IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 January 2023, accepted 26 January 2023, date of publication 3 February 2023, date of current version 17 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3242643

RESEARCH ARTICLE

A Deep Neural Network-Based Spike Sorting With Improved Channel Selection and Artefact Removal

CHRISTIAN O. OKREGHE, (Member, IEEE), MAJID ZAMANI, (Member, IEEE), AND ANDREAS DEMOSTHENOUS, (Fellow, IEEE)
Department of Electronic and Electrical Engineering, University College London, WC1E 7JE London, U.K.

Corresponding author: Christian O. Okreghe (christian.okreghe.16@ucl.ac.uk)

ABSTRACT In order to implement highly efficient brain-machine interface (BMI) systems, high-channel count sensing is often used to record extracellular action potentials. However, the extracellular recordings are typically severely contaminated by artefacts and various noise sources, rendering the separation of multi-unit neural recordings an immensely challenging task. Removing artefact and noise from neural events can improve the spike sorting performance and classification accuracy. This paper presents a deep learning technique called deep spike detection (DSD) with a strong learning ability of high-dimensional vectors for neural channel selection and artefacts removal from the selected neural channel. The proposed method significantly improves spike detection compared to the conventional methods by sequentially diminishing the noise level and discarding the active artefacts in the recording channels. The simulated and experimental results show that there is considerably better performance when the extracellular raw recordings are cleaned prior to assigning individual spikes to the neurons that generated them. The DSD achieves an overall classification accuracy of 91.53% and outperformes Wave_clus by 3.38% on the simulated dataset with various noise levels and artefacts.

INDEX TERMS Artefact removal, channel selection, convolutional neural network (CNN), deep learning, deep spike detection (DSD), extracellular recordings, real-time sorting, spike sorting.

I. INTRODUCTION

Extracellular recordings have been widely used to monitor neuronal activity by implanting multi-electrodes in the cortex and capturing multi-dimensional neural data. The captured data are a mixture of neuronal activities. A processing step, known as *spike sorting*, is necessary to separate the multi-unit activities and assign the captured spikes to their originating neurons [1], [2]. Spike sorting is an invaluable research tool applied in brain-machine interface (BMI) research for studying and decoding neural signals arising from the implanted electrodes and understanding the mechanisms of the brain [3], [4]. It is also essential for deciphering intentions from brain activity in BMIs [5] and improving patient control of prostheses via devices [6], [7].

Due to the negative impacts of numerous noise sources [8] and captured artefacts on the recorded neural data, the performance of the spike sorting processing pipeline is often degraded [9]. In multiple channel recordings with microelectrode arrays (MEAs) [10], [11], a considerable number of channels record pure noise activities [12], whereas others record a substantial amount of noise with neural events and artefacts. Neurons far from the electrode tips are seen as artefacts, which represent spurious neural events from distant active neurons and have a significant negative impact on the recorded signal and consequently on spike sorting performance.

Spike sorting is more challenging than clustering in most other domains due to the above mentioned drawbacks.

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

It is difficult to extract the underlying and discriminative spike features for assigning individual spikes to their originating neurons from combinations of multiple complex signals and variations in waveform shapes. Attempts have previously been made to eliminate common-noise artefacts in recordings, [13], [14], [15], [16], [17], [18]. There is no simple noise model [19], [20].

In most applications, a clean signal is necessary to decipher the neural recordings accurately. Therefore, improving the performance and accuracy of the spike sorting algorithm requires refining the raw data. However, identifying channels containing active neurons and recognizing the authentic spike waveforms from a stream of spikes is very challenging [21].

Deep learning-based algorithms have proven to be highly effective in a variety of domains and applications, including brain decoding and classification. For example, in [22], the principal component analysis network (PCANet) deep learning technique is used to simplify the computationally intensive nature of traditional spike sorting algorithms. In [21], an approach based on deep learning was used to extract online invasive BMI application feature vectors.

In [23], MEA recordings of different action potentials were classified using a supervised deep learning model comprised of long short-term memory (LSTM) and a convolutional neural network (CNN). An efficient unsupervised form of deep learning-based spike sorting with manual labeling is presented in [24]. In [25], a CNN was employed to distinguish between categories of spikes. An L_2-normalized deep convolutional autoencoder (CAE) with spike sorting-aware loss was exploited for feature extraction for fully unsupervised and online spike sorting [26]. In [27], a supervised deep learning was used to distinguish spike events from non-neural events, together with deep learning for offline spike sorting [28].

Deep learning-based algorithms are utilized to denoise neural signals effectively. In [29], a deep encoder-decoder network was used to denoise two-dimensional (2D) images of neuronal physiology, resulting in a significant improvement in the distribution of the signal-to-noise ratio (SNR). CAE have been shown to be superior to the conventional band-pass technique in electroencephalogram (EEG) filtering [30]. In [31], CNN and LSTM networks outperformed conventional wavelet approaches in denoising electrocardiograms (ECG). In [32], an EEG denoising model for raw waveform-based data using a residual convolutional neural network (1D-ResCNN) was introduced and, in an end-to-end manner, utilized to map a noisy EEG signal to a clean EEG signal capable of enhancing the SNR significantly and producing cleaner waveforms. Yang et al. [33] removed ocular artefacts (OA) from EEG recordings utilizing a deep learning network (DLN). There is both an online and an offline component to the method. After removing OAs from the training data, a DLN was trained offline to reconstruct the EEG signals. During the online phase, the DLN model acted as a filter to automatically remove OAs from the corrupted EEG data. In [34], a MATLAB-based open-source toolbox that uses machine learning strategies based on neural networks to label and train models for detecting artefacts in invasive neuronal signals. The authors of [35] use CNN to select channels, after which multiple classes of motor imagery intentions are decoded. In [36], EEG decoding and visualization utilizing deep learning were exploited. It is expected that a convolutional neural network will outperform other methods in these well-established contexts for tasks involving neural data channel selection and artefact removal.

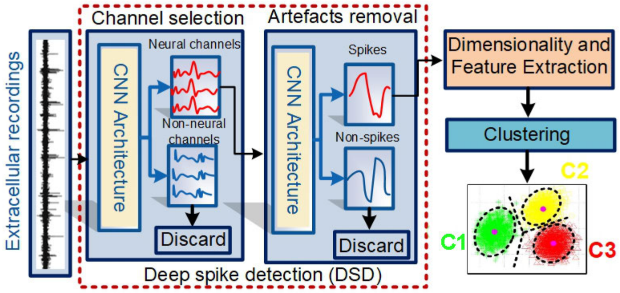


FIGURE 1. Diagram depicting different stages of the proposed spike sorting algorithm, including the deep spike detection (DSD). The end-to-end method consists of three main components: First, the DSD which consists of two phases for both the accurate selection of neural channels and artefact removal from the selected channels, yielding cleaned neural data for the feature extraction stage and subsequently classification of the extracted features utilizing K-means clustering.

