



OPEN

Convolutional spiking neural networks for intent detection based on anticipatory brain potentials using electroencephalogram

Nathan Lutes¹, Venkata Sriram Siddhardh Nadendla² & K. Krishnamurthy^{1✉}

Spiking neural networks (SNNs) are receiving increased attention because they mimic synaptic connections in biological systems and produce spike trains, which can be approximated by binary values for computational efficiency. Recently, the addition of convolutional layers to combine the feature extraction power of convolutional networks with the computational efficiency of SNNs has been introduced. **This paper studies the feasibility of using a convolutional spiking neural network (CSNN) to detect anticipatory slow cortical potentials (SCPs) related to braking intention in human participants using an electroencephalogram (EEG). Data was collected during an experiment wherein participants operated a remote-controlled vehicle on a testbed designed to simulate an urban environment.** Participants were alerted to an incoming braking event via an audio countdown to elicit anticipatory potentials that were measured using an EEG. **The CSNN's performance was compared to a standard CNN, EEGNet and three graph neural networks via 10-fold cross-validation. The CSNN outperformed all the other neural networks, and had a predictive accuracy of 99.06% with a true positive rate of 98.50%, a true negative rate of 99.20% and an F1-score of 0.98.** Performance of the CSNN was comparable to the CNN in an ablation study using a subset of EEG channels that localized SCPs. Classification performance of the CSNN degraded only slightly when the floating-point EEG data were converted into spike trains via delta modulation to mimic synaptic connections.

Significant advancements in computing hardware, such as graphics processing units and field-programmable gate arrays, along with the availability of large datasets, has enabled researchers to develop highly effective neural networks in the last decade. However, training and utilizing these networks often involves a large amount of energy consumption, thus restricting the deployment of neural networks for data, and/or energy, limited settings: typically applications in dynamic/mobile environments. On the contrary, biology-inspired neural networks need only very few or even only one data point to perform at a competitive level compared to “traditional” neural networks (see page 54 in ref¹). Therefore, machine learning architectures more closely resembling biological neural networks are quickly gaining in popularity. One such example is the spiking neural network (SNN)^{2–4} which mimics biological neural networks through its layers composed of *spiking neurons*. These neurons more closely resemble the synaptic connections between neurons in biological neural networks through their emission of aperiodic spikes as opposed to floating point numbers in the case of the traditional artificial neuron. This sparse and discrete behavior of SNNs has been shown to reduce energy consumption by orders of magnitude when implemented on emerging neuromorphic hardware⁵. However, shallow SNNs can be insufficient to detect patterns that occur at random times/locations in tasks such as object detection/segmentation, similar to standard multi-layer perceptrons. This has inspired the development of hybrid convolutional and spiking neural networks, referred to as convolutional spiking neural networks (CSNNs)^{6–8}, which combine the convolutional layer's power of extracting spatio-temporal features with the energy efficiency of spiking neuron layers. In the past few years, CSNNs have received increased attention in diverse applications such as computer vision^{9,10},

¹Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA. ²Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409, USA. ✉email: kkrishna@mst.edu

speech recognition¹¹, hand-gesture recognition¹² and detection of Alzheimer's disease¹³ reinforcing their utility in deciphering complex and multi-dimensional data.

The main contribution of this study is to evaluate the use of CSNNs in *advanced driver-assist systems* (ADAS), specifically those approaches that utilize electroencephalograms (EEGs). ADAS can be summarized as a group of assistive technologies designed to decrease the cognitive load associated with the driving task by assisting with driving and/or parking decisions thus aiding the driver in safely operating their vehicle. This technology has been rapidly introduced in modern vehicles and has shown to greatly improve road safety and reduce traffic accidents¹⁴. EEG is a method of measuring and recording electrical potentials from across various points in the human brain, thus serving as a primary method of discerning a person's current cognitive activity. EEG-based applications are commonly explored in the field of brain-computer-interface (BCI)¹⁵, which has contributed in the development of machine learning models dedicated specifically to the analysis and interpretation of EEG signals, for example "EEGNet"¹⁶. The inclusion of EEG as an auxiliary input source effectively fuses the fields of ADAS and BCI and gives subsequently developed technologies the advantage of an accurate real-time measure of otherwise unknown aspects of the driver state^{17–20} and also allows for the prediction of a driver's intended action (e.g. braking)^{21–23} before it occurs. Literature has reported anticipatory potentials being observed as early as 130 ms²⁴ and 320 ± 200 ms²⁵ before action onset. The present study focuses on the latter advantage of EEG and seeks to train a CSNN as the predictive classifier to detect these anticipatory brain potentials and thus predict braking intention.

Although some initial studies have been made to demonstrate the effectiveness of shallow SNNs in typical BCI applications^{26–32}, the proliferation of other convolutional networks in the realm of BCI (e.g. EEGNet) and the reported success of deep learning methods in EEG decoding problems³³ implies that the inclusion of deep learning methods, such as the addition of convolutional layers, leads to a performance gain in classification tasks involving EEG data. Furthermore, the relative ease with which SNNs and their deep learning counterparts, CSNNs, can be mapped to emerging high-efficiency neuromorphic-computing hardware^{5,34–37} makes them ideally suited for deployment in mobile, energy-limited applications. The use of energy-efficient neuromorphic hardware becomes even more advantageous when implementing various learning methods for online continuous learning or one-shot learning³⁸ in energy-constrained applications.

To the author's knowledge, the potential of CSNNs for EEG-based ADAS has not yet been explored and will be a novel contribution. To achieve a fairer juxtaposition than directly comparing the CSNN's performance in this study to other methods in the literature, additional neural network models were trained on the same dataset to provide clearer context. These models include: i) a CNN of similar architecture; ii) EEGNet; and iii) three graph neural networks (GNNs). The CNN was chosen to be a direct comparison of the spiking architecture to a non-spiking architecture, the EEGNet was chosen as the "state of the art" benchmark model because of its previous history of generalizing better across different BCI paradigms and high performance achievement as compared to existing CNNs and traditional approaches¹⁶. Lastly, the inclusion of GNNs was motivated as an alternative to standard CNN networks because of their similar performance on adjacent EEG decoding tasks^{39–42}.

Related work

Previous studies on BCI-based driver intent detection present a gamut of technical approaches that mainly differ in the pre-processing strategies and various classifiers used. Popular approaches include linear and quadratic discriminant analysis methods. Teng et al.⁴³ combined regularized linear discriminant analysis with the sequential forward-floating search method of feature engineering, Kim et al.⁴⁴ combined EEG, tibialis anterior electromyography (EMG) and brake pedal signal, and Haufe et al.²⁴ compared EEG, EMG and brake pedal response to determine the input features that predict braking intention the fastest. Khaliliardali et al.^{25,45}, on the other hand, used quadratic discriminant analysis in conjunction with bandpass filtering the EEG inputs in a low frequency range of 0.1–1 Hz.

As competitors to discriminant analysis methods, other classification methods prevalent in the literature are shallow and deep neural networks. For example, Hernandez et al.⁴⁶ investigated support vector machines and convolutional neural networks (CNNs) to differentiate normal driving and braking intention EEG signals achieving a reported average accuracy of 71% and 72% for support vector machines and CNNs, respectively. Nguyen et al.⁴⁷ featured a multilayer perceptron neural network in a comparison of EEG band power-based and autoregressive-based feature selection methods, reporting a better accuracy of 91% with the autoregressive based method. Lee et al.⁴⁸ used recurrent convolutional neural networks (RCNNs) for an EEG braking intention decoding task, achieving an AUC score of 0.86. It is evident from the literature that a variety of methods have been used with mixed results. Although there are some examples of neural network usage, the use of SNNs for the braking intention EEG decoding problem is noticeably absent.

The EEG pattern studied here is the contingent negative variation (CNV), which is a type of slow cortical potential (SCP) that occurs prior to movement in the central region of the brain. The CNV, in particular, manifests when a subject is given a warning stimulus followed closely by an imperative stimulus, or stimulus requiring an action. It is featured in previous movement intention literature that also focused on driver braking intent detection^{21,25,45} and has a strong theoretical foundation on being a key anticipatory signal. Using CNV as a measure of how global and local temporal prediction affects expectancy implementation, it is seen that the signal has a larger amplitude when a given stimulus is expected to occur versus when it is unexpected⁴⁹. Furthermore, Mento⁵⁰ provided a comprehensive treatment of CNV as a marker for cognitive expectancy and motor preparation for the warning-imperative stimulus pair paradigm. However, the CNV is not the only EEG pattern used for intention detection in the literature. Event-related desynchronization (ERD) and Bereitschaftspotentials (readiness potentials) are both alternative indications of movement preparation. ERD is an EEG phenomenon occurring in the mu and beta frequency bands up to two seconds before movement is realized. It is marked by a decrease

in the spectral power of EEG within those bands that is not restored until after the movement is completed. On the other hand, Bereitschaftspotentials are slowly building neural signals which occur 1–2 s before movement onset, similar to CNV.

