IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 14, 2021, accepted June 13, 2021, date of publication June 17, 2021, date of current version June 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3089998

Robust Subject-Independent P300 Waveform Classification via Signal Pre-Processing and Deep Learning

RAJEEV SAHAY, (Graduate Student Member, IEEE), AND CHRISTOPHER G. BRINTON, (Senior Member, IEEE)
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA
Corresponding author: Rajeev Sahay (sahayr@purdue.edu)

ABSTRACT Brain Computer Interfaces (BCIs) are capable of processing neural stimuli using electroencephalogram (EEG) measurements to aid communication capabilities. Yet, BCIs often require extensive calibration steps in order to be tuned to specific users. In this work, we develop a subject independent P300 classification framework, which eliminates the need for user-specific calibration. We begin by employing a series of pre-processing steps, where, among other steps, we consider different trial averaging methodologies and various EEG electrode configurations. We then consider three distinct deep learning architectures and two linear machine learning models as P300 signal classifiers. Through evaluation on three datasets, and in comparison to three benchmark P300 classification frameworks, we find that averaging up to seven trials while using eight specific electrode channels on a two-layered convolutional neural network (CNN) leads to robust subject independent P300 classification. In this capacity, our method achieves greater than a 0.20 gain in AUC in comparison to prior P300 classification methods. In addition, our proposed framework is computationally efficient with training time gains of greater than 3x, compared to linear machine learning models, and online evaluation time speedups of up to 2x compared to benchmark methods.

INDEX TERMS Brain–computer interface, deep learning, EEG, P300, machine learning, signal processing.

I. INTRODUCTION

Patients who have suffered lower brain trauma, such as a stroke or a traumatic brain injury (TBI), are often subjects of locked-in syndrome (LIS). LIS prevents patients from moving their extremities resulting in, among a myriad of other challenges, extremely limited communication capabilities. However, neural stimuli from LIS patients can be analyzed to aid communication abilities. Various neural signal processing algorithms [1]–[4] have been proposed for Brain Computer Interfaces (BCIs) [5], which are controlled using neural inputs from the subject's upper brain activity. Such neural inputs are non-invasively collected with electrodes placed at various positions on a subject's scalp using electroencephalogram (EEG) measurements [6]–[8]. The collected signals are then processed in real time on board the BCI.

The P300 signal [9] is a specific type of Event Related Potential (ERP), which results in a momentary increase of electrical activity in the brain. P300 responses are invoked in subjects using the oddball paradigm, where several items are successively shown and a subject is tasked with focusing on one specific *target* item (typically by counting how many times it appears on a changing screen). Following the display of each *target* item, the brain emits a delayed spike approximately 300 ms – 500 ms after processing the optical input. Contrarily, the signal recorded following a *non-target* item is relatively flat. An example of an invoked P300 *target* and *non-target* stimulus is shown in Fig. 1. Despite the apparent difference between target and non-target signals, however, raw P300 signals contain high levels of noise and variability across and within subjects. For example, as shown in Fig. 2, a raw EEG target stimulus is almost indistinguishable from a non-target signal.

Training machine learning models to classify target vs. non-target P300 signals across subjects has proven challenging. This challenge is largely due to the inherent noise caused by electrode amplifiers leading to a lack of discriminative features between raw target and non-target signals. As a

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Wang.

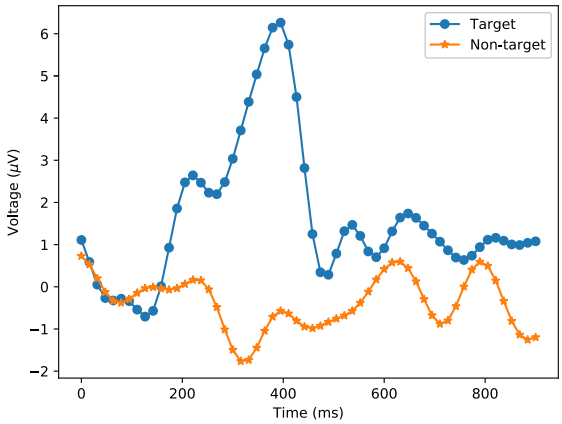

FIGURE 1. Average EEG response for 482 target and 2498 non-target stimuli on Channel Cz. The target response shows an apparent P300 spike compared to the relatively flat non-target response.

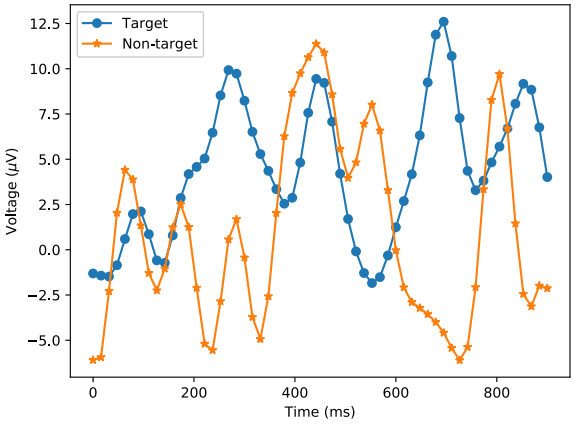

FIGURE 2. EEG response for a single target and non-target stimulus on Channel Cz. Without signal averaging, or other feature engineering techniques, distinguishing target and non-target responses is challenging.

result, BCIs are overwhelmingly trained on subject-specific data that fail to generalize to other users. Although prior work [10]–[14] has attempted to learn common feature spaces shared among various users, they often require a computationally costly domain transformation, achieve modest classification performance on raw EEG time samples, or require additional subject specific calibration. In this work, we construct a computationally efficient subject independent P300 classification framework, which eliminates the needs for subject-specific BCI calibration.

A. RELATED WORK

Traditional P300 classification has largely relied on using pre-processing techniques, such as filtering, trial averaging, and dimensionality reduction, paired with Linear Discriminant Analysis (LDA) classifiers [15]–[19]. Although these methods have been successful for subject-specific models, they are difficult for generalization because better generalization requires higher dimensional input signals, which LDA models often display degraded performance on [20]. Support Vector Machines (SVMs) with linear kernels have also been commonly used for P300 classification [21]–[23]. However, SVMs often require excessive computational costs for processing high dimensional signals and, furthermore, are often used with subject-specific transformed input signals, such as principal components, which hinder the model's ability to effectively learn generalized subject independent feature spaces.

More recent methods have proposed using deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for P300 classification. CNNs represent EEG inputs pictorially to efficiently learn spatially correlated patterns that are difficult for linear models to learn [1], [20], [24]–[27]. However, they often require high-dimensional inputs making them computationally expensive and infeasible to implement on BCIs. Furthermore, CNN classifiers are often subject-specific (i.e., they are trained and tested using data from the same subject) and have not been shown to learn common features for classifying subjects' data that was not included in the training set. We build upon such methods by utilizing deep learning to further minimize online calibration times.

RNNs, on the other hand, have been used to model EEG data sequentially and succinctly to learn temporally correlated data to a higher degree than linear machine learning models. Although RNNs often entail a lower parameter space than CNNs, they have typically shown degraded performance for subject independent classification [11], similar to CNNs and, further, require computationally intense data transformations that have not been shown to work for P300 classification [10]. Finally, convolutional LSTM (ConvLSTM) models, consisting of convolutional and recurrent layers, have been used for single trial P300 classification [28] but are subject-specific and fail to learn a generalized feature space for inter-subject classification.

Furthermore, prior work has largely focused on tuning large-scale deep neural network architectures using minimally pre-processed P300 signals. Contrary to these approaches, we propose using relatively simple pre-processing techniques, which we find significantly reduces the required complexity of the deep learning classifiers. Specifically, our proposed framework allows us to train a low-parameter, yet highly effective, subject independent classifier, which yields faster classification decisions during deployment, while improving the overall classification rate compared to high-parameter deep learning models such as EEGNet [29], multitask autoencoders [30], and deep convolutional neural networks (deep ConvNets) [31].

