BioMedInformatics

MDPI

Article

Real-Time EEG-Based Anaesthesia Monitoring Using Neural Networks on Edge-AI Hardware

Brendan Mullane [1,*], Dominik Przychodni [2] and Chris Crispin-Bailey [3]

[1] Department of Electronic and Computer Engineering, University of Limerick, V94 T9PX Limerick, Ireland
[2] Independent Researcher, County Clare, Ireland; dominik4658@gmail.com
[3] Department of Computer Science, University of York, York YO10 5GH, UK; christopher.crispin-bailey@york.ac.uk
* Correspondence: brendan.mullane@ul.ie

Abstract

Background: Monitoring unconsciousness under general anaesthesia typically relies on indirect physiological indicators rather than direct measures of brain activity. This study investigates a compact, real-time electroencephalography (EEG)-based monitoring system that combines machine learning (ML) and edge artificial intelligence (Edge-AI) for assessing anaesthesia-induced unconsciousness. **Methods**: Multitaper spectral analysis (MSA) was incorporated into the real-time processing pipeline to enable feature extraction from raw EEG prior to inference using LR, LDA, LDA+LR, NN and QNN classifier models. A compact neural network was developed using spectral features from 2 s EEG epochs obtained from an available third-party dataset and evaluated using subject-independent five-fold cross-validation. Across the evaluated classifiers and class-balancing strategies, the neural network trained using the Synthetic Minority Over-sampling Technique (SMOTE) provided the strongest overall performance. The selected model was subsequently quantised to 8-bit precision and deployed as a quantised neural network on an untethered PYNQ-Z2 FPGA module. **Results**: The neural network achieved an accuracy of 0.90 under subject-independent evaluation. Following 8-bit quantisation, the FPGA-deployed model delivered an accuracy of 0.88, while achieving an inference-stage latency below 0.8 ms and thus a computational speed-up exceeding 2.7×. **Conclusions**: These results demonstrate the feasibility of combining subject-independent EEG-based classification with efficient, low-latency Edge-AI implementation for real-time anaesthesia monitoring. The proposed system provides a foundation for the development of EEG-based decision-support tools for anaesthesiologists and has potential for future deployment in resource-constrained or remote monitoring environments.

Keywords: Edge-AI; EEG anaesthesia monitoring; FPGA computing; machine learning; multitaper spectral processing; neural networks

Check for updates

Academic Editor: James C. L. Chow

Received: 5 August 2026
Revised: 18 September 2026
Accepted: 20 September 2026
Published: 24 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Electroencephalography (EEG) is well established as a non-invasive method for monitoring electrical brain activity using electrodes placed on the scalp. It has been widely adopted in clinical and research contexts to assess neurological conditions, sleep quality, cognitive state, and to predict seizures in epilepsy patients [1–7]. With recent advances in signal processing and the growth of Edge Artificial Intelligence (Edge-AI), EEG data can now be leveraged in real-time using compact, energy-efficient computing platforms and applied to anaesthesia monitoring.

The complexity of EEG monitoring systems varies widely, from simple and relatively convenient two-point sensor devices, through to highly complex measurement arrays, such as highlighted in Figure 1. The use of some of these systems in the monitoring of consciousness, both in brain trauma monitoring and in anaesthesia applications, is an active field [8–12].

Monitoring of levels of unconsciousness can be envisaged in many scenarios, as highlighted in Figure 2. Careful monitoring in operating theatre situations is perhaps the most obvious case, where it is necessary to ensure both that consciousness does not resurface during a medical procedure but also that anaesthesia is not over-administered, leading to additional medical consequences. However, the ability to deploy monitoring systems in wider usages is desirable. First responders dealing with head injury, effects of drug or toxic effects, or needing to sedate patients in situ, could greatly benefit from portable versions of such systems.

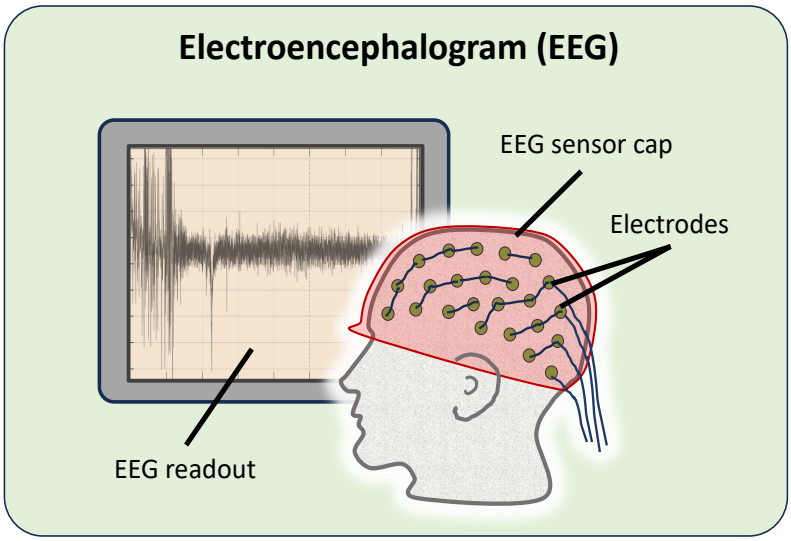


Figure 1. A complex EEG monitoring scenario. A patient wears a tight-fitting EEG sensor cap over the cranium, consisting of electrodes that sense electrical fluctuations attributable to internal brain activity. A network of wires then transfers those signals to an amplification module and a processing element (typically a computer) where the signals are acquired and then subsequently processed for display on an EEG readout.

Whilst immediate feedback is of critical value, the behavior of EEG patterns during such circumstances can have a significant relevance to long term outcomes and subsequent after-care treatment [12–14]. Recording and analysis of such data is therefore of interest, as may be real-time analysis.

General Anaesthesia (GA) induces a controlled, reversible unconscious state necessary for many surgical procedures. Administered by anaesthesiologists using intravenous or inhaled medications, GA suppresses pain perception and awareness while vital signs are monitored [9,15]. Although generally safe, GA carries risks, especially in elderly patients, including pneumonia, stroke, or cardiac events. Traditionally, anaesthesiologists infer consciousness state using indirect indicators, as direct brain monitoring is not standard practice.

This paper presents a real-time EEG-based system for patients undergoing GA to detect Loss of Consciousness (LOC) and Return of Consciousness (ROC) using signal-processing and machine learning (ML) techniques. The system design methods in this study, relied on use of EEG data from an open-access PhysioNet dataset [10,16], where subjects received anaesthetic agents (propofol and/or sevoflurane) to induce unconsciousness. Rather than replacing clinical judgment, the presented embedded AI system design provides a secondary source of insight into patient sedation.

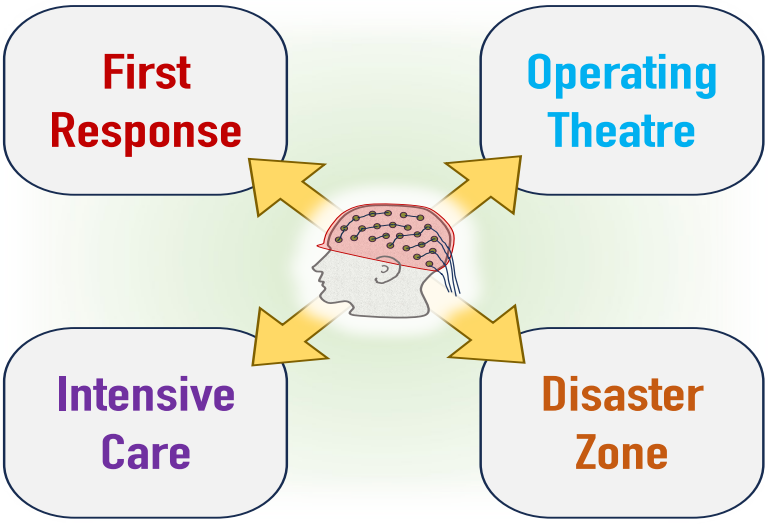


Figure 2. Unconsciousness monitoring scenarios, where feasibility of deployment varies in terms of device form-factor and portability. Deployability is high for operating-theatre anaesthesia and intensive care scenarios, where resources are readily available, but lower for other cases where remoteness or limitations on resources are key constraints.

In this study, a physical implementation is explored whereby EEG signals are captured using a proprietary wearable headband equipped with frontal electrodes, connected to a Cyton open brain–computer interface (OpenBCI) biosensing board [17]. The Cyton module then wirelessly transmits amplified EEG electrode signals for receipt and processing with real-time inference to a chosen host module. Although the headset is capable of capturing multiple EEG channels, the work undertaken in this study focussed upon the use of minimal data-stream evaluations, limiting to a single Fp1 channel, thereby minimising computational effort in the signal preprocessing stages. A more detailed discussion and rationale for this arrangement is given in the later methodology section.

In this paper, the authors explore the use of an FPGA component as host, specifically the PYNQ-Z2 board [18], which integrates an on-board ARM processor core with customisable FPGA logic. This provides the option of implementing any parts of the signal processing pipeline in software (running on the CPU core) or in hardware (utilising custom logic). The motivation for using such an approach is to develop an accurate, low-power system suitable for clinical environments, but without dependence on bulky, expensive and power hungry equipment.

This research work made use of a precomputed multitaper spectrum dataset derived from live EEG data gathered during operating theatre and volunteer tests conducted by the dataset owners [10], as hosted on PhysioNet.org. An important limitation therefore is that, whilst the described hardware implementation provides an end-to-end implementation, that implementation could not be tested with the original raw EEG patient data. Instead, it was implemented to perform the same processing front-end with the same filters, parameters, and configurations described by the original data set provider for their multitaper spectrum generation, and validated as operational with other methods. The authors consider the implemented system to be equivalent but do not provide a one-to-one verification against the data set provider's tool-chain.

The remainder of the paper is organised as follows: A brief review of current work in the field is given in Section 2, while the methodology and focus of this work is detailed in Section 3. Experimental results and discussion are presented with specific attention given to ML aspects in Section 4.1 and Edge-AI insights in Section 4.2. Conclusions are then presented in Section 5, respectively.

Key contributions to this study include:

- Utilising a subject-independent cross-validation ML model evaluation framework.
- Deployment of a quantised neural network (QNN) on a Field Programmable Gate Array (FPGA), demonstrating Edge-AI for real-time EEG-based unconsciousness classification.
- Integration of MSA, ML and QNN processing on the PYNQ-Z2 platform, wirelessly connected to the OpenBCI head gear for a unified low-latency solution.
- Empirical comparison of inference performance and power efficiency on edge (FPGA versus CPU).
- Demonstration of a clinically relevant, low-resource EEG monitoring tool suitable for embedded medical applications.

In this field of work there are a number of papers making quantitative comparisons of different algorithmic methodologies in EEG analysis (e.g., [19,20]), and also other works describing implementations of specific algorithms on FPGA [21,22]. However, there appear to be far fewer studies that collectively compare a number of algorithms, where each is implemented on the same target FPGA platform and with the same EEG signal pre-processing chain. The authors therefore believe that this paper makes a contribution to addressing that knowledge gap and in comparing the performance and trade-offs of a selection of algorithms implemented with the same resource-constrained hardware platform. This work demonstrates that the least favourable software algorithm (NN based on CPU implementation) can actually deliver the best performance trade-offs when implemented in hardware (as an equivalent QNN implemented in custom FPGA logic blocks).

2. Related Works

Several comprehensive review articles have summarised the application of EEG and ML for consciousness monitoring during general anaesthesia [11,12,23], and more widely in related fields [24,25]. Rather than duplicating these extensive reviews, this section focuses on the areas most relevant to the present study and build upon prior research demonstrating the feasibility of using EEG-ML to detect anaesthesia-induced unconsciousness. Specifically, it reviews previous work on EEG-based consciousness monitoring, EEG acquisition methodologies, hardware implementations, and ML approaches for real-time anaesthesia monitoring. These topics provide the background and motivation for the ML framework proposed in this paper.

2.1. EEG-Based Consciousness Monitoring

Depth-of-anaesthesia monitors (DoA) typically use a limited frontal EEG montage (1–2 channels) because frontal signals reliably reflect anaesthesia-related spectral changes associated with loss and recovery of consciousness [26]. The Bispectral Index (BIS), one of the most widely used processed EEG depth monitors, is derived from frontal EEG for use during GA based on correlations between frontal EEG patterns and clinical DoA [27].

Research comparing EEG channel locations for DoA classification showed that frontal locations provided the best discrimination between awake and anaesthetised states, supporting the use of a single frontal channel in monitoring [28].

Many approaches have been proposed for automating the measurement of depth of consciousness/unconsciousness [29,30]. A notable study by Abel et al. [16] introduced a logistic regression classifier trained on multitaper spectral features from healthy volunteers administered propofol based on frontal electrode placement. Their work addressed the scarcity of EEG datasets by releasing high-quality annotated data for public use. A high accuracy was achieved (median AUC of 0.986, accuracy of 0.91), although its deployment was limited to software-based offline analysis. Khan et al. [31] also explored ML models for estimating DoA using a single EEG channel, reporting accuracy of 79% using novel feature

extraction pipelines. A useful study by Schmierer et al. [32] reviewed many deep learning ML methods which model the complex and non-linear relationship between EEG signals and DoA, including the use of Convolution Neural Network (CNN) models [33] that can be computationally intensive for more advanced architectures.

Although these efforts advanced the state of the art, they did not include hardware deployment or considerations for real-time, resource-constrained environments. The ability to deploy such technology for real-time monitoring, with limited resources, is an area of great interest for disaster-emergency response scenarios, use in global regions with limited infrastructure and in first-response situations. To further explore solutions for this domain, this work focuses on efficient, Edge-AI compatible deployment of EEG classifiers using compact neural networks and low-power FPGA hardware. It is demonstrated that real-time inference is feasible with acceptable accuracy trade-offs, even using a single EEG channel.

2.2. EEG Acquisition and System Trade-Offs

EEG hardware varies in channel count, cost, and signal fidelity. Clinical systems often support 32–64 channels, but simpler setups can suffice for targeted applications like anaesthesia monitoring. The OpenBCI Cyton board was chosen for its balance of affordability, wireless connectivity, and compatibility with Python-based workflows [17]. The Cyton offers eight channels and has shown comparable performance to more expensive clinical-grade amplifiers [34]. For high-fidelity EEG capture in portable settings, the Bittium BrainStatus™ system [35] offers 16 channels, wireless operation, and medical device certification.

However, due to cost and form-factor constraints, the OpenBCI platform with a frontal electrode configuration was selected. The PYNQ-Z2 platform (Figure 3) serves as the processing unit. It features a Zynq system-on-chip (SoC) that combines programmable logic with ARM processors, enabling efficient filtering, spectral analysis, and neural network inference. Prior work has demonstrated its suitability for medical-grade signal processing with low error and distortion [36].

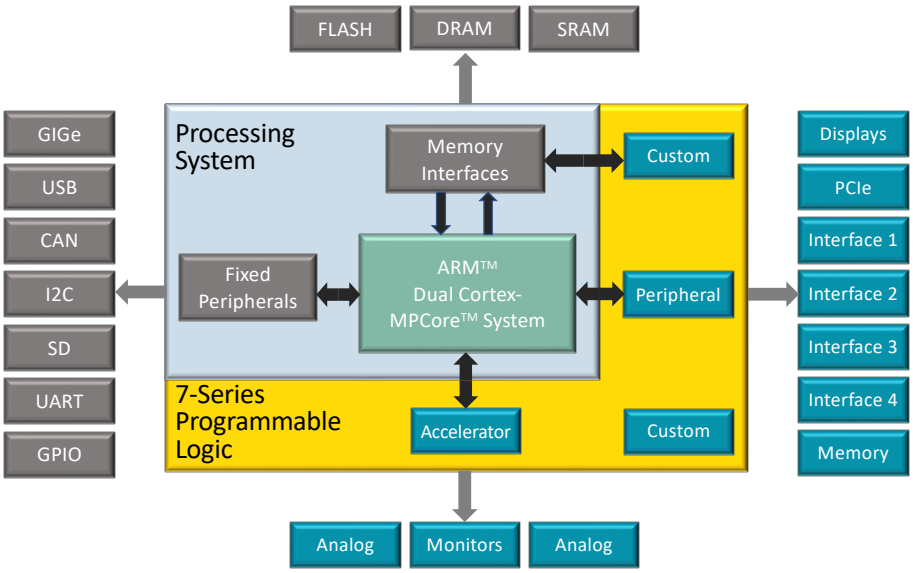


Figure 3. PYNQ programmable SoC architecture showing the ARM processing system and fixed peripheral interfaces, with USB being utilised for connection to the Cyton board, and on-chip FPGA fabric utilised for custom logic implementations (as discussed later). Diagram redrawn based on [18].