Previous research illustrating ERD as a movement preparation indicator include a study to find a suitable classifier for ERD using data from self-determined reaching movement experiments⁵¹, development of a novel algorithm for using ERDs to detect hand movement intention using adaptive wavelet transform⁵² and using ERDs as model inputs to reduce false positives in a motor imagery for rehabilitation application using a two-phase classifier design⁵³. Bereitschaftspotentials have also received attention with advancements made in their understanding and prediction. Nguyen et al.⁵⁴ conducted a study simultaneously integrating acquired EEG and fMRI through computational modeling and determined that reciprocal connections between the SMA and anterior mid-cingulate cortex are important to maintain sustained activity of the readiness potential before movement. Mirzabagherian et al.⁵⁵ developed two convolutional neural networks composed of temporal-spatial, separable and depth-wise layers and used these networks to detect a type of Bereitschaftspotentials known as movement-related cortical potentials (MRCPs), indicating five different hand movements performed by patients with cervical spinal cord injury. Gatti et al.⁵⁶ also studied harnessing MRCPs for movement speed and force intent detection. Mussini and Di Russo⁵⁷ investigated how anxiety can affect anticipatory brain functions by observing its effect on pre-stimulus ERP and the Bereitschaftspotential when performing tasks with and without feedback. Both ERD and Bereitschaftspotentials are popular and useful EEG related signals that could serve as an alternative to slow cortical potentials for movement intention related applications. To the author's knowledge, use of SNNs for predicting ERD or Bereitschaftspotentials is yet to be explored.

Results

Identification of braking intention signature in EEG signals

An example of the pre-processed EEG signals from 19 channels with data markers signifying the temporal locations of the audible countdown commands is shown in Fig. 1a. Each marker is denoted by its associated countdown number from 5 to 1 and ending with “STOP” when the stop command was given. The Cz grand average of the pre-processed data is shown in Fig. 1b along with scalp plots visualizing how the channel grand averages changed over time. The grand averages were calculated by averaging the Cz electrode signal from all participants and across all trials, similar to the procedure followed by Khaliliardali, Chavarriaga, Gheorghe and Millan²⁵. The following observations can be made from Fig. 1.

1. The negative EEG potential, termed contingent negative variation (CNV) potential, started after the “2” count marker and reached the maximum negative value between the “1” count marker and “Stop” command.
2. The negativity rate sharply increased at the “1” count marker and the potential rate became sharply positive midway between the “1” count marker and “Stop” command.
3. Anticipatory potentials were clearly observed before the actual braking action.
4. The more negative potentials were spatially localized in the centro-medial electrodes.

The results obtained are consistent with other past studies on CNV^{21,25,45}.

Classification performance—case 1: 32-bit single precision floating-point EEG input data

The final pre-processed dataset used for training the models included 10702 data segments collected from 15 participants in an experiment described in the Experimental Design section and preprocessed according to the procedure outlined in the Data Preprocessing section: 8573 data segments labelled as class ‘0’ (no intention signal) and 2129 data segments labelled as class ‘1’ (intention signal). The models were evaluated by means of tenfold stratified cross-validation where the training and testing partitions of each fold maintained the original class distribution. The data segments in each fold were shuffled before training and testing. To mitigate the skewed

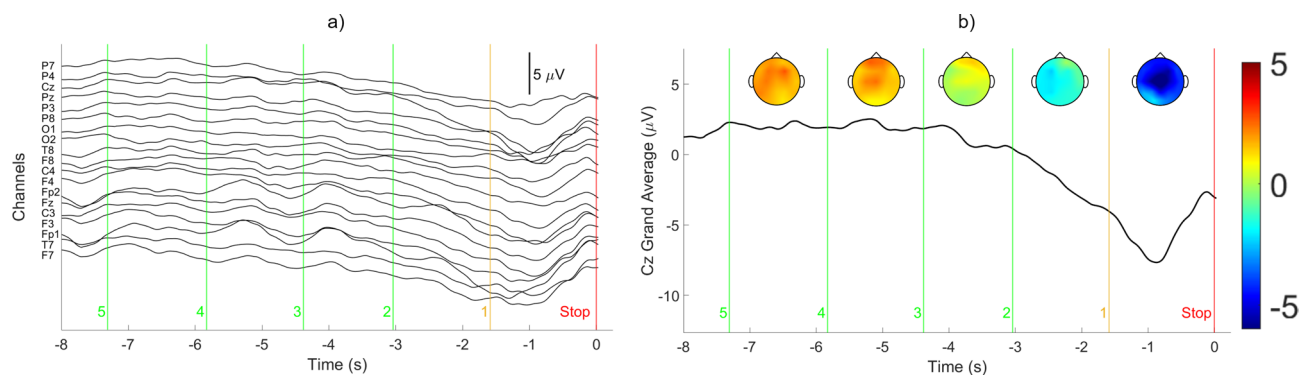


Figure 1. Pre-processed EEG signals. **(a)** Channel potentials with associated countdown and “Stop” command markers and scale. **(b)** Cz grand average signal with scalp maps representing the grand average at the midpoint between two neighboring markers and color bar on the right displaying the potential in μV .

distribution of the classes, wherein approximately 80% of the dataset was negative class and approximately 20% was positive class, the samples belonging to each class were given a weight used in the training loss function. The class weights were calculated as:

$$w_{c,i} = \frac{N_D}{2N_{c,i}} \tag{1}$$

where N_D and $N_{c,i}$ are the total number of data points in the training partition and the number of class i data points in the training partition, respectively.

The convolutional layers in the CSNN and CNN contained a kernel size of 5×5 , a stride of 1, a padding of 0, and 12 and 64 filters for the first and second convolutional layers, respectively. All max pooling layers of both architectures used a 2×2 pooling region with a stride of 2. The sigmoid surrogate function smoothness value, k , for the surrogate backpropagation of the CSNN was chosen as 0.25. The GCN and GCS graph convolutional layers had output sizes of 115, 28, 14 and 3. The GIN had graph convolutional layers with output sizes of 924, 462 and 231. The multi-layer perceptrons used in the GIN contained 5 hidden layers, each with 256 hidden neurons. All GNN architectures utilized in this work were implemented using the Spektral Python package⁵⁸. The CSNN and CNN models were implemented using PyTorch⁵⁹. EEGNet was implemented using tensorflow⁶⁰ with default values with the exception of the length of the 2D convolutional kernel, which was set to 250.

All architectures were trained for up to 1000 epochs per fold with a batch size of 8, and with early termination if the loss did not improve for 50 subsequent epochs. The architectures had the same respective weight initialization for each fold. The CSNN Leaky-Integrate-and-Fire (LIF) layer had a memory decay rate of 0.5, a spiking threshold of 0.5 and used 25 input steps. All models used the 'ADAM' optimizer with learning rate $\gamma = 5e - 4$, running average coefficients $\beta_1 = 0.9$ and $\beta_2 = 0.999$, stability parameter $\epsilon = 1e - 8$, and binary or categorical (for EEGNet) cross entropy as the loss function.

Table 1 shows the mean and standard deviation for predictive accuracy (Acc), true positive rate (TPR), true negative rate (TNR), F1-score, number of epochs trained, and total training time for each model. The table also shows each model's inference time for a single data point. The p -values from two-sample t-tests comparing the CSNN with the other models are shown for each classification metric. No attempt was made to optimize the learning parameters of each network. The results indicate that the CSNN outperformed all the other models in every classification metric category albeit closely followed by EEGNet. The CSNN also showed small standard deviations showcasing the consistency in results across folds. The GNNs exhibited a tendency to get stuck in local minima, evidenced by large standard deviations in all the performance metrics. However, the CSNN had the largest average training time and largest inference time out of all of the models.