B. OUTLINE AND SUMMARY OF CONTRIBUTIONS

In this work, we develop a *subject independent* framework for robust P300 signal classification. Our methodology integrates signal pre-processing techniques with machine learning classifiers to improve subject independent prediction accuracy while reducing the required model complexity and, thus, also reducing the required computational overhead during both

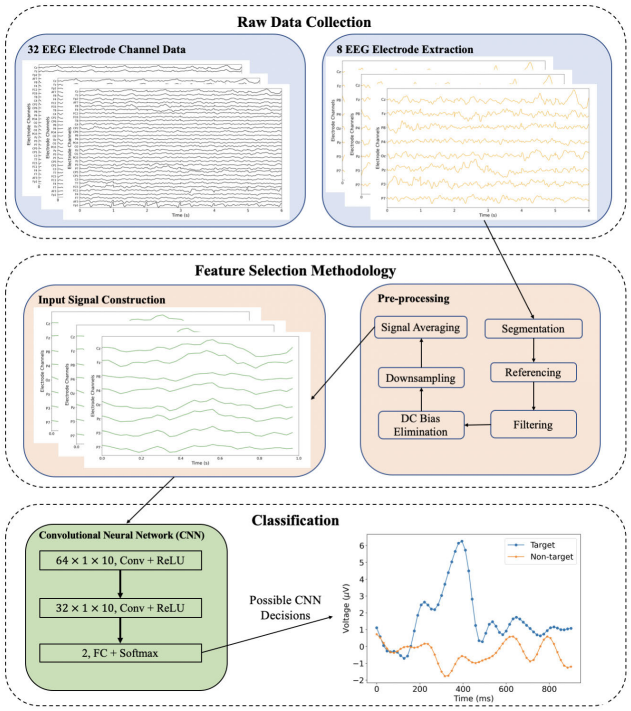

FIGURE 3. Our proposed system structure framework using our most effective feature selection and signal classification techniques. We begin by collecting the raw EEG recording from the subject and extracting the following eight channels: P7, P3, Pz, Oz, P4, P8, Fz, Cz. Next, we perform our proposed feature selection methodology, which consists of signal segmentation, standard referencing, bandpass filtering, DC bias elimination, and subsampling the signal for increased computational efficiency. Finally, we average up to seven signals per event (for target and non-target events) to construct the input signal, which is then classified by our two-layered convolutional neural network (CNN). The CNN classifies each ERP as a target or non-target signal.

development (training) and deployment (real-time decision making). With our methodology in place, we are able to accurately classify P300 signals from subjects whose data has not been used to calibrate the BCI in a computationally efficient manner relative to state-of-the-art baselines. As a result, our method allows for faster communication capabilities by the BCI user during real-world usage.

We begin by discussing our assumptions about the raw EEG data collected by the BCI (Sec. II-A). Next, we outline our feature selection methodology, which consists of steps such as trial averaging, which reduces inherent signal noise, and downsampling, which increases computational efficiency during deployment (Sec. II-B). We also explore using different electrode channel combinations to construct the input signal (Sec II-C), as well as different classifiers including two linear machine learning models and three non-linear deep learning classifiers (Sec. II-D and II-E). We evaluate our methodology on three datasets (Sec. III-A) and empirically explore different channel combinations and trial averaging techniques (Sec. III-B), where we find that using data from eight electrode channels with up to seven averaged signals provides robust subject independent classification accuracy. Using these insights, we demonstrate the robust performance

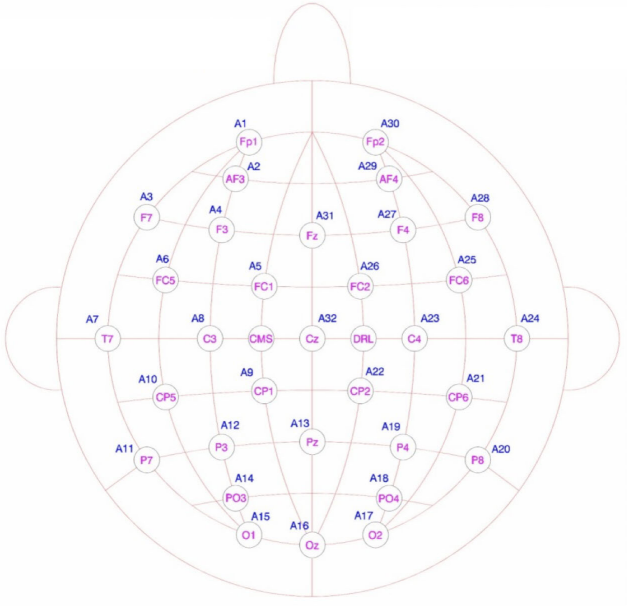

FIGURE 4. 32 channel EEG cap showing the arrangement of each electrode, and its corresponding channel, on the scalp [32].

of our proposed method, compared to three benchmarks (Sec. III-C), and quantify its computational efficiency (Sec. III-D). Lastly, we give concluding remarks and discuss future work in Sec. IV.

II. METHODOLOGY

In this section, we present our proposed P300 framework. We begin by discussing our assumptions about the raw EEG data (Sec. II-A). Then, we outline our novel feature selection algorithm in which we extract salient signal attributes from streams of EEG time-series signals (Sec. II-B). Next, we discuss the utilization of three different electrode configurations for constructing the classification model inputs (Sec. II-C), and we describe our three proposed deep learning classification models as well as two benchmark models, that are traditionally used for P300 classification, which we compare our proposed methods to (Sec. II-D). Lastly, we discuss our evaluation methods used to measure each model's effectiveness (Sec. II-E). Our proposed framework is shown in Fig. 3.

A. EEG DATA COLLECTION

We begin by outlining the form of the initially collected EEG time-series signals. Specifically, we assume that a stream of EEG samples were collected over a period of time using an EEG cap, such as the cap shown in Fig. 4 (although a 32-channel EEG cap is shown, we do not necessarily assume that the data was collected on 32 electrodes), with a sampling rate of f_s. Furthermore, we assume that a P300 signal was evoked using the oddball paradigm at various time instances. From this form of the raw data, we aim to extract all intervals corresponding to target and non-target stimuli and classify them accordingly.

B. PROPOSED FEATURE SELECTION METHODOLOGY

Pre-processing P300 signals for classification requires extensive feature engineering, which can often be computationally intensive leading to slower response times [33]. In this work, we significantly reduce the pre-processing steps required for effective classification, and further, we completely eliminate domain transformations, which are typically used to learn salient P300 signal features. Specifically, for each channel, our proposed data engineering pipeline is as follows. In general, we denote c as the number of electrode channels and s as the sampling rate.

1) *Segmentation*: We begin by extracting 1000 ms of the EEG signal (initially sampled at f_s) following the onset of each target and non-target display.
2) *Referencing*: We reference each signal against the mastoid channels by computing the average of the two mastoid channels and subtracting it from each signal segment.
3) *Filtering*: We filter each signal using a forward-backward Finite Impulse Response (FIR) band-pass filter to remove high frequency artifacts and noise from the waveforms.
4) *DC Bias Elimination*: We eliminate the direct current (DC) bias by averaging the first 100 ms of data in each signal and subtracting it from each signal element.
5) *Downsampling*: We downsample each signal to 32 Hz for computational efficiency, which does not remove salient artifacts of signals with high sampling rates.
6) *Signal Averaging*: We average various numbers of successive target and non-target signals to eliminate noise and variation in signals. The results for averaging different numbers of signals are presented in Sec. III.
7) *Constructing Model Inputs*: We model the inputs for our nonlinear deep learning models as single-channel images consisting of the time-series EEG values from each electrode channel (i.e., each sample was formed into a $c \times s \times 1$ tensor). For the linear models, we aggregate the first two dimensions of the image tensor to form a single vector, which is inputted into the model.

C. CHANNEL COMBINATIONS

We explore the most effective electrodes to extract features from for robust classification. Specifically, in addition to using aggregated signals collected from all 32 electrodes of the EEG cap, we also consider 4-channel and 8-channel configurations where the channels in each configuration are chosen based on the location of electrodes on the scalp that result in the most prominent P300 signal production [34]–[37]. The 4-channel and 8-channel configuration use the following aggregated channels, respectively: Pz, Oz, Fz, Cz and P7, P3, Pz, Oz, P4, P8, Fz, Cz.

D. CLASSIFICATION MODELS

We propose three novel deep learning models to classify P300 signals processed according to our methodology in Sec. II-A. These proposed models consist of (1) convolutional layers, (2) recurrent layers, (3) and both convolutional and recurrent layers. In addition, we describe two linear machine learning models that we use as baselines to compare our proposed models with. These linear models (linear discriminant analysis and support vector machines) are traditionally used in P300 classification [22].

We compare linear machine learning models with non-linear deep learning models in order to demonstrate the necessity of employing a sophisticated classifier on our effectively pre-processed P300 signals. In particular, we find this comparison to be important since (i) linear machine learning models were heavily utilized prior to the advent of deep learning-based classifiers for P300 signal classification and (ii) because the comparison between models will reveal the required complexity needed for the employed classifier. In addition, we also compare our proposed method against three deep learning-based baselines, which are discussed in Sec. III-C.

