sensors

MDPI

*Article*

# Robust Motor Imagery Tasks Classification Approach Using Bayesian Neural Network

**Daily Milanés-Hermosilla [1], Rafael Trujillo-Codorniú [1,2], Saddid Lamar-Carbonell [3], Roberto Sagaró-Zamora [4], Jorge Jadid Tamayo-Pacheco [3], John Jairo Villarejo-Mayor [5] and Denis Delisle-Rodriguez [6,*]**

1 Department of Automatic Engineering, University of Oriente, Santiago de Cuba 90500, Cuba
2 Electronics, Communications and Computing Services Company for the Nickel Industry, Holguín 80100, Cuba
3 Independent Researcher, 33720 Tampere, Finland
4 Department of Mechanical Engineering, University of Oriente, Santiago de Cuba 90500, Cuba
5 Department of Electrical and Electronic Engineering, Federal University of Santa Catarina, Florianopolis 88040-900, SC, Brazil
6 Postgraduate Program in Neuroengineering, Edmond and Lily Safra International Institute of Neurosciences, Santos Dumont Institute, Macaiba 59280-000, RN, Brazil
* Correspondence: denis.rodriguez@isd.org.br

**Abstract:** The development of Brain–Computer Interfaces based on Motor Imagery (MI) tasks is a relevant research topic worldwide. The design of accurate and reliable BCI systems remains a challenge, mainly in terms of increasing performance and usability. Classifiers based on Bayesian Neural Networks are proposed in this work by using the variational inference, aiming to analyze the uncertainty during the MI prediction. An adaptive threshold scheme is proposed here for MI classification with a reject option, and its performance on both datasets 2a and 2b from BCI Competition IV is compared with other approaches based on thresholds. The results using subject-specific and non-subject-specific training strategies are encouraging. From the uncertainty analysis, considerations for reducing computational cost are proposed for future work.

**Keywords:** brain–computer interface; Bayesian Neural Network; variational inference; uncertainty estimation; classification with reject option



**Citation:** Milanés-Hermosilla, D.; Trujillo-Codorniú, R.; Lamar-Carbonell, S.; Sagaró-Zamora, R.; Tamayo-Pacheco, J.J.; Villarejo-Mayor, J.J.; Delisle-Rodriguez, D. Robust Motor Imagery Tasks Classification Approach Using Bayesian Neural Network. *Sensors* **2023**, *23*, 703. https://doi.org/10.3390/s23020703

Academic Editor: Sampsa Vanhatalo

Received: 15 November 2022
Revised: 30 December 2022
Accepted: 5 January 2023
Published: 8 January 2023



## 1. Introduction

A Brain–Computer Interface (BCI) provides a direct communication pathway between the user's brain and external devices/software in order to monitor and/or repair sensorimotor and cognitive functions [1–3]. The BCI's input receives brain commands typically via electroencephalograms (EEG), which is a non-invasive, low-cost, high temporal resolution, and portable technique. Many EEG-based BCIs have been developed by using different paradigms, such as P300 event-related potentials [4,5], steady-state visual evoked potentials (SSVEP) [6,7], and sensorimotor rhythms (SMR) [8–10]. Systems using SMR have demonstrated promising results and great potential for training and recovering motor skills through various types of motor imagery tasks. However, the poor EEG signal-to-noise ratio and intra-and inter-subject variability make the MI classification still a challenge.

Deep learning techniques for MI classification have shown better performance than handcrafted feature extraction methods with respect to accuracy [11–19]. This success of deep learning also appears in other research areas, such as computer vision, image, and language processing, among others. Nevertheless, these approaches are deterministic architectures, and hence they lack strategies to measure classification trust.

The source of uncertainty in deep learning models may come from data input because of noise and inaccurate measurement (aleatoric uncertainty). Uncertainty from other

sources due to lack of knowledge is termed epistemic uncertainty [20,21]. For this reason, uncertainty analysis is crucial to providing robust and reliable classification, which is essential in developments that require low error tolerance, such as medical and autonomous navigation systems. For instance, in these systems, classification mistakes may produce poor decisions, and consequently fatal events. In medical applications, uncertainty analysis has been widely used for disease diagnoses, such as COVID-19 [22], tuberculosis [23], ataxia [24], cancer [25], diabetic retinopathy [26,27], and epileptogenic brain malformations [28]. To the best of our knowledge, strategies for uncertainty analysis have been little explored in MI-based BCIs, excluding a previous study [29] that applied Monte Carlo dropout techniques with this purpose, considered as an approximate Bayesian inference in deep Gaussian processes [21]. The current paper follows a different method by applying BNN.

During classification, the output score vector produced by deterministic architectures is often erroneously interpreted as model confidence (as it appears as probabilities due to the Softmax activation function). Various studies have demonstrated that, even when the highest score is large, the score vector can be a poor predictor of uncertainty. For instance, in [30], the authors perturbed a single pixel in some CIFAR-10 images, and this small change sometimes caused classification mistakes with high output scores, employing well-known models such as VGG [31] and AlexNet [32]. This result increased the safety concerns in critical applications.

Approaches using Bayesian modeling have shown good performance during uncertainty analysis of deep learning models. For instance, the Bayesian Neural Network (BNN) models [33–37] introduce stochastic components over the network parameters $\theta$, simulating various possible models with their probability distribution $p(\theta)$. Although these architectures are computationally more expensive than deterministic ones, they offer prediction uncertainty and consequently more reliable outputs.

This work introduces two BNN architectures termed SCBNN and SCBNN-MOPED for MI classification and uncertainty quantification, which were trained here by using the variational inference procedure. In our study, the SCBNN-MOPED architecture uses the MOPED method from [38] to establish the prior distribution of each weight by employing the weights of the pre-trained equivalent deterministic network. The SCBNN architecture initializes the prior distribution of each weight as a standard Gaussian distribution $\mathcal{N}(0,1)$, which is established by default in the Python's TensorFlow Probability package.

As another novelty, an adaptive threshold scheme is also introduced here to implement the MI classification with a reject option. As an advantage, this scheme uses h a closed formula to calculate the threshold, rather than performing an exhaustive search over the space of possible threshold values as performed in previous works [39–41]. The datasets 2a and 2b from the BCI Competition IV [42] were used for evaluation, considering both subject-specific and non-subject-specific strategies for training. As a result, our approach improved the MI classification with/without a reject option. In terms of efficacy, this proposed threshold scheme performed better with respect to other approaches that estimate the best threshold value by using an uncertainty metric.

The main contributions of this current research are listed as follows:

1. Two BNN architectures by applying the variational inference method for MI classification are proposed, confirming the advantage of using MOPED's prior distribution for Bayesian models. Additionally, an efficient implementation to reduce computational cost for practical real-world applications is proposed.
2. An adaptive threshold scheme based on a closed formula for robust MI classification with a reject option is proposed here.

## 2. Materials and Methods

### *2.1. Datasets Description*

The two popular datasets 2a and 2b from BCI Competition IV were used.

The dataset 2a contains EEG signals with a sampling rate at 250 Hz, recorded during four MI tasks (left hand, right hand, tongue, and foot) from nine healthy subjects over 22 locations (Fz, FC1, FC3, FCz, FC2, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CP1, CPz, CP2, CP4, P1, Pz, P2 and POz). Two sessions on different days were collected for each subject, completing a total of 288 trials per session. In our study, the first and second sessions were used for training and testing, respectively, as in [13,43–47].

The dataset 2b comprises EEG signals collected during the execution of two MI tasks (left hand and right hand) from nine subjects with three bipolar EEG channels (around C3, Cz, and C4), using a sampling rate at 250 Hz. Each subject completed five sessions with a total of 720 trials, comprising 120 trials per session in the first two sessions and 160 trials per session in the remaining three sessions. In our research, the first three sessions were used for training, and the last two sessions for testing, as in [15–18,44,48].

### 2.2. Preprocessing and Data Augmentation

Previous studies [49–54] reported that real or imagined unilateral movement enhances the mu (8 to 12 Hz) and beta (13 to 30 Hz) rhythms over the primary motor cortex in both contralateral and ipsilateral hemispheres, respectively; phenomena known as event-related desynchronization/synchronization (ERD/ERS) [55]. Thus, the EEG signals of each trial were band-pass filtered in our study with a frequency range from 4 to 38 Hz through a fourth-order Butterworth filter [13,14,45,47,56], aiming to preserve the ERD and ERS potentials, rejecting noise and undesirable physiological and non-physiological artifacts. Afterwards, each filtered EEG trial $\mathbf{x}$ was standardized as $(\mathbf{x}(t_i) - \mu(t_i))/\sigma(t_i)$ by applying the electrode-wise exponential moving standardization with a decay factor of 0.999, which computes both mean $\mu(t_i)$ and variance $\sigma^2(t_i)$ values taken at sample $t_i$ [45]. The starting values $\mu(t_0)$ and $\sigma^2(t_0)$ were calculated over periods corresponding to the rest state preceding each trial. In order to rectify outliers, the EEG amplitudes of each trial were first limited to $\mu(t_i) \pm 6\sigma(t_i)$. Finally, the trial crop strategy for data augmentation was employed. For both datasets, crops of 4 s in length each 8 ms from $-0.5$ to 4 s (cue onset at 0 s) over all trials were extracted in our study. Figure 1 shows the EEG preprocessing and data augmentation.