Classification performance—case 2: ablation study with 32-bit single precision floating-point EEG input data from five channels

Laboratory experiments offer the privilege of using state-of-the-art equipment that is typically unhindered in its data collection methods. However, real-world scenarios may present unique challenges where a full 20 channel EEG headset is not possible or practical. To study how a reduced number of available channels would affect the performance of the classifier, a five-channel analysis was performed. The Cz channel and four surrounding channels (Pz, C3, C4 and Fz) were considered for the ablation study. Table 2 shows results for the classification performance with a reduced number of channels.

Classification performance—case 3: delta-modulated spike train input data

The effect of processing the input EEG data from all 19 channels into spike train data before passing to the CSNN network was also studied, namely its impact on classification performance of the network. The filtered and segmented EEG data, normalized to lie in the range of 0 to 1 inclusively, was transformed into a 19 channel array of spike train data by monitoring the change in value of successive data points in each channel. If the value change was greater than a threshold value, a spike '1' was recorded. If not, a '0' was recorded. The result was a binary array of the same dimensions as the floating-point input. The threshold value was varied from 0.05 to 1 which resulted in spike trains of differing densities. The CSNN model used had the same parameters as in Case 1.

Model	Accuracy (%)		TPR (%)		TNR (%)		F1-score		Epochs	Train Time (s)	Inference
	Mean (SD)	p -value	Mean (SD)	p -value	Mean (SD)	p -value	Mean (SD)	p -value	Mean (SD)	Mean (SD)	Time (s)
CSNN	99.06 (0.32)	N/A^a	98.50 (1.05)	N/A^a	99.20 (0.28)	N/A^a	0.98 (0.01)	N/A^a	270 (87)	17756 (5680)	2.533
CNN	79.46 (30.67)	0.088	79.29 (39.66)	0.18	79.50 (39.75)	0.172	0.61 (0.41)	(0.043)	130 (63)	2348 (13)	0.002
EEGNet	97.64 (0.36)	< 0.001	93.61 (2.03)	< 0.001	98.64 (0.38)	0.01	0.94 (0.01)	< 0.001	50 (0)	7238 (0.1)	0.767
GCS	57.09 (28.31)	0.002	36.93 (46.00)	0.003	62.10 (46.75)	0.041	0.13 (0.16)	< 0.001	110 (10)	350 (34)	0.293
GCN	62.03 (27.51)	0.003	44.83 (45.74)	0.007	66.31 (43.79)	0.051	0.21 (0.23)	< 0.001	96 (15)	281 (45)	0.024
GIN	54.66 (17.08)	< 0.001	45.97 (29.57)	< 0.001	56.82 (28.56)	0.002	0.25 (0.10)	< 0.001	82 (29)	926 (328)	0.759

Table 1. Classification performance with floating-point EEG input data (best performance in each classification measure highlighted in bold font).. ^a Proposed model.

Model	Accuracy (%)		TPR (%)		TNR (%)		F1-score		Epochs	Train time (s)	Inference
	Mean (SD)	p-value	Mean (SD)	p-value	Mean (SD)	p-value	Mean (SD)	p-value	Mean (SD)	Mean (SD)	Time (s)
CSNN	99.07 (0.27)	N/A ^a	98.35 (0.90)	N/A ^a	99.24 (0.22)	N/A ^a	0.98 (0.01)	N/A ^a	338 (77)	20772 (4692)	2.494
CNN	99.11 (0.32)	0.618	98.97 (0.91)	0.128	99.15 (0.29)	0.270	0.98 (0.01)	0.582	999 (0)	2365 (12)	0.002
EEGNet	92.22 (0.64)	< 0.001	64.00 (4.10)	< 0.001	99.22 (0.36)	0.856	0.77 (0.03)	< 0.001	50 (0)	7239 (0.1)	0.121
GCS	65.67 (25.34)	0.003	45.0 (41.53)	0.004	71.07 (39.87)	0.063	0.29 (0.26)	< 0.001	118 (36)	7 (1)	0.144
GCN	61.78 (25.57)	0.001	45.0 (43.78)	0.004	62.5 (35.84)	0.01	0.29 (0.30)	< 0.001	165 (99)	7 (2)	0.024
GIN	52.67 (26.05)	< 0.001	60 (45.95)	0.027	50.89 (41.64)	0.005	0.28 (0.22)	< 0.001	94 (38.56)	47 (10)	0.641

Table 2. Five-channel ablation study with floating-point EEG input data (best performance in each classification measure highlighted in bold font).. ^aProposed model.

Table 3 shows the tenfold classification results for Case 3. The results indicate that good predictive performance can still be achieved when the floating-point EEG data were converted into spike trains prior to being input to the network, if a suitable threshold value was selected. Spike train conversion thresholds that are too small or too large constrain the abilities of the CSNN to learn effectively, most likely because features important to the correct classification of data were being obscured. If the threshold was too small, then the spike train becomes saturated, making it difficult for the network to determine the important features. If the threshold was too large, then important features may not be captured at all.

The best results were obtained with a threshold of 0.5. The sensitivity of the classification results to the threshold value was studied using two-sample t-tests comparing the classification measures corresponding to the 0.5 threshold with the measures corresponding to the other thresholds. The t-test results shown in Table 3 indicate significant performance degradation above a threshold of 0.625 and below 0.375. This implies that a range of threshold values could be used to obtain statistically similar results, offering flexibility in the threshold selection. The performance of the CSNN using a threshold value of 0.5 was comparable to the EEGNet results when trained on the floating-point data. A two-sample t-test showed that TNR for the CSNN was statistically better (*p*-value was 0.041), but the accuracy and F1-score metrics were statistically similar (*p*-values were 0.700 and 0.537, respectively). On the other hand, the TPR for EEGNet was significantly better (*p*-value was 0.028).

Discussion

The results presented here show that the CSNN can be used as a classifier for detecting features in EEG data that predict braking intention, which occurs before the actual physical activity. To benchmark the CSNN performance, results were compared to a standard CNN, EEGNet and three GNN models using a 10-fold cross-validation scheme with the CSNN achieving the highest performance and with more consistency. The *p*-values from two-sample t-tests in Table 1 show a significantly higher performance of the CSNN over the GNNs in almost every metric category (except TNR of the GCN network, where the *p*-value was slightly above 0.05). This result is not surprising when the means of the metrics are compared. This fact is in stark contrast to the *p*-values of the CNN and EEGNet, where the CNN has noticeably lower mean values than the CSNN and is statistically similar; however, EEGNet is the closest of any of the other models and is statistically different. This can be explained by the fact that the CNN had enough folds containing results that were nearly identical to the CSNN performance, but also had a couple of folds with poor performance that biased the grand averages. As a result, the overall model performance was not significantly different than the CSNN. On the other hand, although EEGNet performance was competitive and consistent, it did not quite match the CSNN performance on any fold. Therefore, the *p*-values indicated statistically significant higher performance of the CSNN compared with EEGNet. The authors hypothesize that a possible explanation for the CSNN's success is that converting the floating-point numbers to spike trains allows it to filter more efficiently, passing the most important feature maps to the next layer.

Threshold	Accuracy (%)		TPR (%)		TNR (%)		F1-score		Epochs
	Mean (SD)	p-value	Mean (SD)	p-value	Mean (SD)	p-value	Mean (SD)	p-value	Mean (SD)
0.05	90.43 (0.339)	< 0.001	69.74 (2.60)	< 0.001	95.57 (0.53)	< 0.001	0.74 (0.01)	< 0.001	286 (100)
0.25	94.57 (0.65)	< 0.001	82.61 (2.29)	< 0.001	97.54 (0.71)	< 0.001	0.86 (0.02)	< 0.001	208 (38)
0.375	96.89 (0.49)	0.021	89.38 (2.54)	0.035	98.75 (0.37)	0.098	0.92 (0.01)	0.020	243 (51)
0.5	97.56 (0.43)	N/A ^a	91.64 (1.98)	N/A ^a	99.03 (0.23)	N/A ^a	0.94 (0.01)	N/A ^a	270 (45)
0.625	97.27 (0.55)	0.150	91.26 (2.06)	0.646	98.76 (0.47)	0.142	0.93 (0.02)	0.165	184 (40)
0.75	95.80 (0.62)	< 0.001	85.95 (3.19)	< 0.001	98.25 (0.44)	< 0.001	0.89 (0.02)	< 0.001	261 (77)
1	88.83 (1.13)	< 0.001	63.77 (4.79)	< 0.001	95.05 (0.83)	< 0.001	0.69 (0.04)	< 0.001	208 (49)

Table 3. Classification performance with delta modulated spike train input data (best performance in each classification measure highlighted in bold font). ^aThreshold tested.