1) LINEAR DISCRIMINANT ANALYSIS (BASELINE MODEL)

Linear discriminant analysis (LDA) is the most common model used for P300 classification [33] The objective is to determine the classification probability, $P(\mathbf{Y} = y|\mathbf{X} = \mathbf{x})$, of each input sample, $\mathbf{x} \in \mathbb{R}^d$, where the output is given by $y \in \{0, 1\}$ (corresponding to non-target and target classes, respectively) using Bayes' Theorem. Specifically, this is formulated by

$$P(\mathbf{Y} = y|\mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X} = \mathbf{x}|\mathbf{Y} = y)P(\mathbf{Y} = y)}{\sum_{i=0}^{1} P(\mathbf{X} = \mathbf{x}|\mathbf{Y} = i)P(\mathbf{Y} = i)}. \quad (1)$$

The class conditional probability, $P(\mathbf{X} = \mathbf{x}|\mathbf{Y} = y)$, is modeled by the multivariate Gaussian distribution where $P(\mathbf{X} = \mathbf{x}|\mathbf{Y} = 0)$ and $P(\mathbf{X} = \mathbf{x}|\mathbf{Y} = 1)$ share the same covariance matrix, Σ. The PDF, where $|\cdot|$ denotes the determinant operation, is given by

$$P(\mathbf{x}|\mathbf{Y} = y) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_{\mathbf{y}}|}} \exp\left(\frac{-1}{2}(\mathbf{x} - \mu_y)^T \Sigma_{\mathbf{y}}{}^{-1}(\mathbf{x} - \mu_y)\right). \quad (2)$$

Substituting 2 into 1, and taking the logarithm of both sides, yields the LDA model, which is given by

$$P(y|\mathbf{x}) = \left(-\frac{1}{2}(\mathbf{x} - \mu_y)^T \Sigma_{\mathbf{y}}{}^{-1}(\mathbf{x} - \mu_y)\right) + \log P(\mathbf{Y} = y) + \mathbf{C}, \quad (3)$$

where $P(\mathbf{Y} = y)$ can be estimated from the data. Equation (3) can also be written as follows:

$$P(y|\mathbf{x}) = \mathbf{w}_y^T \mathbf{x} + \mathbf{w}_{y0} + \mathbf{C}, \quad (4)$$

where $\mathbf{w}_y^T$ and $\mathbf{w}_{y0}$ are the model parameters fitted during training. The class prediction of a sample, $\mathbf{x} \in \mathbb{R}^d$, is then given from 4 using the learned parameters estimated from the training data.

2) SUPPORT VECTOR MACHINE (BASELINE MODEL)

Training support vector machines (SVMs) on high dimensional EEG data is often extremely computationally costly making it infeasible to implement in BCIs. Therefore, successful implementations often deploy SVMs on a reduced dimensional representation (via Principal Component Analysis) of the training data such as in [21] and [23]. We compare our proposed deep learning method with such SVMs trained on Principal Components (PCs). Specifically, given a training dataset matrix, $\mathcal{X}_{tr}$, (where each row is a single realization $\mathbf{x} \in \mathbb{R}^n$) and its corresponding covariance matrix, Σ, the j^{th} PC, for $j = 1, 2, \ldots, n$, is given by $a_j = \mathbf{v}_j^T \mathcal{X}_{tr}^T$ where $\mathbf{v}_j$ is the eigenvector corresponding to the j^{th} largest eigenvalue. Our experiments reduce the dimensionality of the data to $k = 25$ PCs, which accounts for approximately 80% of variability in each model setup, to alleviate computational power and training time.

The Linear SVM aims to learn a separating hyperplane, which maximizes the margin between itself and the PCs. Specifically, for each training sample, $\mathbf{x}_i \in \mathbb{R}^k$, $i = 1, 2, \ldots, n$, along with its corresponding output, $y \in \{-1, 1\}^n$, the Linear SVM model is given by

$$\hat{y} = \mathbf{w}^T \mathbf{x} + b, \tag{5}$$

where $\mathbf{w} \in \mathbb{R}^k$ and $b \in \mathbb{R}$ are the model parameters learned during training. The model is then minimized by employing the hinge loss in the objective function

$$\min_{\mathrm{w,b}} \frac{1}{2} ||\mathbf{w}^T \mathbf{w}|| + \sum_i (y_i(\mathbf{w}^T \mathbf{x}_i + b))^+. \tag{6}$$

The optimal $\mathbf{w}$ and b found in (6) is then used to construct the model shown in (5). Finally, given an input testing sample, $\mathbf{x} \in \mathbb{R}^k$, and optimized parameters, $\mathbf{w} \in \mathbb{R}^k$ and $b \in \mathbb{R}$, the SVM prediction is given by

$$\mathrm{sgn}(\mathbf{w}^T \mathbf{x} + b), \tag{7}$$

where $\mathrm{sgn}(\cdot)$ is the sign of the resulting vector corresponding to target and non-target responses.

3) CONVOLUTIONAL NEURAL NETWORK (PROPOSED ARCHITECTURE)

Convolutional neural networks (CNNs) have achieved state-of-the-art performance in a myriad of tasks, such as computer vision [38] and neuroimaging [39], due to their superb abilities to learn and analyze spatially correlated patterns. Their successes have also been shared in EEG signal processing but with the requirement of high parameter models, which consume excessive training time, and have not been shown to generalize across subjects, making them infeasible for BCI implementation. Herein, we propose a low-parameter, two-layered CNN model with small kernel dimensions that are highly efficient to train. The first and second layers contain 64 and 32 feature maps, respectively, each with 35% dropout to avoid overfitting. Each layer consists of 1×10 dimensional kernels and employ the Rectified Linear Unit (ReLU) activation function, which is given by

$$\sigma(a) = \max\{0, a\}. \tag{8}$$

The output of each feature map is given by

$$\sigma\big(\mathbf{v} * \mathbf{a} + b\big), \tag{9}$$

where $*$ denotes convolution, $\mathbf{a}$ is the input into the convolutional layer, $\mathbf{v}$ is the kernel whose parameters are learned during training, and b is a bias threshold.

4) RECURRENT NEURAL NETWORK (PROPOSED ARCHITECTURE)

Recurrent Neural Networks (RNNs) employ feedback systems to create memory within deep learning models. Specifically, given an input, $\mathbf{x} \in \mathbb{R}^d$, a recurrent layer, with k units, calculates each hidden state, $\mathbf{h} \in \mathbb{R}^k$ (where each element of $\mathbf{h}$ is represented by h_t for $t = 1, 2, \ldots, k$), and layer output, $y \in \mathbb{R}^k$, for the subsequent unit, t, according to the following formulation (with a random initial state for $\mathbf{h}^0$):

$$\boldsymbol{a}^{(t)} = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \tag{10}$$

$$\boldsymbol{h}^{(t)} = tanh(\boldsymbol{a}^{(t)}) \tag{11}$$

$$\boldsymbol{o}^{(t)} = \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \tag{12}$$

$$\boldsymbol{y}^{(t)} = \sigma(\boldsymbol{o}^{(t)}), \tag{13}$$

where $\mathbf{W}$ denote parameters linking hidden layers, $\mathbf{U}$ denote parameters used for connecting the input to the hidden layers, and $\mathbf{V}$ denote parameters connecting the hidden state to the layer's output, and $\sigma(\cdot)$ is an activation function.

Long-short-term-memory (LSTM) cells [40] extend the idea of recurrent layers and were initially designed to mitigate the vanishing gradient problem, but they were later found to deliver strong performance on time-series data [41] and, therefore, have been used to classify P300 signals. LSTM cells differ from the foregoing recurrent behavior by introducing three operations: *input gates* to prevent the hidden state of the respective LSTM unit from learning irrelevant inputs; *forget gates* to eliminate input features that non-relevant features after concatenating x_t with h_t; and *output gates*, which contain the outputs of the LSTM layer and are inputted into the next layer. During forward propagation, the *input gate*, $g_i^{(t)}$, and *forget gate*, $f_i^{(t)}$, for cell i at time step t are given by:

$$g_i^{(t)} = \sigma^1\left(\sum_j \mathbf{V}_{i,j}^g \mathbf{x}_j^{(t)} + \sum_j \mathbf{W}_{i,j}^g \mathbf{h}_j^{(t-1)} + \mathbf{b}_i^g\right) \tag{14}$$

$$f_i^{(t)} = \sigma^1\left(\sum_j \mathbf{V}_{i,j}^f \mathbf{x}_j^{(t)} + \sum_j \mathbf{W}_{i,j}^f \mathbf{h}_j^{(t-1)} + \mathbf{b}_i^f\right), \tag{15}$$

where $\sigma^1(\cdot)$ is the logistic sigmoid activation function given (element-wise) by $\sigma^1(\mathbf{a}) = 1/(1 + \exp(\mathbf{a}))$ and $\mathbf{V}^{g,f}$, $\mathbf{W}^{g,f}$, and $\mathbf{b}^{g,f}$ are the input weights, recurrent weights, and bias vector, respectively, for the input and forget gates that act on the current input vector, $\mathbf{x}^{(t)}$, and current hidden layer vector,

$\mathbf{h}^{(t)}$. The internal state of the cell, $s_i^{(t)}$, is calculated using (14) and (16) together:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(\sum_j \mathbf{V}_{i,j} \mathbf{x}_j^{(t)} + \sum_j \mathbf{W}_{i,j} \mathbf{h}_j^{(t-1)} + \mathbf{b}_i \right), \tag{16}$$

where $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{b}$ are the input weights, recurrent weights, and input bias vector, respectively. Finally, the *output gate*, $p_i^{(t)}$, and cell output, $q_i^{(t)}$, is given by:

$$p_i^{(t)} = \sigma\left(\sum_j \mathbf{V}_{i,j}^0 \mathbf{x}_j^{(t)} + \sum_j \mathbf{W}_{i,j}^0 \mathbf{h}_j^{(t-1)} + \mathbf{b}_i^0 \right), \tag{17}$$

$$q_i^{(t)} = \tanh(s_i^t) p_i^t. \tag{18}$$

The model parameters are optimized by minimizing the objective function shown in (20) using the backpropagation algorithm. Our RNN model consisted of a single LSTM layer with 32 cells, with input $\mathbf{x} \in \mathbb{R}^{c \times s}$, followed by a fully connected[1] ReLU layer containing 64 units.