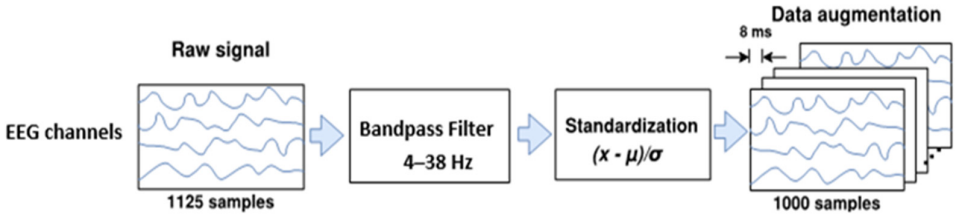


**Figure 1.** The raw EEG preprocessing and data augmentation.

### 2.3. Baseline Architecture

A shallow architecture based on a Convolutional Neural Network (SCNN) [57] was used in our work as a baseline deterministic network, as shown in Figure 2. This architecture contains two convolutional layers and a dense classification layer. The first convolution has an input tensor of $1000 \times c \times 1$, where $c$ is the number of EEG channels. The first convolution applies a temporal convolution with a $45 \times 1$ filter and 40 channels, giving an output tensor of $478 \times c \times 40$ after performing a down-sampling with a stride of 2. The second convolutional layer realizes a spatial convolution composed of 40 channels and a $1 \times c$ filter, which performs a convolution amongst all EEG channels. Following the spatial convolution, the model executes a sequence of transformations as follows: a batch normalization layer, an activation function with square non-linearity, an average pooling layer with $45 \times 1$ sliding windows, and a max-pooling layer with $8 \times 1$ pool size, $8 \times 1$ stride, and a logarithmic activation function. Lastly, the classification layer composed of a dense layer using the Softmax activation function receives 2160 features, which are translated into a $s \times 1$ prediction vector, $s$ being the number of classes.

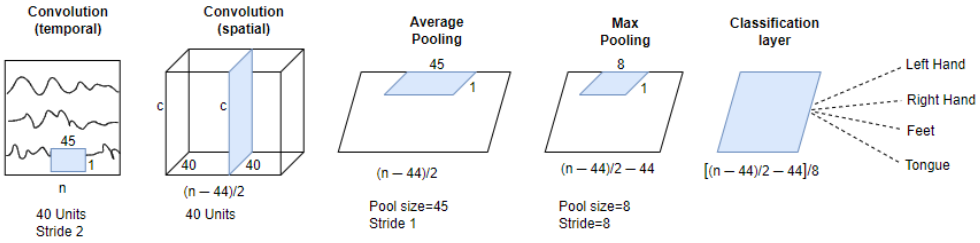


**Figure 2.** Shallow Convolutional Neural Network as baseline architecture.

Although dropout layers and the "MaxNorm" regularization have been previously used in the baseline architecture [57] to avoid overfitting, we did not use them in BNN models that are already more robust and less prone to overfitting. Furthermore, the "Early Stopping" and the decay of learning rate techniques were used. Additionally, the Adam optimizer and the Categorical Cross-Entropy as a cost function were employed.

### *2.4. Bayesian Neural Networks*

BNNs [58] provide a probabilistic interpretation of deep learning models by placing distributions over their weights. In BNNs, the weights are no longer considered as fixed values, but rather as random variables sampled according to a distribution whose parameters are learned in the training stage. This makes each prediction different for the same input, and for many forward passes, the average behavior produces relevant results. Notably, the variability of these predictions also allows assessment of the model's confidence.

Let $D = \{X,\ Y\}$ be a training set with inputs $X = \{\mathbf{x}_1,\ \cdots,\ \mathbf{x_n}\}$ and expected outputs or targets $Y = \{\mathbf{y}_1,\ \cdots,\ \mathbf{y_n}\}$, where $\mathbf{x_n} \in \mathbb{R}^d$ and $\mathbf{y_i} \in \mathbb{R}^C$, $C$ being the number of classes. The Bayesian modeling aims to approximate the posterior distribution of weights $p(\mathbf{w}|D)$ that generated the output vector $Y$. Following this approach, the prior distribution $p(\mathbf{w})$ that represents the initial beliefs about the weights has to be established.

If $p(\mathbf{w}|D)$ is known, then for a given input $\hat{\mathbf{x}}$, the predictive distribution $p\left(\hat{\mathbf{y}} \middle| \hat{\mathbf{x}},\ D\right)$. can be determined from the outputs $p\left(\hat{\mathbf{y}} \middle| \hat{\mathbf{x}},\ \mathbf{w}\right)$. corresponding to a specific set of weights $\mathbf{w}$ as follows:

$$p(\hat{\mathbf{y}} \mid \hat{\mathbf{x}},\ D) = \int p(\hat{\mathbf{y}}|\ \hat{\mathbf{x}},\ \mathbf{w})\ p(\mathbf{w}|D)d\mathbf{w} \tag{1}$$

The integral in (1) can be estimated by using the Monte Carlo method. For this, it is necessary to perform $T$ evaluations of the neural network on the same input $\hat{\mathbf{x}}$ and weights $\mathbf{w}_t$ sampled from the posterior distribution $p(\mathbf{w}|D)$. As a result, instead of a single output, we obtain $T$ outputs from the model $\left\{\hat{\mathbf{y}}_t; 1 \le t \le T\right\}$. According to [21], $T = 50$ is a good balance in terms of complexity and accuracy. The set $\left\{\hat{\mathbf{y}}_t\right\}$ can be interpreted as a sample of the predictive distribution. The final prediction of BNN on the input $\hat{\mathbf{x}}$ is seen as the sample mean of the predictive distribution:

$$\mathbb{E}\left[p\left(\hat{\mathbf{y}}\right)\right] \approx \frac{1}{T}\sum_{t=1}^{T} p\left(\hat{\mathbf{x}},\ \mathbf{w}_t\right) = \frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{y}}_t; \qquad \mathbf{w}_t \sim p(\mathbf{w}|D) \tag{2}$$

In neural networks, the uncertainty measures how reliable a model is when making predictions. The statistical dispersion of the predictive distribution $p\left(\hat{\mathbf{y}} \middle| \hat{\mathbf{x}},\ D\right)$. is one indicator of uncertainty. A "large" dispersion in $p\left(\hat{\mathbf{y}} \middle| \hat{\mathbf{x}},\ D\right)$. means that the neural network produces less confident results. In contrast, a "small" dispersion indicates that the network always provides similar results independently of the sampled weights $\mathbf{w}_t$ .

To summarize, if the posterior distribution is known, it is possible to obtain, for each input $\hat{\mathbf{x}}$, an approximation of the predictive distribution $p\left(\hat{\mathbf{y}} \middle| \hat{\mathbf{x}}, D\right)$. This then allows us to estimate the BNN prediction and an uncertainty measure. However, finding and sampling the posterior distribution for complex models, such as neural networks, is a computationally intractable problem because of their high dimensionality and non-convexity. To address this issue, two popular approaches have been introduced previously in other studies, such as (1) variational inference [33] and (2) Monte Carlo dropout [21]. The next section only describes the former method, which was used in our approach.

### 2.5. Variational Inference

In variational inference, rather than sampling from the exact posterior distribution $p(\mathbf{w}|D)$, a variational distribution $q(\mathbf{w}, \theta)$ is used, parametrized by a vector $\theta$. The $\theta$ values are then learned in such a way that $q(\mathbf{w}, \theta)$ is close as possible to $p(\mathbf{w}|D)$ in a Kullback–Leibler (KL) divergence sense. It is known that minimizing the KL divergence between $q(\mathbf{w}, \theta)$ and $p(\mathbf{w}|D)$ is equivalent to maximizing the evidence lower bound (ELBO) function, denoted as $\mathcal{L}(\theta)$, which serves as the objective function to train the model.

$$\mathcal{L}(\theta) = \int q(\mathbf{w}, \theta) \ln p(Y|X, \mathbf{w}) d\mathbf{w} - D_{KL}(q(\mathbf{w}, \theta), p(\mathbf{w})) \quad (3)$$

Maximizing the first term in (3) encourages $q(\mathbf{w}, \theta)$ to explain the data well, while minimizing the second term encourages $q(\mathbf{w}, \theta)$ to be close to $p(\mathbf{w})$.