Despite the CSNN's success in classification, it had the longest average training time and largest inference time by a large margin. While computational efficiency could be improved by deploying CSNNs on neuromorphic hardware, those gains were not realized in this study where only a von Neumann computer was utilized. Neuromorphic chips, such as the Intel Loihi chip, have been shown to produce faster training times than von Neumann processors and deep learning accelerators⁵.

Results of the five-channel ablation study indicate that a few strategically chosen channels may be sufficient with the CSNN's classification performance being nearly identical to that attained with all 19 channels. It can also be seen that the CNN performance increased considerably compared to the results shown in Table 1 with all 19 channels. Although the mean performance of the CNN was substantially lower when all 19 channels were used, the *p*-values in Table 1 indicate that the results were not significantly different from the CSNN. As noted previously, a couple of folds with poor performance biased the CNN grand average. Therefore, the boost in CNN performance to the level of CSNN in the ablation study is not altogether surprising and could simply have resulted from more consistent performance across folds when using only five channels.

The spike train conversion results shown in Table 3 have significant implications because converting the floating-point EEG data to spike trains at the outset could allow for additional energy savings, thereby taking full advantage of any neuromorphic computing hardware used to implement the CSNN for large, complex datasets and/or in real-time applications. Findings from this study can be exploited in future work to implement the CSNN on a neuromorphic platform to study the actual energy efficiency and feasibility for on-line learning in real-time applications.

The Cz grand average calculation and the analysis itself is based on the assumption of zero phase shift of the CNV pattern relative to the external stimuli. Lew et al.²¹ and Khaliliardali et al.²⁵ discuss in great detail SCPs, of which CNVs are a subset. In these studies, the SCP was determined to begin as early as 1.5 s before the onset of movement and therefore the crucial aspects of the CNV should be contained within the data between the "1" count marker and the stop marker. Variations of the exact timing of the pattern will occur between trials but as mentioned above, the grand average pattern obtained in this study are consistent with the results reported in the literature. Also, it is well known that EEG has poor spatial resolution. For this reason, the CNV pattern in Fig. 1 appears to occur across the entire brain instead of the central area as expected. Results with a higher number of channels than the 19 channels considered in this study have reported greater regional localization²⁵. Furthermore, ablation study results presented in Table 2 confirm that the CNV is occurring in the central area.

The EEG data was collected from 15 participants with the dataset containing a total of 3244 trials that were cleaned and segmented into 10,702 data segments. Although the number of participants in this study is relatively small, it is on par with other studies in the literature. The classification experiment conducted in-house involved using a simulated-realistic testbed with the participant operating a remote-controlled vehicle using a live video feed under ideal conditions. Although this study focused on a narrow frequency band of 0.1–1 Hz as suggested by Garipelli et al.²² and similar to other studies^{21,25,45}, a much wider 0.1–45 Hz bandpass filter has also been investigated^{44,46,49}. A possible extension to this work could examine the performance of the CSNN using the full neural dynamics. It would also be interesting to study the performance of the CSNN with participants operating under cognitive stress, for example, when under fatigue or in the presence of distractions or to study the abilities of the CSNN in other EEG-BCI applications such as P300, motor imagery, motor-related cortical potentials and steady-state evoked potentials. Exploring the use of CSNNs for movement intention detection using Bereitschaftspotentials, which are well-known as an indicator of movement preparation within the brain, much like CNVs, is also compelling. Their use in braking intention, or other driving related task intention, using CSNNs would be another interesting research direction.

Methods

Convolutional spiking neural networks

CSNNs are deep networks comprised of standard convolutional layers that extract feature maps from the input data before passing these feature maps to subsequent spiking layers. These combination layers are referred to here as "convolutional-spiking" layers. For classification, the output layer is composed of a fully connected layer with linear activation function followed by a spiking layer. Figure 2a shows a schematic of the specific CSNN used in this study. It is comprised of two convolutional-spiking layers followed by the output layer. Each convolutional-spiking layer is comprised of a two-dimensional convolution layer followed by a two-dimensional max pooling layer and ending with an output spiking layer. The spiking layer in each convolutional-spiking layer is composed of a tensor of LIF neurons having the same shape as the shape of the input to the layer. In the output layer, the fully connected layer and the subsequent output LIF spiking layer both have two neurons for the two classes in the EEG dataset. The predicted output of the CSNN was determined by counting the number of spikes output by these two neurons and setting the predicted label to the class represented by the neuron which produced the most spikes. A tie would result in a predicted class of '0'.

Due to the non-differentiable nature of the output of spiking neurons, training SNNs is difficult and requires special approaches beyond simply using standard backpropagation. If the spiking behavior of a neuron is represented as:

$$S[t] = \Theta(U_{mem}[t] - \theta) \quad (2)$$

where $\Theta(\cdot)$ is the heavy-side function, $U_{mem}[t]$ is the membrane potential of the neuron and θ is the spiking threshold, the derivative of (2) with respect to U_{mem} is the dirac delta function:

$$\frac{\partial S}{\partial U_{mem}[t]} = \delta(U_{mem}[t] - \theta) \in \{0, \infty\} \quad (3)$$

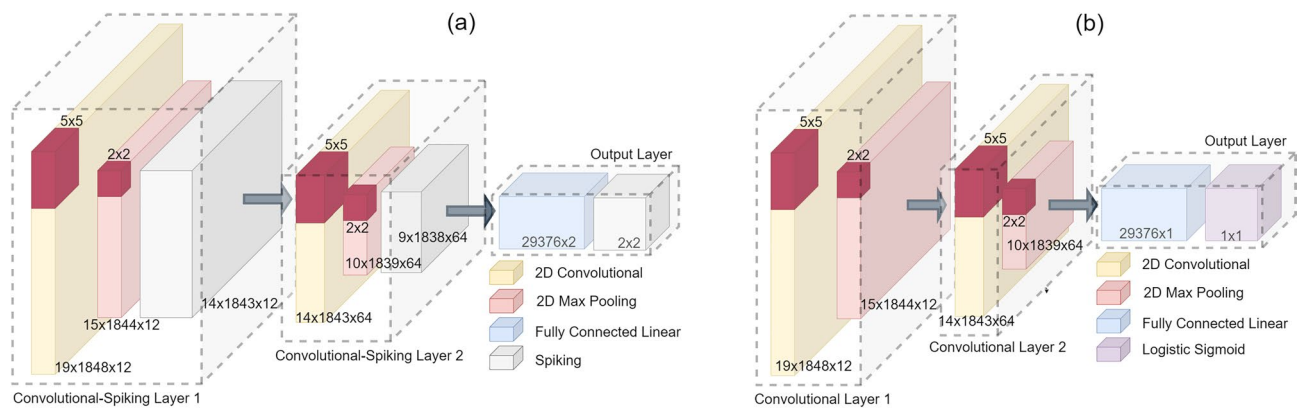


Figure 2. Schematic of neural network architectures. (a) CSNN. (b) CNN.

which is defined as zero for all time except where $U_{mem} = \theta$ at which it is infinity. This leads to the “dead neuron” problem for training using backpropagation. To mitigate this, the surrogate gradient approach is employed wherein during the “backward-pass”, when the gradient of the loss function due to the network parameters is being computed, the heavy-side function is approximated using a sigmoidal function, thereby creating a readily differential function. The exact function used as the surrogate in this paper is described as:

$$\tilde{S} = \frac{U_{mem}[t] - \theta}{1 + k|U_{mem}[t] - \theta|} \quad (4)$$

where k is known as the ‘slope’ and determines the smoothness of the surrogate function. The derivative of (4) is then obtained as:

$$\frac{\partial \tilde{S}}{\partial U_{mem}[t]} = \frac{1}{(k|U_{mem}[t] - \theta| + 1)^2} \quad (5)$$

From (5) it can be seen that as k increases, (5) converges to (3). For a more detailed explanation of SNNs and their training, the reader is referred to⁶¹.

Convolutional neural network

For the sake of a direct spiking vs. non-spiking comparison, a CNN composed of a largely identical architecture as that of the CSNN is considered to fully quantify any differences in performance that may arise by the addition of the spiking layers. As shown in Fig. 2b, the CNN has two convolutional layers, each including a max pooling layer, followed by a fully connected linear layer and ending with a logistic sigmoid output layer for class prediction.