This paper introduces a deep learning-based spike sorting termed deep spike detection (DSD) with improved spike detection accuracy. As shown in Figure 1, it embeds two convolutional neural networks into the conventional spike processing pipeline for the selection of the active neural channels and the removal of artefacts from the selected channels. The proposed method utilizes deep two-dimensional (2D) and one-dimensional (1D) CNNs. The DSD extracts discriminative spike and artefact features from the input channel to exclusively remove artefacts from extracellular recordings. A feature vector is constructed by concatenating a batch of waveforms with a length equal to the number of samples to produce the input vector. A batch of waveforms is classified as a neural channel by the channel selection if at least one waveform in the batch represents a spike. In contrast, it only qualifies a batch as an artefact if all waveforms inside the batch are artefacts. The artefact removal also determines whether a batch contains spikes and discards artefact events if so. By removing artefact channels and artefacts, the spike sorting performance increases since the classification algorithm only processes neural channels containing clean spike events. After removing the artefacts, the active neurons are sorted using principal component analysis (PCA) feature vectors and the K-means classification algorithm. The main contributions of this paper are summarized as follows:

- Embedding deep learning algorithms into spike processing pipeline to select active neural channels and extract discriminative spike waveforms.
- Complete removal of artefacts from the selected channels utilizing an optimized complementary CNN model

with deep spike waveform identification capability based on morphological characteristics.
- Enhanced feature learning for efficient hardware implementation to improve classification performance.

The rest of the paper is structured as follows: Section II describes the proposed DSD algorithm and methodology. The results are presented in Section III, followed by a discussion in Section IV. Finally, Section V makes some concluding remarks.

II. MATERIALS AND METHODS

A. OVERVIEW OF THE ALGORITHM

The proposed algorithm comprises three processing steps, as shown in Figure 1. The first step is a two-stage DSD algorithm, which is followed by a PCA feature extractor that also reduces the dimensionality of cleaned neural events by retaining only few features. The extracted features of events representing neural data are subsequently classified using the K-means clustering algorithm.

The DSD algorithm is a hierarchical structure consisting of two phases capable of accurately detecting spikes in extracellular recordings. The hierarchical structure is constructed by combining two designed CNN-based networks to accurately select neural channels and detect spikes in the selected neural channels, respectively. The performance of the proposed method was evaluated using both simulated and experimental datasets.

B. DATASETS

The effectiveness of the proposed method was evaluated using the simulated labeled dataset made available by [37]. The dataset consists of twenty subsets with disparate levels of noise and similarity between the spikes. Depending on the degree of similarity between the spikes, the subset data is classified into four types of difficulty, namely Easy1, Easy2, Difficult1, and Difficult2. Four noise levels of standard deviation are contained within each type of difficult data (0.05, 0.10, 0.15, and 0.20). The datasets are available with spike times, associated labels, and degrees of overlap between spikes. Using the ground-truth spike times stated in each dataset, 48 sampling points of spikes were extracted, making them appropriate for testing the proposed DSD algorithm and classification performance (see Section III-C).

In this study, both non-human primates (NHPs) and human patients' experimental datasets were used. The human experimental datasets were obtained from [38] as well as [39], [40], where micro-wires and Utah arrays were used to record from human patients. Two sets of micro-wires were inserted into the hippocampus of two human patients, and two Utah arrays, each containing 100 electrodes placed in a 10 × 10 grid, were inserted into the posterior parietal cortex of two other patients.

The NHPs dataset was obtained from collaborative research in computational neuroscience (CRCNS) [41], [42]. Implanted Utah arrays were used to record data from a macaque monkey's primary visual cortex. In addition to the CRCNS dataset, publicly available datasets captured from two rhesus macaques (X and B) using single micro-electrodes were also used. All datasets contain 48 sampled, preprocessed, labeled events since deep learning models require labeled data for supervised learning tasks for training and validation. Table 1 provides detailed information on the experimental datasets.

TABLE 1. Details of the experimental datasets.

Specie	Recording Device	Brain Area	No. of Sessions
Humans	Utah array	Posterior parietal cortex	7
Humans	Utah array	Posterior parietal cortex	7
Humans	Micro-wires	Anterior hippocampus	1
Humans	Micro-wires	Anterior hippocampus	2
NHPs	Utah array	Primary motor cortex	2
NHPs	Utah array	Primary motor cortex	1
NHPs	Single micro-electrodes	Superior parietal lobule	10

The datasets were divided into a training set and a testing set. The training set contains 70% of the data needed to optimize the parameters and the hyperparameters of the proposed DSD algorithm. The testing set, which is made up of the remaining 30%, was used to evaluate the generalization capability of the DSD performance on unseen data. In order to prevent biases during training, a more balanced dataset was created by subsampling the data of both classes to generalize well to the distribution. Generalizing well essentially means being able to learn factors inherent to that distribution so that one can perform well on any data sampled from that distribution. The training samples of neural events are almost twice as many as the training examples of artefact events. Consequently, to avoid bias during training, an equal number of samples were randomly chosen from each class. Furthermore, validation data were used to avoid overfitting by using early stopping criteria.

Since labeled data are required for supervised learning and classification tasks, ground truth labeling is obtained by employing KiloSort [8]. KiloSort is a MATLAB-based offline automated spike sorting algorithm that effectively clusters multi-channel neural spike signals based on the geometric layout of the electrode array. KiloSort is optimally used with Phy, an open-source manual clustering Python library with a graphical interface designed to improve the manual refinement of automatic spike sorting. Adjustments are made, which are mainly focused on channel selection attributes and the sensitivity of neuron identification. These are based on each channel's visualization of the overlaid waveform clusters to assess how many clusters are present, the interspike interval distribution of any potential single spike, and whether each potential single unit is stationary (i.e., present for the majority of the recording session). The Phy is only used to manually curate Kilosort's automatic spike sorting, merging (or splitting) to combine (or to divide) two clusters in order to unify spikes from the same neuron or separate spikes from distinct neurons, and labeling. Clusters that contain only artefact events are labelled as "artefact", while spike events are labelled as "spike."

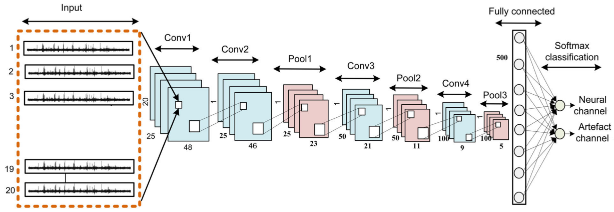


FIGURE 2. **The CNN model architecture utilized for channel selection. The neural data is successively processed to identify the neural and artefact channels. Input is convolved using trained kernel functions in order to get informative feature maps. The convolved feature maps are downsampled by the pooling layer, and the output decision is generated by the Softmax classifier. The corresponding sizes are annotated.**

C. CHANNEL SELECTION

Channel selection is the first phase of the DSD. In this paper, the channel selection phase is a deep learning algorithm that uses the CNN [43] algorithm to track and select neural data recording channels while discarding artefact channels. The number of channels varies according to the probe used to record extracellular recordings. The channel selection network architecture is illustrated in Figure 2.