5) CONVOLUTIONAL LSTM NETWORK (PROPOSED ARCHITECTURE)

Our last proposed deep learning model is the convolutional LSTM recurrent neural network (CRNN) inspired from [28] for subject-specific single-trial P300 classification. CRNN models are considered to capture spatially and temporally correlated data resulting in increased classification performance. In this work, we construct a CRNN model, with input $\mathbf{x} \in \mathbb{R}^{c \times s \times 1}$, consisting of two ReLU convolutional layers with 64 and 32 feature maps, respectively, each with a 1×10 kernel, followed by a 16 cell LSTM layer.

E. TRAINING AND PERFORMANCE EVALUATION

Each proposed deep learning model is trained using the Adam optimizer with a learning rate of 0.001 and 100 epochs. Furthermore, the output layer of each proposed deep learning model consists of a two-unit dense layer with the softmax normalization function given by

$$\sigma^1(\mathbf{h})_i = \frac{e^{\mathbf{h}_i}}{\sum_{k=1}^2 e^{\mathbf{h}_k}}, \tag{19}$$

where $\mathbf{h}$ is the model output, prior to softmax normalization, termed logits. The softmax normalization results in a probabilistic interpretation of the outputs where $\underset{i}{\text{argmax}} f_i(\mathbf{h})$ is the model's assigned classification for any given input sample $\mathbf{x}$. Finally, each model is optimized by minimizing the categorical cross entropy function, which is given by

$$\mathcal{L} = -\sum_{i=1}^{2} y_i \log(\sigma^1(\hat{y}_i)), \tag{20}$$

where y_i is the ground-truth label, $\hat{y}$ is the classifier's assigned label.

[1]The output of the fully connected layer, at each unit, is given by $[\mathbf{w}^T \mathbf{x} + b]^+$, where $\mathbf{w} \in \mathbb{R}^k$ and $b \in \mathbb{R}$ are the weight vector and bias, respectively, learned during training, and $\mathbf{x} \in \mathbb{R}^k$ is the output from the preceding layer.

III. RESULTS

Here, we evaluate our proposed methods from Sec. II. We begin by introducing our employed datasets (Sec. III-A). Next, we quantify the performance of each considered model under multiple feature averaging techniques and electrode configurations (Sec. III-B). We then show detailed metrics outlining the strength of the strongest performing model for each considered dataset (Sec. III-C). Finally, we discuss the computational feasibility of our proposed methods (Sec. III-D).

A. DATASETS

We evaluate our proposed method on three publicly available P300 datasets, which are described below. Each dataset was collected by invoking a visual stimulus using the oddball paradigm on each subject. Moreover, the number of subjects in each dataset, denoted by s, differs, allowing us to effectively assess the performance of our methodology on subjects whose data has not been exposed to the training algorithm. In this capacity, we train each considered model using s-fold cross validation, where each model was trained using data from $s-1$ subjects and tested on data collected from the excluded subject for all s training combinations. The results from each individual subject on each model were aggregated to produce an averaged result to measure performance as shown in Sec. III-B.

Dataset A: This dataset consists of $s = 4$ subjects and is constructed using a subset of the data (from healthy subjects only) presented in [16]. The data were collected by randomly illuminating one of six items from a menu for 100 ms. Each one of four subjects were tasked with counting the number of illuminations of one specific target image. EEG data was collected from each subject over several sessions using a 32-channel EEG cap with a sampling frequency of 2048 Hz. The four subjects were all males (ages 30 ± 2.3 years).

Dataset B: This dataset consists of $s = 16$ subjects and was constructed by acquiring EEG data from 16 healthy young adults (ranging in age from 22-30 years old) with no history of neurological, physical, or psychiatric illness [12]. The data were collected on a 16-channel, active Ag/AgCl electrodes, EEG cap (with a sampling frequency of 256 Hz) using a P300 speller board by illuminating target characters for 100 ms with an inter-stimulus interval of 150 ms.

Dataset C: Similar to Dataset A, this dataset was constructed using only the healthy subjects from the data presented in [42] for a total of $s = 42$ subjects. The subjects ranged between 19 and 35 years in age. Each subject was tasked with counting the number of target characters, which would illuminate for 100 ms on a P300 speller board. The EEG signals were captured using a g.USBamp 8-channel EEG cap with a sampling frequency of 256 Hz.

B. MODEL EVALUATION

Each of the three electrode configurations were evaluated by averaging up to 15 EEG segments (epochs). Figs. 5-9 show

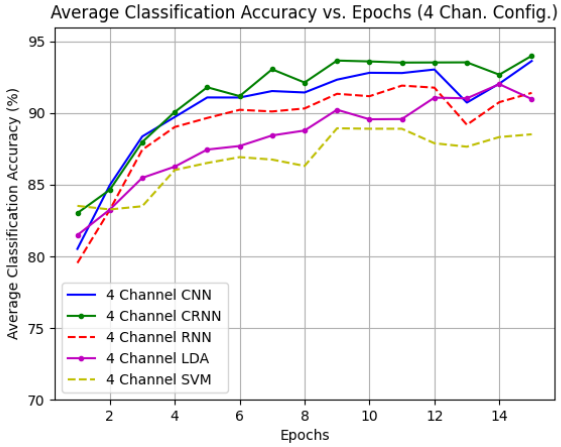

FIGURE 5. Average classification performance at various averaged epochs using the 4-channel electrode configuration on Dataset A. We see that the deep learning models, and in particular the CRNN, deliver better performance compared to traditional linear machine learning models.

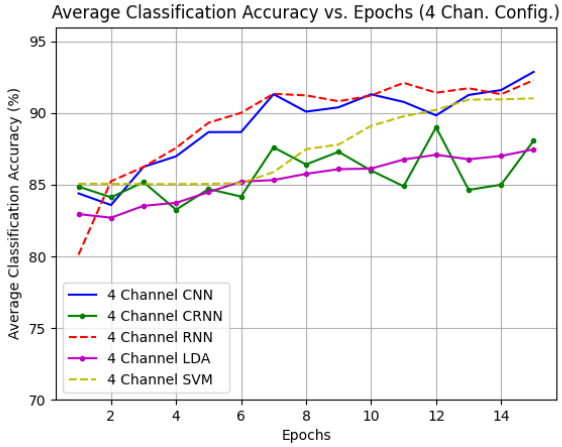

FIGURE 6. Average classification performance at various averaged epochs using the 4-channel electrode configuration on Dataset B. Here, we see that the CNN and RNN are the strongest overall performing models.

each considered model's performance when averaging various numbers of waveforms on Datasets A and B. Note that the average classification accuracy in each figure was calculated by averaging the accuracy of all subjects in their respective datasets when their data was used for testing the respective model (individual subject performance is discussed in further detail below). In all three electrode configurations, our proposed deep learning models outperform traditional linear machine learning models and, in particular, we see that the CNN is consistently one of the strongest performing models.

As shown in Figs. 5 and 6, the highest average classification performance using the 4-channel configuration, for each model, is achieved when a larger number of epochs are averaged together. This is expected as averaging more signals results in lower noise variance and a higher signal to noise ratio (SNR). Although averaging less than eight segments results in lower performance for each model on the 4-channel configuration, each of our proposed deep learning models outperform the SVM and LDA on the raw EEG time samples. However, the lower performance across all tested models indicates that the four selected electrodes may eliminate salient subject-independent P300 features, which are captured on other electrodes and required for classifying P300 signals across subjects.

The 8-channel electrode configuration significantly outperforms the 4-channel configuration in that the deep learning models are able to achieve high accuracy when averaging a small number of segments on both Dataset A and B. For example, the CNN trained on averages of six waveforms achieve an average of 95.05% accuracy across the four subjects in Dataset A and an average accuracy of 92.18% across the 16 subjects in Dataset B. The CNN's ability to learn generalized features on low numbers of averaged epochs indicates the potential for rapid real time P300 classification on deployed BCIs as opposed to linear methods, which require longer processing times for better performance and

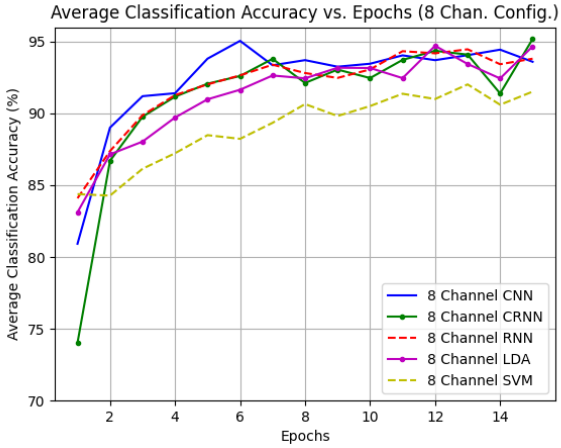

FIGURE 7. Average classification performance at various averaged epochs using the 8-channel configuration on Dataset A. Averaging six epochs and using a CNN for classification delivers the highest average accuracy across subjects and significantly outperforms linear classifiers.

tend to learn subject-specific signal artifacts. Furthermore, contrary to the 4-channel configuration, the CNN is the best performing overall model across each segmentation value in the 8-channel configuration indicating that learning discriminative features from averaging a small number of waveforms is possible using both efficient pre-procesing and an efficient classification model. Lastly, each model's ability to deliver relatively high classification accuracy using the 8-channel configuration indicates that salient features required for effective P300 classification are captured on merely eight electrodes.