EEGNet

EEGNet is a single CNN architecture designed for classification tasks across multiple EEG-based BCI domains (P300 visual-evoked potentials, error-related negativity responses, movement-related cortical potentials and sensory motor rhythms)¹⁶. It consists of two blocks of convolutional layers followed by a dense layer with finally a softmax layer. It is compact in terms of the number of model parameters (see Fig. 3). In the first block, two convolutional operations are done in sequence. This block starts with a temporal convolution to learn frequency filters followed by “depthwise” convolution to learn frequency-specific spatial filters. The second block also includes two convolutional operations. The first is another “depthwise” convolution to individually learn the temporal feature map and the second is “pointwise” convolution to optimally combine the feature maps. These two convolutions are combined into one layer, termed “Separable 2D Convolution”. The output of the second block of layers is then flattened and passed to the dense and softmax layer for generation of the predicted class.

Graph neural networks

GNNs are a specialized version of neural networks designed to operate on graph data. A graph is a grouping of data with defined internal relationships (edges) between objects (nodes) where these relationships may or may not be euclidean in nature. Mathematically, a graph is typically represented as: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ where \mathcal{V} represents a finite set of nodes having length $|\mathcal{V}| = N$, \mathcal{E} is a set of edges between the nodes, and $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix containing the edge weights. Graph data is input to a GNN as a matrix of node feature vectors: $X \in \mathbb{R}^{N \times n}$, where N is the number of nodes and n is the number of node features, along with its adjacency matrix, A , and sometimes a set of edge features, E . Some graph operations also include the diagonal degree matrix D where $D_{ii} = \sum_j A_{ij}$. For more details on graph theory and a detailed survey providing a comprehensive overview of GNNs, the reader is referred to^{62,63}. Because of the spatial relationship between electrodes, EEG data can naturally be represented as graphs with each data input sharing the same node and edge structure, thereby differing only in node data. Expressing the data as graphs allows for the non-euclidean spatial relationships to be exploited as extra information available to the classifier. For this reason, EEG-BCI applications have used GNNs previously, to notable effect.

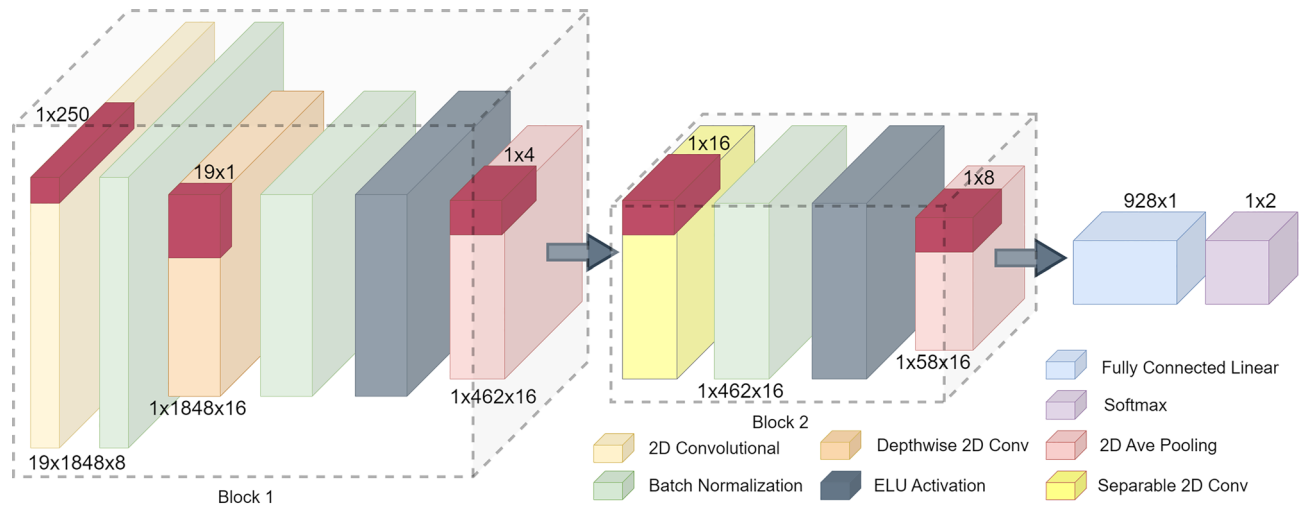


Figure 3. Schematic diagram of EEGNet.

As shown in Fig. 4, the three GNN architectures used in this work differ only in their initial graph processing layers, whilst universally sharing the last three layers. Each network possessed a global attention sum pool graph aggregation layer which feeds into a fully-connected hidden layer with rectified linear unit (ReLU) activation. The final layer is a classification output layer consisting of one neuron with a logistic sigmoid activation function. The global attention sum pool layer computes

$$X' = \sum_{i=1}^N \alpha_i X_i, \quad \alpha = \text{softmax}(Xa) \quad (6)$$

where a is a trainable weight vector, X is the layer input tensor, N is the number of nodes in the input graph, and the softmax operation is applied over nodes instead of features.

As shown in Fig. 4a, the first architecture, GCNConv network⁶⁴, consists of four graph convolutional (GCN) layers. The GCN layers perform the operation:

$$Y = \hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} XW + b \quad (7)$$

where Y is the output of the layer, $\hat{A} = A + I$ is the adjacency matrix of the input graph plus the identity matrix of appropriate shape, $\hat{D} = \sum_j \hat{A}_{ij}$ is the degree matrix, X is the layer input tensor, W is the layer weights, and b is the layer matrix.

Figure 4b shows the second architecture, which is the GCSCConv network. This consists of four GCS layers, which are GCN layers with an added, trainable skip connection. The GCS layer operation is described by:

$$Y = D^{-1/2} A D^{-1/2} XW_1 + XW_2 + b \quad (8)$$

where Y is the output of the layer, D is the degree matrix, A is the adjacency matrix, X is the node feature matrix, W_1 and W_2 are the two sets of layer weights, and b is the layer bias.

The third architecture is the GIN architecture⁶⁵ and is shown in Figure 4c. This architecture contains three graph isometric network (GIN) layers where each layer performs the following operation for each node in the input matrix:

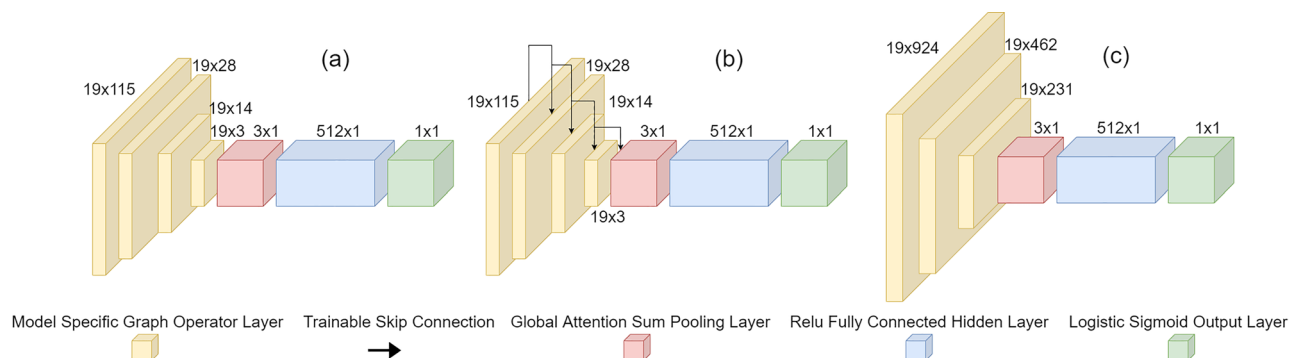


Figure 4. Schematic diagrams of GNNs. (a) GCNConv, (b) GCSCConv, (c) GINConv.

$$Y_i = MLP \left((1 + \epsilon) \cdot x_i + \sum_{j \in N(i)} x_j \right) \quad (9)$$

where $MLP(\cdot)$ is a multi-layer perceptron, ϵ is a learned parameter and x_i is the i th node of the input matrix.

The adjacency matrix, which would be common to all data samples, for all the three GNNs, was calculated as follows:

$$A = |P| - I \quad (10)$$

where $A \in \mathbf{R}^{N \times N}$ is the adjacency matrix, $|P| \in \mathbf{R}_+^{N \times N}$ is the absolute value of the Pearson's correlation coefficient of the dataset and $I^{N \times N}$ is the identity matrix.

Experimental design

The total number of participants in the experiment was 15 (13 male, 2 female). They consisted of Missouri University of Science and Technology students and professors, all in healthy condition. The participants had normal or corrected to normal vision and had normal hearing. The experiment received approval from the University of Missouri Institutional Review Board, and all experiments were performed in accordance with relevant guidelines and regulations. Written informed consent was obtained from all subjects and/or their legal guardian(s) prior to their participation. Further, written informed consent was obtained for publication of identifying information/images in an online open-access publication.