2.3. Hardware-Based EEG Monitoring Systems

Whilst there are a variety of front-end signal acquisition systems available, such as OpenBCI, the implementation of complete EEG monitoring systems still requires signal conditioning, data processing capability, decision making 'intelligence', and ultimately a human interface feedback element. Many systems designed for what might be termed 'permanently located deployment' are based upon large equipment footprints, often based upon desktop PC (or equivalent) scales of technology at the back-end, and having ample power and space available. The most obvious example of this is the operating theatre or the intensive care unit (ICU). However, there has been a thread of work in the field that relates to the optimisation of systems at the hardware level with the clear opportunity to improve cost factors including size, portability, power consumption, and other criteria, including recent reviews of these topics [37]. Thus the potential for wider deployment of EEG monitoring capabilities, outside of theatre or ICU scenarios can be facilitated.

Examples for such work include dedicated but experimental integrated circuit designs [38], which typically achieve high levels of integration, low power and low physical size. FPGA platforms are more widely prevalent in related research projects, where rapid design changes can be applied and tested without the complexity of chip design flows. These works report varying power consumption figures for implementations of milliwatts up to tens of watts, and iteration times and latencies typically ranging from hundreds of milliseconds to seconds [21,39–41].

Recent works, including this paper, emphasise the value of real-time performance. Importantly, in the domain of anaesthesia monitoring, the concept of real-time system performance is bounded by clinical expectations rather than pure technology optimisation. Feedback on DoA in the timescales of seconds is entirely suitable for an operating theatre monitoring situation, given that the purpose is typically to ensure anaesthesia levels are adjusted manually by the anaesthetist with response times of the patient being in similar timescales. Higher rates of inference are still of interest however. Post-operative study and analysis of EEG state and subsequent information on pre-operative recovery, e.g., detection of burst-suppression events for instance, could lead to more sophisticated monitoring systems, more tailored post-operative treatments, and better long term patient outcomes [13]. Whilst such data can be analysed off-line in a post-operative mode, the rapid recording and capture of initial diagnostic indicators and AI-enhanced feedback is likely to be valuable in the immediate post-operative period.

2.4. Machine Learning and FPGA Acceleration

EEG analysis has traditionally used logistic regression, Support Vector Machines (SVM), and more recently, deep neural networks such as CNNs and Recurrent Neural Networks (RNNs). A comprehensive review by Roy et al. [42] identified CNNs and RNNs as dominant in EEG classification tasks, yet emphasised the trade-offs between model complexity and deployment feasibility.

Recent studies have shown that FPGA-based hardware acceleration can significantly improve inference speed and reduce power consumption. Kostarelos et al. [43] developed an FPGA-based system for traumatic brain injury detection, achieving 2× speedups over CPU implementations. This work aligns with this direction by leveraging the FINN framework (Fast Inference Engine for Neural Networks) [44] to compile QNNs to FPGA logic. FINN's integration with Vivado High Level Synthesis (HLS) and Brevitas FPGA tools [45] allows for scalable, low-precision neural network deployment with minimal accuracy degradation [46].

Although previous studies have demonstrated the feasibility of EEG-based ML or consciousness detection, comparatively fewer studies have investigated subject-independent

evaluation across clinical cohorts while also examining the influence of different class-balancing strategies on classifier performance. These limitations motivated the methodology adopted in this work. The contributions of this work are threefold. First, a subject-independent ML framework is developed and evaluated using five-fold cross-validation across volunteer and clinical operating room EEG cohorts. Second, the influence of multiple class-balancing strategies and ML classifiers are investigated to identify the most effective approach for consciousness detection. Finally, the selected algorithm is implemented as an FPGA-based Edge-AI system incorporating spectral processing and a QNN model, demonstrating the feasibility of real-time EEG-based anaesthesia monitoring.

3. Methodology

This section outlines the rationale, dataset processing and approach used to acquire, process, and classify EEG data for detecting anaesthesia-induced unconsciousness.

3.1. Rationale

A core aim of this study was to assess how different ML inference methods would perform when applied to the problem of classifying EEG data into conscious or unconscious subject status in the presence of anaesthesia. The expectation was that there would be variations in accuracy, complexity, iteration time, and resource utilisation. The implication would then be that those factors would allow informed choices about optimal implementation of such a system in deployment scenarios as outlined earlier.

Five ML inference models were examined:

- LR: Logistic Regression
- LDA: Linear Discriminant Analysis
- LDA+LR : Combination of LR and LDA
- NN : Standard Neural Network implementation
- QNN: Quantised Neural Network

LDA was included both as a standalone linear classifier and as a supervised feature transformation method (LDA+LR) within a structured progression of models ranging from simple linear baselines (LR) to nonlinear neural network approaches (NN and QNN). Alternative techniques such as locality preserving projection (LPP) preserve local manifold structure and may better capture nonlinear EEG relationships [47]. However, with the emphasis on computational efficiency for hardware deployment, these five models were chosen due to their supervised nature and computational simplicity.

The first four cases, LR, LDA, LR+LDA, and NN, were implemented as software coded algorithms running on an ARM Cortex-A9 core embedded within a Zynq-7000 SoC device (XC7Z020-1CLG400C) and hosted on a PYNQ-Z2 development board (see details in Acknowledgements section). The QNN implementation utilises reduced precision representation of values within a neural network, based upon quantitation to 8-bit quantities, and was implemented as a custom hardware module using FPGA logic block fabric instead of a CPU model. The expectation here was that by quantising the NN values the accuracy would reduce, but it would also permit reductions in hardware cost. Provided the trade-off was favourable, this would then present a useful option for reducing hardware footprint of a future system.

Each inference implementation relies upon training their respective algorithm with the same, appropriately selected, EEG dataset, tested with the same assumptions about pre-processing front-end and spectral analysis. This is described in detail in Section 3.2, where the methodology of multi-tapered spectrum is described as a technique for translating raw EEG data into a dynamically updated power spectra, which updates at the end of each two-second data window, applied consecutively across the whole of each patient anaesthesia data run.

The ML models were evaluated using subject-independent cross-validation, with performance assessed using Accuracy, Precision, Sensitivity, Specificity, F1-score, and AUC. Following comparative evaluation, the best-performing model was further analysed using aggregated confusion matrices, precision–recall analysis, and prediction probability distributions before deployment as a QNN on the FPGA-based Edge-AI platform. The remaining subsections describe the dataset handling aspects and the ML models deployed for classification, including the final QNN case with quantisation and FPGA implementation.

3.2. Dataset and Preprocessing

The dataset used in this study is an open-source compilation from PhysioNet [10], which includes anonymised EEG spectral recordings from two cohorts undergoing GA: a controlled volunteer group and a clinical operating room (OR) group. The dataset access is restricted to authorised users bound by a Health Data Use Agreement to safeguard privacy and avoid individual identification. The volunteer group included 10 healthy individuals who underwent anaesthetic administration in a controlled experimental setting, while their responsiveness was directly monitored. The volunteers received increasing doses of propofol while responding to auditory cues to identify Loss of Consciousness (LOC) and Return of Consciousness (ROC) events. The OR group consisted of 44 patients receiving anaesthesia care in a surgical context, where 27 of these patients were administered solely propofol, while 17 other patients received sevoflurane (alone or in addition to propofol). The anonymised subject IDs are tabulated in Table 1, where they are grouped in to anaesthesia subsets and separated into OR and volunteer cohorts. Two subjects from the OR propofol + sevoflurane cohort (212 and 630) were excluded from the experimental study following a data quality assessment because their EEG recordings exhibited substantial inconsistencies between the provided reference consciousness annotations and the observed EEG spectral characteristics. Since the source of these discrepancies could not be established, these recordings were not considered suitable, as reliable reference labels are essential for supervised learning. More information about the exclusion of these two subjects is discussed in Appendix B.

Table 1. Dataset subjects by category of anaesthesia and modality. Items in curly brackets represent two individual subjects in the mixed cohort excluded from the experimental work.

Cohort	Subject ID's
OR (Propofol) (27 subjects)	54, 62, 65, 154, 223, 373, 455, 511, 695, 697, 703, 713, 724, 788, 790, 796, 806, 808, 811, 834, 839, 856, 858, 867, 908, 937, 962
OR (Sevoflurane) (single subject)	222
OR (Propofol with Sevoflurane) (16 subjects, 2 excluded)	132, 133, 143, 208, 228, 244, 281, 300, 731, 1014, 1029, 1051, 1069, 1071, {212, 630}
Volunteer (Propofol) (10 subjects)	02, 03, 04, 05, 07, 08, 09, 10, 13, 15

Patient status was classified as conscious or unconscious based on surgical timestamps corresponding to induction, surgery initiation and surgery completion [16]. The EEG signals were recorded based on a frontal electrode (Fp1) configuration, downsampled to 250 Hz where necessary and then converted to the frequency domain using a Fourier Transform (FT) utilising Multitaper Spectral Analysis (MSA).

MSA presents a refined technique that leverages a series of orthogonal tapered window functions, typically Slepian sequences (also called discrete prolate spheroidal sequences,

DPSS). Slepian sequences are designed to maximise energy concentration within a specified bandwidth W in the frequency domain, while being limited to a finite time window of length N. This makes them ideal for analysing signals where both time and frequency resolution are important. A derivation of the multitaper equations is provided in [48], however it is important to note that the sequences are orthonormal over the interval $[0, N-1]$ as defined in (1), which ensures that each taper provides an independent view of the signal's spectrum.

$$\sum_{n=0}^{N-1} v_k[n] v_j[n] = \delta_{kj}, \tag{1}$$

$$\begin{aligned} \text{where:} \quad N &= \text{length of the sequence,} \\ v_k[n] &= \text{k-th taper,} \\ \delta_{kj} &= \text{Kronecker delta.} \end{aligned}$$

Figure 4 shows the principle of preprocessing the raw EEG signal data, with an initial detrending phase which removes low frequency offsets due to baseline signal drift, after which the data is segmented into two-second windows, in this implementation without overlap. Each window is then further processed to yield a detailed frequency spectrum suitable for inference usage. An example of the frequency derivation, via the application of four Slepian tapers to an EEG signal, is illustrated in Figure 5. The frequency derivation methodology involves the combination of the signal with each DPSS taper, followed by the computation of individual taper spectra through the Fast Fourier Transform (FFT).

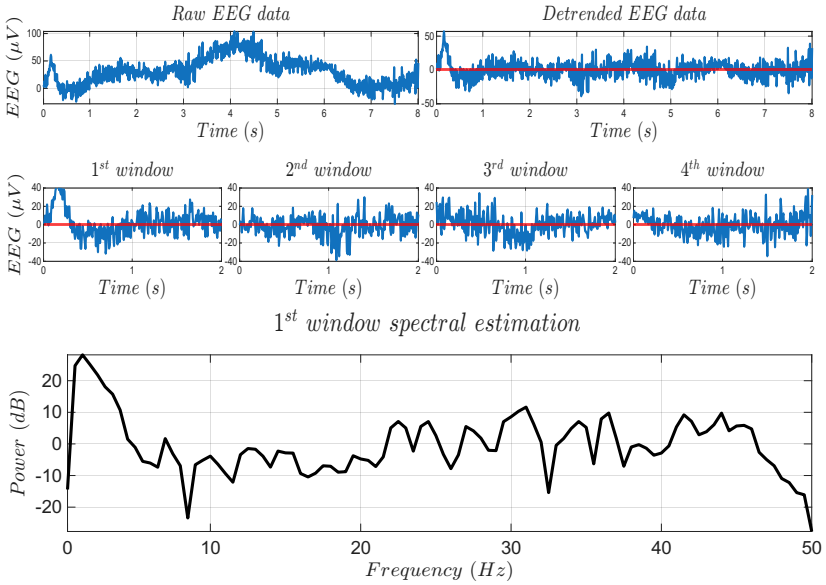


Figure 4. Data processing of the EEG signal leading to a multitaper spectrum.

As Figure 5 illustrates, the initial acquired digital signal for a signal $x[n]$ at (a) is subject to a set of K Slepian tapers $v_k[n]$ at stage (b), yielding the tapered signals (2) at (c). These signals are then transformed at stage (d) using the FFT (3) to give the frequency representation of the signal under each taper. Finally, at stage (e) the power spectral density (PSD) estimate from each taper is then calculated in (4). The orthogonality of DPSS tapers permits the averaging of these single-taper spectra to obtain the multitaper spectral estimate as defined in (5), resulting in a comprehensive multitaper spectrum.

$$x_k[n] = x[n] \cdot v_k[n], \quad k = 1, 2, \ldots, K, \tag{2}$$

$$X_k(f) = \mathrm{FFT}(x_k[n]), \tag{3}$$

$$S_k(f) = |X_k(f)|^2, \tag{4}$$

$$\hat{S}(f) = \frac{1}{K}\sum_{k=1}^{K} S_k(f) = \frac{1}{K}\sum_{k=1}^{K} |X_k(f)|^2. \tag{5}$$

The raw EEG signal is sampled at 250 Hz and data processing involves linear detrending and segmentation into 2 s epochs without any overlapping intervals. In order to match the pre-processing used in the available dataset composition, the computation of spectral power across the frequency range is generated using (in this case) five tapers, where MSA is applied to each data segment and returns the signal power for 100 frequency bins, ranging from 0 to 49.5 Hz in 0.5 Hz intervals.

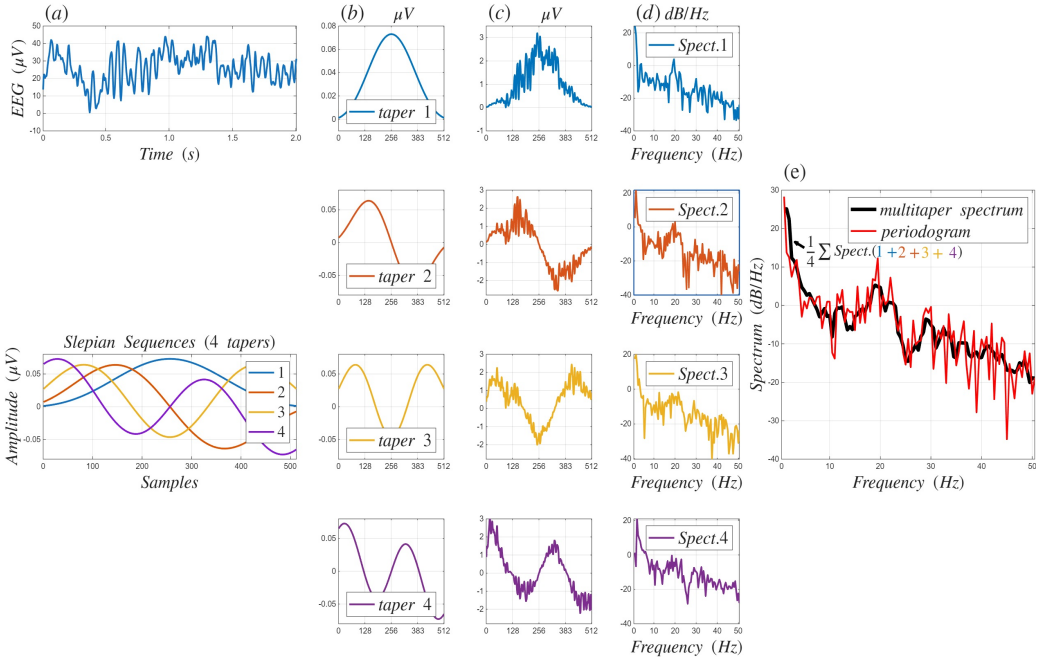


Figure 5. General example of MSA applied to EEG signal, where (**a**) shows the time-domain EEG signal, (**b**) shows four DPSS tapers applied to the input signal, with the resulting individual outputs shown in (**c**). Following on, the four DPSS taper outputs are then processed to yield individual FFTs as shown in (**d**), and then finally summed and averaged to form a single multitaper spectrum as illustrated in (**e**).