Given a dataset of neural signal separated into labeled trials, $D^i = \{(x^1, y^1), \ldots\ldots, (x^{N_i}, y^{N_i})\}$ where N_i denotes the total number of recorded labeled trials and x^i represent the i^{th} feature vector and y^i the corresponding class label, the goal is to train a neural network,

$$f(x_i; \theta) : x_i \in R^{b \times w} \rightarrow y^i \tag{1}$$

on D^i such that the output of the parametric decoder is used to assign the correct label y_i to each raw data feature vector x_i by learning the parameters θ, iteratively from the training data, i.e the i^{th} sample is a neural channel if $y_i = 1$ and $y_i = 0$ is an artefact channel. The θ represents the parameters of the channel selection CNN while loss is measured by the cross-entropy with b and w are the height and width respectively. Since the channel selection CNN operates on batches of data, the batch size b used is 20.

In this study, the complete dataset consists of 32,235,500 labeled waveforms that yield 1,611,775 labeled input feature vectors. The input is 48 × 20 neural data in 2D matrix form, denoted as x above, where 48 is the sample waveform segment for a time duration of 1.6 ms. The construction of a single feature vector, denoted by x, is accomplished by concatenating a batch of b waveforms of length w, which ultimately results in the creation of a feature vector $x_i \in R^{b \times w}$.

The feature vector x is labeled a neural channel if one of the concatenated events is identified as a spike activity. Alternatively, x is labeled as an artefact channel if all the concatenated events constitute artefact activities (Figure 2). The goal is not to determine how many or which waveforms in the batch contain spikes, but rather to determine which batch of waveforms yields likelihood of containing a spike event.

The designed channel selection CNN model is shown in Figure 2. It takes each feature vector x as described above and processes it through the neural network architecture shown in Figure 2, which consists of four convolutional layers, three pooling layers, a fully connected layer with one hidden layer, and the classifier layer is a softmax layer with a neural channel and an artefact channel as output. The details of the CNN layers used in channel selection are also annotated in Figure 2.

During forward propagation, the convolution operation is performed by sliding each filter across the width and height of x of the input 2D data with a stride of 1. This yields 2D convolved feature maps, which are subjected to nonlinear activation maps for additional processing. The activation function employed is rectified linear units (ReLUs),

$$f(x) = max(x, 0). \tag{2}$$

Two sequential convolutional layers (conv1 and conv2) with a kernel size of 3 × 1 and strides of 1 × 1 extracted the temporal and spatial features, respectively. The subsequent convolutional layers (conv3 and conv4) employ time-based convolution. Fully connected (FC) layers were the final two layers. The first two FC feature maps include 500 and 100 neurons, respectively. The last layer is fully connected to two outputs i.e. neural or artefact channel. These kernel sizes were selected in accordance with the spatial extent of single-unit activities. For an illustration of how many and what size filters make up each convolutional layer, see Figure 2.

With the exception of the first convolutional layer, a pooling layer followed each convolutional layer. Max pooling layers are added to reduce the dimensionality and extract the most complex features from the convolved feature map. Prior to executing downsampling, zero-padding of convolutional layer 3 and convolutional layer 4 was chosen across the width without data change.

Batch normalization is chosen as a regularization technique to improve classification accuracy. The normalization method standardizes each layer of the network input of the CNN for each batch of data to a mean value of 0 and a variance of 1. Dropout [44] is another regularization method, in which some input neurons' values are randomly set to zero. Finally, the cross-entropy cost function is modified by including an L_2 regularization term, which ensures that all weight parameters have small values to avoid a single weight parameter from dominating the classification decision.

The optimization problem is solved using mini-batch gradient descent with momentum (Mini-batch SGDM), a popular optimization technique for updating the biases and weights. For the model learning, Mini-batch SGDM was used for the iterative learning, with a learning rate starting at 0.1 and being tuned piecewise, decreasing by 10 for every 5 training epochs. The Mini-batch SGDM was chosen to be 0.90. With a step size of 0.2, grid search was performed from 0 to 5 using L_2 regularization and the optimal was found to be 1.8. An early stopping criterion was employed to prevent overfitting on the testing data at each epoch by monitoring the validation error. If the error increases or remains unchanged over six successive epochs, the training is terminated. Finally,

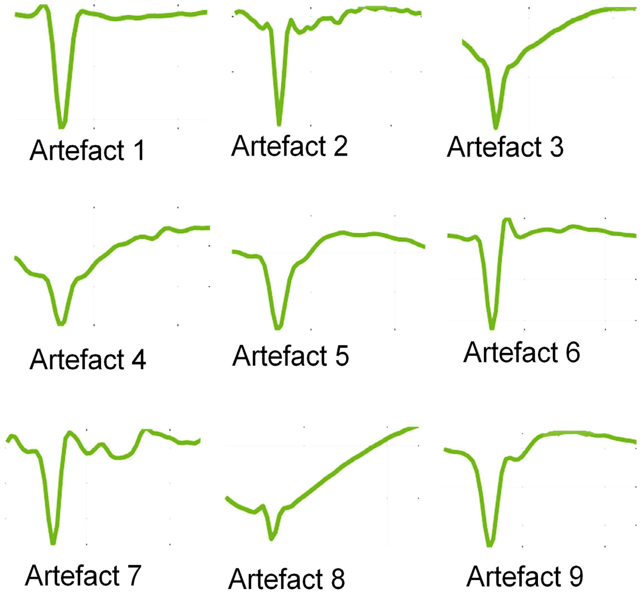


FIGURE 3. Templates of artefacts (Artefact 1, Artefact 2 ....Artefact 9).

dropout regularization was used to prevent overfitting by deciding whether or not to discard an input neuron with a probability of 0.5. The trained network assigns one of two labels, as neural and artefact, to an input feature vector.