High electrode configurations are often used for EEG signal processing because they consist of high-resolution attributes believed to improve BCI algorithms. However, our results demonstrate that using time samples from more than eight electrodes degrades classification performance in both deep learning and linear machine learning classification models. To demonstrate this, we show the performance of our

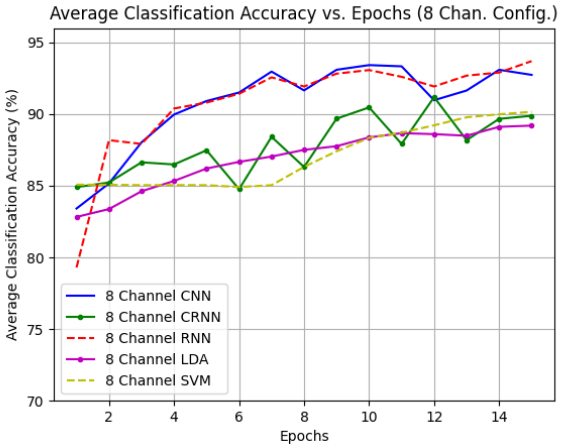

FIGURE 8. Average classification performance at various averaged epochs using the 8-channel configuration on Dataset B. We see that the CNN and RNN are again the strongest performing models on average.

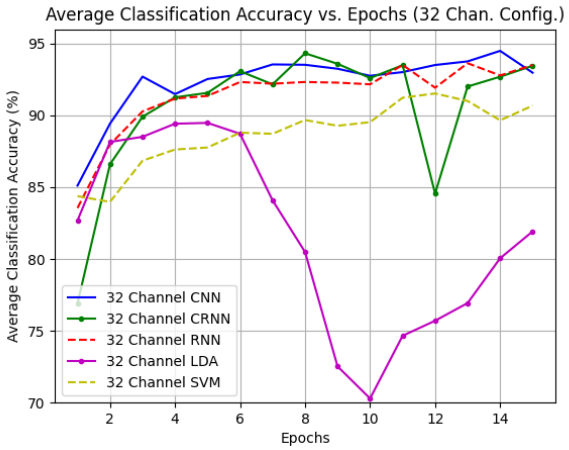

FIGURE 9. Average classification performance at various averaged epochs using the 32-channel electrode configuration on Dataset A. Similar to Figs. 5 - 8, we see that the CNN is, on average, the best performer in terms of classification accuracy.

considered classifiers using the 32-electrode configuration on Dataset A and the 16-electrode configuration on Dataset B in Figs. 9 and 10, respectively. Here, the lower performance of the SVM and LDA models is expected as linear models have exhibited degraded performance when processing high-dimensional EEG inputs [20]. However, our considered deep learning models also attain lower classification performance, compared to the lower electrode configurations indicating that they have higher difficulty in learning discriminative features when they process signals from higher numbers of electrodes. Moreover, the superior ability of the deep learning models, compared to linear models, is consistent with the other electrode channel configurations.

As a general trend, the P300 classification accuracy tends to increase when higher numbers of epochs are averaged. Yet, we see that models trained on more than eight channels (i.e., as in Figs. 9 and 10), sometimes deviate from this trend. Specifically, in these cases, the classification accuracy drops on certain classifiers (in particular on recurrent deep learning-based models and the LDA classifier) before increasing for higher averaged epochs. We believe that this is due to the LDA failing to effectively separate target and non-target trials in high-dimensional signals due to the overlap in the variance of the two classes, leading to a lack of effective separability (consistent with the findings of [20]). Furthermore, the recurrent-based classifiers trained on high dimensional inputs may learn ineffective time correlations on certain channels, which do not provide salient characteristics for effective distinction between target and non-target signals, thus leading to lower classification performance in certain cases.

C. SUBJECT PERFORMANCE

We now evaluate the efficacy of our method on our three considered datasets. In doing so, we compare our proposed method to three baseline P300 signal classification frameworks: EEGNet [29], multitask autoencoders

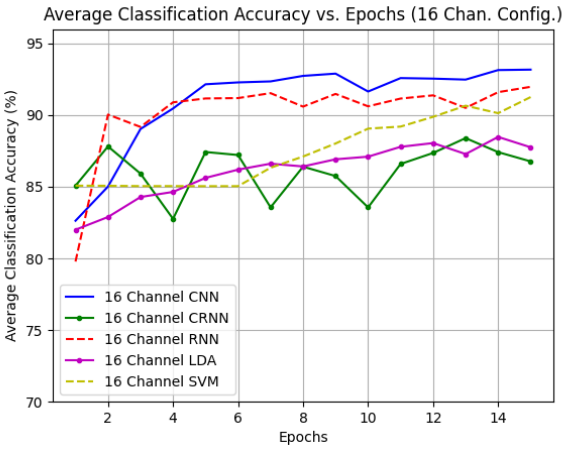

FIGURE 10. Average classification performance at various averaged epochs using the 16-channel configuration on Dataset B. Similar to Fig. 8, the performance of the SVM and LDA increase after dropping.

(MT AE) [30], and deep convolutional neural networks (deep ConvNets) [31]. The EEGNet is a multi-layered binary CNN classifier, with each layer consisting of either temporal, spatial, or pointwise filters along with batch normalization, dropout, and downsampling (via pooling). The deep ConvNet is similar to the EEGNet, with the exception of using several more convolutional layers in its construction. Lastly, the MT AE trains a deep autoencoder (consisting of convolutional and recurrent LSTM layers) on the collected ERP signals and then uses the autoencoder's latent variable representation to train a binary classifier to distinguish between target and non-target signals.

We evaluate each dataset independently by training the model using $s-1$ subjects from the corresponding dataset. Then, we use the outstanding subject for evaluation, where we first determine the number of true positives (TP), false positives (FP), True Negatives (TN), and False Negatives (FN).

TABLE 1. Comparison of our proposed method with three benchmarks. Each metric corresponds to the average value over each subject in the dataset (as well as its standard deviation) when that subject was used as the testing data. We see that the proposed method is always the best performing except on Dataset C, where the deep ConvNet performs equivalently to the proposed method. The proposed method also provides a faster classification time, quantified in Table 5.

Dataset	Method	Accuracy (%)	Recall	Precision	Error	F-measure	AUC
A	**Proposed**	$\mathbf{95.05 \pm 5.26}$	$\mathbf{0.12 \pm 0.05}$	$\mathbf{0.93 \pm 0.05}$	$\mathbf{0.31 \pm 0.32}$	$\mathbf{0.21 \pm 0.08}$	$\mathbf{0.87 \pm 0.16}$
A	EEGNet [29]	78.47 ± 8.43	0.06 ± 0.06	0.35 ± 0.08	1.39 ± 0.59	0.08 ± 0.07	0.57 ± 0.06
A	MT AE [30]	83.97 ± 0.87	0.01 ± 0.00	0.39 ± 0.11	1.03 ± 0.03	0.01 ± 0.00	0.51 ± 0.01
A	Deep ConvNet [31]	81.76 ± 5.95	0.10 ± 0.08	0.39 ± 0.13	1.17 ± 0.40	0.15 ± 0.11	0.71 ± 0.19
B	**Proposed**	$\mathbf{92.18 \pm 7.78}$	$\mathbf{0.12 \pm 0.05}$	$\mathbf{0.83 \pm 0.17}$	$\mathbf{0.52 \pm 0.53}$	$\mathbf{0.21 \pm 0.05}$	$\mathbf{0.85 \pm 0.08}$
B	EEGNet [29]	84.57 ± 2.74	0.01 ± 0.01	0.29 ± 0.22	1.04 ± 0.09	0.01 ± 0.02	0.51 ± 0.02
B	MT AE [30]	83.32 ± 6.96	0.01 ± 0.04	0.06 ± 0.11	1.12 ± 0.46	0.01 ± 0.04	0.51 ± 0.02
B	Deep ConvNet [31]	78.49 ± 2.39	0.03 ± 0.01	0.20 ± 0.04	1.46 ± 0.14	0.05 ± 0.01	0.52 ± 0.01
C	**Proposed**	$\mathbf{100.00 \pm 0.00}$	$\mathbf{0.17 \pm 0.01}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.30 \pm 0.02}$	$\mathbf{1.00 \pm 0.00}$
C	EEGNet [29]	88.98 ± 2.23	0.09 ± 0.02	0.82 ± 0.05	0.63 ± 0.12	0.16 ± 0.04	0.72 ± 0.06
C	MT AE [30]	82.79 ± 1.04	0.01 ± 0.01	0.49 ± 0.31	0.99 ± 0.02	0.01 ± 0.01	0.51 ± 0.01
C	**Deep ConvNet [31]**	$\mathbf{100.00 \pm 0.00}$	$\mathbf{0.17 \pm 0.01}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.30 \pm 0.02}$	$\mathbf{1.00 \pm 0.00}$

TABLE 2. Subject-specific metrics on each considered model using the 8-channel configuration on Dataset A. Rows in bold indicate the best performing model for its corresponding subject.