The application of MSA yields a spectral representation that helps with the accuracy of traditional FT-based methods, but also avoids the introduction of ringing artefacts, providing a clearer and more reliable insight into the signal's frequency content.

Taking inspiration from the MSA study in [48] and to visualise the brain signal characteristics from a patient undergoing GA, Figure 6a is an example of an EEG data recorded from the Fp1 electrode of a volunteer undergoing propofol-induced GA. The EEG was recorded during a surgical procedure, with the displayed data section lasting ∼1600 s, and where the anaesthetic was administered at ∼200 s. The right-hand figure illustrates a close-up of a 2 s data window beginning at 1000 s, corresponding to the time the patient was under GA. Figure 6b is the MSA derived EEG spectrogram, where the PSD uses a dB scale colourmap. Under GA, alpha-band activity (8–12 Hz) is predominantly observed

in the frontal region, whereas in the awake state, this frontal alpha activity is markedly reduced [49]. A close-up of the PSD estimate at 1000 s is displayed on the right-hand side, showing pronounced alpha-band activity.

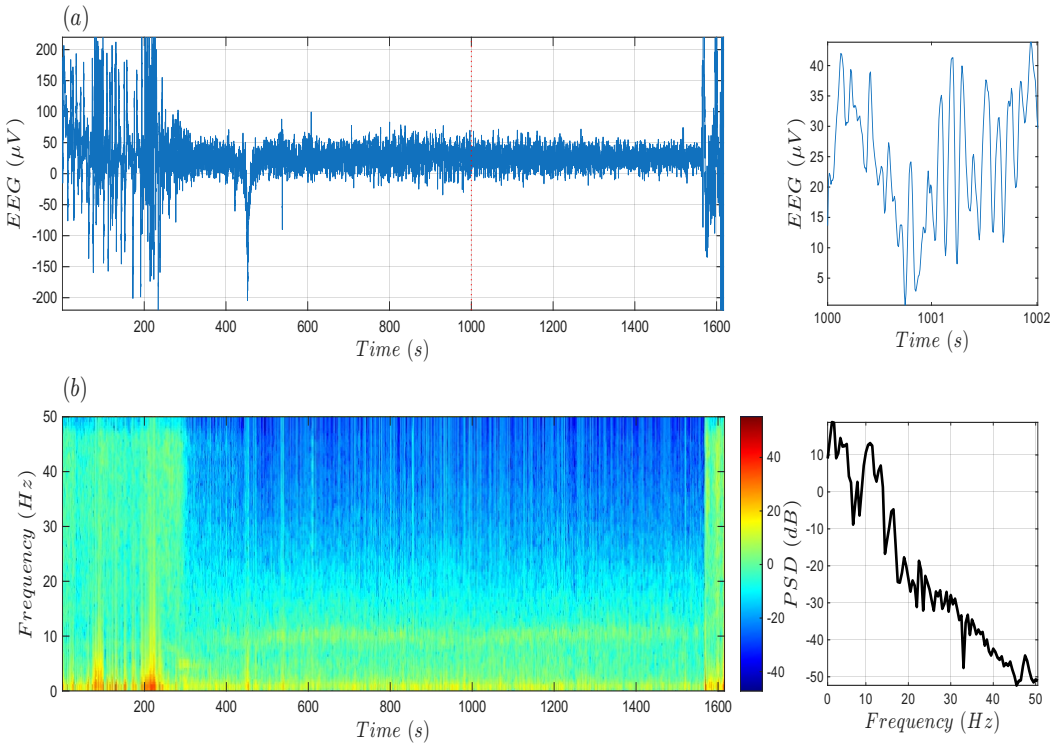


Figure 6. Illustration of EEG spectrogram for a patient undergoing GA. (**a**) Time-domain EEG trace; (**b**) spectrogram derived from MSA. The right-side panels display a close-up of a 2 s interval at 1000 s, highlighted by the red dashed line in (**a**).

3.3. Treatment of Signal Artefacts

It is well known that EEG electrode sensing comes with unwanted extraneous signal behaviours. These include poor electrode contact with the skin, effects of movement of the subject, and muscle related signals being picked up by the electrode: eye blinking being a particularly prevalent cause in these scenarios.

In this study, the public repository dataset provided was labelled only for conscious versus unconscious patient status, and no additional information was available to attempt to train ML models or to apply mitigations to identify such transient effects. Consequently this work does not attempt to identify or manage such effects but rather to achieve best overall accuracy within that unrefined signal environment.

It is recognised however that more sophisticated systems might be trained to identify such artefacts based upon standard reference models, and future work might find benefit in applying those techniques to unlabelled data to improve both training and testing outcomes.

3.4. Class Balancing

The dataset containing the anonymous EEG recordings are included in multiple files, linked by a unique case ID. The raw EEG signal is not provided and instead the files are structured to contain the spectrogram information and labeled to equate the state the subject was in at the time with '1' indicating consciousness and '0' indicating unconsciousness. Further information indicates the time in seconds since the start of the EEG recording along with a set of data points to represent the 100 spectral frequency bins.

The volunteer cohort consists of 10 separate recordings, each averaging about 2 h and 37 min in length, whilst the OR cohort consists of 44 different recordings, each averaging about 59 min in length. The data in Figure 7 captures the level of class 0 to class 1 imbalance across the OR and Volunteer cohorts, where the imbalance is measured as the absolute difference between the number of valid class 0 and class 1 labels as a percentage of all n class labels in a given subject i, as noted in Equation (6).

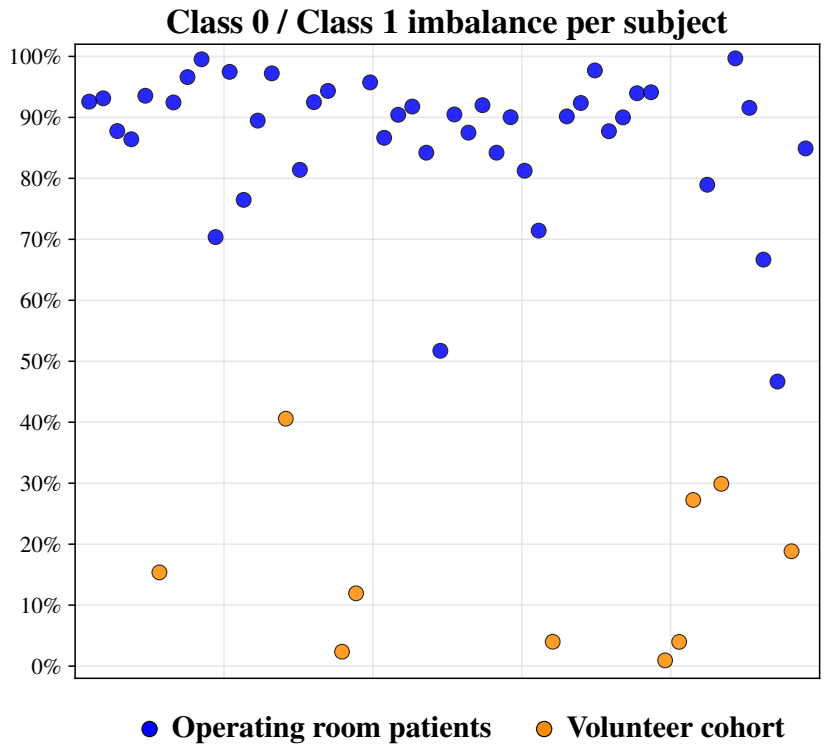


Figure 7. Class 0 to Class 1 imbalance ratios for OR patients and Volunteer cohorts.

$$imbalance_i = |\frac{c_0}{n} - \frac{c_1}{n}| \times 100\% \tag{6}$$

where: c_0 = class 0 count for subject i,
c_1 = class 1 count for subject i,
n = total count for subject i.

It is noticeable that the majority of the OR cohort recordings occur during the patient's class 0 (unconscious) state as the imbalance percentages are very high. In comparison, the volunteer cohort recordings are reasonably more balanced between the class 1 (conscious) and class 0 (unconscious) states. This large skew, especially in the distribution of the OR cohort data can lead to a negative effect on the ML classifier being trained and is termed an 'imbalanced dataset'. A 2D example representation for one of the OR patient's data is illustrated in in Figure 8a, where the majority class 0 is coloured blue, and the minority class 1 is coloured red. To address the dataset class imbalance, both Random Undersampling (RUS) and Synthetic Minority Oversampling Technique (SMOTE) methods are utilised [50]. The undersampled data (Figure 8b) reduces the majority class by randomly removing samples, while SMOTE generates synthetic minority class samples (Figure 8c) by interpolating between existing instances.

These resampling strategies were selected as straightforward and widely used approaches to mitigate severe class imbalance during training, with the aim of stabilising optimisation and improving class representation. Prior to model classification, the spectral

features were scaled using the median and interquartile range to ensure standardisation and to minimise the impact of outliers.

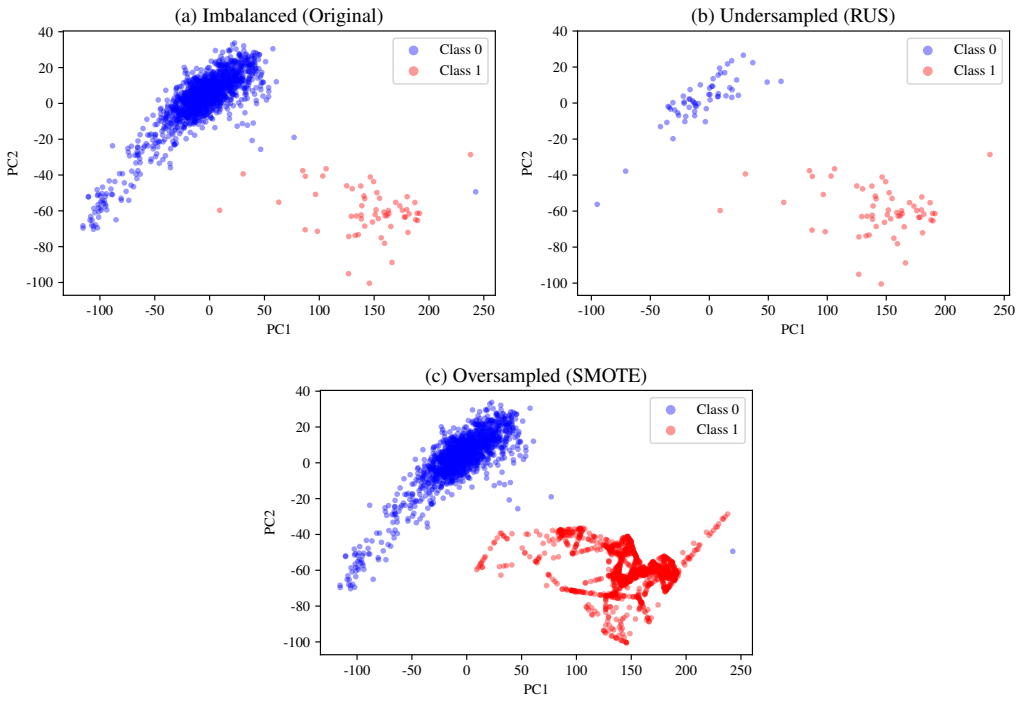


Figure 8. 2D representation of Imbalanced, Undersampled and Oversampled datasets, where (**a**) is the Original imbalanced dataset, (**b**) shows the RUS undersampling outcome, and (**c**) shows oversampling using the SMOTE technique.

3.5. ML Five-Fold Evaluation Approach

For ML model analysis, EEG data from the volunteer and OR cohorts were combined into a single dataset and evaluated using a five-fold subject-independent cross-validation framework. Subjects were assigned to five mutually exclusive folds, such that all EEG spectral samples from an individual subject were allocated exclusively to a single fold. As subjects contributed varying numbers of EEG samples, an initial assignment followed by a swap-based optimisation procedure was employed to distribute subjects across the five folds, minimising deviation from the target number of samples per fold while maintaining complete subject independence. During each cross-validation iteration, four folds were used for model training and the remaining fold was reserved for testing. This evaluation strategy prevented subject-level data leakage and provided a rigorous assessment of model generalisation to previously unseen individuals.

To investigate the effect of class-balancing strategies on classifier performance, three five-fold datasets were generated. The Original dataset retained the natural class distribution within both the training and test partitions, preserving the inherent imbalance between conscious and unconscious EEG samples. The RUS dataset was generated by randomly under-sampling the majority class within the training partition of each fold to produce a balanced training set, while the corresponding test partition remained unchanged. Similarly, the SMOTE dataset was created by applying SMOTE to the training partition of each fold to synthetically increase the minority class and achieve class balance. In all cases, the test partition retained its original class distribution to provide an unbiased evaluation under realistic class prevalence. Performance metrics were calculated for each fold and are reported as the mean and standard deviation across the five cross-validation folds using radar plots from the tabulated results included in Appendix A.

3.6. ML Models

As mentioned earlier, five key models were evaluated as part of this study:

- Logistic Regression (LR): A standard baseline for binary classification.
- Linear Discriminant Analysis (LDA): Used both as a classifier and for feature reduction.
- LDA+LR: A hybrid model approach using LDA for input feature transformation followed by LR classification.
- Neural Network (NN): A fully connected multi-layer perceptron.
- Quantised Neural Network (QNN): The same NN architecture converted into an 8-bit fixed-point representation for FPGA deployment.

The implementation methodology for each of the five evaluated models are described in the following subsections.

3.6.1. Logistic Regression (LR)

This is a supervised ML algorithm designed for binary classification tasks, aiming to predict one of two possible outcomes based on a set of input features. In contrast to linear regression which predicts a continuous value, LR estimates the probability of an input belonging to a specific class [51]. Mathematically, LR models the probability $P(y = 1 \mid x)$ of a binary outcome y given a feature vector x, using the sigmoid function:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-\left(w^\top x + b\right)}}, \tag{7}$$

where: $w =$ weight vector,
$b =$ bias term,
$x =$ input features (e.g., spectral power).

In this work, LR was implemented as a (Python) software coded solution running on the ARM CPU core.

3.6.2. Linear Discriminant Analysis (LDA)

This is a supervised learning technique primarily used for classification and dimensionality reduction, particularly effective when dealing with normally distributed data and linearly separable classes [52]. In binary classification tasks, LDA seeks to find a linear boundary that best separates the two classes based on the input features.

The LDA technique models the data by assuming that each class generates data according to a Gaussian distribution with a shared covariance matrix, but different class means. The algorithm computes a discriminant function that projects the input data onto a line such that the between-class variance is maximised and the within-class variance is minimised. In EEG-based ML, LDA is widely used due to its computational efficiency, robustness to noise, and interpretability. It is particularly effective when the number of features is large relative to the number of observations, a common scenario in EEG analysis. In this work, LDA was also implemented as a (Python) software coded solution running on the ARM CPU core.

3.6.3. LDA+LR Combination

LDA+LR builds on approaches that combine linear feature extraction with statistical classification in EEG analysis [53]—this study extended the analysis by implementing a hybrid model that integrates Linear Discriminant Analysis (LDA) for feature extraction

with a Logistic Regression (LR) classifier as a unified pipeline. Again, the implementation was software coded on the CPU core.