D. ARTEFACT REMOVAL

The channel selection phase identifies the channels that record neural events. Additionally, these channels record a substantial amount of noise and artefacts. It is impossible to get rid of every possible source of artefacts from recordings. Therefore, it is absolutely necessary to distinguish neural spike events from other types of events, such as artefacts, in order to achieve the highest possible level of data quality and unit yield in the neural channels of choice. Neurons far from the electrode tips are regarded as artefacts [45], which represent spurious neural events from distantly active neurons and have a significant negative impact on the recorded signal. Individual microelectrodes can become rather close to one another in arrays or bundles of microwires. Since the volume of recordings on different channels can overlap, it is possible to capture the same unit on more than one channel simultaneously, resulting in duplicate spike events, a source of artefacts that is often overlooked. In addition, electrical interference, cable and head movement, broken cables, etc. within the recording setup could all result in spurious spike events [46]. All of these technical glitches originate from sources other than neurons, and can produce artefacts that look very similar to spikes of neurons, and can be recorded on several channels simultaneously. Importantly, an artefact in this context is defined as any waveform that did not exceed the threshold of the network setting in this work. Some templates of the artefacts captured in this work are shown in Figure 3. As was demonstrated in the preceding paragraph, corrupted segments are detrimental to any analysis and therefore must be identified and dealt with in order to produce reliable results. This includes artefact removal or discarding the segment. By analyzing the recorded data closely, it is found that artefacts across channels frequently share very similar event shapes.

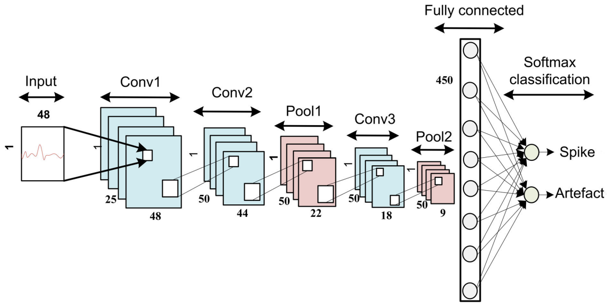


FIGURE 4. Architecture of the artefact removal. Convolutional and pooling layers are utilized to convolve and downsample the input. Number of kernels and size are specified for each convolutional and pooling layer. Zero padding is used during the first convolutional layer only. A stride of one and kernel size of 1 × 2 is utilized.

The initial assumption is that the electrode arrays record a variety of spike waveforms, either recognized as authentic spikes or spurious (or artefact) waveforms. By implementing artefact removal, a reversing process is introduced to remove spurious waveforms from the active neural channels that have an adverse impact on the overall classification performance. The labeled training data in Section II-B was used to achieve this goal of designing supervised learning. The architecture of the artefact removal is shown in Figure 4. The temporal characteristics of the spike waveform have a significant effect on accuracy. For this reason, a 1D CNN was trained on the temporal pattern.

The requirements for the architecture must be able to identify and extract the most crucial abstract features and not be limited to specific feature categories. The artefact removal CNN model receives 48 samples as input. The 1D CNN utilizes three convolutional layers and two pooling layers followed by a fully connected layer to process each data segment, and the softmax function is employed in the final layer, which outputs a value denoting the probability of a spike or artefact.

Unique features are extracted in the convolutional layer. Each filter is convolved across the width of the input segment and then slides with a stride of 1, to extract the most abstract features that differentiate the spike waveforms from artefacts. This produces 1D feature maps that have been convolved. Smaller strides result in more comprehensive and dense feature extraction and do not exclude an excessive amount of information. Consequently, the value of the stride was fixed at 1. Using a smaller convolution kernel allows for more feature extraction with less data than when using a larger convolution kernel. Then, nonlinearity is introduced using an activation layer. By introducing nonlinearity into the data and employing the ReLU $f(x) = max(x, 0)$, the optimization problem is solved. Max-pooling is also utilized to eliminate superfluous data and reduce computational complexity.

The cross-entropy cost function is minimized using an L_2 regularization term. In addition, batch normalization is utilized to standardize the intermediate outputs of the model to a zero mean and unit variance for each mini-batch of training inputs. The hyperparameters are tuned using Mini-batch SGDM, as described in Section II-C.

The softmax function was deployed in the final network layer. The softmax classifier is used to predict whether the input data is a spike or an artefact. The final connected layer has two output dimensions, i.e, spike or artefact.

E. FEATURE EXTRACTION AND CLUSTERING

The essential part of this study is to show that removing artefact events improves the spike sorting performance even with the simplest clustering algorithm. Typically, neural data generated too far from the recording electrodes' tips and artificial events are regarded as artefacts. Consequently, it is difficult for any classification algorithm to distinguish them as distinct clusters. However, the proposed DSD algorithm notably isolates the artefacts. After removing artefacts, clustering is therefore a trivial task.

PCA is used as a feature extractor in this work on the spike data. Using the eigenvectors with the most variability in the acquired data, the high-dimensional spike events are projected onto 2D or 3D principal component space using PCA. The feature vectors are Z-normalized to ensure that all features have a mean of zero and a standard deviation of one.

The first stage is to transform the spike waveform data into fewer dimensions by extracting the most crucial features that differentiate the waveforms, thus forming clusters. Using this method, meaningful features were successfully extracted even from signals with high structural similarity. The low dimensional projection of PCA does not capture enough discriminatory power. Therefore, in this study, a criterion that keeps a certain amount of variability in the presented data intact and constructs low dimensional feature vectors is applied. The number of principal components (features) selected is 85% of the variability of the data. The criterion of keeping 85% of the variability intact resulted in at most seven or eight principal components.

The K-means algorithm is then used to classify the features with a squared Euclidean distance metric to initialize the centres of the specified number of clusters. The maximum number of predicted clusters in this study is three. For near-optimal clustering analysis, the K-means tool in MATLAB was utilized with 10 iterations.

III. RESULTS

In this paper, three metrics were used to evaluate the sorting performance. The metrics for precision and recall will be reported for the channel's selection and artefact removal, while accuracy will be used for clustering performance.

Precision (P) is determined by dividing the number of true positive spikes (TPS) by the total number of all detected spikes, which corresponds to the sum of (TPS) and false positive spikes (FPS) due to artefacts and overlapping perturbations. Mathematically, it is expressed as;

$$P = \frac{TPS}{TPS + FPS}. \tag{3}$$

The recall (R) is calculated by dividing the number of true positive spikes (TPS) by the total number of detected spikes, including (TPS) and the false negative spikes (FNS) which show the spiking activities comprising those that were detected and those that were incorrectly labeled as artefact. R is calculated using

$$R = \frac{TPS}{TPS + FNS}. \tag{4}$$

The classification accuracy (CAcc) is the sum of the true positives for each cluster and the total number of activities for each cluster (i.e., a generalization of recall), expressed as

$$CAcc = \frac{\sum_{i=1}^{N} TPS_i}{\sum_{i=1}^{N}(TPS_i + FNS_i)} \tag{5}$$

where N is the number of clusters.

Algorithm implemented in MATLAB R2021b on a Windows platform personal computer with an Intel(R) Core (TM) i5-10310U CPU running at 2.21 GHz, 16 GB of RAM, and an Intel(R) UHD GPU. The training phase of the models was written, implemented, and trained in MATLAB R2021b on a 2 x AMD EPYC 7543 32-Core CPU, 512GB RAM, 4 x NVidia 80Gb Tesla A100 GPU server.