Although in general guessing $q(\mathbf{w}, \theta)$ is complex, to adopt an independent Gaussian distribution $w \sim \mathcal{N}(\mu, \sigma^2)$ for each weight is a simple and common approach that often works. Additionally, with respect to the prior distribution, a standard normal distribution $w \sim \mathcal{N}(0, 1)$ is commonly used. In contrast, the MOPED method [38] proposes to specify prior distributions $p(w)$ from deterministic weights $(w_d)$ derived from a pretrained DNN with the same architecture.

### 2.6. Bayesian Neural Models for Uncertainty Estimation and MI Classification

In this work, two BNN architectures (SCBNN and SCBNN-MOPED) derived from the SCNN baseline are proposed. They only differ in the way the prior distribution of the weights is established. The former initializes the prior distribution of weights as a standard Gaussian distribution $\mathcal{N}(0, 1)$, whereas the second uses the MOPED method. For SCBNN-MOPED, the prior distribution of each weight $w$ is initialized as an independent Gaussian distribution with the mean $\mu$ taken from the pretrained weights $w_d$ of the SCNN architecture and a scale $\sigma = 0.1|w_d|$.

Both models were obtained by starting from a deterministic network SCNN and translating it onto a Bayesian network by using the Flipout estimator [59] from TensorFlow. We preferred the Flipout estimator instead of using reparameterization [60] because it offers a significant low variance, albeit it uses roughly twice that of floating point operations. Table 1 shows the main differences between SCNN and the two proposed Bayesian neural architectures, following the terminology used in the "TensorFlow probability" library of Python for better comparison [33].

**Table 1.** Comparison between SCNN and the two proposed Bayesian neural architectures.

| Layer | SCNN | SCBNN and SCBNN-MOPED |
|---|---|---|
| Temporal Convolution | Conv2D | Conv2DFlipout |
| Spatial Convolution | Conv2D | Conv2DFlipout |
| Dropout Layer | Yes | No |
| Classification Layer | Dense | DenseFlipout |

### *2.7. Uncertainty Measures*

Several uncertainty measures can be used to quantify the model uncertainty [33,39,40,61]. For better understanding, three well-known metrics—predictive entropy ($\mathbb{H}$), mutual information ($\mathbb{I}$), and margin of confidence ($\mathbb{M}$)—were first presented as follows.

Let $C$ be the total number of classes and $\hat{\mathbf{y}}_t = (\hat{y}_{1t}, \hat{y}_{2t}, \cdots, \hat{y}_{Ct})$ the model output for a given input $\hat{\mathbf{x}}$ in a stochastic forward pass $t$; $1 \leq t \leq T$. Let $\hat{\mathbf{y}}^* = (\hat{y}_1^*, \hat{y}_2^*, \cdots, \hat{y}_C^*)$ be the average of predictions $\left\{\hat{\mathbf{y}}_t; 1 \leq t \leq T\right\}$.

**Predictive Entropy** $\mathbb{H}$. This metric captures through Equation (4) the average on the amount of information contained in the predictive distribution, reaching its maximum value $(log_2 C)$ when all classes have equal uniform probability (maximum uncertainty, in other words). In contrast, it reaches zero value (the minimum) when a unique class has probability equal to 1, confirming a certain prediction.

$$\mathbb{H} \approx -\sum_{j=1}^{C} \hat{y}_j^* \log_2(\hat{y}_j^*) \tag{4}$$

To facilitate the comparison across various datasets, the predictive entropy was normalized as follows: $\mathbb{H}_n = \mathbb{H}/log_2 C$, $\mathbb{H}_n \in [0, 1]$.

**Mutual Information** $\mathbb{I}$. This measures the epistemic uncertainty by capturing the model's confidence from its output.

$$\mathbb{I} \approx \mathbb{H} - \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{C} \hat{y}_{jt} log_2(\hat{y}_{jt}) \tag{5}$$

**Margin of Confidence** $\mathbb{M}$. Let $c = argmax\ \hat{y}_j^*$ be the predicted class. The most intuitive form of measuring uncertainty is through analyzing the difference between two predictions of highest confidence. For this, in each forward pass $t$, the difference $d_t$ between the output $\hat{y}_{ct}$ (predicted class) and the other output of highest score value $\underset{j \neq c}{max} \hat{y}_{jt}$ is calculated. These differences are then averaged as follows:

$$\mathbb{M} = \frac{1}{T}\sum_{t=1}^{T} d_t; \qquad d_t = \hat{y}_{ct} - \underset{j \neq c}{max} \hat{y}_{jt} \tag{6}$$

### *2.8. Classification with a Reject Option*

The uncertainty values, calculated by using any of the aforementioned measures, provide to classifiers the ability to accept or reject inputs. For instance, a high uncertainty means that the classifier may be performing random predictions; therefore, the associated input should be rejected. This type of classification is known as classification with a reject option [36,62], which is of great importance for applications that require low error tolerance. Classification with a reject option generalizes the decision problem of class predictions, and also determines whether the prediction is reliable (accept) or unreliable (reject).

The process of refraining from providing a response or discarding an input when the system does not have enough confidence in its prediction is a topic of interest that has been addressed over the last 60 years [63]. Recent methods [39–41] have implemented the classification of rejection from an established threshold $Th$ by using any uncertainty metric, rejecting inputs that present an uncertainty value above $Th$.

The criteria used in [39] incorporate the ground-truth label, model prediction, and uncertainty value to evaluate the selected threshold. In this approach, the map of *correct* and *incorrect* values is computed by matching the ground truth labels with the model's predictions. Then, given a threshold $Th$, the predictions are classified as *certain* and *uncertain*,

providing four combinations to predict an input as follows: incorrect–uncertain (iu), correct–uncertain (cu), correct–certain (cc), and incorrect–certain (ic).

Let $N_{iu}$, $N_{cu}$, $N_{cc}$ and $N_{ic}$ be the total number of each type of predictions in each subset, where $N$ is the total number of predictions, and $R_c$ is the proportion of certain predictions with respect to the total number of predictions:

$$R_c(Th) = \frac{P(\text{certain})}{N} = \frac{N_{cc} + N_{ic}}{N} \tag{7}$$

Then, the following criteria measure the effectiveness of an uncertainty estimator and a threshold selector [39]:

The correct–certain ratio $R_{cc}$ measures the ability of a classifier to accurately classify non-rejected samples:

$$R_{cc}(Th) = \frac{P(\text{correct} \cap \text{certain})}{P(\text{certain})} = \frac{N_{cc}}{N_{cc} + N_{ic}} \tag{8}$$

The overall accuracy of the uncertainty estimation can be measured through the Uncertainty Accuracy $(UA)$, which indicates the ability of a classifier to accurately classify non-rejected samples, and reject misclassified samples:

$$UA(Th) = \frac{N_{cc} + N_{iu}}{N} = 1 - \frac{N_{ic} + N_{cu}}{N} \tag{9}$$

$UA(Th)$ penalizes both the incorrect–certain and correct–uncertain predictions, and its highest values suggest better thresholds. Thus, this criterion can be also used to compare between different threshold values, in order to increase the reliability, effectivity, and feasibility for practical applications.

In this study, the correct–uncertain ratio $R_{cu}$ was proposed to calculate the accuracy of rejected predictions, as shown in Equation (10). This criterion was incorporated with the objective of evaluating how close the accuracy on rejected predictions was with respect to the hypothetical accuracy that would be obtained if the classifier provided random predictions.

$$R_{cu}(Th) = \frac{P(\text{correct} \cap \text{uncertain})}{P(\text{uncertain})} = \frac{N_{cu}}{N_{cu} + N_{iu}} \tag{10}$$

### *2.9. Adaptive Uncertainty Threshold*

To select the cut-off threshold, recent works [39–41] propose to evaluate all threshold values for a given uncertainty measure, such as predictive entropy. For this purpose, this study measured the performance by using each threshold, taking into account a criterion of quality as $UA$, consequently keeping the best threshold. This strategy has high computational cost because of its expensive and exhaustive search. As a contribution, we introduced a low-computational-cost novel threshold scheme based on a closed formula, rather than an exhaustive search.

Our threshold scheme uses Equation (6) to compute the margin of confidence $\mathbb{M}$, which takes values close to zero when the prediction has high uncertainty. It is worth mentioning that if $\mathbb{M}$ is not significantly higher than zero, there will be no significant difference between the two highest confidence predictions of the model. Therefore, we reduced the labeling problem on the prediction of an input $\hat{\mathbf{x}}$ as uncertain to a hypothesis test for the mean $\mathbb{E}\left[D\left(\hat{\mathbf{x}}\right)\right]$, where $D\left(\hat{\mathbf{x}}\right)$ denotes all possible differences $\{d_t = \hat{y}_{ct} - \underset{j \neq c}{max}\hat{y}_{jt}\}$ that could be obtained from the forward passes associated with the input $\hat{\mathbf{x}}$. As a result, if the prediction

is certain, the null hypothesis $H_0$: $\mathbb{E}\left[D\left(\hat{\mathbf{x}}\right)\right] \leq 0$ must be rejected, whereas the alternative hypothesis $H_1$: $\mathbb{E}\left[D\left(\hat{\mathbf{x}}\right)\right] > 0$ must be accepted. The test statistic is:

$$\zeta = \frac{\mathbb{M}}{\sigma_d / \sqrt{T}} \tag{11}$$

where $\sigma_d$ is the standard deviation of the sample $\{d_t\}_{t=1}^{T}$, and $T$ is the number of forward passes over each input $\hat{\mathbf{x}}$. It is worth remarking that the rejection region for the null hypothesis is the right tail of the distribution.