3.6.4. Neural Network (NN)

The NN model is based upon a class of ML models that consist of interconnected layers of nodes (neurons) that learn complex mappings between input features and target outputs through iterative optimisation. In EEG analysis, NNs are particularly effective for capturing nonlinear relationships and spatiotemporal patterns in brain signals. NNs are widely used in EEG-based applications such as emotion recognition, seizure detection, and brain–computer interfaces (BCIs) [54,55]. Their ability to learn hierarchical representations makes them suitable for raw or minimally processed EEG data, although they typically require large datasets and careful regularisation to avoid overfitting.

In this work, a two-layer NN was developed which comprised of 100 input features, a single fully connected hidden layer containing 32 neurons with ReLU activation, and a single output neuron with a sigmoid activation for binary classification as shown in Figure 9. The network was trained for 300 epochs using a batch size of 64, using the Adam optimiser ($\beta_1 = 0.9, \beta_2 = 0.9999$) with a learning rate of 0.0002, and binary cross-entropy loss. Reproducibility was ensured by fixing the PyTorch (torch.manual_seed(0)) and NumPy (np.random.seed(0)) random seeds (PyTorch version 1.10.2, NumPy version 1.19.0). For each subject-independent cross-validation fold, model training was performed only on the corresponding training partition, with the class-balancing procedure (RUS or SMOTE) applied to each subject ID in the training data folds. Feature scaling (RobustScaler) was fit only on each fold's training partition and applied to transform the corresponding test fold, to avoid information leakage between folds. The model achieving the highest balanced accuracy was retained for evaluation on the independent test fold maintaining its original class distribution. The relatively compact 100-32-1 architecture was selected following evaluation of alternative configurations (e.g., 16–64 hidden units and one vs. two hidden layers), which showed no substantial performance gains for larger or deeper models while increasing model complexity and CPU resource usage. This was done with the objective of achieving a suitable balance between classification performance and computational complexity for subsequent FPGA deployment, thus making it practical for Edge-AI deployment. The trained NN runs on the CPU core as part of the Pytorch Modeling Framework [56].

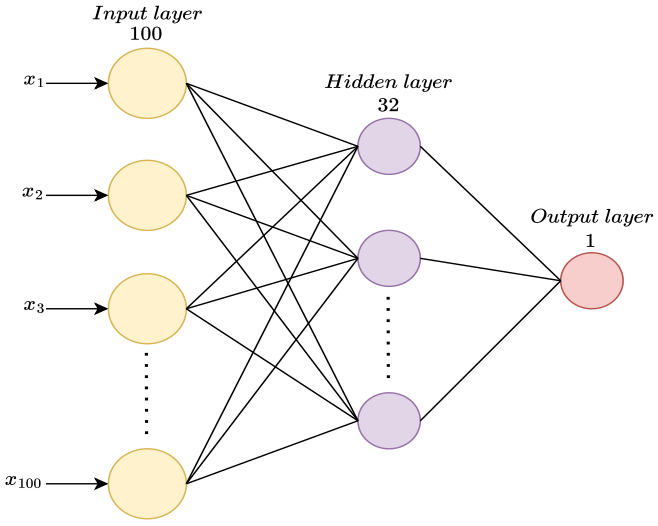


Figure 9. Two-layer neural network model.

3.6.5. Quantised Neural Network (QNN)

This is based upon a class of neural models where the weights, activations, and sometimes gradients are represented using low-precision numerical formats, such as 8-bit integers instead of 32-bit floating-point values [57]. This quantisation significantly reduces the computational complexity, memory footprint, and energy consumption, making QNNs highly suitable for real-time EEG processing on resource-constrained devices. In QNNs, the standard NN operations are modified to use quantised values:

$$\hat{y} = f(Q(W) \cdot Q(x) + Q(b)), \tag{8}$$

where: $Q(.)$ = quantisation function,
W, x, b = weights, inputs, and biases,
$f(.)$ = nonlinear activation function, often also quantised.

QNNs maintain competitive performance compared to full-precision models, especially when combined with techniques like post-training quantisation (PTQ) [58] or quantisation-aware training (QAT) [59]. Following selection of the full-precision NN architecture, an 8-bit QNN design was developed using QAT implemented with the Brevitas library in PyTorch. The same network architecture 100-32-1 was maintained, while weights and activations were quantised to 8-bit precision. The resulting QNN was exported through ONNX for FPGA deployment on the PYNQ-Z2 platform. The same optimiser, learning rate, epochs and batch size parameters were used as for the full-precision NN training procedure. QAT was selected over PTQ to better account for quantisation effects during optimisation and to preserve performance under low-precision constraints.

QNNs are well-suited for EEG applications, offering fast and energy-efficient inference that facilitates deployment on microcontrollers and FPGAs. Unlike the other models, QNN was compiled as a hardware accelerator core, implemented using the FPGA logic block fabric, and does not run on the CPU. Fuller details are given in the next Section 3.7.

3.7. Field-Programmable Gate Array (FPGA) and Hardware Validations

An FPGA features a matrix of configurable logic blocks for hardware acceleration, enabling offloading of repetitive, compute-intensive tasks from the CPU, allowing it to perform other tasks and to reduce overall power utilisation. The CPU may itself be part of the same chip, allowing a highly integrated solution with small physical size. FPGAs are traditionally programmed with hardware description languages like Verilog or VHDL, but increasingly employ High-Level Synthesis (HLS) for transforming high-level behavioral descriptions into synthesisable designs [60]. HLS utilises languages such as C and C++, which simplifies embedded development. The work presented in this paper uses HLS (AMD Vitis version 2022.2) to operate a NN on an FPGA applied to the PYNQ-Z2 platform.

The implementation methodology used in this work assumes a physical implementation, whereby EEG signals are captured using a proprietary wearable headband equipped with frontal electrodes, connected to a Cyton open brain–computer interface (OpenBCI) biosensing board [17] as shown in Figure 10. The Cyton module wirelessly transmits amplified signals for receipt and processing with real-time inference to the host module: a PYNQ-Z2 board [18], which integrates an ARM processor with FPGA logic. The spectral features supplied by the third-party dataset were generated from non-overlapping two second EEG epochs, using the NiTime Python package. Each epoch was linearly detrended using bandpass filtering prior to multitaper spectral estimation over the 0–50 Hz frequency range, using a time-bandwidth product of three and five tapers. To ensure consistency between

the spectral features used for model development and those generated during real-time operation, the MSA stage of the real-time pipeline was configured using the corresponding spectral estimation parameters and same Python (version 3.6) libraries. In all cases, Python code executing on the on-board FPGA CPU core is used for signal processing, including filtering and multitaper spectral analysis (MSA). The final stage is then performed by a selected ML classification method implemented in either CPU-bound coding or as custom FPGA logic with a hybrid CPU+FPGA model.

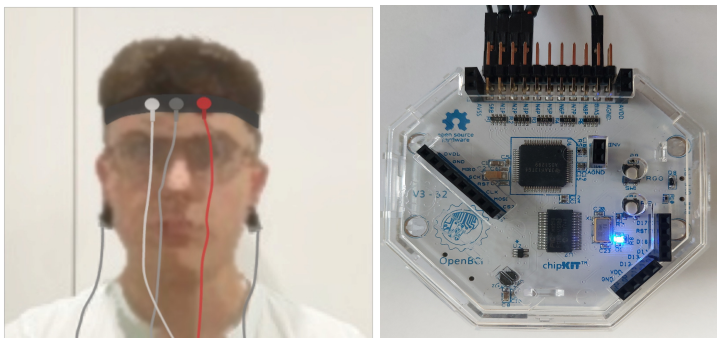


Figure 10. Example Headband configuration (**left**) with electrodes (blurred photo enhanced to show sensor positioning) capturing EEG data from Cyton biosensing board (**right**). Data is then transmitted wirelessly to the host platform under development.

Importantly, due to the practical constraints of actual live patient testing outside of a clinical research setting, the prototype host platform was evaluated for performance capabilities (speed, power, accuracy and performance) in two ways: Inference methods, running on the FPGA-hosted CPU core, were evaluated via the use of a publicly available dataset [10] which provides a precomputed MSA data stream for a number of patients from clinical and control settings. The MSA pre-processing stages implemented for this system's evaluations were configured with the same processing parameters used by the dataset originators and tested independently as a 'front-end' stage to validate its operation. One of the consequences of using the pre-existing dataset is that only a single EEG data channel is available, however as the intention in this work was to utilise a single channel, Fp1, and evaluate how well that simplest arrangement would perform, this was not an issue.

One key question (and motivation) relating to the trade-offs between software versus hardware NN implementations relates to scaling complexity. In this paper only a single EEG channel is used; however, multiple channels might be desirable. At the same time a more sophisticated algorithm might also be employed. Both of these aims imply more artificial neurons in an NN or QNN solution. In very general terms, doubling the complexity of an NN solution implemented in software is likely to increase computational time (serial by nature in a uni-processor) by a factor of 2 to 4 times. However, the layered and tree-like architecture of a QNN implemented in hardware means that most of its additional complexity is added in parallel to existing neurons, and computational time cost is therefore likely to be minimally increased even with many EEG sensor channels. Studies of the scaling complexity of artificial neural networks, and FPGA related optimisations, suggest there is significant opportunity here for future development [61–63].

In this study, a complete design flow was implemented on an FPGA, which spanned signal acquisition, frequency-domain transformation, and MSA to facilitate real-time EEG signal processing and classification using data from the OpenBCI headband. That hardware implementation was then tested for functional real-time correctness of operation in order to validate its ability to acquire live signals via the headset, convert data channels into taper spectra data blocks of the same size and format as those provided in the open dataset, and then to apply those data-blocks in the analytical stage chosen for each test scenario (LDA,

NN, QNN, etc.). The correctness and performance accuracy of the algorithms employed was tested separately using the taper spectra blocks already preprocessed as this was how they were supplied in the open dataset used.

4. Results and Discussion

4.1. ML Model Application and Results

The ML models described in Section 3.6 were evaluated using the subject-independent cross-validation framework outlined in the methodology section. To facilitate a comparison of classifier performance across the three class-balancing strategies that are SMOTE, RUS and Original datasets, Figure 11 presents a radar plot of the Precision, Specificity, Sensitivity, F1-score, AUC and Accuracy scores obtained across the five folds for the LR, LDA, LDA+LR and NN models. The data used for these plots is provided in Appendix A along with mean and standard deviation data for the five-fold cross-validation experiments (see Tables A1, A2, and A3, respectively).

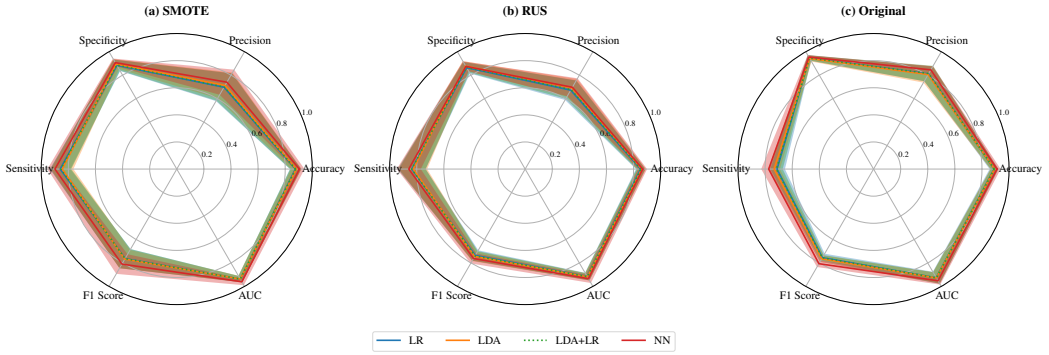


Figure 11. Radar plots comparing the mean performance of the LR, LDA, LDA+LR, and NN classifiers for the (**a**) SMOTE, (**b**) RUS, and (**c**) Original training strategies. Polygon vertices represent the mean values obtained from subject-independent five-fold cross-validation, while the shaded regions indicate ±1 standard deviation across the five test folds.

These plots provide a visual summary of the performance profile of each classifier, highlighting the trade-offs between evaluation metrics, and compares the average performance of the four evaluated classifiers under the three training strategies. The polygon vertices correspond to the mean performance metrics obtained from subject-independent five-fold cross-validation, while the shaded regions represent the corresponding standard deviations across the five folds. The two-layer NN model consistently achieved the highest mean performance across all evaluation metrics, while exhibiting relatively small variability for sensitivity and specificity scores, indicating robust and consistent generalisation to previously unseen subjects. The Original (imbalanced) training strategy performs reasonably well, but its sensitivity and F1-scores are noticeably lower than those obtained with SMOTE. Meanwhile RUS class balancing improves sensitivity relative to the Original training set, but does not match the balance and overall performance achieved by SMOTE, particularly for sensitivity, specificity and AUC scores. Based on these findings, the two-layer NN model, trained using the SMOTE balancing strategy, was selected for detailed performance analysis.

The confusion matrix, probability distribution and precision–recall curve analyses presented in the following sections were generated by aggregating the predictions from the five independent test folds, thereby evaluating classifier performance across all previously unseen subjects.

4.1.1. Confusion Matrix Test Results

Following the cross-validation analysis, the performance of the selected NN-SMOTE classifier was further evaluated using confusion matrices generated from the aggegrated predictions of the imbalanced distributions in the five independent test folds. To assess model performance across different acquisition settings and anaesthetic protocols, separate confusion matrices were constructed for three distinct patient cohorts: (i) the Volunteer cohort, in which unconsciousness was induced using propofol; (ii) the operating room (OR) cohort receiving propofol-based anaesthesia; and (iii) the OR cohort receiving a combination of propofol and sevoflurane.

Figure 12a–c presents the confusion matrices for all three cohorts with the corresponding performance metrics summarised in Table 2. The classifier demonstrated consistently strong performance across all cohorts, achieving accuracies between 88.89% and 91.3% and AUC values ranging from 0.9472 to 0.9801. Good performance was observed for the OR propofol cohort, with an accuracy of 91.24%, sensitivity of 96.28%, specificity of 90.96%, and an AUC of 0.9801.

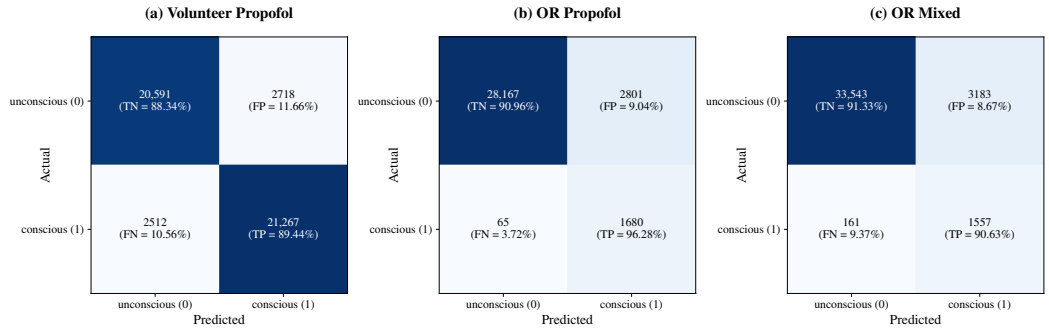


Figure 12. NN-SMOTE confusion matrices for the three cohorts (**a**) Volunteer cohort where propofol is used to induce anaesthesia, (**b**) OR cohort receiving propofol-based anaesthesia and (**c**) OR cohort receiving a combination of propofol and sevoflurane.

Table 2. Cohort comparison table.