A. CLASSIFICATION ACCURACY OF CHANNEL SELECTION

The channel selection CNN assigns one of two labels, as neural or artefact, to a feature vector consisting of 20 waveforms. The trained model of the channel selection algorithm was evaluated on the test data. It performs well in classifying the feature vectors across all subjects, with few false negatives. The model has wrongly classified only four channels out of seven hundred and ninety-two channels, as shown in Table 2, which shows that the model has great generalization capability. Its performance was consistently evaluated on the individual channels to evaluate the effectiveness of the model across all channels.

In the recording sessions with the micro-wire implanted device, seven of the channels did not record any data. From the remaining nine, eight could record neural activity. All feature vectors of channel 1 and channels 3-9 were labeled as neural channels. Channel 2 did not record any neural data and was therefore labeled an artefact channel. The algorithm detected both neural and artefact channels with more than 95% accuracy in each channel. With an average accuracy of 99.5%, the channel selection correctly selects the channels recording neural activity.

In order to correctly train the CNN model to be able to achieve the best classification performance, hyperparameters need to be optimized. The batch size is one of the

TABLE 2. Results for channels selection performance evaluation on human patients and NHPs datasets.

Dataset/subjects	No of channels	Neural channels	Artefacts channels	False positives	False negatives
Utah array (humans)	576	109	467	2	0
Mircowires (humans)	16	15	1	0	0
Utah array (NHP)	192	190	2	0	2
Microelectrode (NHP)	8	8	0	0	0

main hyperparameters that need to be tuned. To evaluate the impact of batch size and learning rate on classification performance, several experiments with different parameter sets were performed. This study examines the impact of varying the learning rates while the batch size remains unchanged. Table 3 depicts the detailed results. On the contrary, attempts were made to compare experiments with varying batch sizes while maintaining the same learning rate. Table 4 displays the results. As demonstrated, batch size and learning rate are interdependent and can have a significant effect on classification performance.

TABLE 3. Imapct of learning rate on classification performance.

Batch size	Learning rate	Test accuracy
20	0.00001	0.9827
20	0.0002	0.9876
20	0.0001	0.9891
20	0.002	0.9902
20	0.001	0.9915
20	0.05	0.5764
20	0.1	0.4980
20	0.2	0.4373

Figure 5 shows the classification accuracy with a learning rate of 0.001 when the channel selection CNN model is trained and evaluated for each batch size. From Table 4, the accuracy is comparable for the same batch size.

TABLE 4. Imapct of batch size on classification performance.

Batch size	Learning rate	Test accuracy
10	0.001	0.9732
20	0.001	0.9751
30	0.001	0.9844
40	0.001	0.9853
50	0.001	0.9893
60	0.001	0.9925
80	0.001	0.9914
100	0.001	0.9962

During most training trials, with the same learning rate of 0.001, the classification accuracy and batch accumulation time increase with increasing batch size (Figure 5a). The average time it takes for all channels to accumulate waveforms is depicted in Figure 5b.

It is essential to optimize the trade-off between accurate classification and batch size selection. When the batch size was set at 20, the classification accuracy was 97.5%. With increasing iterations steps and a batch size of 65 prior to saturation, the classification accuracy was 99.5%. Comparing the two plots reveals that the channel selection CNN requires 270 ms to construct a feature vector with a batch size of 10. This yields an acceptable classification rate of 97%. The accuracy of batch accumulation is a significant factor in online decoding. Therefore, in order to perform online decoding, it is essential to construct a feature vector and track neural signals from each channel. The batch size decision may be less constrained for offline spike sorting.

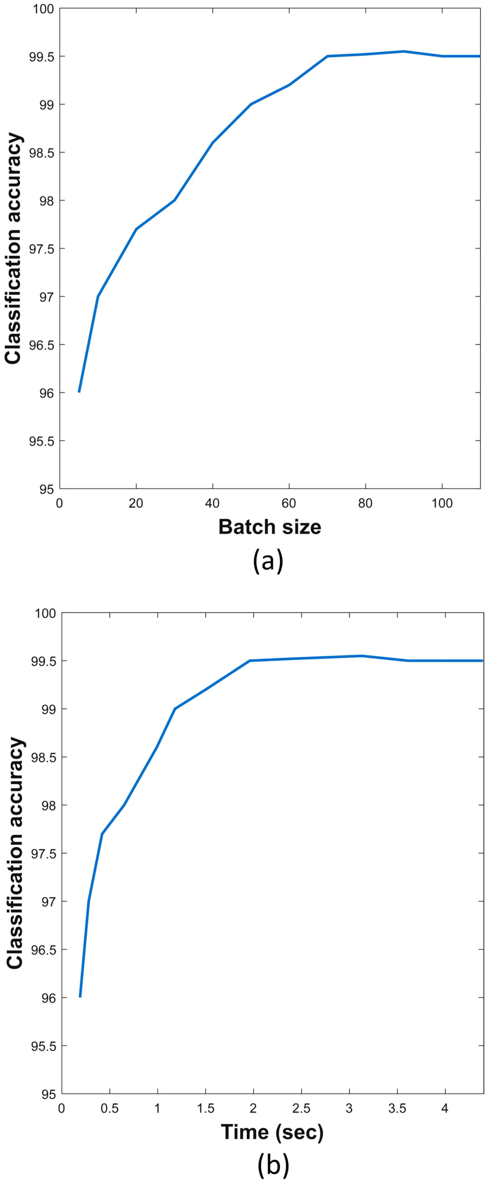


FIGURE 5. Impact of batch size and accumulation time on classification performance.

From the experimental analysis demonstrated as shown in Table 3, the best accuracy and the lowest loss were achieved with a learning rate of 0.001 and batch size of 20. A batch size of 20 is selected for this work, while the selection of batch size is application dependent.