Subject	Model	TP	FP	TN	FN	Accuracy	Recall	Precision	Error	F-measure	AUC
1	**CNN**	**67**	**9**	**371**	**0**	**97.9866**	**0.1530**	**0.8816**	**0.1343**	**0.2607**	**0.9882**
	RNN	67	32	348	0	92.8412	0.1614	0.6768	0.4776	0.2607	0.9579
	CRNN	62	20	360	5	94.4072	0.1469	0.7561	0.3731	0.2460	0.9364
	LDA	58	2	378	9	97.5391	0.1330	0.9667	0.1642	0.2339	0.9302
	SVM	38	7	373	29	91.9463	0.0925	0.8444	0.5373	0.1667	0.7744
2	**CNN**	**66**	**1**	**378**	**1**	**99.5516**	**0.1486**	**0.9851**	**0.0299**	**0.2583**	**0.9912**
	RNN	67	8	371	0	98.2063	0.1530	0.8933	0.1194	0.2612	0.9894
	CRNN	63	13	366	4	96.1883	0.1469	0.8289	0.2537	0.2495	0.9530
	LDA	59	23	356	8	93.0493	0.1422	0.7195	0.4627	0.2374	0.9100
	SVM	35	3	376	32	92.1525	0.0852	0.9211	0.5224	0.1559	0.7572
3	**CNN**	**68**	**1**	**433**	**17**	**96.5318**	**0.1357**	**0.9855**	**0.2118**	**0.2386**	**0.8988**
	RNN	58	4	430	27	94.0270	0.1189	0.9355	0.3647	0.2109	0.8366
	CRNN	57	7	427	28	93.2563	0.1178	0.8906	0.4118	0.2080	0.8272
	LDA	70	30	404	15	91.3295	0.1477	0.7000	0.5294	0.2439	0.8772
	SVM	27	7	427	58	87.4759	0.0595	0.7941	0.7647	0.1107	0.6508
4	CNN	14	2	396	64	86.1345	0.0341	0.8750	0.8462	0.0657	0.5872
	RNN	23	14	384	55	85.5042	0.0565	0.6216	0.8846	0.1036	0.6298
	CRNN	**25**	**11**	**387**	**53**	**86.5546**	**0.0607**	**0.6944**	**0.8205**	**0.1116**	**0.6464**
	LDA	8	3	395	70	84.6639	0.0199	0.7273	0.9359	0.0386	0.5475
	SVM	10	21	377	68	81.3025	0.0258	0.3226	1.1410	0.0478	0.5377

Using these metrics, we calculate the accuracy, recall, precision, error, F-measure, and the Area Under the Curve (AUC) of the receiver operating characteristic curve. The calculations for each metric is shown in (21) - (25) below.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{21}$$

$$\text{Recall} = \frac{TP}{TP + TN} \tag{22}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{23}$$

$$\text{Error} = \frac{FP + FN}{TP + FN} \tag{24}$$

$$\text{F-measure} = 2\frac{Recall \times Precision}{Recall + Precision} \tag{25}$$

Table 1 shows the performance of our proposed framework in comparison to the three considered baselines. Note that we only consider averaging up to seven signals using the 8-channel electrode configuration when evaluating our proposed method, since that combination demonstrated the best balance between a low number of required signals and strong classification performance in Figs. 5 - 10. In Table 1, we see that our proposed method is consistently the strongest performing model in comparison to the three considered baselines. In addition, we see that the performance of each method is somewhat lower, overall, on Dataset B, indicating

TABLE 3. Resulting *p*-values obtained from two-tailed paired-sample t-test between the mean accuracy from our method in comparison to the considered baselines. At a significance level of $\alpha = 0.05$, we see that our results are statistically significant in eight out of the nine cases.

Dataset	Method	p-value
A	EEGNet [29]	0.0204
A	MT AE [30]	0.0228
A	Deep ConvNet [31]	0.0158
B	EEGNet [29]	0.0016
B	MT AE [30]	0.0020
B	Deep ConvNet [31]	2.779×10^{-6}
C	EEGNet [29]	1.1415×10^{-30}
C	MT AE [30]	7.1564×10^{-52}
C	Deep ConvNet [31]	1.0

that the dataset contains less discriminative features in its signals compared to Datasets A and C. Furthermore, we see from Table 1 that both the proposed method and the deep ConvNet perform equivalently on Dataset C, achieving perfect performance across on subject, while the EEGNet and the MT AE also achieve high classification rates on the dataset. We believe this is due to the dataset eliminating bad trials by design (i.e., eliminating trials from the dataset with excessive noise or subject movement during collection), thus increasing the ability to effectively discriminate between target and non-target signals.

To assess the significance of our improvements over the baselines, we conduct pairwise statistical tests between the accuracies obtained by our method and the baselines, and find that the mean improvement in accuracy is statistically significant (with $p < 0.05$) in eight out of the nine cases. Specifically, Table 3 shows the calculated p-values from the two-tailed paired-sample t-test between our proposed method and each considered baseline on each dataset. We see that at a significance level of $\alpha = 0.05$, our improvements are statistically significant in comparison to all baselines on Dataset A and Dataset B. On Dataset C, the mean accuracy from our method is statistically significant in comparison to the EEGNet and the MT AE, but we obtain a p-value of 1.0 when calculating the significance between the Deep ConvNet and our method due to their identical performance.

For a closer examination into the performance of each considered model, we present the results of each considered classifier on every subject in Dataset A in Table 2. Here, we see that the CNN outperforms each of the other considered classification models for Subjects 1-3, but the CRNN is the best performer for classifying signals obtained from Subject 4. However, Subject 4 resulted in lower classification performance regardless of the model as demonstrated by the lower accuracy and AUC values calculated for that subject. [16] claimed that the lower performance experienced by this subject could be due to mental fatigue resulting in low quality data collection. Therefore, assessing the CRNN is the best performing model for subject 4 may not be justifiable due to the overall lower quality of data collected for that subject. Furthermore, as a general trend, the RNN is the second best classification model for each subject followed by the CRNN (except in the case of Subject 4 for the aforementioned reason). The LDA and SVM models achieve the lowest performance for each subject indicating that they are not capable of learning generalizable properties of P300 signals to the same degree as non-linear deep learning models. We found that the 4-channel and 32-channel configurations follow the same trend in which the CNN and RNN are among the best performing classifiers followed by the CRNN and then the LDA and SVM models.

D. MODEL EFFICIENCY

Beyond the classification performance of each model, we also evaluate the computational overhead of each considered classifier.[2] Specifically, we calculate both the training time of each classifier and the evaluation time of samples during deployment.

TABLE 4. Model training times for each model on dataset A. Note that the model used to test each subject was trained on data from the other three subjects.

Channels	Model	Avg. Training Time (s)	Avg. Acc. (%)
4	CNN	13.25 ± 0.35	91.09
	RNN	10.87 ± 0.08	90.23
	CRNN	104.95 ± 2.24	91.68
	LDA	0.10 ± 0.03	87.70
	SVM	2.06 ± 0.57	86.93
8	CNN	17.62 ± 0.55	95.05
	RNN	15.49 ± 0.13	92.65
	CRNN	184.24 ± 4.58	92.60
	LDA	0.13 ± 0.05	91.65
	SVM	2.20 ± 0.40	88.23
32	CNN	47.21 ± 4.99	92.85
	RNN	46.03 ± 0.98	92.33
	CRNN	631.25 ± 15.00	93.08
	LDA	0.89 ± 0.08	88.72
	SVM	8.50 ± 3.25	88.80

TABLE 5. Average online evaluation times (and their standard deviations) for each model and dataset on a single sample. We see that our proposed method delivers the fastest evaluation time per sample while retaining the highest average AUC.