Cohort	Accuracy	Sensitivity	Specificity	AUC
Volunteer (Propofol)	0.8889	0.8944	0.8834	0.9472
OR (Propofol)	0.9124	0.9628	0.9096	0.9801
OR (Propofol + Sevoflurane)	0.9130	0.9063	0.9133	0.9618

These findings indicate that the proposed classifier generalises well across both controlled volunteer recordings and clinical operating-room environments despite differences in anaesthetic protocols. The strong performance observed within the OR propofol cohort suggests that the proposed methodology is particularly well suited to clinical EEG recordings, while the comparable results obtained for mixed-agent anaesthesia indicate encouraging robustness across different anaesthetic protocols.

From a clinical perspective, false negative predictions (i.e., classifying a conscious patient as unconscious) are generally considered more consequential than false positive predictions, as failure to detect intraoperative awareness may expose patients to unintended awareness and the associated risk of psychological distress. In contrast, false positive predictions may prompt additional clinical assessment or a temporary adjustment of anaesthetic delivery, representing a more conservative response that can be readily verified using existing physiological monitoring and clinical judgment.

In this context, the NN-SMOTE classifier demonstrated consistently high sensitivity across the clinical operating room cohorts, achieving sensitivities of 96.28% for the propofol

cohort and 90.63% for the mixed propofol–sevoflurane cohort, while maintaining specificities of 90.96% and 91.33%, respectively. These results indicate that the classifier successfully identified the majority of conscious EEG epochs while maintaining a balanced false-positive rate. The consistently high AUC values (>0.94) together with sensitivity and specificity scores across all cohorts demonstrate that the classifier maintains robust discrimination despite differences in anaesthetic protocol.

This data reflects a good balance between sensitivity and specificity for an anaesthesia decision-support system. By prioritising the detection of conscious states while maintaining a low rate of false classifications, the proposed approach aims to reduce the likelihood of missed awareness events while limiting unnecessary clinical interventions which is preferable to maximise patient safety [64]. It is important to note that the classifier is intended to complement, rather than replace, existing intraoperative monitoring by providing an additional source of information to support anaesthetic management.

4.1.2. Violin Plot with Confidence Interval

To examine class separability, violin plots of the predicted probabilities for consciousness were generated for each true class. Figure 13 illustrates the distribution of predicted probabilities produced by the selected NN-SMOTE classifier for class 0 (unconscious) and class 1 (conscious) EEG epochs aggregated across the five subject-independent test folds. The predicted probabilities are strongly concentrated near 0 for unconscious epochs and near 1 for conscious epochs, demonstrating excellent separation between the two classes. Only a small proportion of predictions occur near the decision threshold, indicating that the classifier makes high-confidence decisions for the majority of previously unseen EEG samples. The narrow 95% confidence intervals (CI) around the median further demonstrate the stability of the classifier outputs across the aggregated evaluation dataset.

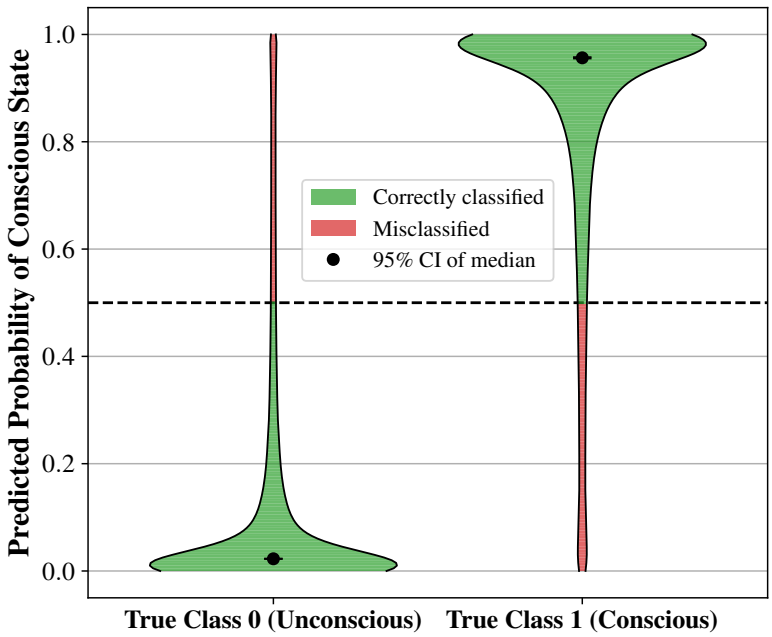


Figure 13. Violin plot of predicted scores for the NN-SMOTE model, where the black circle denotes the median predicted probability.

4.1.3. Precision–Recall Curve

A Precision–Recall (PR) analysis was performed to further evaluate model performance given the class imbalance associated with the OR cohort. Figure 14 presents the aggregated PR curve for the proposed NN-SMOTE classifier obtained from the five subject-

independent test folds. The classifier achieved an Average Precision (AP) of 0.870, substantially exceeding the no-skill baseline precision of 0.230, which corresponds to the prevalence of conscious EEG epochs within the evaluation dataset. The curve demonstrates that high precision is maintained across a broad range of recall values, indicating that the classifier is capable of detecting conscious states, while limiting false positive predictions over a wide range of decision thresholds.

The PR analysis further demonstrates the flexibility of the proposed classifier. While the default decision threshold of 0.5 was adopted throughout this study, the PR curve indicates that the operating threshold may be adjusted to achieve alternative balances between precision and recall. For example, the maximum F1-score was obtained at a threshold of approximately 0.667. Such threshold optimisation may allow future implementations to tailor classifier performance according to specific clinical priorities, such as maximising awareness detection or reducing false alarms.

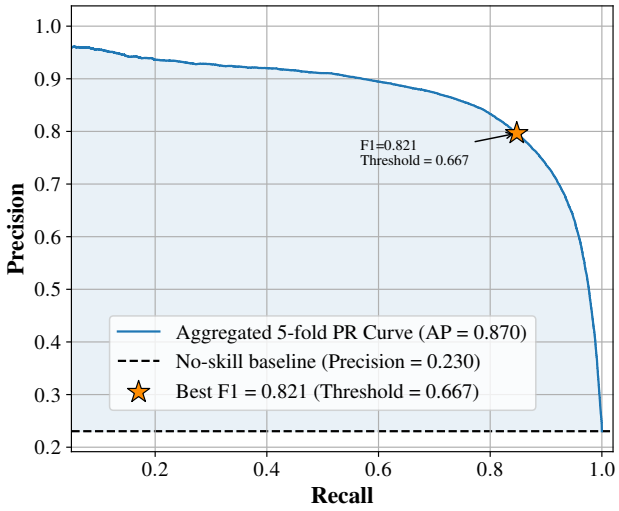


Figure 14. Precision–Recall curve generated from the aggregated prediction probabilities of the NN-SMOTE classifier across the five subject-independent cross-validation folds. The curve illustrates classifier performance over the full range of decision thresholds, with an Average Precision (AP) of 0.870.

The choice of decision threshold therefore depends on the relative cost of these error types. In contexts where false alarm indications must be minimised: for example, when premature declarations of regained consciousness carry clinical risks, a higher threshold may be appropriate. Conversely, if missing conscious episodes are deemed more critical, a lower threshold then would provide more sensitive detection.

It must be noted that the proposed system is not intended to function as a standalone diagnostic device, but rather as a decision-support tool that provides continuous probabilistic estimates of patient consciousness. Model outputs are designed to complement existing physiological monitoring and clinician judgement, enabling anaesthesiologists to identify potential awareness events earlier and intervene when appropriate. Decision thresholds may be adjusted depending on clinical context, patient risk profile, and anaesthetic strategy, allowing sensitivity and specificity to be tuned to clinically relevant operating points.

4.2. Edge-AI Implementation Results

The prototype Edge-AI design in Figure 15 represents the flow of data being collected at the electrode all the way to the RGB LEDs that indicate the consciousness level. The EEG single Fp1 electrode data is acquired by the OpenBCI Cyton module and wirelessly transmitted to a dongle attached to the PYNQ-Z2 board. PYNQ contains a programmable

system that integrates both FPGA programmable logic (PL) and an ARM Cortex-A9 CPU processor system (PS). The PYNQ Linux OS running on CPU supports Python code development. A Python EEGAnalyser class object uses a predict() function that waits for exactly 2 s of data equating to 500 points at a sampling frequency of 250 Hz. This single-channel EEG data is then pre-processed by applying a Butterworth bandpass and ac noise notch filter to remove unwanted components. Once the data is filtered, the power spectrum is then estimated using MSA and then passes this output data to the QNN classifier designed on the FPGA PL.

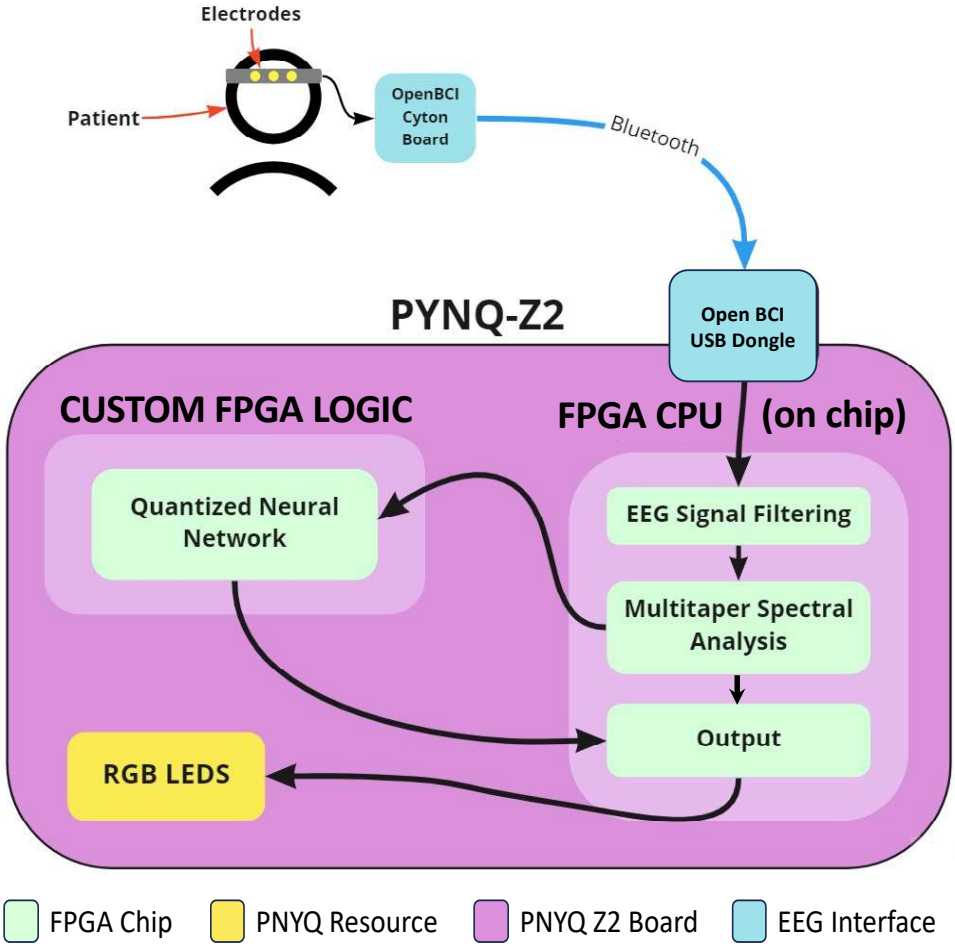


Figure 15. Design implementation and data flow, showing initial data acquisition stage (Cyton open-BCI Board), Bluetooth wire-free link to main host system, signal preprocessing and spectral transformation via on-board CPU, and final inference stage via QNN implementation. Note that (as shown in earlier Figure 3), all data processing elements (both ARM CPU core and custom logic) are embedded within the same single FPGA chip and are therefore all implemented on FPGA either as custom logic or as code executing on the CPU core.

A FINN compiler [65] generates the QNN inference engine on the FPGA fabric and is used in conjunction with Vivado HLS tools to build the logic design. A Python function is used to quantise the input data to unsigned 8-bit values ranging from 0∼255, while the QNN class code shown in Figure 16 uses PyTorch and layers from the Brevitas library [45] that migrate to the FPGA. The QNN consists of the same compact two-layer network as for the CPU-deployable NN, however the key difference is that the network uses 8-bit hardware operations. The QNN FPGA information in Table 3 provides the breakdown for the resources used in the FPGA custom logic for the QNN design. As observed, the logic used is very small in comparison to the total number of resources available on the XC7Z020

FPGA programmable logic fabric. This data suggests that the design could be ported to a smaller FPGA device such as the XC7Z007 device which has ≈ 8x less resources on the chip and more scope for power reductions.

Table 3. Xilinx/AMD XC7Z020 resource utilisation for QNN design.

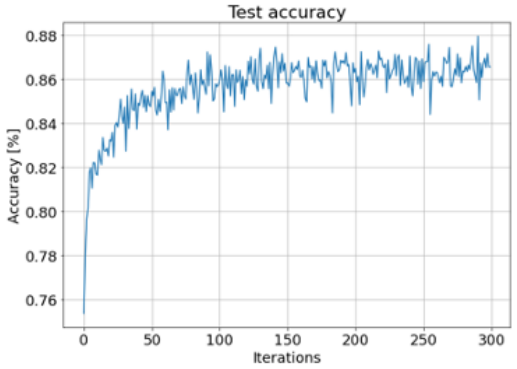
Resource	Total	% Utilisation [1]
LUT	1386	2.61%
LUT (as logic)	1374	2.58%
LUT (as memory)	12	0.07%
Registers as Flip Flops	1210	1.14%
DSPs	1	0.45%
BRAM	1	0.71%

[1] As a percentage of total resources available on the FPGA fabric, as reported by Vivado compilation output.

The QNN test accuracy results seen in Figure 16, shows good performance up to 88% which could be improved by more refinement on the NN and by gathering more data. The FINN tool generates a driver which allows users to interact with the model through Python. Unlike the CPU operated NN inference model, which consumes approximately 228 mW of power, the QNN based inference model deployed on the PYNQ-Z2 board consumed 156 mW of on-chip power during operation.

A notable difference in performance is observed for the NN running on the CPU vs. the NN post-quantisation running on the FPGA logic fabric seen in Table 4. There is a small change in the model's ability to correctly predict consciousness, but inference time does become significantly faster.

The QNN FPGA implementation achieves almost two-thirds reduction in evaluation time, with the additional benefit of a useful reduction in power consumption: For the inference stage, performing equal number of inferences per second (one per each two-second data block as described earlier), then overall ML-stage power consumption reduces from 228 mW (NN) to 156 mw (QNN), approximately ≈ 31.5% reduction. Although the overall system power savings (228 vs. 156 mW) and total evaluation times (remaining in the low 300 ms range) are more modest, the key observation is that implementing the ML stage in custom logic achieves similar accuracy with lower overall power demand, reduced CPU loading and and no increase (rather a decrease) in overall evaluation time.




```
from brevitas.nn import QuantReLU, QuantLinear, QuantSigmoid
import torch.nn as nn
# Neural Network model class
class EEGNet(nn.Module):
    def __init__(self, input_size, hidden_layer,
                 output_size, weight_bit_width=8, act_bit_width=8):
        super(EEGNet, self).__init__()
        self.fc1 = QuantLinear(input_size, hidden_layer, bias=True,
                               weight_bit_width=weight_bit_width)
        self.act1 = QuantReLU(weight_bit_width=act_bit_width)
        self.fc2 = QuantLinear(hidden_layer, output_size,  bias=False,
                               weight_bit_width=weight_bit_width)
    def forward(self, x):
        out = self.fc1(x)
        out = self.act1(out)
        out = self.fc2(out)
        return out
```

Figure 16. QNN accuracy results (**left**), and QNN model class (**right**).