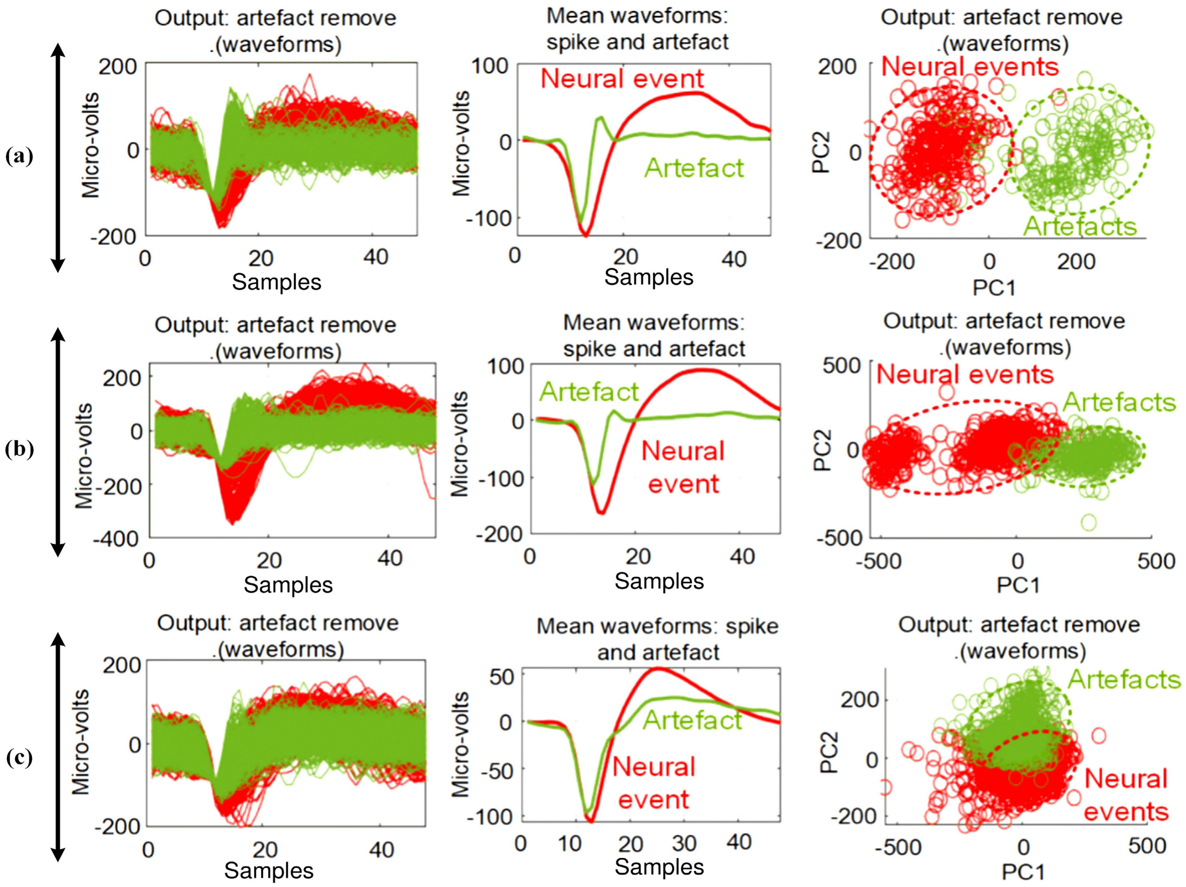


FIGURE 6. Three different examples of artefact removal are shown. Artefact events are shown in green, while neural events are shown in red. (a) neural and artefact are fully separated. (b) neural and artefact are partially overlapped. (c) neural and artefact are having higher overlapped. PC1 and PC2 are used 2D for projection of spike clusters.

B. CLASSIFICATION ACCURACY OF ARTEFACT REMOVAL

The channel selection phase achieves a 99.5% rate of accuracy in identifying channels with neuronal activity, i.e., only four wrong predictions out of a total of seven hundred and ninety-two channels (see Table 2). The trained model of artefact removal was evaluated on identified neural channels by the channel selection phase performance to determine the robustness of the model to evaluate the likelihood that a neural waveform is a spike event or artefact. The artefact CNN model is trained on different waveform types from a diverse set of subjects in the training set. The model was exposed to variability in classification of previously sorted spikes utilizing KiliSort, a MATLAB-based application, with each waveform assigned a binary label of $y_i = 1$ if it is a spike and $y_i = 0$ if it is an artefact. The artefact removal performance remains consistent during all evaluations, with an accuracy (recall) of 88.9% and 95.4%. The overall classification accuracy (CAcc) of the artefact removal is 92.3%.

For visualization purposes, three different samples were selected. As the results show, Figure 6 depicts the waveforms with predicted labels for each class and the PCA-based projection of waveforms onto 2D feature space. The performance of the artefact removal on three different types of recording channels was shown.

Figure 6a depicts the performance of the artefact removal on the channel where spike events and artefacts are sufficiently separated. In this case, it is easy for even a simple clustering algorithm to accurately identify the two clusters.

In Figure 6b, neural events and artefacts are partially overlapped in PCA space. Hence, conventional spike sorting pipeline clustering algorithms will struggle to automatically extract discriminative features to distinguish between two clusters and systematically fail when spike events and artefacts overlap. However, Figure 6c shows another type of channel where neural events and artefacts are almost completely overlapped. This will be a very challenging task for a clustering algorithm to clearly define the differences between the neural waveforms of two clusters due to the overlapping artefact. The artefact removal has identified the spike events for each neuron across all channels and discarded the artefacts. It aided the classification process simplifies as shown in Figure 6c. Despite increasing artefacts and overlapped waveforms as shown in the output waveforms (spike and artefact) in PCA space, the artefact removal in this work successfully isolates the overlapped clusters as shown in the mean waveform (spike and artefact) plots. The results reveal that the addition of the artefact removal algorithm

TABLE 5. Comparison of classification accuracy on simulated dataset. The number in parentheses represents the number of overlapped spikes, (FP - False Positives).

Dataset	Noise levels	Cluster 1	Cluster 2	Cluster 3	No of spikes	Misses (Wave_Clus, Proposed)	FP (Wave_Clus, Proposed)
Easy1	0.05	1165 (250)	1157 (275)	1192 (260)	3514 (785)	17 (193), 0 (20)	711, 190
	0.10	1151 (248)	1134 (264)	1237 (257)	3522 (769)	2 (177), 2 (33)	57, 11
	0.15	1132 (242)	1188 (272)	1157 (270)	3477 (784)	145 (215), 43 (57)	14, 5
	0.20	1198 (279)	1128 (248)	1148 (269)	3474 (796)	714 (275), 118 (95)	10, 2
	0.25	1094 (237)	1089 (229)	1115 (246)	3298 (712)	13 (211), 5 (34)	7, 3
	0.30	1162 (294)	1164 (285)	1149 (267)	3475 (846)	20 (203), 6 (0)	4, 1
	0.35	1208 (285)	1137 (269)	1189 (278)	3534 (832)	15 (174), 3 (1)	1, 0
	0.40	1079 (238)	1158 (261)	1149 (242)	3386 (741)	0 (168), 0 (0)	4, 0
Easy2	0.05	1130 (274)	1113 (257)	1167 (260)	3410 (791)	0 (174), 0 (0)	0, 0
	0.10	1160 (269)	1146 (280)	1214 (277)	3520 (826)	0 (191), 0 (1)	2, 0
	0.15	1181 (265)	1098 (237)	1132 (261)	3411 (763)	10 (173), 14 (3)	1, 0
	0.20	1186 (262)	1188 (278)	1152 (271)	3526 (811)	376 (256), 44 (18)	5, 1
Difficult1	0.05	1115 (244)	1113 (256)	1155 (267)	3383 (767)	1 (210), 3 (133)	63, 18
	0.10	1164 (260)	1155 (269)	1129 (281)	3448 (810)	0 (191), 0 (175)	10, 7
	0.15	1159 (275)	1172 (260)	1141 (277)	3472 (812)	8 (203), 10 (190)	6, 16
	0.20	1136 (267)	1099 (257)	1179 (266)	3414 (790)	184 (219), 170 (171)	2, 5
Difficult2	0.05	1120 (271)	1109 (274)	1135 (284)	3364 (829)	0 (182), 0 (2)	1, 0
	0.10	1187 (230)	1136 (238)	1139 (252)	3462 (720)	0 (152), 0 (10)	5, 3
	0.15	1142 (284)	1113 (262)	1185 (263)	3440 (809)	3 (186), 5 (23)	4, 0
	0.20	1151 (260)	1195 (277)	1147 (240)	3493 (777)	262 (228), 121 (41)	2, 0

shows robust learning capability as a pre-processing step that improves the performance of the clustering algorithm.