$\mathbb{M}$ is a sample mean of $T = 50$ random values $\{d_t\}_{t=1}^{T}$, making $\mathbb{M}$ follow a normal distribution with variance $\sigma_d^2/T$. Therefore, we assume that the random variable $\zeta$ follows a standard normal distribution. The quantile $z_{1-\alpha} = \Phi^{-1}(1-\alpha)$ is used as the critical value of the test, where $\Phi$ is the cumulative distribution function (CDF) of the standard normal distribution, and $1-\alpha$ is the confidence level. The null hypothesis is rejected for $\zeta > z_{1-\alpha}$. Thus, if the prediction is certain, $\mathbb{M}$ must be greater than $\sigma_d z_{1-\alpha}/\sqrt{T}$. This way, the proposed adaptive threshold can be calculated as follows:

$$T_{\mathbb{M}}\left(\hat{\mathbf{x}}\right) = \frac{\sigma_d z_{1-\alpha}}{\sqrt{T}} \tag{12}$$

As a highlight, our adaptive threshold scheme highlights four observations. First, the threshold scheme is statistically based on a closed formula. Second, the threshold is not fixed; it varies depending on the sample variance $\sigma_d^2$ of $\{d_t\}_{t=1}^{T}$. In other words, this novel threshold is adaptive to the predictive distribution characteristics of each input $\hat{\mathbf{x}}$. Third, our scheme is conservative, as it does not classify as uncertain those inputs in which the predictions for each forward pass are consistent ($\sigma_d \approx 0$). However, increasing the confidence level $1-\alpha$, the threshold scheme behaves less conservatively. Fourth, our threshold scheme can be considered as universal, as it does not require prior knowledge about the data, depending exclusively on the predictive distribution.

### *2.10. Experiments*

The proposed methods were evaluated with three experiments, using subject-specific and non-subject-specific classification strategies.

The first experiment analyzed the accuracy of the Bayesian Neural Network models SCBNN and SCBNN-MOPED for MI classification, compared with their deterministic counterpart SCNN [57].

The second experiment assessed the quality using the proposed adaptive threshold scheme within the SCBNN-MOPED model. The quality was measured by looking into accepted and rejected partitions of the testing set, measuring how well each partition could be classified. Ideally, the rejected partition should be hard to classify, with significantly better accuracy than random guessing.

The last experiment evaluated the proposed threshold scheme with respect to [39] as a benchmark, as it seeks the optimum threshold via exhaustive search. The aim was to demonstrate that our proposed adaptive threshold may achieve the state-of-the-art performance without brute force.

All experiments were designed to meet the requirements for the BCI Competition IV dataset, never using the testing sets during training or validation. Figure 3 shows the difference between both subject-specific and non-subject-specific strategies, respectively, where the former restricts itself to the training data of each subject (see Figure 3a), whereas the latter calibrates the classifier by using the training data from all subjects (see Figure 3b).

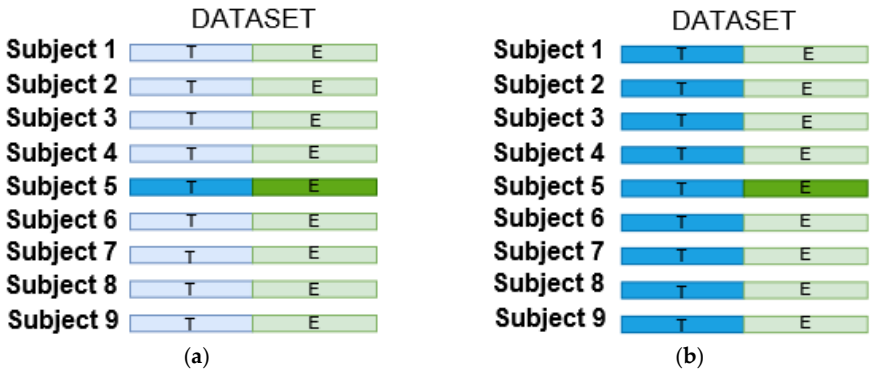


**Figure 3.** The training ( $\mathcal{T}$ in blue) and testing ( $\mathcal{E}$ in green) set selections for both strategies: (**a**) subject-specific; (**b**) non-subject-specific.

In all experiments, a repeated holdout validation over the same testing set was carried out, considering both subject-specific and non-subject-specific classification strategies. For each repetition, a subset $\mathcal{V}_i$ of the training set $\mathcal{T}$ was randomly picked, guaranteeing identical class ratios in $\mathcal{V}_i$ with respect to the class ratios in $\mathcal{T}$. The subset $\mathcal{V}_i$ was used for validation purposes, which is necessary to build deep learning models. For training purposes, the remaining set $\mathcal{T}_i = \mathcal{T} \setminus \mathcal{V}_i$ was used. Once the model was trained for each pair $\mathcal{T}_i$ and $\mathcal{V}_i$, it was evaluated on the testing set $\mathcal{E}$. To reduce the influence of random factors, this process was repeated 16 times for each model, reporting the average of some metrics, such as accuracy, $R_{cc}$, among others.

Specifically, the accuracy was used for evaluation in the first experiment, whereas the criteria $R_c$, $R_{cc}$ and $R_{cu}$ (see Equations (7), (8) and (10)) were used in the second experiment, assessing the quality of accepted and rejected partitions on the testing set. As in [39], we used $UA$ (Equation (9)) as an evaluation criterion in the third experiment.

For the third experiment, the method in [39] was replicated by using the SCBNN-MOPED architecture and the normalized predictive entropy $\mathbb{H}_n$ as the uncertainty measure. As in [39–41], the optimum threshold was obtained by making the uncertainty threshold $Th$ vary over the interval [0.05, 1] and calculating $UA$ over the validation set $\mathcal{V}_i$ for each $Th$. Once the optimal threshold was found, its performance was evaluated with respect to the criterion $UA$ by using the testing set $\mathcal{E}$. Similar to the other experiments, 16 validation sets ($\mathcal{V}_1 \cdots, \mathcal{V}_{16}$) were used to compute the overall assessment of the method in [39].

An HPC using a Dell PowerEdge R720 server with four 2.40 GHz Intel Xeon processors and 48 GB was used for all experiments. An NVIDIA GM107L Tesla M10 GPU with 32 GB memory was used to train and test the Bayesian models by using the Python TensorFlow 2.3.0 framework.

## 3. Results and Discussion

### *3.1. MI Classification Using SCBNN and SCBNN-MOPED Models*

Tables 2–5 show the results for subject-specific and non-subject specific strategies by using the datasets 2a and 2b, respectively, where a confidence level $\mathbf{1} - \boldsymbol{\alpha}$ at 0.95 by using the Wilcoxon signed rank test was considered.

**Table 2.** Subject-specific classification results (% accuracy) on dataset 2a.

| Model | A01 | A02 | A03 | A04 | A05 | A06 | A07 | A08 | A09 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|
| **SCNN** [57] | **83.81** | **51.97** | 91.48 | 73.82 | 69.82 | 53.90 | **91.17** * | **81.87** * | **82.39** * | **75.58** |
| **SCBNN** | 62.95 | 27.48 | 78.90 | 41.34 | 35.75 | 36.95 | 56.12 | 66.27 | 68.92 | 52.74 |
| **SCBNN-MOPED** | 83.64 † | 51.72 † | **91.50** † | **74.53** † | **70.84** † | **55.28** † | 89.42 † | 80.63 † | 81.12 † | 75.41 † |

(*) indicates significant differences with 95% confidence level; (†) indicates significant differences between the SBBNN and SCBNN-MOPED.

**Table 3.** Subject-specific classification results (% accuracy) on dataset 2b.

| Model | B01 | B02 | B03 | B804 | B05 | B06 | B07 | B08 | B09 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|
| **SCNN** [57] | 75.43 | **55.36** | 52.09 | 94.96 | **87.60** | 79.71 | **79.77** | 87.87 | **84.69** * | 77.50 |
| **SCBNN** | 69.92 | 50.46 | 51.53 | 94.56 | 84.23 | 78.87 | 77.76 | 86.10 | 79.13 | 74.73 |
| **SCBNN-MOPED** | **75.97** † | 54.41 † | **52.36** | **95.44** * | 87.38 † | **80.38** † | 79.37 † | **88.52** † | 83.95 † | **77.53** † |

(*) indicates significant differences with 95% confidence level; (†) indicates significant differences between the SBBNN and SCBNN-MOPED.