This is an excellent result when considering the flexibility of system deployment and the classifier's robustness to quantisation. Indeed, as Table 4 indicates, the QNN FPGA model out-performs all other models evaluated for inference time, overall power usage and energy per inference, with almost no disadvantage for accuracy as compared to all

of the other models. Although the NN model achieved marginally better accuracy, the QNN FPGA implementation achieved a 2.7× reduction in ML-stage inference time and significantly lower power consumption, making it suitable for battery-powered wearable devices. With a measured ML-stage inference time of 0.78 ms and total system power consumption of 156 mW, the system consumes approximately 122 µJ per inference, as evaluated using (9).

$$\text{Energy per inference(uJ)} = \text{Power(mW)} \times \text{Inference time(ms)}. \quad (9)$$

Such low-power performance enables long-duration monitoring without recharging and supports continuous operation in portable EEG systems within operating rooms or outpatient environments. Examples of the operational system are shown in Figure 17. Once the EEG data stream is live, the PYNQ device will either show a green light to indicate that the QNN is predicting consciousness from the single EEG electrode data, or a red light indicating the subject has entered a state of unconsciousness. Although in this existing implementation, there are only two measures of consciousness, it will be possible in future systems to refine this to a degree of representation of level of consciousness on a suitable scale, such as the bispectral index (BIS) scale [66].

Table 4. FPGA vs. CPU performance test cases.

Model	Accuracy	Inference Time [1]	Power [2]	uJ/Event [3]
QNN (FPGA)	0.880	0.78 ms	156 mW [4]	122 uJ
NN (CPU)	0.903	2.10 ms	228 mW	478 uJ
LR (CPU)	0.877	1.01 ms	228 mW	230 uJ
LDA (CPU)	0.882	1.04 ms	228 mW	237 uJ
LDA+LR (CPU)	0.878	1.81 ms	240 mW	434 uJ

Data shows that significant speedups and power reductions are possible for an FPGA implementation with only a small reduction in accuracy. [1] ML stage inference time only, [2] Total power cost above idle, assuming continuous operation, [3] Estimated microJoules per classification event, [4] Comprising 25 mW dynamic power, 131 mW static power.

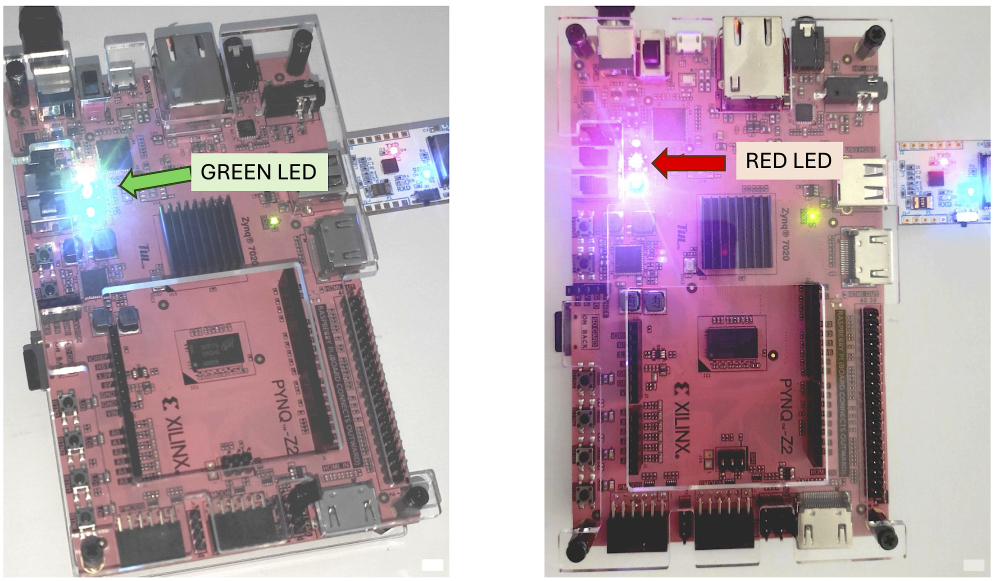


Figure 17. PYNQ Predicting consciousness and unconsciousness states. (**Left**) hand photo represents a 'green light' conscious state, and the (**right**) hand photo is the system predicting 'red light' unconsciousness. The size of the main board is 138 mm (length) × 85 mm (width) × 18 mm (height) and weighs 90 g.

The overall pipeline latency is dominated by the multitaper spectral analysis (MSA), which requires 306 ms to process each sliding EEG window, as noted in Table 5. Since the signal-processing stages (filtering and MSA) are identical for all implementations, differences in execution time are solely due to the ML inference models. For consistency, all reported latency measurements correspond to single-sample inference (batch size = 1).

Table 5. Timing measures of various pipeline stages.

Stage	Timing
Butterworth filter	7.85 ms
Notch filter	2.28 ms
Multitaper spectral analysis	306 ms
Inference engine	0.78 to 2.10 ms
Total	≈318 ms

All tests run on the PYNQ-Z2 board with 650 MHz CPU clock frequency, with custom FPGA logic for the QNN.

The overall time delay is then approximately 318 ms for the total pipeline including inference from the ML models. This suggests that an inference refresh rate of around 3.2 inferences per second could be achieved with simple adjustments. In some contexts, higher inference refresh rates may be important. For example: detection and monitoring of burst suppression events and other short transient signal features or where use of finer-grained AI interpretation of levels of consciousness is desired. In the context of anaesthesia monitoring, where EEG-based indices are commonly computed over windows of hundreds of milliseconds or longer, this end-to-end latency remains compatible with real-time continuous monitoring.

Future work that aims to achieve faster update rates would profitably focus upon accelerating the MSA stage, perhaps by using a custom FPGA logic implementation of the whole or sub-sections of this task. Trade-offs between MSA parameter selection vs. speed and accuracy are of particular interest. Typically, FFT algorithms contain high degrees of exploitable parallelism in hardware embodiments [67,68], and exploration of parallelism within the computational structure when transposed into logic circuits would undoubtedly yield much more significant speedups. Furthermore, extending the approach to multi-channel EEG analysis represents potential for future work to progress further, as single-channel EEG may not fully capture all inter-patient variability or spatially distributed cortical dynamics.

5. Conclusions

In this paper, a compact and potentially portable real-time system was presented for monitoring unconsciousness under anaesthesia by analysis of the power spectrum characteristics of electrical brain activity. A multitaper spectrum method was used to compute the initial spectra and then various inference models were used to evaluate them. The NN and QNN models achieved comparably high classification performance, and outperformed the conventional LR, LDA, and LDA+LR classifiers across many principal evaluation metrics. While maintaining prediction performance relatively similar to the full-precision NN, the QNN achieved substantially lower power consumption and faster inference. This confirms that QNN would therefore be the preferred solution for FPGA-based real-time deployment, based upon the evaluated hardware models.

Beyond the primary technical outcomes, this study has demonstrated several other important outcomes. First of all, the relatively small form factor of the device (even as a prototype), and its stand-alone capability, demonstrate that monitoring of degrees of unconsciousness can be deployed in small, potentially hand-held, and self-powered devices.

This opens up the possibility of applications beyond well-provisioned operating theatres, and toward deployability in third-world regions, war-zones, disaster-relief situations, and also in first-responder scenarios.

Where first-response situations are encountered, the ability to monitor the varying levels of consciousness of a casualty and to monitor effects of emergency sedation, in a portable device, could be extremely valuable [69]. The combination of intelligent EEG-consciousnesses sensing with multi-sensing headband technologies such as pulsometer, oxiometer and respiratory sensing [70], and wire-free connectivity to a base unit, would certainly allow a sophisticated patient status monitor to assist in these scenarios.

This study encountered several limitations in the analysis and application of ML techniques to EEG data. A primary limitation was sourcing a high-quality and sufficiently large dataset. This study relied on an open-access PhysioNet dataset which was the sole open-access resource available at the time. While containing both volunteer and OR cohorts and exposure to two anaesthetic agents, it does not fully represent the variability encountered in routine clinical practice. The inclusion of both controlled volunteer recordings and intraoperative patient data introduces diversity in acquisition conditions and anaesthetic exposure; however, broader demographic variation, comorbidities, and additional anaesthetic agents are not comprehensively represented. Accordingly, the reported results should be interpreted as benchmark performance within this mixed dataset, and further validation on independent and prospectively collected clinical datasets is necessary to establish broader generalisability and to establish robustness for real-world clinical use. Another significant constraint was the preprocessed nature of the available data; without access to the raw data, it was not possible to explore and compare various analytical methods fully.

While the present study demonstrates methodological feasibility and hardware-efficient implementation, validation for clinical deployment would require additional steps beyond the current evaluation. These include prospective operating room studies in which the proposed system is evaluated alongside established DoA indices, such as BIS and clinician-in-the-loop assessments to examine usability, interpretability, and alarm burden. Such staged validation would allow assessment of real-world performance while ensuring that the system functions as a decision-support tool that augments, rather than replaces clinical judgment.

Furthermore, practical device testing of the developed classifier presents its own set of hurdles. Such testing would necessitate ethical approval, collaboration with a qualified anaesthesiologist, and the recruitment of volunteers willing to undergo GA, all of which are essential for further refining and validating the effectiveness of automated classifier's. An interesting conclusion is that whichever model is chosen, parameters may be tuned to suit particular situations. There are opportunities here to refine accuracy performance by accounting for subject age, medical background, and types of anaesthesia employed. In particular, the authors have noted differences in spectral behaviour between propofol and mixed anaesthesia groups in the dataset evaluated in this paper, and believe that there is potential here for anaesthesia-aware classification to further improve performance.

This study has illustrated the potential for future work aimed at developing a fully deployable small-form-factor monitoring device. This would require reduced power and size, but also offer the opportunity to create a more sophisticated device that can potentially operate untethered and in situations such as first-responder contexts and disaster emergency response scenarios as well as offering more flexibility in standard clinical settings.

The authors conclude that significant advances in this area can be achieved with further exploration of the topic in the coming years, leading to smaller and more powerful monitoring systems with many new deployability contexts.

Author Contributions: B.M.: conceptualisation, data curation, formal analysis, investigation, methodology, supervision, writing—original draft preparation, writing—review and editing, validation and visualisation. D.P.: conceptualisation, data curation, formal analysis, investigation, methodology, software and visualisation. C.C.-B.: data curation, formal analysis, investigation, methodology, writing—original draft preparation, writing—review and editing, validation and visualisation. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study. The research exclusively utilised the publicly available EEG database containing electroencephalography (EEG) multitaper spectra and associated conscious/unconscious labels. As no new clinical data or personally identifiable information was collected by the authors, this study falls under the category of non-human subjects research exemption.

Informed Consent Statement: Written informed consent was obtained from the individual depicted in the photograph of Figure 10 for publication of the image. The individual has consented to the use of the photograph for illustrating the EEG headset and sensor positioning.

Data Availability Statement: The dataset presented in the study is available in [10]—only registered users who sign the PhysioNet Restricted Health Data Use Agreement can access the files. The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: During the preparation of this manuscript, generative AI tools (OpenAI GPT-5.2 and GPT-5.5) were used for language refinement, structural organisation and formatting assistance. All of the scientific methodology, experimental design, data analysis and results were undertaken and verified by the authors, who accept full responsibility for the content of this manuscript. The PYNQ-Z2 board was sourced from Farnell Ireland Ltd., December 2020, Republic of Ireland, and contains an ARM Cortex-A9 core (Arm Holdings Plc), embedded within a Zynq-7000 (XC7Z020-1CLG400C) SoC device (Advanced Micro Devices Inc.).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
AUC	Area Under Curve
BCI	Brain Computer Interface
BIS	Bispectral Index
CPU	Central Processor Unit
CNN	Convolution Neural Network
DoA	Depth of Anaesthesia
DPSS	Discrete Prolate Spheroidal Sequences
EEG	Electroencephalography
FT	Fourier Transform
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GA	General Anaesthesia
HLS	High Level Synthesis
ICU	Intensive Care Unit
LDA	Linear Discriminant Analysis
LOC	Loss of Consciousness
LPP	Locality Preserving Projection
LR	Logistic Regression
ML	Machine Learning
MSA	Multitaper Spectral Analysis

NN	Neural Network
OR	Operating Room
PTQ	Post-Training Quantisation
PSD	Power Spectral Density
RNN	Recurrent Neural Network
ROC	Return of Consciousness
RUS	Random Undersampling
SMOTE	Synthetic Minority Oversampling Technique
SoC	System-on-Chip
SVM	Support Vector Machines
QAT	Quantisation-Aware Training
QNN	Quantised Neural Network

Appendix A

This appendix contains the five-fold cross-validation table results for the ML models attached to the radar plots in Figure 11 for the SMOTE, RUS and Original datasets. Each column (Fold 1–Fold 5) corresponds to a different held-out test fold in the five-fold cross-validation scheme. Folds were split at the subject level to ensure subject independence between training and test data. For each fold as shown in Table A1, SMOTE was applied only to the corresponding training partition (the remaining four folds) to address class imbalance, while the held-out test fold retained its original class distribution. Similarly for Tables A2 and A3 results, subject-independent RUS class-balancing and original (imbalanced) data were used in the training folds, while the held-out test fold retained its original class distribution.

Table A1. SMOTE dataset—model results.

Model	Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std. Dev
LR	Accuracy	0.9563	0.8735	0.8341	0.8601	0.8603	0.8768	0.0417
	Precision	0.8746	0.7484	0.5504	0.7311	0.5813	0.6972	0.1185
	Specificity	0.9588	0.9209	0.8257	0.8674	0.8478	0.8841	0.0489
	Sensitivity	0.9479	0.7272	0.8681	0.8429	0.9145	0.8601	0.0758
	F1 Score	0.9098	0.7376	0.6737	0.7830	0.7108	0.7630	0.0816
	AUC	0.9846	0.9136	0.9113	0.9316	0.9481	0.9378	0.0269
LDA	Accuracy	0.9579	0.8731	0.8492	0.8595	0.8712	0.8822	0.0388
	Precision	0.8830	0.7590	0.5820	0.7352	0.6041	0.7127	0.1100
	Specificity	0.9621	0.9276	0.8522	0.8721	0.8621	0.8952	0.0424
	Sensitivity	0.9442	0.7048	0.8368	0.8301	0.9102	0.8452	0.0825
	F1 Score	0.9126	0.7309	0.6865	0.7797	0.7262	0.7672	0.0785
	AUC	0.9844	0.9116	0.9091	0.9303	0.9493	0.9370	0.0278
LDA+LR	Accuracy	0.9569	0.8723	0.8384	0.8577	0.8628	0.8776	0.0412
	Precision	0.8766	0.7471	0.5592	0.7276	0.5860	0.6993	0.1157
	Specificity	0.9595	0.9209	0.8346	0.8656	0.8504	0.8862	0.0468
	Sensitivity	0.9481	0.7224	0.8358	0.8393	0.9166	0.8560	0.0778
	F1 Score	0.9110	0.7345	0.6758	0.7795	0.7150	0.7632	0.0811
	AUC	0.9844	0.9116	0.9091	0.9303	0.9493	0.9370	0.0278
NN	Accuracy	0.9496	0.8866	0.8439	0.9381	0.9058	0.9048	0.0378
	Precision	0.8400	0.7370	0.5718	0.8662	0.6849	0.7400	0.1070
	Specificity	0.9441	0.9037	0.8467	0.9380	0.9019	0.9069	0.0347
	Sensitivity	0.9676	0.8338	0.8324	0.9382	0.9229	0.8990	0.0557
	F1 Score	0.8993	0.7824	0.6779	0.9007	0.7863	0.8093	0.0836
	AUC	0.9878	0.9258	0.9063	0.9815	0.9726	0.9602	0.0295

Table A2. RUS dataset—model results.

