. Electr.*, **69** (2023), 1005–1016. https://doi.org/10.1109/TCE.2023.3263672
27. T. Thanarajan, Y. Alotaibi, S. Rajendran, K. Nagappan, Improved wolf swarm optimization with deep-learning-based movement analysis and self-regulated human activity recognition, *AIMS Mathematics*, **8** (2023), 12520–12539. https://doi.org/10.3934/math.2023629
28. S. Rajagopal, T. Thanarajan, Y. Alotaibi, S. Alghamdi, Brain tumor: Hybrid feature extraction based on UNET and 3DCNN, *Comput. Syst. Sci. Eng.*, **45** (2023), 2093–2109. https://doi.org/10.32604/csse.2023.032488

AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)