The objective of the experiment was to induce a predictable response in the participants such that any anticipatory signals that may occur can be reliably measured and recorded using an EEG. The experiment simulated a real-world driving environment wherein the participants operated an open-source remote-controlled robot called JetBot (built using Waveshare's Jetbot AI Kit and Nvidia's 4GB Jetson-Nano) on a novel testbed designed to simulate urban roadways (see picture inserted in Fig. 5). The testbed boundary was marked with standard masking tape and the track material was Delxo's anti-slip tape (with 80-grit granularity) to provide additional traction to the JetBot wheels. The participants navigated the JetBot in the testbed lanes using a Logitech G29 Driving Force racing wheel and pedal setup, while watching a live video feed cast to a computer monitor from an on-board camera. The JetBot was programmed to drive at a constant speed without the participant pressing the acceleration pedal, necessitating only the use of the steering wheel and brake pedal for full control. There were no other 'vehicles' or obstructions on the testbed lanes and participants were free to navigate anywhere within the testbed boundaries.

The EEG signals of the participants were recorded using a Neuroelectrics ENOBIO 20 EEG headset. The electrode setup used was the international 10–20 standard and the sampling frequency during the experiment was 500 Hz. Data was collected from 19 channels by applying a high conductivity Signagel saline gel on the electrodes to increase the quality of data capture. The data acquisition software used was the Neuroelectrics NIC2 software, which featured its own EEG signal quality monitor. The quality monitor assessed the EEG signal by computing a quality index (QI) that was dependent on: (i) line noise, which was defined as electrical noise originating from surrounding power lines; (ii) main noise, which was defined as the signal power of the standard EEG band; and (iii) offset, which was the mean value of the waveform. Specifically, QI was calculated as⁶⁶:

$$QI(t) = \tanh \left(\sqrt{\left(\frac{\zeta_L(t)}{W_{\zeta_L}} \right)^2 + \left(\frac{\zeta_m(t)}{W_{\zeta_m}} \right)^2 + \left(\frac{O(t)}{W_O} \right)^2} \right) \quad (11)$$

where $\zeta_L(t)$ and W_{ζ_L} denote the line noise and line noise normalizing weight ($= 100 \mu V$), respectively, $\zeta_m(t)$ and W_{ζ_m} denote the main noise and main noise normalizing weight ($= 250 \mu V$), respectively, and $O(t)$ and W_O denote the offset and offset normalizing weight ($= 280 mV$), respectively. The NIC2 software indicators used a color scheme to indicate different levels of QI. A green indicator meant that the signal had a QI between 0 and 0.5, an orange indicator meant a QI of between 0.5 and 0.8, and a red indicator meant a QI of 0.8 to 1. For this experiment, green indicators for all channels was the standard; however, brief periods of orange indicator were

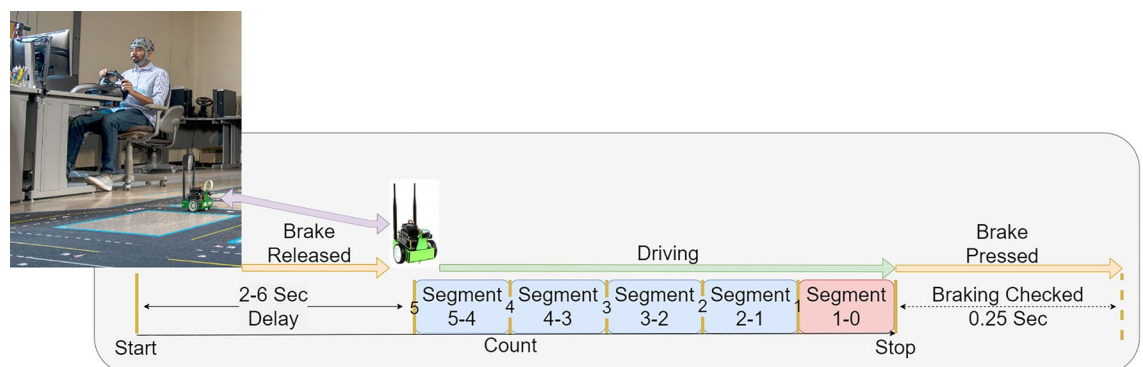


Figure 5. Experimental design illustration of a trial. Photo by Micheal Pierce/Missouri S & T.

considered acceptable. The data was filtered using a 60 Hz filter during capture to help reduce electrical noise and all channels were captured with reference to the Common Mode Sense channel (20th channel) which was fixed to the participant's right ear lobe.

The experimental design was based, in part, on the experiment conducted in²⁵ and is illustrated in Fig. 5. Each participant underwent eight sets of 30 trials each for a total of 240 trials, with short 5–10 min breaks between sets. Each trial consisted of a set of audible commands issued by MATLAB that included a “Start” command, upon hearing which the participant would release the brake allowing the JetBot to move, followed by a count-down from 5 to 1 and ending with a “Stop” command, when the participant would immediately stop the JetBot by pressing the brake. To ensure that the participants responded in a timely fashion, activity at the brake pedal was monitored and any trial where the brake pedal depression did not register a numerical reading higher than 0.05% of its total depression range within 0.25 seconds of the issuance of the “Stop” command were marked for removal. Trials where the participant braked too early were manually marked for removal as well. The EEG recording would begin concurrently with the issuance of the “Start” command, markers corresponding to the countdown numbers would be applied to the data concurrently with each audio count and the EEG recording would stop at detection of braking action or after the 0.25 s delay to check brake pedal depression.

Data preprocessing

The data was processed in several steps using the open source EEG toolkit EEGLAB⁶⁷ along with the TBT plugin⁶⁸ for only the trials that were not marked for removal during the experiments (see Fig. 6).

- Each trial was:
 - Spectrally filtered using a FIR bandpass filter from 0.1 Hz to 1 Hz as suggested in²².
 - Cleaned using EEGLAB's built-in automated cleaning function “Clean_RawData and ASR” (Artificial Subspace Reconstruction) to remove bad channels under the following criteria: If the channel: (i) was flat for more than 5 s; (ii) correlated at less than 0.8 to an estimate based on nearby channels; and (iii) contained more than four standard deviations of line noise relative to its signal. The ASR corrected bad data periods containing high-amplitude artifacts⁶⁹, such as eye blinks, and its maximum acceptable 0.5 s window standard deviation limit was set to a conservative 20 standard deviations. The values used in this step were the standard default values in EEGLAB and were also used as part of a pre-processing scheme in a previous study⁷⁰.
 - Segmented by slicing the data according to the markers corresponding to the countdown numbers or the “Stop” command. For example, as shown in Fig. 5, the first segment consisted of taking only the data that occurred between the “5” count marker and the “4” count marker. The segments from 5 to 1 were given a “0” label or were regarded as not containing an intention signal and the segment between “1” and “Stop” was labelled as “1” and was regarded as containing the signal of interest. Each segment was then baseline corrected by subtracting the mean value of the segment from every value in the segment.
- Each segment was then further cleaned using the TBT plugin to remove high amplitude noise. Channels were removed from the segment if either of the following two criteria was met for a data period duration of 10% of an segment or more⁷⁰: (i) if they exceeded $\pm 100 \mu\text{V}$ in magnitude, or (ii) if the joint probabilities (i.e., probabilities of activity) exceeded 3 standard deviations for local or global thresholds. If either criterion was met for less than 10% of an segment, then the offending data period was removed and subsequently interpolated. Segments with more than 50% of channels removed (i.e., 10 channels) were omitted entirely. Any removed channels were re-interpolated after the cleaning was finished for consistency in input dimension. Similar segment-by-segment cleaning strategies utilizing the TBT plugin have been employed in a previous study⁴⁹.
- Every data point was padded to a uniform length of 1848, the size of the largest dataset in a trial, by appending zeros to the end. Padding data should not significantly change the prediction results and is commonly done when using machine learning algorithms on datasets containing images having different sizes. Furthermore, padding the data should less significantly alter the dataset versus truncating data, which allows for the possibility of deletion of key features.
- Finally, each channel was normalized such that the data lay within a range of 0 to 1. The particular equation used was:

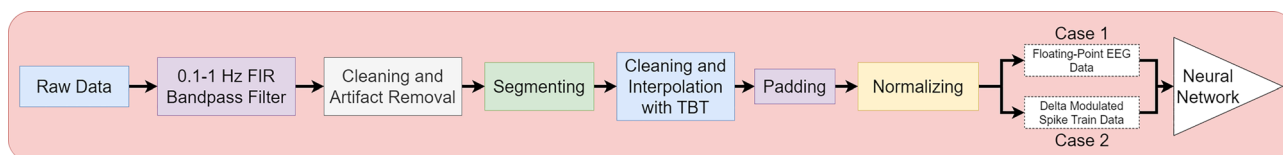


Figure 6. Data processing pipeline.

$$\bar{X}_i = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)} \quad (12)$$

where i is the i th data channel and X is the data vector corresponding to that channel, $\max(\cdot)$ and $\min(\cdot)$ are the maximum and minimum channel values per segment, respectively.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Code availability

The Python code written to obtain the results is publicly available at <https://github.com/sid-nadendla/BI-Spike-EEG>.