Model	Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std. Dev
LR	Accuracy	0.8388	0.8622	0.8187	0.9142	0.8272	0.8522	0.0343
	Precision	0.7255	0.7789	0.6195	0.6958	0.5409	0.6271	0.0834
	Specificity	0.8951	0.9133	0.8023	0.9034	0.8046	0.8638	0.0496
	Sensitivity	0.6973	0.7384	0.8627	0.9612	0.9170	0.8353	0.1017
	F1 Score	0.7111	0.7581	0.7212	0.8072	0.6804	0.7356	0.0436
	AUC	0.8725	0.9016	0.9124	0.9710	0.9199	0.9155	0.0321
LDA	Accuracy	0.8341	0.8619	0.8301	0.9200	0.8465	0.8603	0.0315
	Precision	0.7419	0.7858	0.6411	0.7149	0.5736	0.6914	0.0754
	Specificity	0.9049	0.9182	0.8220	0.9128	0.8292	0.8774	0.0426
	Sensitivity	0.6875	0.7256	0.8518	0.9514	0.9151	0.8263	0.1035
	F1 Score	0.7137	0.7545	0.7316	0.8163	0.7052	0.7443	0.0398
	AUC	0.8770	0.9003	0.9146	0.9719	0.9245	0.9177	0.0315
LDA+LR	Accuracy	0.8398	0.8604	0.8214	0.9162	0.8371	0.8550	0.0330
	Precision	0.7278	0.7727	0.6238	0.7011	0.5566	0.6764	0.0770
	Specificity	0.8963	0.9099	0.8058	0.9058	0.8150	0.8666	0.0462
	Sensitivity	0.6974	0.7407	0.8632	0.9614	0.9252	0.8376	0.1027
	F1 Score	0.7123	0.7564	0.7243	0.8108	0.6951	0.7398	0.0408
	AUC	0.8770	0.9003	0.9146	0.9719	0.9245	0.9177	0.0315
NN	Accuracy	0.8464	0.8756	0.8367	0.9278	0.8497	0.8672	0.0329
	Precision	0.7225	0.7977	0.6456	0.7334	0.5776	0.6960	0.0764
	Specificity	0.8887	0.9193	0.8187	0.9195	0.8286	0.8749	0.0435
	Sensitivity	0.7398	0.7701	0.8852	0.9637	0.9340	0.8586	0.0887
	F1 Score	0.7326	0.7837	0.7466	0.8329	0.7138	0.7619	0.0422
	AUC	0.8729	0.9292	0.9359	0.9861	0.9470	0.9342	0.0365

Table A3. Original dataset—model results.

Model	Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std. Dev
LR	Accuracy	0.9163	0.8928	0.8927	0.8262	0.9184	0.8893	0.0334
	Precision	0.9146	0.7439	0.7601	0.7885	0.8469	0.8108	0.0626
	Specificity	0.9815	0.9302	0.9247	0.9234	0.9683	0.9456	0.0244
	Sensitivity	0.6889	0.7535	0.7870	0.6162	07151	0.7121	0.0584
	F1 Score	0.7859	0.7487	0.7733	0.6918	0.7754	0.7550	0.0339
	AUC	0.9598	0.9403	0.9365	0.8259	0.9658	0.9257	0.0511
LDA	Accuracy	0.9212	0.8875	0.8903	0.8412	0.9202	0.8921	0.0292
	Precision	0.9186	0.7251	0.7476	0.7967	0.8389	0.8054	0.0690
	Specificity	0.9819	0.9229	0.9184	0.9209	0.9653	0.9419	0.0265
	Sensitivity	0.7097	0.7556	0.7978	0.6690	0.7363	0.7337	0.0433
	F1 Score	0.8008	0.7400	0.7719	0.7273	0.7843	0.7648	0.0274
	AUC	0.9613	0.9407	0.9382	0.8324	0.9646	0.9274	0.0487
LDA+LR	Accuracy	0.9165	0.8887	0.8922	0.8432	0.9171	0.8915	0.0269
	Precision	0.9197	0.7338	0.7592	0.8144	0.8433	0.8141	0.0650
	Specificity	0.9828	0.9273	0.9245	0.9310	0.9676	0.9466	0.0239
	Sensitivity	0.6857	0.7454	0.7853	0.6536	0.7110	0.7162	0.0458
	F1 Score	0.7856	0.7395	0.7720	0.7252	0.7715	0.7588	0.0226
	AUC	0.9613	0.9407	0.9382	0.8234	0.9646	0.9274	0.0487
NN	Accuracy	0.9130	0.9271	0.9145	0.8603	0.9394	0.9109	0.0270
	Precision	0.8648	0.8487	0.7947	0.8325	0.8864	0.8454	0.0310
	Specificity	0.9675	0.9617	0.9333	0.9348	0.9750	0.9545	0.0172
	Sensitivity	0.7230	0.7986	0.8526	0.6995	0.7942	0.7736	0.0554
	F1 Score	0.7875	0.8229	0.8226	0.7602	0.8378	0.8062	0.0283
	AUC	0.9565	0.9765	0.9599	0.8883	0.9774	0.9517	0.0328

Appendix B

This appendix highlights the exclusion reasoning for subject IDs {212, 630} from the OR mixed cohort in this study. We examine three cases to illustrate the point.

In case one, as shown in Figure A1, (subject ID 208) the data is considered valid by examining the spectral characteristics (top) and annotated labelling from the dataset used in this experimental study (middle). Here, it can be observed that the subject's data is labelled as conscious for a short time at the beginning of the procedure and then labelled as unconscious there after. Viewing the spectral plot on top, one can observe short increases of spectral power which are not accounted for in the binary classified labels, but maybe due to body movements, burst suppression activities, etc. that forms the basis of future work study. These type of characteristics are entirely expected, and therefore this patient data was not excluded from the study in this manuscript.

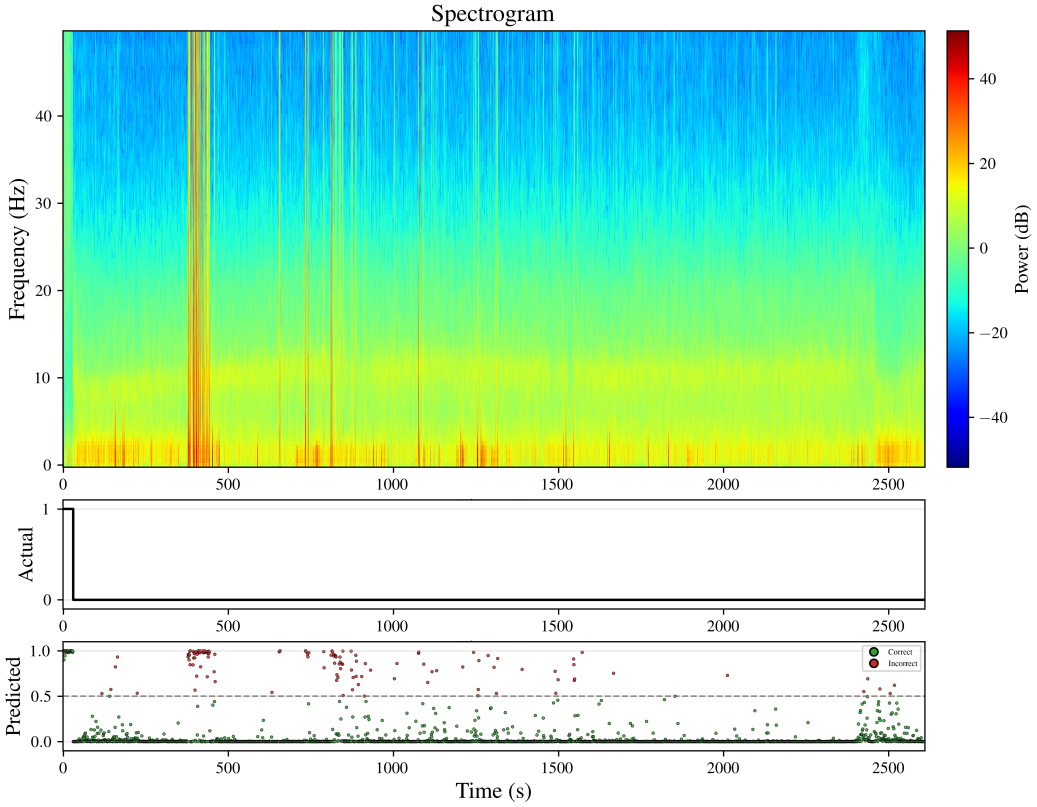


Figure A1. Case-1: Spectrogram of subject {208} (**top**), actual labels (**middle**) and model predictions (**bottom**).

Conversely, the spectrogram in Figure A2 and Figure A3 for case 2 and 3 (Subject IDs 212 and 630, respectively), as obtained from the third-party dataset, illustrates the inconsistency of labelling between the actual annotated reference labels (middle plots) and the observed spectral characteristics. These inconsistencies are especially noticeable for subject {212} over the 2 s EEG intervals from time 350 s to 900 s and similarly for subject {630} from 0 to 1400 s. We concluded that the labelling could not be validated without access to the original patient data recordings and decided to exclude those two cases from the study results. All other subject data as provided in the available dataset and shown in Table 1 are included.

The subject data was also input to the NN-SMOTE model with the predictions (bottom plots) confirming that the model is unable to reliably predict consciousness due to the inconsistencies of labelling.

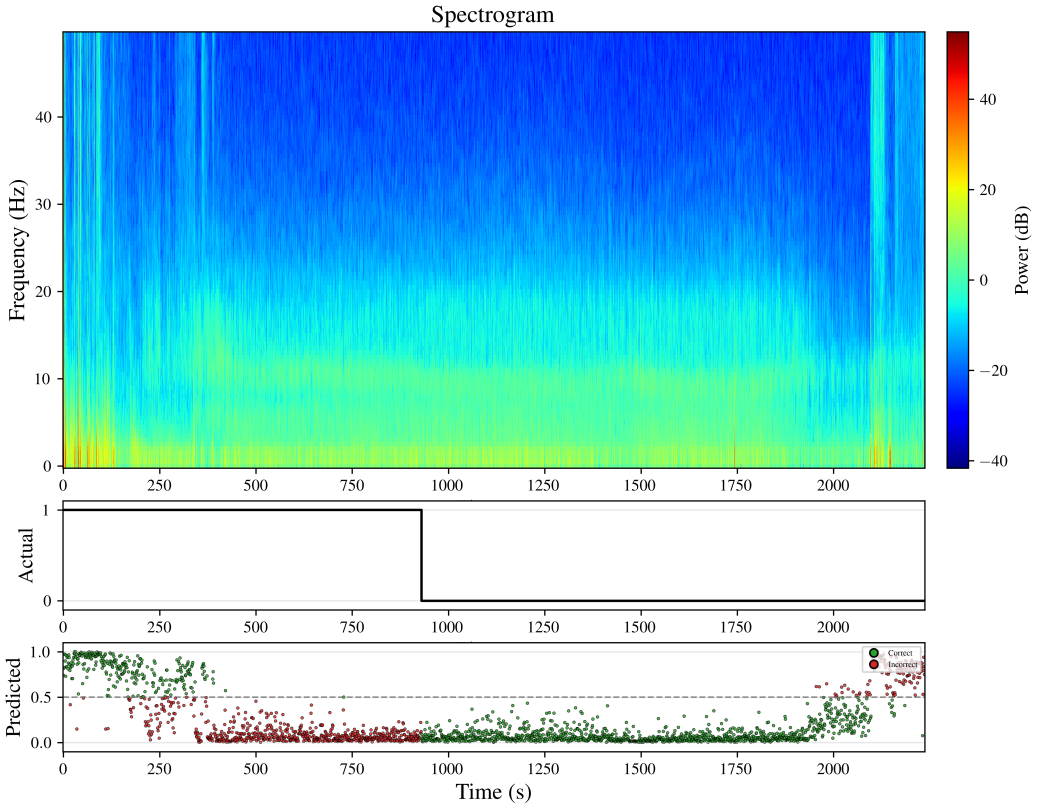


Figure A2. Case-2: Spectrogram of subject {212} (**top**), labelling (**middle**) and model predictions (**bottom**).

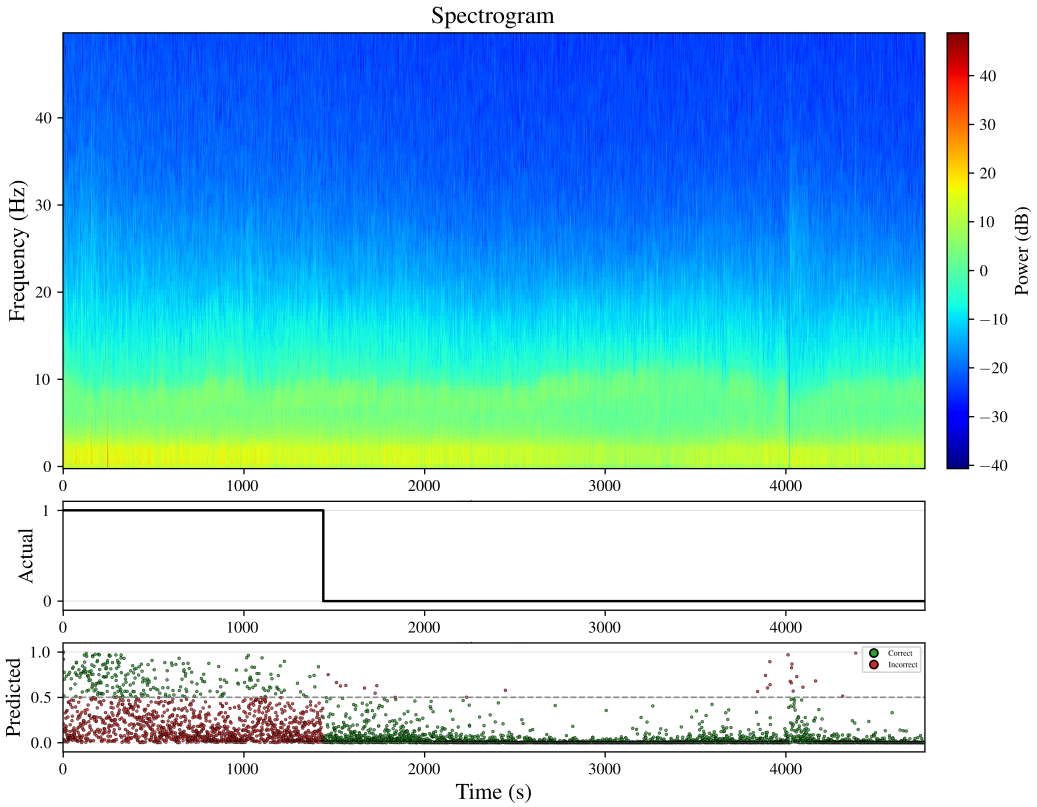


Figure A3. Case-3: Spectrogram of subject {630} (**top**), labelling (**middle**) and model predictions (**bottom**).