Dataset	Method	Avg. Eval. Time (ms)	Avg. AUC
A	**Proposed**	$\mathbf{49.9 \pm 6.4}$	$\mathbf{0.87 \pm 0.16}$
A	EEGNet [29]	99.9 ± 18.7	0.57 ± 0.06
A	MT AE [30]	86.9 ± 11.2	0.51 ± 0.01
A	Deep ConvNet [31]	115.4 ± 36.2	0.71 ± 0.19
B	**Proposed**	$\mathbf{61.2 \pm 10.0}$	$\mathbf{0.85 \pm 0.08}$
B	EEGNet [29]	95.2 ± 14.8	0.51 ± 0.02
B	MT AE [30]	76.7 ± 6.3	0.51 ± 0.02
B	Deep ConvNet [31]	119.8 ± 77.1	0.52 ± 0.01
C	**Proposed**	$\mathbf{65.4 \pm 19.5}$	$\mathbf{1.00 \pm 0.00}$
C	EEGNet [29]	106.5 ± 60.8	0.72 ± 0.06
C	MT AE [30]	92.0 ± 48.3	0.51 ± 0.01
C	Deep ConvNet [31]	100.0 ± 12.4	1.00 ± 0.00

We begin by evaluating the training times of each classifier that we consider in our proposed method (during offline calibration). The training time of each model depends on several factors such as the number of subjects in the training dataset, the number of trials collected per subject, and number of electrode channels used to construct the input signals. Since each dataset we consider varies widely in these factors, we select Dataset A to serve as an example of the difference experienced in training times between models and electrode configurations. Table 4 shows the average training times of each considered classifier, using different electrode configurations, on Dataset A.

As shown in Table 4, the linear models require significantly less training time compared to the deep learning models. However, the computational efficiency enjoyed by these models is compromised by consistently lower classification performance. Among the deep learning models, on the other hand, both the CNN and RNN have

[2]The computational overhead was calculated when running each model on an NVIDIA Tesla P100 GPU with 16 GB of memory.

relatively low and comparable computational costs accompanied by high average accuracies. The CRNN results in stronger classification performance than either considered linear model but is highly computationally costly requiring up to 631.25 seconds for training while achieving similar performance to both the CNN and RNN for each channel configuration.

Interestingly, our results show that reducing the number of channels used for classification significantly reduces model training time and, in some cases, improves accuracy. For example, the CNN trained using the 8-channel configuration provides a 2.66x speedup in model training, over the CNN trained on the 32-channel configuration, while boosting the average accuracy from 92.85% to 95.05%. The RNN provides a similar speedup of 2.97x while retaining approximately equivalent classification performance when using 8 channels instead of 32. The speedup is even more apparent for the CRNN and SVM models, which provide a 3.42x and 4.42x speedup, respectively, when using the 8-channel configuration in place of the 32 channels. Similar to the RNN, the average classification performance remains approximately equivalent when eliminating features collected from the additional 16 electrodes. For the LDA, the training time remains below one second for both the 8 and 32-channel configurations while actually improving average classification performance when only using 8 channels. However, the increased accuracy in the 8-channel case is significantly lower than the performance achieved by the deep learning models trained on the same channel configuration.

As expected with reduced dimensional inputs, the models trained on the 4-channel configurations provided significant speedups (3.56x, 4.23x, 6.01x, 8.86x, and 4.41x for the CNN, RNN, CRNN, LDA, and SVM, respectively) compared to the models trained on the 32-channel configuration. However, the higher classification performance was achieved using 8-channels indicating that the features lost when reducing the 8-channel configuration to the 4-channel configuration results in eliminating vital features required for learning general P300 properties for effective inter-subject classification. Overall, the CNN trained using the 8-channel configuration provides the best balance between computational efficiency and classification robustness as it achieves the highest average accuracy across all considered models while mitigating the need for excessive training times required by the 32-channel configuration.

Lastly, we consider the average classification time of a sample from each dataset during deployment on our proposed model (using the 8-channel electrode configuration on the CNN). Table 5 summarizes these results and shows a comparison of evaluation times to the three considered benchmarks. Here, we see that our proposed method delivers the fastest online evaluation times (with up to a 2x speedup in some cases) while also resulting in the most robust classification performance. Through analyzing both the offline training and online evaluation times, we find that our proposed method results in the lowest computational efficiency among our considered datasets and benchmarks, thus resulting in faster communication capabilities by the user during deployment.

IV. CONCLUSION

In the scope of this work, we proposed a novel EEG processing pipeline capable of learning intrinsic P300 signal properties. In particular, we demonstrated that our proposed pre-processing methods paired with novel deep learning classification models effectively distinguish target and non-target P300 signals on one particular subject when the model was trained using aggregated data from various other subjects. Our results demonstrated that each considered deep learning model is capable of learning intrinsic P300 signal properties to a greater extent than linear machine learning models, such as the Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA) algorithms, which are traditionally used for P300 classification. Among the three considered deep learning architectures, our results showed a similar performance for both the CNN and RNN where the CNN was shown to be slightly more robust in each experimental setup. The CRNN, although shown to be more robust than either linear model, was consistently the lowest performing deep learning model, on average, across all subjects.

In addition to the CNN achieving the best performance, our experiments revealed the ability of deep learning models to learn intrinsic P300 signal properties without requiring a large amount of signal averaging leading to faster decision making by BCIs while processing fewer epochs. Furthermore, in addition to achieving robust performance with as few as seven averaged trials, using data from eight electrodes was shown to be both more (or equivalently) robust and less computationally costly than using data from all 32 electrodes for classification. The reduced computational overhead stems directly from the reduced cardinality of the input features, which often scales proportionally with the required training time. The equivalent classification performance achieved using eight electrodes indicates that salient P300 signal features are captured on this particular subset of channels as classification performance is not degraded when eliminating the measurements captured on the remaining electrodes.

As noted in prior work, the P300 response evoked by certain subjects with disabilities can differ from expected P300 signals, so the learned feature space captured for different subjects with particular disabilities may differ. In future work, we anticipate exploring and learning these common feature spaces associated with various neurological disorders. Furthermore, an extension of our methodology can be applied using deep neural network autoencoders, which can be used to reconstruct noise-free representations of noisy P300 signals on raw EEG signals. In this capacity, autoencoders can dramatically decrease computational complexity and potentially further eliminate pre-processing steps beyond what was eliminated in this work. Finally, we anticipate applying active learning algorithms to our models in which incorrect predictions made on a particular subject are used to refine the BCI's classification capabilities for real-time calibration. Active

learning would not only result in accurate pre-trained models, but it would also allow real-time algorithmic improvement in BCIs for P300 classification. Ultimately, finding a common feature space for P300 signals across various subjects, while keeping computational costs low, is crucial for designing BCI prototypes that can work without subject-specific calibration. This work steps towards alleviating this challenge by showing the feasibility of capturing common P300 characteristics from a set of subjects using deep learning models.

REFERENCES

[1] Z. Oralhan, "3D input convolutional neural networks for P300 signal detection," *IEEE Access*, vol. 8, pp. 19521–19529, 2020.

[2] J. Jin, Z. Chen, R. Xu, Y. Miao, X. Wang, and T.-P. Jung, "Developing a novel tactile P300 brain-computer interface with a cheeks-stim paradigm," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 9, pp. 2585–2593, Sep. 2020.

[3] T. Zeyl, E. Yin, M. Keightley, and T. Chau, "Adding real-time Bayesian ranks to error-related potential scores improves error detection and auto-correction in a P300 speller," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 1, pp. 46–56, Jan. 2016.

[4] T. Wilaiprasitporn, A. Ditthapron, K. Matchaparn, T. Tongbuasirilai, N. Banluesombatkul, and E. Chuangsuwanich, "Affective EEG-based person identification using the deep learning approach," *IEEE Trans. Cognit. Develop. Syst.*, vol. 12, no. 3, pp. 486–496, Sep. 2020.

[5] B. Z. Allison, E. W. Wolpaw, and J. R. Wolpaw, "Brain–computer interface systems: Progress and prospects," *Expert Rev. Med. Devices*, vol. 4, no. 4, pp. 463–474, Jul. 2007.

[6] P. Sawangjai, S. Hompoonsup, P. Leelaarporn, S. Kongwudhikunakorn, and T. Wilaiprasitporn, "Consumer grade EEG measuring sensors as research tools: A review," *IEEE Sensors J.*, vol. 20, no. 8, pp. 3996–4024, Apr. 2020.

[7] M. Teplan, "Fundamental of EEG measurement," *Meas. Sci. Rev.*, vol. 2, no. 2, pp. 1–11, 2002.

[8] E. Ratti, S. Waninger, C. Berka, G. Ruffini, and A. Verma, "Comparison of medical and consumer wireless EEG systems for use in clinical trials," *Frontiers Hum. Neurosci.*, vol. 11, p. 398, Aug. 2017.

[9] M. L. Avantaggiati, V. Ogryzko, K. Gardner, A. Giordano, A. S. Levine, and K. Kelly, "Recruitment of p300/CBP in p53-dependent signal pathways," *Cell*, vol. 89, no. 7, pp. 1175–1184, Jun. 1997.