Received: 18 July 2023; Accepted: 11 April 2024

Published online: 17 April 2024

References

- Christensen, D. V. *et al.* 2022 roadmap on neuromorphic computing and engineering. *Neuromorph. Comput. Eng.* **2**, 022501. <https://doi.org/10.1088/2634-4386/ac4a83> (2022).
- Tan, C., Sarlija, M. & Kasabov, N. Spiking neural networks: Background, recent development and the neucube architecture. *Neural Process. Lett.* **52**, 1675–1701. <https://doi.org/10.1007/s11063-020-10322-8> (2020).
- Dora, S. & Kasabov, N. Spiking neural networks for computational intelligence: An overview. *Big Data Cognit. Comput.* **5**, 67. <https://doi.org/10.3390/bdcc5040067> (2021).
- Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T. & Maida, A. Deep learning in spiking neural networks. *Neural Netw.* **111**, 47–63. <https://doi.org/10.1016/j.neunet.2018.12.002> (2019).
- Davies, M. *et al.* Advancing neuromorphic computing with Loihi: A survey of results and outlook. *Proc. IEEE* **109**, 911–934. <https://doi.org/10.1109/jproc.2021.3067593> (2021).
- Lee, C., Srinivasan, G., Panda, P. & Roy, K. Deep spiking convolutional neural network trained with unsupervised spike-timing-dependent plasticity. *IEEE Trans. Cognit. Dev. Syst.* **11**, 384–394. <https://doi.org/10.1109/TCDS.2018.2833071> (2019).
- Vaila, R., Chiasson, J. & Saxena, V. Feature extraction using spiking convolutional neural networks. *Proc. Int. Conf. Neuromorph. Syst.* <https://doi.org/10.1145/3354265.3354279> (2019).
- Lee, C., Panda, P., Srinivasan, G. & Roy, K. Training deep spiking convolutional neural networks with STDP-based unsupervised pre-training followed by supervised fine-tuning. *Front. Neurosci.* <https://doi.org/10.3389/fnins.2018.00435> (2018).
- Barchid, S., Mennesson, J. & Djeraba, C. Deep spiking convolutional neural network for single object localization based on deep continuous local learning. In: *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*. <https://doi.org/10.1109/cbmi50038.2021.9461880> (2021).
- Matsugu, M., Mori, K., Ishii, M. & Mitarai, Y. Convolutional spiking neural network model for robust face detection. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.* Vol. 2. 660–664. <https://doi.org/10.1109/ICONIP.2002.1198140> (2002).
- Dong, M., Huang, X. & Xu, B. Unsupervised speech recognition through spike-timing-dependent plasticity in a convolutional spiking neural network. *PLOS ONE* <https://doi.org/10.1371/journal.pone.0204596> (2018).
- Xing, Y., Di Caterina, G. & Soraghan, J. A new spiking convolutional recurrent neural network (SCRNN) with applications to event-based hand gesture recognition. *Front. Neurosci.* <https://doi.org/10.3389/fnins.2020.590164> (2020).
- Turkson, R. E., Qu, H., Mawuli, C. B. & Eghan, M. J. Classification of Alzheimer's disease using deep convolutional spiking neural network. *Neural Process. Lett.* **53**, 2649–2663. <https://doi.org/10.1007/s11063-021-10514-w> (2021).
- Swief, A. & El-Habrouk, M. A survey of automotive driving assistance systems technologies. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. 1–12. <https://doi.org/10.1109/IDAP.2018.8620826> (2018).
- Rashid, M. *et al.* Current status, challenges, and possible solutions of eeg-based brain–computer interface: A comprehensive review. *Front. Neurobot.* <https://doi.org/10.3389/fnbot.2020.00025> (2020).
- Lawhern, V. J. *et al.* EEGNet: A compact convolutional neural network for EEG-based brain–computer interfaces. *J. Neural Eng.* **15**, 056013. <https://doi.org/10.1088/1741-2552/aace8c> (2018).
- Chuang, C.-H., Lai, P.-C., Ko, L.-W., Kuo, B.-C. & Lin, C.-T. Driver's cognitive state classification toward brain computer interface via using a generalized and supervised technology. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*. <https://doi.org/10.1109/ijcnn.2010.5596835> (2010).
- Lin, F.-C., Ko, L.-W., Chen, S.-A., Chen, C.-F. & Lin, C.-T. EEG-based cognitive state monitoring and prediction by using the self-constructing neural fuzzy system. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. <https://doi.org/10.1109/iscas.2010.5536955> (2010).
- Zheng, W.-L. & Lu, B.-L. A multimodal approach to estimating vigilance using EEG and forehead EOG. *J. Neural Eng.* **14**, 026017. <https://doi.org/10.1088/1741-2552/aa5a98> (2017).
- Hajinoroozi, M., Zhang, J. & Huang, Y. Prediction of fatigue-related driver performance from EEG data by deep Riemannian model. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. <https://doi.org/10.1109/embc.2017.8037774> (2017).
- Lew, E., Chavarriaga, R., Silvoni, S. & Millán, J. D. R. Detection of self-paced reaching movement intention from EEG signals. *Front. Neuroeng.* <https://doi.org/10.3389/fneng.2012.00013> (2012).
- Garipelli, G., Chavarriaga, R. & del R Millán, J. Single trial analysis of slow cortical potentials: A study on anticipation related potentials. *J. Neural Eng.* **10**, 036014. <https://doi.org/10.1088/1741-2560/10/3/036014> (2013).
- Lew, E. Y. L., Chavarriaga, R., Silvoni, S. & Millán, J. D. R. Single trial prediction of self-paced reaching directions from EEG signals. *Front. Neurosci.* <https://doi.org/10.3389/fnins.2014.00222> (2014).
- Haufe, S. *et al.* EEG potentials predict upcoming emergency brakings during simulated driving. *J. Neural Eng.* **8**, 056001. <https://doi.org/10.1088/1741-2560/8/5/056001> (2011).
- Khalilardali, Z., Chavarriaga, R., Gheorghe, L. A. & Millán, J. Action prediction based on anticipatory brain potentials during simulated driving. *J. Neural Eng.* **12**, 066006. <https://doi.org/10.1088/1741-2560/12/6/066006> (2015).
- Honzik, V. & Mouček, R. Spiking neural networks for classification of brain–computer interface and image data. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 3624–3629. <https://doi.org/10.1109/BIBM52615.2021.9669864> (2021).