C. CLASSIFICATION ACCURACY OF THE CLUSTERING ALGORITHM

The channel selection in conjunction with the artefact removal nearly cleaned the neural data of artefact events in two complementary steps. To extract the most abstract features, the neural waveform data is transformed to fewer dimensions using PCA, which differentiates the waveforms forming the clusters. The extracted features were fed into the K-means algorithm to identify the neural waveforms that are associated with each cluster on a single channel (i.e., the maximum number of clusters is 3). In addition to showing good reliability and robustness under varying noise levels, the results demonstrate the proposed method's ability to extract features that cluster well and are easily separated from one another. If more advanced methods for feature extraction are used, the accuracy is very likely to improve.

1) SIMULATED DATASET

To evaluate the effectiveness and classification accuracy (CAcc) of the proposed pipeline, comparisons were made with a conventional but powerful method, Wave_clus [47], with the simulated dataset. Table 5 and Table 6 display the outcomes for the data with varying degrees of noise.

Table 5 shows the number of false negative spikes (FNS) and false positive spikes (FPS) compared with Wave_clus. The DSD gives high accuracy in both cases where overlapping spikes shown in parentheses are included and excluded. In a few cases, the method described in this paper performed suboptimally.

Table 6 shows the classification accuracy in comparison with Wave_clus. The proposed method demonstrated an accuracy greater than 93%, in contrast to the Wave_clus spike sorting algorithms, which demonstrated an overall classification accuracy of less than 89% when the level of noise was greater than 0.3 in the Easy1 sub dataset. Moreover, the pipeline produced the best performance on Difficult2, the most difficult subset from the simulated dataset, with an average spike sorting classification accuracy of 91.48% as compared to Wave_clus's 81.89%. In most cases, the pipeline showed superior performance over Wave_clus, except in Difficult1, where Wave_clus outperformed the proposed method when the noise levels were 0.15 and 0.2. This poor performance is one of a few cases where the DSD scored sub-optimally due to the higher number of false positives in the pipeline than Wave_clus, as shown in Table 5. Furthermore, the Wave_clus used powerfull feature extractor, dimensionality reduction and classification (discrete wavelet transform (DWT)>Kolmogorov-Smirnov (KS)>superparamagnetic (SPC)) in contrast to DSD utilizing only K-means. On the contrary, the average classification accuracy of the pipeline is 91.53%, an improvement of 3.38% over Wave_clus. Compared to Wave_clus, the advantages of the proposed method were more apparent. Across all datasets, Wave_clus deteriorated as noise levels increased.

TABLE 6. Comparison of classification accuracy on simulated dataset, the proposed DSD algorithm versus Wave_clus [47] and SpikeDeep-classifier [28].

Dataset	Noise levels	No of spikes	Wave_Clus	SpikeDeep-classifier	Proposed
Easy1	0.05	3514	98.82	99.07	99.13
	0.10	3522	98.81	99.32	99.31
	0.15	3477	98.72	99.10	99.13
	0.20	3474	98.55	98.85	98.93
	0.25	3298	97.32	97.13	97.44
	0.30	3475	89.50	96.75	97.38
	0.35	3534	82.12	93.34	93.23
	0.40	3386	82.60	94.72	95.67
Easy2	0.05	3410	96.88	96.84	97.31
	0.10	3520	91.62	91.30	92.40
	0.15	3411	91.30	92.24	92.23
	0.20	3526	84.52	83.32	84.62
Difficult1	0.05	3383	89.10	96.79	97.22
	0.10	3448	93.44	93.24	95.24
	0.15	3472	66.95	70.64	57.68
	0.20	3414	75.19	67.70	67.73
Difficult2	0.05	3364	93.52	95.57	96.57
	0.10	3462	94.05	83.65	98.66
	0.15	3440	83.16	81.86	91.87
	0.20	3493	56.17	77.85	78.83
Average			88.15	90.56	91.53