[10] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Learning representations from EEG with deep recurrent-convolutional neural networks," 2015, *arXiv:1511.06448*. [Online]. Available: http://arxiv.org/abs/1511.06448

[11] R. Maddula, J. Stivers, M. Mousavi, S. Ravindran, and V. de Sa, "Deep recurrent convolutional neural networks for classifying P300 BCI signals," *GBCIC*, vol. 201, Sep. 2017, pp. 18–22.

[12] B. Abibullaev and A. Zollanvari, "Learning discriminative spatiospectral features of ERPs for accurate brain–computer interfaces," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 5, pp. 2009–2020, Sep. 2019.

[13] J. Jin, S. Li, I. Daly, Y. Miao, C. Liu, X. Wang, and A. Cichocki, "The study of generic model set for reducing calibration time in P300-based brain–computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 1, pp. 3–12, Jan. 2020.

[14] O.-Y. Kwon, M.-H. Lee, C. Guan, and S.-W. Lee, "Subject-independent brain–computer interfaces based on deep convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 3839–3852, Oct. 2020.

[15] M. Fira, "Detection of P300 in a BCI speller," in *Proc. Int. Conf. Hybrid Inf. Technol.* Berlin, Germany: Springer, 2011, pp. 481–487.

[16] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens, "An efficient P300-based brain–computer interface for disabled subjects," *J. Neurosci. methods*, vol. 167, pp. 25–115, Feb. 2008.

[17] M. R. Meshriky, S. Eldawlatly, and G. M. Aly, "An intermixed color paradigm for P300 spellers: A comparison with gray-scale spellers," in *Proc. IEEE 30th Int. Symp. Computer-Based Med. Syst. (CBMS)*, Jun. 2017, pp. 242–247.

[18] J. Qu, F. Wang, Z. Xia, T. Yu, J. Xiao, Z. Yu, Z. Gu, and Y. Li, "A novel three-dimensional P300 speller based on stereo visual stimuli," *IEEE Trans. Human-Mach. Syst.*, vol. 48, no. 4, pp. 392–399, Aug. 2018.

[19] D. B. Ryan, G. Townsend, N. A. Gates, K. Colwell, and E. W. Sellers, "Evaluating brain-computer interface performance using color in the P300 checkerboard speller," *Clin. Neurophysiol.*, vol. 128, no. 10, pp. 2050–2057, Oct. 2017.

[20] A. Farahat, C. Reichert, C. M. Sweeney-Reed, and H. Hinrichs, "Convolutional neural networks for decoding of covert attention focus and saliency maps for EEG feature visualization," *J. Neural Eng.*, vol. 16, no. 6, Oct. 2019, Art. no. 066010.

[21] Y. Li, H. Liu, and S. Wang, "Exploiting EEG channel correlations in P300 speller paradigm for brain-computer interface," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 6, pp. 1653–1662, 2016.

[22] H. Mirghasemi, R. Fazel-Rezai, and M. B. Shamsollahi, "Analysis of P300 classifiers in brain computer interface speller," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2006, pp. 6205–6208.

[23] S. Tayeb, A. Mahmoudi, F. Regragui, and M. M. Himmi, "Efficient detection of P300 using kernel PCA and support vector machine," in *Proc. 2nd World Conf. Complex Syst. (WCCS)*, Nov. 2014, pp. 17–22.

[24] H. Cecotti and A. Graser, "Convolutional neural networks for P300 detection with application to brain-computer interfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 433–445, Mar. 2011.

[25] H. Shan, Y. Liu, and T. Stefanov, "A simple convolutional neural network for accurate P300 detection and character spelling in brain computer interface," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1604–1610.

[26] S. Kundu and S. Ari, "Fusion of convolutional neural networks for P300 based character recognition," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Dec. 2019, pp. 155–159.

[27] H. Du, I. Jouny, Y.-C. Yu, and S. Wang, "Exploring P300-based biometric for individual identification based on convolutional neural networks," in *Proc. IEEE Signal Process. Med. Biol. Symp. (SPMB)*, Dec. 2018, pp. 1–3.

[28] R. Joshi, P. Goel, M. Sur, and H. A. Murthy, "Single trial P300 classification using convolutional LSTM and deep learning ensembles method," in *Proc. Int. Conf. Intell. Hum. Comput. Interact.* Allahabad, India: Springer, 2018, pp. 3–15.

[29] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A compact convolutional neural network for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 42, pp. 117–134, Jul. 2018.

[30] A. Ditthapron, N. Banluesombatkul, S. Ketrat, E. Chuangsuwanich, and T. Wilaiprasitporn, "Universal joint feature extraction for P300 EEG classification using multi-task autoencoder," *IEEE Access*, vol. 7, pp. 68415–68428, 2019.

[31] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, Nov. 2017.

[32] P. Praamstra. *Headcaps*. Accessed: Apr. 2, 2020. [Online]. Available: https://www.biosemi.com/headcap.htm

[33] J. T. Philip and S. T. George, "Visual P300 mind-speller brain-computer interfaces: A walk through the recent developments with special focus on classification algorithms," *Clin. EEG Neurosci.*, vol. 51, no. 1, pp. 19–33, Jan. 2020.

[34] H. Serby, E. Yom-Tov, and G. F. Inbar, "An improved P300-based brain-computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 1, pp. 89–98, Mar. 2005.

[35] E. W. Sellers and E. Donchin, "A P300-based brain–computer interface: Initial tests by ALS patients," *Clin. Neurophysiol.*, vol. 117, no. 3, pp. 538–548, Mar. 2006.

[36] F. Piccione, F. Giorgi, P. Tonin, K. Priftis, S. Giove, S. Silvoni, G. Palmas, and F. Beverina, "P300-based brain computer interface: Reliability and performance in healthy and paralysed participants," *Clin. Neurophysiol.*, vol. 117, no. 3, pp. 531–537, Mar. 2006.

[37] D. J. Krusienski, E. W. Sellers, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, "Toward enhanced P300 speller performance," *J. Neurosci. Methods*, vol. 167, no. 1, pp. 15–21, Jan. 2008.

[38] S. Dey, A. K. Singh, D. K. Prasad, and K. D. Mcdonald-Maier, "SoCodeCNN: Program source code for visual CNN classification using computer vision methodology," *IEEE Access*, vol. 7, pp. 157158–157172, 2019.
[39] L. Zou, J. Zheng, C. Miao, M. J. Mckeown, and Z. J. Wang, "3D CNN based automatic diagnosis of attention deficit hyperactivity disorder using functional and structural MRI," *IEEE Access*, vol. 5, pp. 23626–23636, 2017.
[40] S. Hochreiter and J. J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 80–1735, 1997.
[41] A. Graves, "Supervised sequence labelling," in *Supervised Sequence Labelling With Recurrent Neural Networks*. Berlin, Germany: Springer, 2012, pp. 5–13.
[42] E. Santamaría-Vázquez, V. Martínez-Cagigal, F. Vaquerizo-Villar, and R. Hornero, "EEG-inception: A novel deep convolutional neural network for assistive ERP-based brain-computer interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 12, pp. 2773–2782, Dec. 2020.

RAJEEV SAHAY (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from The University of Utah, Salt Lake City, UT, USA, in 2018, and the M.S. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2021, where he is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering. Since Summer 2020, he has been an Intern at the Department of Statistical Sciences, Sandia National Laboratories. In Summer 2021, he was an Intern with the Massachusetts Institute of Technology Lincoln Laboratory, Advanced SATCOM Group. His current research interests include adversarial machine learning, uncertainty quantification, biomedical signal processing, and wireless communication networks. He is a member of the Phi Eta Sigma National Honor Society and the Tau Beta Pi Engineering Honor Society. While at The University of Utah, he was a recipient of the Allan & Judith Jennings Endowed Scholarship, the Parker-Hannifan Endowed Scholarship, and the Simon Ramo Endowed Scholarship. He also received two Undergraduate Research Opportunities Program (UROP) grants.

CHRISTOPHER G. BRINTON (Senior Member, IEEE) received the B.S. degree (valedictorian) in electrical engineering from The College of New Jersey, in 2011, and the M.S. degree and the Ph.D. degree (Hons.) in electrical engineering from Princeton University, in 2013 and 2016, respectively. He is currently an Assistant Professor with the School of Electrical and Computer Engineering, Purdue University. Prior to that, he was the Associate Director of the EDGE Laboratory and a Lecturer of electrical engineering at Princeton University. His research interests include the intersection of machine learning and networked systems, specifically in distributed machine learning, behavioral signal processing, and data-driven network optimization. He is the Co-Founder of Zoomi Inc., a big data startup company that has provided learning optimization to more than one million users worldwide. He holds a U.S. patent in machine learning for individualized learning. His book *The Power of Networks:* 6 *Principles That Connect our Lives and associated Massive Open Online Courses (MOOCs)* have reached over 400 000 students to date. Since joining Purdue ECE in Fall 2019, he has won the 2019 Purdue Seed for Success Award, the 2020 Purdue ECE Outstanding Faculty Mentor Award, and the 2020 Ruth and Joel Spira Outstanding Teacher Award.