27. Pals, M. *et al.* Demonstrating the viability of mapping deep learning based EEG decoders to spiking networks on low-powered neuromorphic chips. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*. 6102–6105 <https://doi.org/10.1109/EMBC46164.2021.9629621> (2021).
28. Luo, Y. *et al.* EEG-based emotion classification using spiking neural networks. *IEEE Access* **8**, 46007–46016. <https://doi.org/10.1109/ACCESS.2020.2978163> (2020).
29. Singanamalla, S. K. & Lin, C.-T. Spiking neural network for augmenting electroencephalographic data for brain computer interfaces. *Front. Neurosci.* <https://doi.org/10.3389/fnins.2021.651762> (2021).
30. Kumarasinghe, K., Kasabov, N. & Taylor, D. Brain-inspired spiking neural networks for decoding and understanding muscle activity and kinematics from electroencephalography signals during hand movements. *Sci. Rep.* <https://doi.org/10.1038/s41598-021-81805-4> (2021).
31. Yan, Z., Zhou, J. & Wong, W.-F. EEG classification with spiking neural network: Smaller, better, more energy efficient. *Smart Health* **24**, 100261. <https://doi.org/10.1016/j.smhl.2021.100261> (2022).
32. Shah, D., Wang, G. Y., Doborjeh, M., Doborjeh, Z. & Kasabov, N. Deep learning of EEG data in the neucube brain-inspired spiking neural network architecture for a better understanding of depression. *Neural Inform. Process.* https://doi.org/10.1007/978-3-030-36718-3_17 (2019).
33. Chen, X. *et al.* Toward open-world electroencephalogram decoding via deep learning: A comprehensive survey. *IEEE Signal Processing Magazine* **39**, 117–134. <https://doi.org/10.1109/MSP.2021.3134629> (2022).
34. Merolla, P. A. *et al.* A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **345**, 668–673. <https://doi.org/10.1126/science.1254642> (2014).
35. Jo, S. H. *et al.* Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* **10**, 1297–1301. <https://doi.org/10.1021/nl904092h> (2010).
36. Gopalakrishnan, R., Chua, Y. & Kumar, A. J. S. Hardware-friendly neural network architecture for neuromorphic computing. *arXiv Preprint arXiv:1906.08853* (2019).
37. Ivanov, D., Chezhegov, A., Kiselev, M., Grunin, A. & Larionov, D. Neuromorphic artificial intelligence systems. *Front. Neurosci.* <https://doi.org/10.3389/fnins.2022.959626> (2022).
38. Scherr, F., Stöckl, C. & Maass, W. One-shot learning with spiking neural networks. *BioRxiv*. <https://doi.org/10.1101/2020.06.17.156513> (2020).
39. Demir, A., Koike-Akino, T., Wang, Y., Haruna, M. & Erdogmus, D. EEG-GNN: Graph neural networks for classification of electroencephalogram (EEG) signals. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. <https://doi.org/10.1109/embc46164.2021.9630194> (2021).
40. Jia, S., Hou, Y., Shi, Y. & Li, Y. Attention-based graph ResNet for motor intent detection from raw EEG signals. <https://doi.org/10.48550/ARXIV.2007.13484> (2020).
41. Hou, Y. *et al.* GCNS-Net: A graph convolutional neural network approach for decoding time-resolved EEG motor imagery signals. <https://doi.org/10.48550/ARXIV.2006.08924> (2020).
42. Zhong, P., Wang, D. & Miao, C. EEG-based emotion recognition using regularized graph neural networks. <https://doi.org/10.48550/ARXIV.1907.07835> (2019).
43. Teng, T., Bi, L. & Liu, Y. EEG-based detection of driver emergency braking intention for brain-controlled vehicles. *IEEE Trans. Intell. Transport. Syst.* **19**, 1766–1773. <https://doi.org/10.1109/tits.2017.2740427> (2018).
44. Kim, I.-H., Kim, J.-W., Haufe, S. & Lee, S.-W. Detection of braking intention in diverse situations during simulated driving based on EEG feature combination. *J. Neural Eng.* **12**, 016001. <https://doi.org/10.1088/1741-2560/12/1/016001> (2014).
45. Khalilardali, Z. *et al.* Real-time detection of driver's movement intention in response to traffic lights. <https://doi.org/10.1101/443390> (2019).
46. Hernandez, L. G., Mozos, O. M., Ferrandez, J. M. & Antelis, J. M. EEG-based detection of braking intention under different car driving conditions. *Front. Neuroinform.* <https://doi.org/10.3389/fninf.2018.00029> (2018).
47. Nguyen, T.-H. & Chung, W.-Y. Detection of driver braking intention using EEG signals during simulated driving. *Sensors* **19**, 2863. <https://doi.org/10.3390/s19132863> (2019).
48. Lee, S.-M., Kim, J.-W. & Lee, S.-W. Detecting driver's braking intention using recurrent convolutional neural networks based EEG analysis. In *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*. <https://doi.org/10.1109/acpr.2017.86> (2017).
49. Duma, G. M. & Granzio, U. & Mento, G. Should I stay or should I go? How local-global implicit temporal expectancy shapes proactive motor control: An HDEEG study. *NeuroImage* <https://doi.org/10.1016/j.neuroimage.2020.117071> (2020).
50. Mento, G. The passive CNV: Carving out the contribution of task-related processes to expectancy. *Front. Hum. Neurosci.* <https://doi.org/10.3389/fnhum.2013.00827> (2013).
51. Planelles, D., Hortal, E., Costa, Á., Úbeda, A., Iáez, E. & Azorín, J. Evaluating classifiers to detect arm movement intention from EEG signals. *Sensors* **14**, 18172–18186. <https://www.mdpi.com/1424-8220/14/10/18172> (2014).
52. Chamanzar, A., Shabany, M., Malekmohammadi, A. & Mohammadinejad, S. Efficient hardware implementation of real-time low-power movement intention detector system using FFT and adaptive wavelet transform. *IEEE Trans. Biomed. Circuits Syst.* **11**, 585–596 (2017).
53. Song, M., Oh, S., Jeong, H., Kim, J. & Kim, J. A novel movement intention detection method for neurorehabilitation brain-computer interface system. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 1016–1021 (2018).
54. Nguyen, V., Breakspear, M. & Cunnington, R. Reciprocal Interactions of the SMA and Cingulate Cortex Sustain Premovement Activity for Voluntary Actions. *Journal Of Neuroscience*. **34**, 16397–16407 (2014). <https://www.jneurosci.org/content/34/49/16397>.
55. Mirzabagherian, H., Menhaj, M., Suratgar, A., Talebi, N., Abbasi Sardari, M. & Sajedin, A. Temporal-spatial convolutional residual network for decoding attempted movement related EEG signals of subjects with spinal cord injury. *Comput. Biol. Med.* **164**, 107159. <https://www.sciencedirect.com/science/article/pii/S0010482523006248> (2023).
56. Gatti, R., Atum, Y., Schiaffino, L., Jochumsen, M. & Manresa, J. Prediction of hand movement speed and force from single-trial EEG with convolutional neural networks. *BioRxiv*. <https://www.biorxiv.org/content/early/2019/11/07/492660> (2019).
57. Di Russo, F. & Mussini, E. Reduction of anticipatory brain activity in anxious people and regulatory effect of response-related feedback. *PsyArXiv*. <https://doi.org/10.31234/osf.io/ekh45> (2022).
58. Grattarola, D. & Alippi, C. Graph neural networks in tensorflow and keras with spektral (application notes). *IEEE Comput. Intell. Mag.* **16**, 99–106. <https://doi.org/10.1109/mci.2020.3039072> (2021).
59. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* (Wallach, H. *et al.* eds.). Vol. 32. 8024–8035 (Curran Associates, Inc., 2019).
60. Abadi, M. *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. <https://www.tensorflow.org/>. (2015).
61. Eshraghian, J. K. *et al.* Training spiking neural networks using lessons from deep learning. *arXiv Preprint arXiv:2109.12894* (2021).
62. Wilson, R. J. *Introduction to Graph Theory* (Prentice Hall, 2015).
63. Wu, Z. *et al.* A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **32**, 4–24. <https://doi.org/10.1109/tnnls.2020.2978386> (2021).
64. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. <https://doi.org/10.48550/ARXIV.1609.02907> (2016).
65. Xu, K., Hu, W., Leskovec, J. & Jegelka, S. How powerful are graph neural networks? <https://doi.org/10.48550/ARXIV.1810.00826> (2018).

66. Neuroelectrics. <https://www.neuroelectrics.com>. Accessed 18 May 2022 (2022).
67. Delorme, A. & Makeig, S. Eeglab: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J. Neurosci. Methods* **134**, 9–21. <https://doi.org/10.1016/j.jneumeth.2003.10.009> (2004).
68. Ben-Shachar, M. S. *mattansb/tbt: Channel you inner error*. <https://doi.org/10.5281/zenodo.5948294> (2022).
69. Mullen, T. R. *et al.* Real-time neuroimaging and cognitive monitoring using wearable dry EEG. *IEEE Trans. Bio-Med. Eng.* **62**, 2553–2567. <https://doi.org/10.1109/TBME.2015.2481482> (2015).
70. Bagdasarov, A. *et al.* Spatiotemporal dynamics of EEG microstates in four- to eight-year-old children: Age- and sex-related effects. <https://doi.org/10.31234/osf.io/x35uf> (2022).

Author contributions

N.L., V.S.S.N. and K.K. conceptualized and designed the method, performed analysis and prepared the manuscript. N.L. collected and processed the EEG data, and developed the software code to generate the results.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024