**Table 4.** Non-subject-specific classification results (% accuracy) on dataset 2a.

| Model | A01 | A02 | A03 | A04 | A05 | A06 | A07 | A08 | A09 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|
| **SCNN** [57] | 72.29 | 39.26 | 81.59 | 60.90 | **54.03** * | 51.58 | **74.70** * | **77.11** | **76.86** | 65.37 |
| **SCBNN** | 60.14 | 29.19 | 63.85 | 40.87 | 32.98 | 38.57 | 40.61 | 62.42 | 58.89 | 47.50 |
| **SCBNN-MOPED** | **82.44** * | **47.42** * | **83.25** * | **62.31** * | 49.73 † | **52.79** † | 70.53 † | 76.67 † | 75.74 † | **66.77** * |

(*) indicates significant differences with 95% confidence level; (†) indicates significant differences between the SBBNN and SCBNN-MOPED.

**Table 5.** Non-subject-specific classification results (% accuracy) on dataset 2b.

| Model | B01 | B02 | B03 | B04 | B05 | B06 | B07 | B08 | B09 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|
| **SCNN** [57] | 66.94 | 55.66 | **54.08** * | 93.78 | **79.09** | 80.81 | **74.18** * | 90.05 | **83.42** | 75.33 |
| **SCBNN** | 69.53 | 56.07 | 52.69 | 92.81 | 74.98 | 78.60 | 72.75 | 89.48 | 80.58 | 74.17 |
| **SCBNN-MOPED** | **73.32** * | **57.22** * | 53.10 | **94.08** * | 78.57 † | **81.52** * | 73.20 | **90.38** | 83.15 † | **76.06** * |

(*) indicates significant differences with 95% confidence level; (†) indicates significant differences between the SBBNN and SCBNN-MOPED.

For both datasets, the SCBNN-MOPED architecture achieved significantly better results than the SCBNN for almost all subjects. For dataset 2a, the accuracy improvement was greater than 22% and 19% (on average) in the subject-specific and non-subject-specific strategy, respectively. In general, our results were consistent with the findings in [38], showing that the MOPED method helped the training convergence and improved the accuracy of Bayesian neural models.

The SCBNN-MOPED model also achieved better mean accuracy than the deterministic SCNN model, except for the subject-specific classification in dataset 2a. Tables 4 and 5 show that, for non-subject-specific classification, SCBNN-MOPED significantly outperformed the SCNN model with respect to overall accuracy, and for almost all subjects.

Because the SCBNN-MOPED architecture achieved superior performance, the next experiments were carried out discarding the SCBNN architecture.

### *3.2. Classification with Reject Option Using SCBNN-MOPED Architecture*

Figures 4 and 5 show the performance by applying the proposed adaptive threshold for MI classification with a reject option, considering both subject-specific and non-subject-specific classifications, respectively.

Figures 4 and 5 show that a larger number of predictions were rejected in dataset 2a independently of the training strategies. As expected from an adaptive and effective threshold, the accuracy achieved on the accepted predictions was higher than considering the full testing set without rejection. This was notable, using $R_{cc}$ as criterion for comparison with the results shown in Tables 2–5. This supports that by rejecting uncertain predictions, the classification with a reject option can improve both accuracy and reliability, which is of great importance in systems for real-life applications.

The $\boldsymbol{R_{cu}}$ criterion is another way to measure the quality of our proposed threshold scheme, looking into the accuracy obtained over the rejected predictions. For dataset 2b and both training strategies, $\boldsymbol{R_{cu}}$ of 50% was achieved, showing that the classification accuracy over rejected predictions and random guessing was not different, confirming the

optimal performance using the proposed threshold scheme. For dataset 2a, $R_{cu}$ of 40% was obtained, while a random classifier would obtain 25%.

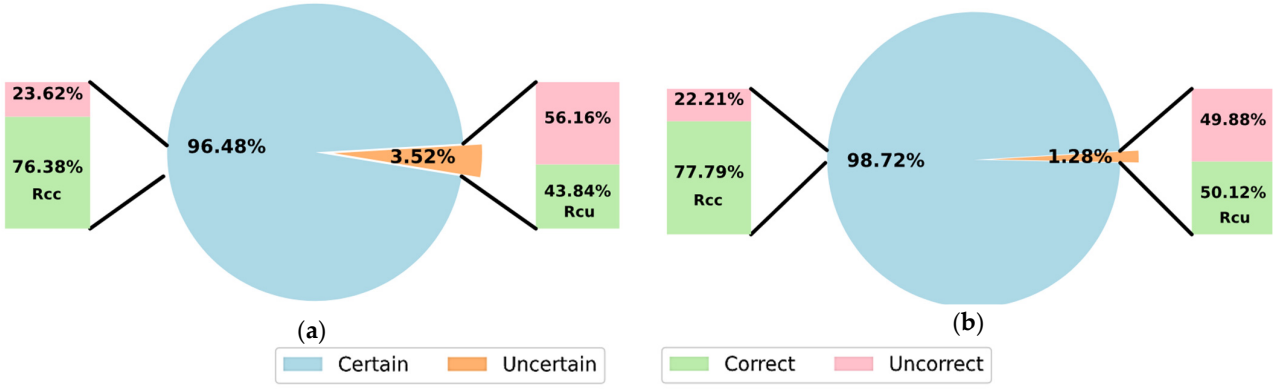


**Figure 4.** Results using SCNN-MOPED with the adaptive threshold scheme for MI classification with reject option, considering the subject-specific strategy, (**a**) dataset 2a; (**b**) dataset 2b.

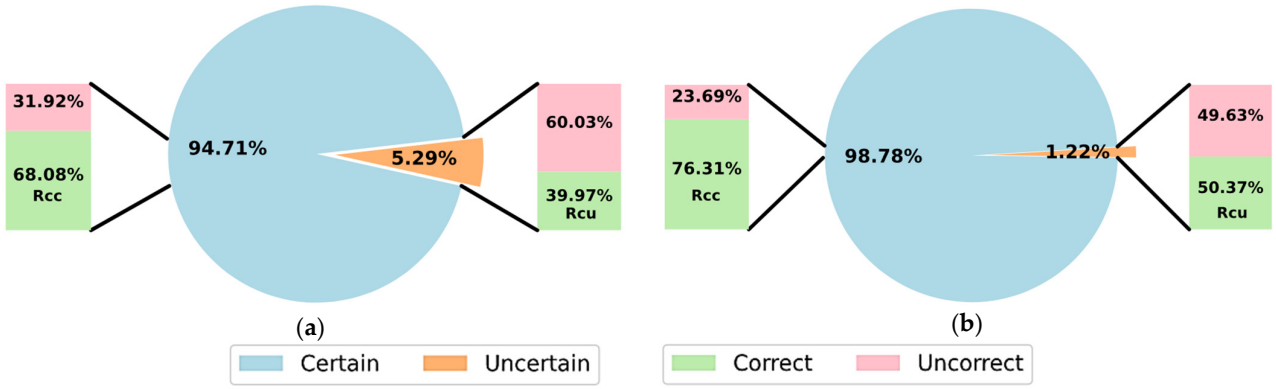


**Figure 5.** Results using SCNN-MOPED with the adaptive threshold scheme for MI classification with reject option, considering the non-subject-specific strategy, (**a**) dataset 2a; (**b**) dataset 2b.

It is worth noting that our approach based on an adaptive threshold is conservative, achieving significant low percentage of rejected predictions in comparison to the incorrect predictions. For instance, 33.23% of inputs were incorrectly classified in dataset 2a with a non-subject-specific strategy, and only 1.22% was associated as uncertain by our adaptive threshold. However, by increasing the confidence level $\mathbf{1} - \boldsymbol{\alpha}$, the number of rejected predictions can be increased. This strategy has limitations, though, as our proposed threshold scheme only considers epistemic uncertainty, and it may also overlook other sources of uncertainty.

### *3.3. Comparison with Other Methods Based on Thresholds*

Tables 6 and 7 show the $\boldsymbol{UA}$ values obtained by our proposed adaptive threshold ($\boldsymbol{UA}(\boldsymbol{T}_{\mathbb{M}})$), as well as the maximum or optimal values reached by [39] via exhaustive search and normalized predictive entropy ($\boldsymbol{U}, \boldsymbol{A}, (\boldsymbol{T}_{\mathbb{H}})$).

Recall that the $\boldsymbol{UA}$ criterion makes it possible to evaluate different cut-off thresholds. By inspecting the values in Tables 6 and 7, we conclude that our proposed threshold scheme allows us to achieve state-of-the-art performance. Moreover, by relying exclusively on the predictive statistical distribution, our scheme becomes universal and independent of the nature of data.