References

1. Miladinović, A.; Ajčević, M.; Busan, P.; Jarmolowska, J.; Deodato, M.; Mezzarobba, S.; Battaglini, P.P.; Accardo, A. EEG changes and motor deficits in Parkinson's disease patients: Correlation of motor scales and EEG power bands. *Procedia Comput. Sci.* **2021**, *192*, 2616–2623. [CrossRef]
2. NHS. Electroencephalogram (EEG). Available online: https://www.nhs.uk/conditions/electroencephalogram (accessed on 4 August 2026).
3. Bouafif, L. Monitoring of Anesthesia by Bispectral Analysis of EEG signals. *Comput. Math. Methods Med.* **2021**, *2021*, 9961998. [CrossRef]
4. Zeng, W.; Shan, L.; Su, B.; Du, S. Epileptic seizure detection with deep EEG features by convolutional neural network and shallow classifiers. *Front. Neurosci.* **2023**, *17*, 1145526. [CrossRef]
5. Ahmad, I.; Wang, X.; Javeed, D.; Kumar, P.; Samuel, O.W.; Chen, S. A Hybrid Deep Learning Approach for Epileptic Seizure Detection in EEG signals. *IEEE J. Biomed. Health Inform.* **2023**, *30*, 1019–1029. [CrossRef]
6. Lechat, B.; Hirotsu, C.; Appleton, S.; Younes, M.; Adams, R.J.; Vakulin, A.; Hansen, K.; Zajamsek, B.; Wittert, G.; Catcheside, P.; et al. A novel EEG marker predicts perceived sleepiness and poor sleep quality. *Sleep* **2022**, *45*, zsac051. [CrossRef]
7. Xiang, C.; Fan, X.; Bai, D.; Lv, K.; Lei, X. A resting-state EEG dataset for sleep deprivation. *Sci. Data* **2024**, *11*, 427. [CrossRef]
8. Siuly, S.; Li, Y.; Zhang, Y. Electroencephalogram (EEG) and Its Background. In *EEG Signal Analysis and Classification: Techniques and Applications*; Springer International Publishing: Cham, Switzerland, 2016; pp. 3–21. [CrossRef]
9. Tan, T. General Anaesthesia. Available online: https://www.drtonytan.com/post/general-anaesthesia (accessed on 4 August 2026).
10. Abel, J.; Badgeley, M.; Meschede-Krasa, B.; Schamberg, G.; Garwood, I.; Lecamwasam, K.; Chakravarty, S.; Zhou, D.; Keating, M.; Purdon, P.; et al. Multitaper spectra recorded during GABAergic anesthetic unconsciousness (version 1.0.0). *PhysioNet* **2021**, RRID:SCR_007345. [CrossRef]
11. Voss, L.; Sleigh, J. Monitoring consciousness: The current status of EEG-based depth of anaesthesia monitors. *Best Pract. Res. Clin. Anaesthesiol.* **2007**, *21*, 313–325. [CrossRef]
12. Kreuzer, M. EEG Based Monitoring of General Anesthesia: Taking the Next Steps. *Front. Comput. Neurosci.* **2017**, *11*, 56. [CrossRef]
13. Pawar, N.; Barreto Chang, O.L. Burst Suppression During General Anesthesia and Postoperative outcomes: Mini Review. *Front. Syst. Neurosci.* **2022**, *15*, 767489. [CrossRef]
14. Chan, M.T.; Cheng, B.C.; Lee, T.M.; Gin, T. CODA Trial Group. BIS-guided anesthesia decreases postoperative delirium and cognitive decline. *J. Neurosurg. Anesthesiol.* **2013**, *25*, 33–42.
15. Clinic, M. General Anesthesia. Available online: https://www.mayoclinic.org/tests-procedures/anesthesia/about/pac-20384568 (accessed on 4 August 2026).
16. Abel, J.H.; Badgeley, M.A.; Meschede-Krasa, B.; Schamberg, G.; Garwood, I.C.; Lecamwasam, K.; Chakravarty, S.; Zhou, D.W.; Keating, M.; Purdon, P.L. Machine learning of EEG spectra classifies unconsciousness during GABAergic anesthesia. *PLoS ONE* **2021**, *16*, e0246165. [CrossRef]
17. Open-BCI. Cyton Biosensing Board (8-CHANNELS). Available online: https://shop.openbci.com/products/cyton-biosensing-board-8-channel (accessed on 4 August 2026).
18. AMD. PYNQ: Python productivity. Available online: https://www.pynq.io (accessed on 4 August 2026).
19. Altıntop, Ç.G. Enhancing Diagnostic Accuracy of Neurological Disorders Through Feature-Driven Multi-Class Classification with Machine Learning. *Diagnostics* **2025**, *15*, 2132. [CrossRef]
20. Guerrero, M.C.; Parada, J.S.; Espitia, H.E. EEG signal analysis using classification techniques: Logistic regression, artificial neural networks, support vector machines, and convolutional neural networks. *Heliyon* **2021**, *7*, e07258. [CrossRef]
21. Alam, M.S.; Siddiqui, Y.; Hasan, M.; Farooq, O.; Himeur, Y. Energy-Efficient FPGA Based Sleep Apnea Detection Using EEG Signals. *IEEE Access* **2024**, *12*, 40182–40195. [CrossRef]
22. Xie, Y.; Majoros, T.; Oniga, S. FPGA-Based Hardware Accelerator on Portable Equipment for EEG Signal Patterns Recognition. *Electronics* **2022**, *11*, 2410. [CrossRef]
23. Kaiser, H.A.; Knapp, J.; Sleigh, J.; Avidan, M.S.; Stuber, F.; Hight, D. The quantitative EEG in electroencephalogram-based brain monitoring during general anesthesia. *Anaesthesist* **2021**, *70*, 531–547. [CrossRef]
24. Gkintoni, E.; Aroutzidis, A.; Antonopoulou, H.; Halkiopoulos, C. From neural networks to emotional networks: A systematic review of EEG-based emotion recognition in cognitive neuroscience and real-world applications. *Brain Sci.* **2025**, *15*, 220. [CrossRef]
25. Akor, P.A.; Enemali, G.; Muhammad, U.; Singh, R.R.; Larijani, H. An Attention-Residual Convolutional Network for Real-Time Seizure Classification on Edge Devices. *Sensors* **2025**, *25*, 6855. [CrossRef]
26. Jagadeesan, N.; Wolfson, M.; Chen, Y.; Willingham, M.; Avidan, M.S. Brain monitoring during general anesthesia. *Trends Anaesth. Crit. Care* **2013**, *3*, 13–18. [CrossRef]
27. Mashour, G.A.; Rosow, C. Monitoring the depth of anesthesia. In *Handbook of Clinical Neurophysiology*; Elsevier: Amsterdam, The Netherlands, 2008; Volume 8, pp. 899–908. [CrossRef]

28. Dubost, C.; Humbert, P.; Benizri, A.; Tourtier, J.P.; Vayatis, N.; Vidal, P.P. Selection of the Best Electroencephalogram Channel to Predict the Depth of Anesthesia. *Front. Comput. Neurosci.* **2019**, *13*, 65. [CrossRef]
29. Carneiro, R.A.A.G.; Pereira, L.A.G. Depth of Anesthesia Monitoring and Artificial Intelligence. *Curr. Anesthesiol. Rep.* **2025**, *15*, 19. [CrossRef]
30. Musialowicz, T.; Lahtinen, P. Current Status of EEG-Based Depth-of-Consciousness Monitoring During General Anesthesia. *Curr. Anesthesiol. Rep.* **2014**, *4*, 251–260. [CrossRef]
31. Khan, F.H.; Ashraf, U.; Bin Altaf, M.A.; Saadeh, W. A Patient-Specific Machine Learning based EEG Processor for Accurate Estimation of Depth of Anesthesia. In *Proceedings of the 2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*; IEEE: New York, NY, USA, 2018; pp. 1–4. [CrossRef]
32. Schmierer, T.; Li, T.; Li, Y. Harnessing machine learning for EEG signal analysis: Innovations in depth of anaesthesia assessment. *Artif. Intell. Med.* **2024**, *151*, 102869. [CrossRef]
33. Ferreira, A.L.; Vide, S.; Nunes, C.S.; Neto, J.; Amorim, P.; Mendes, J. Implementation of Neural Networks to Frontal Electroencephalography for the Identification of the Transition Responsiveness/Unresponsiveness During Induction of General Anesthesia. *IRBM* **2021**, *42*, 390–397. [CrossRef]
34. Frey, J. Comparison of an Open-hardware Electroencephalography Amplifier with Medical Grade Device in Brain-computer Interface Applications. In *Proceedings of the 3rd International Conference on Physiological Computing Systems—PhyCS*; SciTePress: Setúbal, Portugal, 2016. [CrossRef]
35. Bittium. Bittium BrainStatus™ Wireless EEG Amplifier. Available online: https://www.bittium.com/medical/bittium-brainstatus (accessed on 4 August 2026).
36. Arif, R.; Wijaya, S.K.; Prajitno, P.; Gani, H.S. Development of electroencephalography (EEG) data acquisition system based on FPGA PYNQ. *AIP Conf. Proc.* **2019**, *2092*, 020026. [CrossRef]
37. Xie, Y.; Oniga, S. A comprehensive review of hardware acceleration techniques and convolutional neural networks for EEG signals. *Sensors* **2024**, *24*, 5813. [CrossRef]
38. Saadeh, W.; Khan, F.H.; Altaf, M.A.B. Design and Implementation of a Machine Learning Based EEG Processor for Accurate Estimation of Depth of Anesthesia. *IEEE Trans. Biomed. Circuits Syst.* **2019**, *13*, 658–669. [CrossRef]
39. Chen, K.; Wu, Z.; Xu, Y.; Wang, R.; Jiang, S.; Chen, L.; Zhao, F.; Mao, Y.; Li, M. A neuromorphic system based on spiking-timing-dependent plasticity for evaluating wakefulness and anesthesia states using intracranial EEG signals. *View* **2025**, *6*, 20240138. [CrossRef]
40. Ingole, D.; Holaza, J.; Takács, B.; Kvasnica, M. FPGA-based explicit model predictive control for closed-loop control of intravenous anesthesia. In *Proceedings of the 2015 20th International Conference on Process Control (PC)*; IEEE: New York, NY, USA, 2015; pp. 42–47.
41. Abood, L.H.; Karam, E.H.; Issa, A.H. FPGA Implementation of Single Neuron PID Controller for Depth of Anesthesia Based on PSO. In *Proceedings of the 2018 Third Scientific Conference of Electrical Engineering (SCEE)*; IEEE: New York, NY, USA, 2018; pp. 247–252.
42. Roy, Y.; Banville, H.; Albuquerque, I.; Gramfort, A.; Falk, T.H.; Faubert, J. Deep learning-based electroencephalography analysis: A systematic review. *J. Neural Eng.* **2019**, *16*, 051001. [CrossRef]
43. Kostarelos, F.; MacNamee, C.; Mullane, B. A Hardware Implementation of a qEEG-Based Discriminant Function for Brain Injury Detection. In *Proceedings of the 2021 IEEE Biomedical Circuits and Systems Conference (BioCAS)*; IEEE: New York, NY, USA, 2021; pp. 1–6. [CrossRef]
44. Umuroglu, Y.; Fraser, N.J.; Gambardella, G.; Blott, M.; Leong, P.; Jahre, M.; Vissers, K. FINN: A Framework for Fast, Scalable Binarized Neural Network Inference. In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 65–74. [CrossRef]
45. Pappalardo, A. Xilinx/Brevitas. Available online: https://zenodo.org/records/15375017 (accessed on 4 August 2026).
46. Blott, M.; Preußer, T.B.; Fraser, N.J.; Gambardella, G.; O'Brien, K.; Umuroglu, Y.; Leeser, M.; Vissers, K. FINN-R: An End-to-End Deep-Learning Framework for Fast Exploration of Quantized Neural Networks. *ACM Trans. Reconfigurable Technol. Syst.* **2018**, *11*, 16. [CrossRef]
47. Ghosh, S.; Chatterjee, A.; Munshi, S. UaBMA-OLPP: A Novel Manifold-Inspired Technique for sEMG-based Hand Movement and Object Grasp Recognition. *IEEE J. Biomed. Health Inform.* **2025**, *30*, 5392–5399. [CrossRef]
48. Babadi, B.; Brown, E.N. A Review of Multitaper Spectral Analysis. *IEEE Trans. Biomed. Eng.* **2014**, *61*, 1555–1564. [CrossRef]
49. Purdon, P.L.; Pierce, E.T.; Mukamel, E.A.; Prerau, M.J.; Walsh, J.L.; Wong, K.F.K.; Salazar-Gomez, A.F.; Harrell, P.G.; Sampson, A.L.; Cimenser, A.; et al. Electroencephalogram signatures of loss and recovery of consciousness from propofol. *Proc. Natl. Acad. Sci.* **2013**, *110*, E1142–E1151. [CrossRef]
50. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
51. Cox, D.R. The Regression Analysis of Binary Sequences. *J. R. Stat. Soc. Ser. B* **1958**, *20*, 215–242. [CrossRef]
52. Huberty, C.J. Discriminant Analysis. *Rev. Educ. Res.* **1975**, *45*, 543–598. [CrossRef]

53. Sharma, N. Single-trial P300 Classification using PCA with LDA, QDA and Neural Networks. *arXiv* **2017**, arXiv:1712.01977.
54. Craik, A.; He, Y.; Contreras-Vidal, J.L. Deep learning for electroencephalogram (EEG) classification tasks: A review. *J. Neural Eng.* **2019**, *16*, 031001. [CrossRef]
55. Fernandes, J.V.M.R.; Alexandria, A.R.D.; Marques, J.A.L.; Assis, D.F.D.; Motta, P.C.; Silva, B.R.D.S. Emotion Detection from EEG Signals Using Machine Deep Learning Models. *Bioengineering* **2024**, *11*, 782. [CrossRef]
56. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY USA, 2019; pp. 8024–8035.
57. Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; Bengio, Y. Quantized Neural Networks: Training neural networks with low precision weights and activations. *J. Mach. Learing Res.* **2017**, *18*, 6869–6898.
58. Cecotti, H.; Dhaliwal, D.; Singh, H.; Meena, Y.K. Post-Training Quantization in Brain-Computer Interfaces based on Event-Related Potential Detection. *arXiv* **2024**, arXiv:2410.07920.
59. Hua Zhong, S.; Shi, J.; Wang, Y. Knowledge-guided quantization-aware training for EEG-based emotion recognition. *J. Vis. Commun. Image Represent.* **2025**, *108*, 104415. [CrossRef]
60. Kastner, R.; Matai, J.; Neuendorffer, S. Parallel Programming for FPGAs. *arXiv* **2018**, arXiv:1805.03648.
61. Freire, P.; Srivallapanondh, S.; Spinnler, B.; Napoli, A.; Costa, N.; Prilepsky, J.E.; Turitsyn, S.K. Computational Complexity Optimization of Neural Network-Based Equalizers in Digital Signal Processing: A Comprehensive Approach. *J. Light. Technol.* **2024**, *42*, 4177–4201. [CrossRef]
62. Ngo, D.; Park, H.C.; Kang, B. Edge Intelligence: A Review of Deep Neural Network Inference in Resource-Limited Environments. *Electronics* **2025**, *14*, 2495. [CrossRef]
63. Andronic, M.; Constantinides, G.A. NeuraLUT-Assemble: Hardware-Aware Assembling of Sub-Neural Networks for Efficient LUT Inference. In *Proceedings of the 2025 IEEE 33rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*; IEEE: New York, NY, USA, 2025; pp. 208–216.
64. Brook, K.; Agarwala, A.V.; Li, F.; Purdon, P.L. Depth of anesthesia monitoring: An argument for its use for patient safety. *Curr. Opin. Anesthesiol.* **2024**, *37*, 689–696. [CrossRef]
65. Xilinx/AMD. FINN. Available online: https://xilinx.github.io/finn/quickstart.html (accessed on 4 August 2026).
66. Mathur, S.; Patel, J.; Goldstein, S.; Hendrix, J.M.; Jain, A. Bispectral Index. In *StatPearls [Internet]*; StatPearls Publishing: Treasure Island, FL, USA, 2023.
67. Guo, Y.; Wang, Z.; Hong, Q.; Luo, H.; Qiu, X.; Liang, L. A 60-mode high-throughput parallel-processing FFT processor for 5G/4G applications. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2022**, *31*, 219–232. [CrossRef]
68. Kaya, Z.; Garrido, M.; Takala, J. Memory-based FFT architecture with optimized number of multiplexers and memory usage. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *70*, 3084–3088. [CrossRef]
69. Bell, J.; Laasch, H.U.; Wilbraham, L.; England, R.; Morris, J.; Martin, D. Bispectral index monitoring for conscious sedation in intervention: Better, safer, faster. *Clin. Radiol.* **2004**, *59*, 1106–1113. [CrossRef]
70. Arnal, P.J.; Thorey, V.; Debellemaniere, E.; Ballard, M.E.; Bou Hernandez, A.; Guillot, A.; Jourde, H.; Harris, M.; Guillard, M.; Van Beers, P.; et al. The Dreem Headband compared to polysomnography for electroencephalographic signal acquisition and sleep staging. *Sleep* **2020**, *43*, zsaa097. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.