**Table 6.** Comparison between the proposed adaptive threshold and the method from [39], using subject-specific classification.

| Subject | Dataset 2a | | Dataset 2b | |
|---|---|---|---|---|
| | $UA(T_{\mathbb{M}})$ | $UA(T_{\mathbb{H}})$ | $UA(T_{\mathbb{M}})$ | $UA(T_{\mathbb{H}})$ |
| 01 | **83.58** | 83.27 | **75.95** | 75.70 |
| 02 | 53.62 | **55.41** * | **54.37** * | 53.17 |
| 03 | **91.50** | 90.95 | **52.34** * | 51.04 |
| 04 | **75.41** | 75.08 | **95.44** | 95.11 |
| 05 | **71.99** | 71.63 | **87.39** | 86.66 |
| 06 | 56.75 | **60.21** * | **80.39** | 80.16 |
| 07 | **89.41** | 88.03 | **79.38** | 79.16 |
| 08 | **80.92** | 80.66 | **88.52** | 87.89 |
| 09 | 81.28 | **81.94** | **83.91** * | 82.23 |
| **Mean** | 76.05 | **76.35** | **77.52** * | 76.79 |

(*) indicates significant differences with 95% confidence level.

**Table 7.** Comparison between the proposed adaptive threshold and the method from [39], using non-subject-specific classification.

| Subject | Dataset 2a | | Dataset 2b | |
|---|---|---|---|---|
| | $UA(T_{\mathbb{M}})$ | $UA(T_{\mathbb{H}})$ | $UA(T_{\mathbb{M}})$ | $UA(T_{\mathbb{H}})$ |
| 01 | **82.46** | 82.07 | 73.30 | **73.32** |
| 02 | 51.50 | **54.01** * | 57.18 | **57.23** |
| 03 | **83.62** | 83.54 | 53.08 | **53.10** |
| 04 | 63.85 | **65.46** * | 94.08 | **94.09** |
| 05 | 51.97 | **55.89** * | 78.58 | 78.58 |
| 06 | 54.23 | **56.39** * | 81.49 | **81.54** |
| 07 | **71.11** * | 69.55 | **73.21** | 73.20 |
| 08 | 77.36 | **77.68** | 90.38 | 90.38 |
| 09 | 76.25 | **76.60** | 83.14 | **83.16** |
| **Mean** | 68.04 | **69.02** * | 76.05 | 76.07 |

(*) indicates significant differences with 95% confidence level.

## 4. Considerations for Reducing Computational Cost

All evaluations considering the classification accuracy and uncertainty analysis throughout this current work were carried out by using data augmentation, as shown in Figure 1. The accuracy was obtained as a ratio between correct classified crops and the total crops, as proposed in the CAgross method [19]. Others studies followed different strategies. For instance, a single prediction for each trial was obtained in the CAST method [19] by using majority voting, whereas in [45] the authors averaged the predictions over all crops. This analysis of the crops requires an extra computational cost, which increases even more with the use of Bayesian architectures, as it realizes $T$ forward passes to obtain the prediction over the same input.

Given the nature of the problem that is addressed in our study, the EEG variability of patterns associated with MI tasks, and the large intra- and inter-subject differences can be observed through the ERD/ERS potentials throughout. As a result, in certain crops, the aforementioned phenomena were not detected appropriately by the model, producing incorrect predictions. Figure 6a shows the predictive distribution histograms of crops over a specific trial on Subject A01 from dataset 2a, where the initial crops were classified correctly as left-hand MI, but after approximately crop number 45, they were incorrectly predicted as right-hand MI. Figure 6b shows that this change correlated with the increase of uncertainty. Observing this behavior, we believe that, when considering the predictions over all crops predicting the MI of each trial, it is not the most appropriate strategy, since all crops do not have the same certainty. Therefore, it is preferable to consider the crops with less uncertainty.

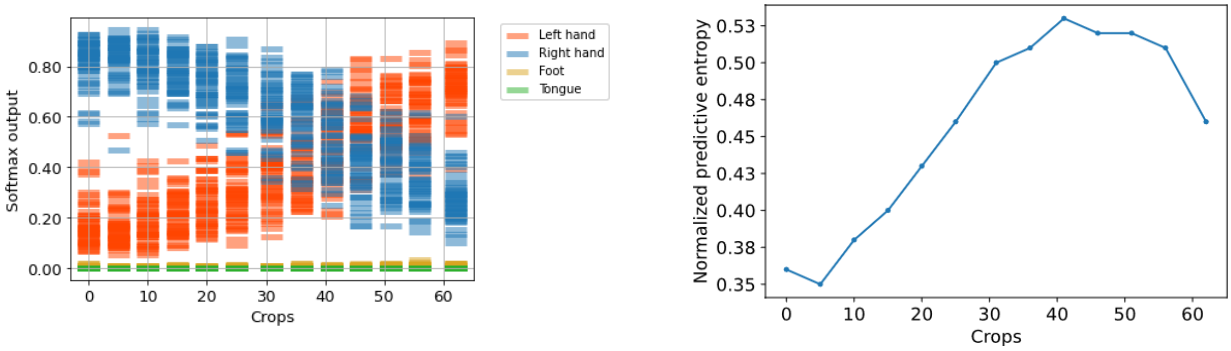


**Figure 6.** (**a**) Predictive distribution histograms of crops; (**b**) normalized predictive entropy of crops.

We hypothesize that if the predictions with less uncertainty are determined over the same time interval for all trials on the validation set, similar behavior may be also obtained in the testing set. This allows us to choose only crops contained in that interval for classification with/without a reject option, reducing the computational cost with high accuracy.

Figure 7 shows the normalized predictive entropy on a validation set from database 2a. We observed that the central crops covering a time interval from 1.724 to 5.756 s regularly presented the lowest uncertainty with respect to the crops located at the extremes. It is worth mentioning that this finding was also observed in dataset 2b.

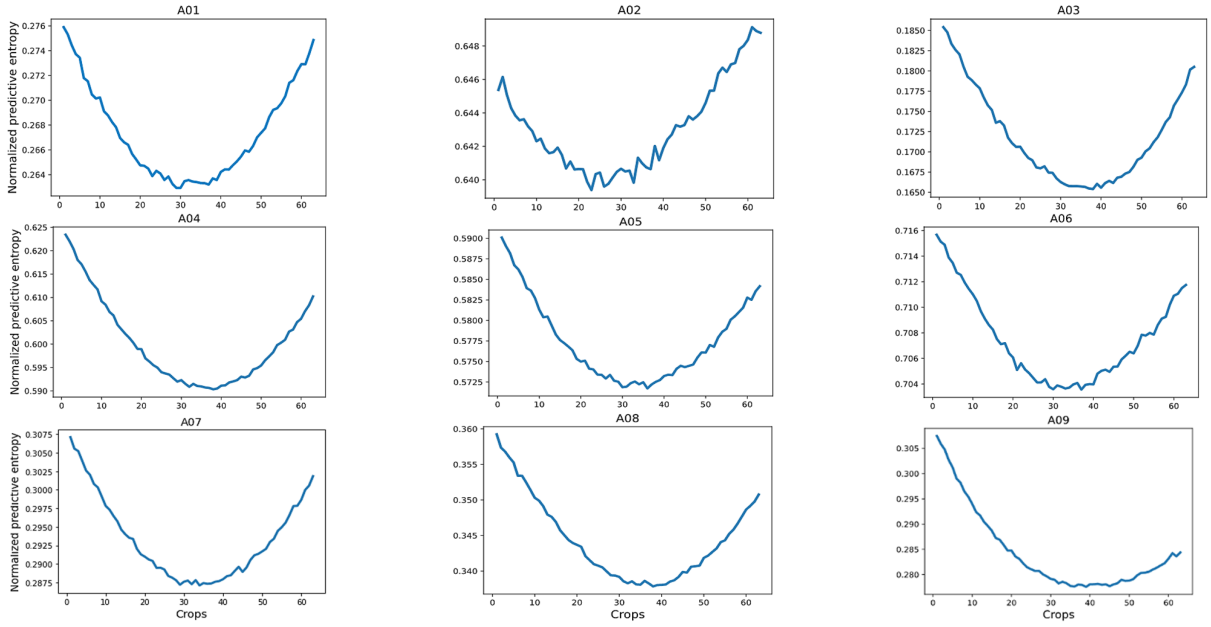


**Figure 7.** Average of the normalized predictive entropy for each crop on subjects in dataset 2a.

From our findings, we selected the five central crops of each trial, aiming to reduce computational cost during the evaluation step. Then, we averaged the predictions of these five central crops to rank each trial. Table 8 shows the classification accuracy without/with a reject option achieved on the selected crops of each trial (from the testing set), using SCBNN-MOPED architecture and subject-specific strategy with each database. For a better assessment of the established criterion, we also added the achieved results from Tables 2 and 3, using all the crops.

**Table 8.** Performance of SCBNN-MOPED (% accuracy) by using the selected crops for subject-specific classification.

| Subject | 2a | | | 2b | | |
|---|---|---|---|---|---|---|
| | All Crops | Five Central Crops | | All Crops | Five Central Crops | |
| | | Without Rejection | With Rejection | | Without Rejection | With Rejection |
| 01 | 83.64 | 84.27 | 84.42 | 75.97 | 76.07 | 76.23 |
| 02 | 51.72 | 52.50 | 53.08 | 54.41 | 54.96 | 54.91 |
| 03 | 91.50 | 91.95 | 92.07 | 52.36 | 52.32 | 52.28 |
| 04 | 74.53 | 75.91 | 76.61 | 95.44 | 95.59 | 95.60 |
| 05 | 70.84 | 71.31 | 72.28 | 87.38 | 87.15 | 87.31 |
| 06 | 55.28 | 56.36 | 56.87 | 80.38 | 80.96 | 81.14 |
| 07 | 89.42 | 90.04 | 90.32 | 79.37 | 79.69 | 79.80 |
| 08 | 80.63 | 81.49 | 81.73 | 88.52 | 88.73 | 88.90 |
| 09 | 81.12 | 82.25 | 82.51 | 83.95 | 84.32 | 84.55 |
| **Mean** | 75.41 | 76.23 | 76.65 | 77.53 | 77.75 | 77.85 |

The results show that the established criterion to reduce the computational cost by using the five central crops provides advantages during MI classification without/with a reject option. This proposal increases the possibilities of real implementations using Bayesian classifiers for MI tasks. However, for practical implementation, other criteria must be considered, since neural network models require large hardware resources. This analysis is outside the scope of this current work.

## 5. Conclusions

The advantage of using the MOPED method to improve MI classification by employing BNN has been verified in this work. The uncertainty quantification on predictions by using the margin of confidence provided valuable information, which enhanced the MI classification. In addition, the proposed adaptive threshold scheme allowed us to obtain a more robust, effective and adequate classifier. The proposed scheme, formulated in a closed equation, is based exclusively on the predictive statistical distribution, which makes it a universal method. Thus, it can be extended to other classification problems. Finally, a criterion to reduce the computational cost based on the uncertainty analysis was proposed, which increases the possibilities of practical implementations of Bayesian classifiers in MI-based BCIs.

**Author Contributions:** Conceptualization, methodology and data processing, D.M.-H., S.L.-C. and R.T.-C.; supervision, R.T.-C. and R.S.-Z.; original draft writing, D.M.-H. and R.T.-C.; draft revision and editing, D.D.-R., R.S.-Z., J.J.T.-P. and J.J.V.-M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Science Project "Development of upper limb exoskeletons for rehabilitation task" PN223LH004-026 of High Ministry of Science and Technology from Cuba Republic.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** This research was conducted with the support of the Supercomputer of the Universidad de Oriente, Santiago de Cuba, which is part of the Cuban Centre for Academic Supercomputing (HPC-Cuba), supported by the VLIR-UOS JOINT project and the Iberoamerican Supercomputing Network (RICAP). The authors also would like to thank the collaboration of Rolando Trujillo Rasua for useful comments and suggestions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References


1. Ang, K.K.; Guan, C. Brain-computer interface for neurorehabilitation of upper limb after stroke. *Proc. IEEE* **2015**, *103*, 944–953. [CrossRef]
2. Ang, K.K.; Guan, C. Brain-Computer Interface in Stroke Rehabilitation. *J. Comput. Sci. Eng.* **2013**, *7*, 139–146. [CrossRef]
3. Chaudhary, U.; Birbaumer, N.; Ramos-Murguialday, A. Brain-computer interfaces for communication and rehabilitation. *Nat. Rev. Neurol.* **2016**, *12*, 513–525. [CrossRef] [PubMed]
4. Kübler, A.; Furdea, A.; Halder, S.; Hammer, E.M.; Nijboer, F.; Kotchoubey, B. A brain-computer interface controlled auditory event-related potential (P300) spelling system for locked-in patients. *Ann. N. Y. Acad. Sci.* **2009**, *1157*, 90–100. [CrossRef] [PubMed]
5. Pires, G.; Nunes, U.; Castelo-Branco, M. Statistical spatial filtering for a P300-based BCI: Tests in able-bodied, and patients with cerebral palsy and amyotrophic lateral sclerosis. *J. Neurosci. Methods* **2011**, *195*, 270–281. [CrossRef]
6. Zhu, D.; Bieger, J.; Garcia-Molina, G.; Aarts, R.M. A survey of stimulation methods used in SSVEP-based BCIs. *Comput. Intell. Neurosci.* **2010**, *2010*, 702357. [CrossRef]
7. Lesenfants, D.; Habbal, D.; Lugo, Z.; Lebeau, M.; Horki, P.; Amico, E.; Pokorny, C.; Gómez, F.; Soddu, A.; Müller-Putz, G.; et al. An independent SSVEP-based brain-computer interface in locked-in syndrome. *J. Neural Eng.* **2014**, *11*, 035002. [CrossRef]
8. Ang, K.K.; Guan, C.; Phua, K.S.; Wang, C.; Zhou, L.; Tang, K.Y.; Joseph, G.J.E.; Kuah, C.W.K.; Chua, K.S.G. Brain-computer interface-based robotic end effector system for wrist and hand rehabilitation: Results of a three-armed randomized controlled trial for chronic stroke. *Front. Neuroeng.* **2014**, *7*, 00030. [CrossRef]
9. Wolpaw, J.R.; Birbaumer, N.; McFarland, D.J.; Pfurtscheller, G.; Vaughan, T.M. Brain-computer interfaces for communication and control. *Clin. Neurophysiol.* **2002**, *113*, 767–791. [CrossRef]
10. Pfurtscheller, G.; Guger, C.; Müller, G.; Krausz, G.; Neuper, C. Brain oscillations control hand orthosis in a tetraplegic. *Neurosci. Lett.* **2000**, *292*, 211–214. [CrossRef]
11. Craik, A.; He, Y.; Contreras-Vidal, J.L. Deep learning for electroencephalogram (EEG) classification tasks: A review. *J. Neural Eng.* **2019**, *16*, 031001. [CrossRef]
12. 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]
13. Liao, J.J.; Luo, J.J.; Yang, T.; So, R.Q.Y.; Chua, M.C.H. Effects of local and global spatial patterns in EEG motor-imagery classification using convolutional neural network. *Brain Comput. Interfaces* **2020**, *7*, 1–10. [CrossRef]
14. Amin, S.U.; Alsulaiman, M.; Muhammad, G.; Bencherif, M.A.; Hossain, M.S. Multilevel weighted feature fusion using convolutional neural networks for EEG motor imagery classification. *IEEE Access* **2019**, *7*, 18940–18950. [CrossRef]
15. Dai, M.; Zheng, D.; Na, R.; Wang, S.; Zhang, S. EEG classification of motor imagery using a novel deep learning framework. *Sensors* **2019**, *19*, 551. [CrossRef]
16. Tabar, Y.R.; Halici, U. A novel deep learning approach for classification of EEG motor imagery signals. *J. Neural Eng.* **2016**, *14*, 016003. [CrossRef]
17. Xu, G.; Shen, X.; Chen, S.; Zong, Y.; Zhang, C.; Yue, H.; Liu, M.; Chena, F.; Che, W. A deep transfer convolutional neural network framework for EEG signal classification. *IEEE Access* **2019**, *7*, 112767–112776. [CrossRef]
18. Li, F.; He, F.; Wang, F.; Zhang, D.; Xia, Y.; Li, X. A Novel Simplified Convolutional Neural Network Classification Algorithm of Motor Imagery EEG Signals Based on Deep Learning. *Appl. Sci.* **2020**, *10*, 1605. [CrossRef]
19. Roy, S.; Chowdhury, A.; McCreadie, K.; Prasad, G. Deep learning based inter-subject continuous decoding of motor imagery for practical brain-computer interfaces. *Front. Neurosci.* **2020**, *14*, 918. [CrossRef]
20. Gal, Y. Uncertainty in Deep Learning. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2016.
21. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1050–1059.
22. Ghoshal, B.; Tucker, A. On Cost-Sensitive Calibrated Uncertainty in Deep Learning: An application on COVID-19 detection. In Proceedings of the 34th International Symposium on Computer-Based Medical Systems (CBMS), Aveiro, Portugal, 7–9 June 2021; pp. 503–509.
23. Abideen, Z.U.; Ghafoor, M.; Munir, K.; Saqib, M.; Ullah, A.; Zia, T.; Tariq, S.A.; Ahmed, G.; Zahra, A. Uncertainty assisted robust tuberculosis identification with bayesian convolutional neural networks. *IEEE Access* **2020**, *8*, 22812–22825. [CrossRef]
24. Stoean, C.; Stoean, R.; Atencia, M.; Abdar, M.; Velázquez-Pérez, L.; Khosravi, A.; Nahavandi, S.; Acharya, U.R.; Joya, G. Automated detection of presymptomatic conditions in Spinocerebellar Ataxia type 2 using Monte Carlo dropout and deep neural network techniques with electrooculogram signals. *Sensors* **2020**, *20*, 3032. [CrossRef] [PubMed]
25. Jungo, A.; Meier, R.; Ermis, E.; Blatti-Moreno, M.; Herrmann, E.; Wiest, R.; Reyes, M. On the effect of inter-observer variability for a reliable estimation of uncertainty of medical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Granada, Spain, 16–20 September 2018; pp. 682–690.
26. Jungo, A.; Meier, R.; Ermis, E.; Blatti-Moreno, M.; Herrmann, E.; Wiest, R.; Reyes, M. DR| GRADUATE: Uncertainty-aware deep learning-based diabetic retinopathy grading in eye fundus images. *Med. Image Anal.* **2020**, *63*, 101715.
27. Leibig, C.; Allken, V.; Ayhan, M.S.; Berens, P.; Wahl, S. Leveraging uncertainty information from deep neural networks for disease detection. *Sci. Rep.* **2017**, *7*, 17816. [CrossRef] [PubMed]

28. Gill, R.S.; Caldairou, B.; Bernasconi, N.; Bernasconi, A. Uncertainty-Informed Detection of Epileptogenic Brain Malformations Using Bayesian Neural Networks. In Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Shenzhen, China, 13–17 October 2019; pp. 225–233.
29. Milanés-Hermosilla, D.; Codorniú, R.T.; López-Baracaldo, R.; Sagaró-Zamora, R.; Delisle-Rodriguez, D.; Villarejo-Mayor, J.J.; Núñez-Álvarez, J.R. Monte Carlo Dropout for Uncertainty Estimation and Motor Imagery Classification. *Sensors* **2021**, *21*, 7241. [CrossRef] [PubMed]
30. Su, J.; Vargas, D.V.; Sakurai, K. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.* **2019**, *23*, 828–841. [CrossRef]
31. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
33. Duerr, O.; Sick, B.; Murina, E. *Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability*; Manning Publications: Shelter Island, NY, USA, 2020.
34. Steinbrener, J.; Posch, K.; Pilz, J. Measuring the uncertainty of predictions in deep neural networks with variational inference. *Sensors* **2020**, *20*, 6011. [CrossRef]
35. Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U.R.; et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion* **2021**, *76*, 243–297. [CrossRef]
36. Lorieul, T. Uncertainty in Predictions of Deep Learning Models for Fine-Grained Classification. Ph.D. Thesis, Université Montpellier, Montpellier, France, 2020.
37. De Waal, A.; Steyn, C. Uncertainty measurements in neural network predictions for classification tasks. In Proceedings of the 2020 IEEE 23rd International Conference on Information Fusion (FUSION), Rustenburg, South Africa, 6–9 July 2020; pp. 1–7.
38. Krishnan, R.; Subedar, M.; Tickoo, O. Specifying weight priors in bayesian deep neural networks with empirical bayes. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 4477–4484.
39. Mobiny, A.; Yuan, P.; Moulik, S.K.; Garg, N.; Wu, C.C.; Van Nguyen, H. Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Sci. Rep.* **2021**, *11*, 5458. [CrossRef]
40. Mobiny, A.; Singh, A.; Van Nguyen, H. Risk-aware machine learning classifier for skin lesion diagnosis. *J. Clin. Med.* **2019**, *8*, 1241. [CrossRef]
41. Cicalese, P.A.; Mobiny, A.; Shahmoradi, Z.; Yi, X.; Mohan, C.; Van Nguyen, H. Kidney Level Lupus Nephritis Classification Using Uncertainty Guided Bayesian Convolutional Neural Networks. *IEEE J. Biomed. Health Inform.* **2020**, *25*, 315–324. [CrossRef]
42. Tangermann, M.; Müller, K.-R.; Aertsen, A.; Birbaumer, N.; Braun, C.; Brunner, C.; Leeb, R.; Mehring, C.; Miller, K.J.; Müller-Putz, G.R.; et al. Review of the BCI competition IV. *Front. Neurosci.* **2012**, *6*, 55. [CrossRef]
43. Ang, K.K.; Chin, Z.Y.; Zhang, H.; Gaun, C. Filter bank common spatial pattern (FBCSP) in brain-computer interface. In Proceedings of the International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; pp. 2390–2397.
44. Ang, K.K.; Chin, Z.Y.; Wang, C.; Guan, C.; Zhang, H. Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. *Front. Neurosci.* **2012**, *6*, 39. [CrossRef]
45. Schirrmeister, R.T.; Springenberg, J.T.; Fiederer, L.D.J.; Glasstetter, M.; Eggensperger, K.; Tangermann, M.; Hutter, F.; Burgard, W.; Ball, T. Deep learning with convolutional neural networks for EEG decoding and visualization. *Hum. Brain Mapp.* **2017**, *38*, 5391–5420. [CrossRef]
46. Sakhavi, S.; Guan, C. Convolutional neural network-based transfer learning and knowledge distillation using multi-subject data in motor imagery BCI. In Proceedings of the 8th International IEEE/EMBS Conference on Neural Engineering (NER), Shanghai, China, 25–28 May 2017; pp. 588–591.
47. Amin, S.U.; Alsulaiman, M.; Muhammad, G.; Mekhtiche, M.A.; Hossain, M.S. Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion. *Future Gener. Comput. Syst.* **2019**, *101*, 542–554. [CrossRef]
48. Sun, B.; Zhao, X.; Zhang, H.; Bai, R.; Li, T. EEG Motor Imagery Classification With Sparse Spectrotemporal Decomposition and Deep Learning. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 541–551. [CrossRef]
49. Pfurtscheller, G.; Neuper, C. Motor imagery activates primary sensorimotor area in humans. *Neurosci. Lett.* **1997**, *239*, 65–68. [CrossRef]
50. Pfurtscheller, G.; Neuper, C. Motor imagery and direct brain-computer communication. *Proc. IEEE* **2001**, *89*, 1123–1134. [CrossRef]
51. Pfurtscheller, G.; Neuper, C.; Brunner, C.; da Silva, F.L. Beta rebound after different types of motor imagery in man. *Neurosci. Lett.* **2005**, *378*, 156–159. [CrossRef]
52. Ramoser, H.; Muller-Gerking, J.; Pfurtscheller, G. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 441–446. [CrossRef]
53. Pfurtscheller, G.; Brunner, C.; Schlögl, A.; da Silva, F.L. Mu rhythm (de) synchronization and EEG single-trial classification of different motor imagery tasks. *NeuroImage* **2006**, *31*, 153–159. [CrossRef] [PubMed]
54. Hung, C.-I.; Lee, P.-L.; Wu, Y.-T.; Chen, L.-F.; Yeh, T.-C.; Hsieh, J.-C. Recognition of motor imagery electroencephalography using independent component analysis and machine classifiers. *Ann. Biomed. Eng.* **2005**, *33*, 1053–1070. [CrossRef] [PubMed]

55. Pfurtscheller, G.; Da Silva, F.L. Event-related EEG/MEG synchronization and desynchronization: Basic principles. *Clin. Neurophysiol.* **1999**, *110*, 1842–1857. [CrossRef] [PubMed]
56. Wu, H.; Niu, Y.; Li, F.; Li, Y.; Fu, B.; Shi, G.; Dong, M. A Parallel Multiscale Filter Bank Convolutional Neural Networks for Motor Imagery EEG Classification. *Front. Neurosci.* **2019**, *13*, 1275. [CrossRef]
57. Hermosilla, D.M.; Codorniu, R.T.; Baracaldo, R.L.; Zamora, R.S.; Rodriguez, D.D.; Albuerne, Y.L.; Alvarez, J.R.N. Shallow Convolutional Network Excel for Classifying Motor Imagery EEG in BCI applications. *IEEE Access* **2021**, *9*, 98275–98286. [CrossRef]
58. Neal, R.M. *Bayesian Learning for Neural Networks*; Springer Science & Business Media: Berlin, Germany, 2012; Volume 118.
59. Wen, Y.; Vicol, P.; Ba, J.; Tran, D.; Grosse, R. Flipout: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
60. Kingma, D.P.; Salimans, T.; Welling, M. *Variational Dropout and the Local Reparameterization Trick, Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2015; pp. 2575–2583.
61. Kendall, A.; Gal, Y. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 5574–5584.
62. Mena, J.; Pujol, O.; Vitrià, J. Uncertainty-based rejection wrappers for black-box classifiers. *IEEE Access* **2020**, *8*, 101721–101746. [CrossRef]
63. Chow, C. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theory* **1970**, *16*, 41–46. [CrossRef]