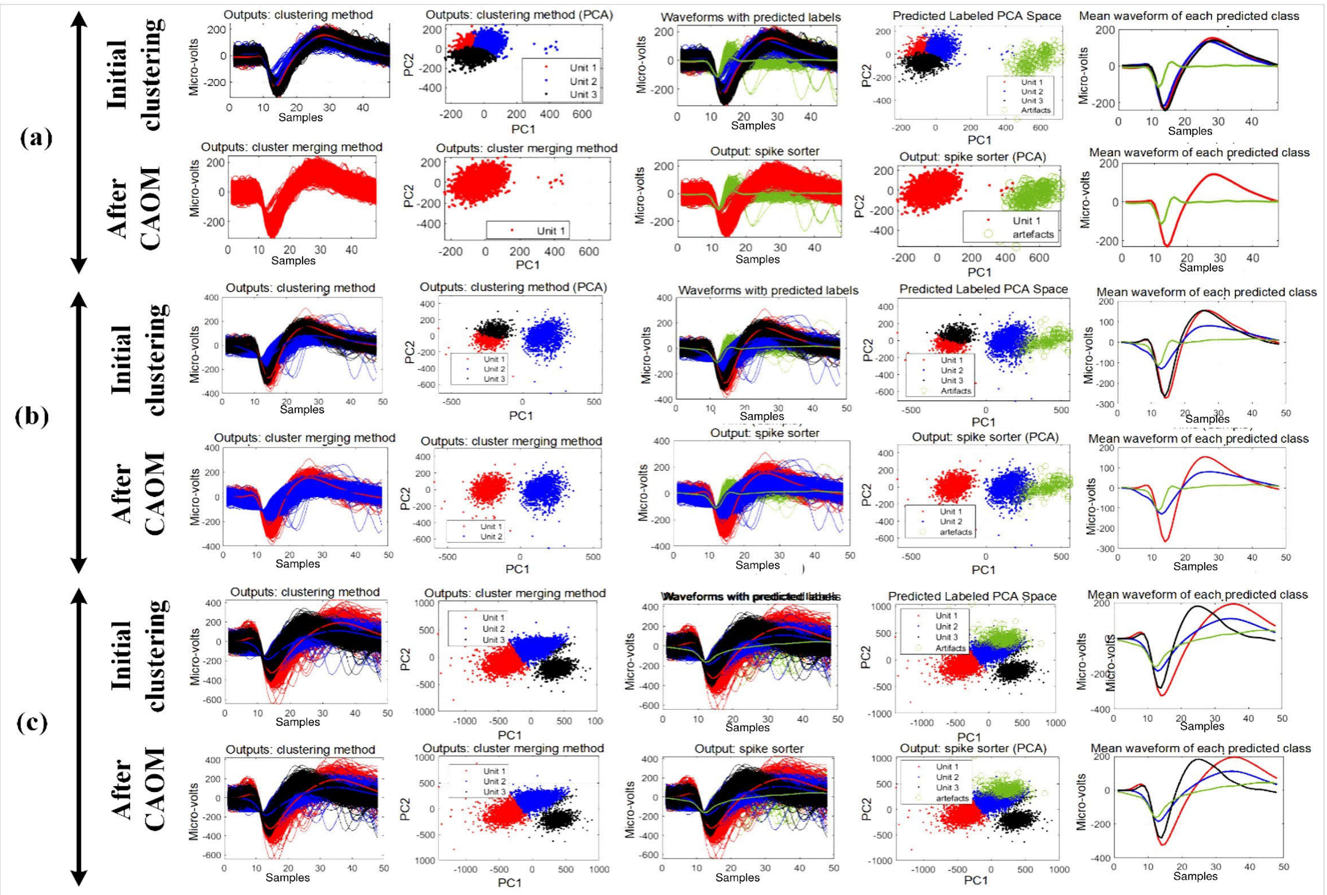


FIGURE 7. Classification results. Artefact events are shown in green, while neural events are shown in other colours. (a) Three neural units merged into one cluster (b) Three neural units merged into two (c) A channel with three distinct units. In each example, the K-means clustering algorithm's results are displayed in the first row of each example, and the CAOM's results are displayed in the second row.

The performance of the DSD algorithm was also compared to that of SpikeDeep-classifier [28], a deep learning-based spike sorting algorithm, on datasets from [37]. As shown in Table 6, the DSD algorithm classifier maintains a slightly higher accuracy of 1% than SpikeDeep-classifier and obtains a comparable number of true-positive clusters. A potential reason for this slightly higher accuracy is that the focused class of artefacts in this work has a more negative impact on the spike sorting pipeline compared to the background activities of SpikeDeep-classifier [28], despite the advanced feature extraction and clustering of SpikeDeep-classifier. Furthermore, the DSD demonstrate a high level of reliability with a large dataset and functions in a truly multichannel environment. The results show that the DSD algorithm is more robust than the SpikeDeep-classifier algorithm, with slightly fewer hits but consistently fewer false-positive clusters.

The computational cost of algorithms is an important factor for online applications with microelectrode arrays with hundreds of channels. Therefore, a lower computational cost makes it more suitable as a feature extractor for BMI decoding applications. The total number of parameters in DSD is 30% less than SpikeDeeptor and SpikeDeep-classifier. The number of floating point operations (FLOPs) and multiply-and-accumulate operations (MACs) of the DSD are 30% less than SpikeDeeptector and SpikeDeep-classifier's FLOPS and MACS. The DSD is 25% less computationally intensive than SpikeDeep-classifier and thus will be more easily implemented on a hardware resource-constrained device as a real-time spike sorting processor employing a deep neural network.

2) EXPERIMENTAL DATASET

Experimental datasets are used to evaluate the proposed DSD pipeline. As shown in Table 2, only one hundred and nine of the five hundred and seventy-six channels in the human dataset (Utah array) were predicted to be neural channels. Only a small number of channels record neural activity from multiple neural sources that are each located in different brain areas. However, only three neural sources were captured on a small number of channels. The optimal number of clusters on the recording channels is predicted using K-means clustering in conjunction with cluster accept or merge (CAOM).

The main idea is to assess the degree of similarity and difference pairwise by examining each pair of clusters that are mutually closest to one another and determining whether the two clusters should be merged. Only six channels out of one hundred and seven were predicted to have a different number of clusters. For any number of units, the overall accuracy achieved is greater than 91%. Moreover, the performance of the classification algorithm and CAOM across all individual recording sessions remains stable.

A visual inspection and evaluation of the quality of the obtained clusters are demonstrated. Figure 7 shows the classification algorithm in conjunction with CAOM. Figure 7a depicts the output of the K-means algorithm with three neural units merged into one. The CAOM output is displayed in the second row. The similarity between units is determined according to CAOM criteria. In addition, the second row of figures depicts the mean waveforms and PCA projection in 2D. Figure 7b illustrates the output that is predicted by the K-means algorithm after three neural units have been merged into two. The CAOM output is displayed in the second row. Figure 7c presents the results that were predicted by the K-means algorithm, which consists of three clusters. In this particular instance, each of the three clusters has been considered a distinct cluster.

IV. DISCUSSION

In this study, a DSD is introduced to deal with the problem of noisy channels and artefact removal. Most spike sorting studies do not consider artefact channels and artefacts, despite the fact that removing them is one way to improve data quality. However, in a setting designed for recording, it is not possible to get rid of all of the factors that could be disruptive. Consequently, it is crucial to examine the negative impacts of artefacts in neural data. Two deep neural network classifiers are trained to select channels recording neural events and separate spikes from artefacts in the selected channels in real-time extracellular electrophysiological recordings using a pool of previously spike-sorted and labelled data.

Unlike the SpikeDeeptector [27], the DSD channel selection is focused on a defined class of artefacts. It is less generic than the SpikeDetector and more sensitive to deep and temporal spikes. Though the purpose of this study was to enrich the potential of classic spike detection with a simple implementation of a deep learning-based spike detection algorithm, it unequivocally demonstrates that it is robust to the defined class of artefacts. It has a better classification score and is more efficient than SpikeDeeptector. The DSD CNN design has the advantage of evaluating the entire recording space by a convolutional filter convolving across the entire spatial dimension, capturing spatial features more effectively.

The DSD algorithm is unique in comparison to the SpikeDeeptector [27] method. The DSD makes use of CNN architecture to extract both spatial and temporal features from spike signals simultaneously recorded in multi-channel electrodes while the SpikeDeeptector uses convolution only for discovering time dependencies along the spike/artefact samples and possible dependencies between different samples. Acknowledging how the temporal features of spike signals are extremely mitigated by artefacts, the DSD CNN design focuses on extracting the spatial feature of the spike data, adapts to non-stationary data, and is, therefore, well suited for acute recordings. By incorporating a generalized prototype loss-based on distance cross-entropy, the algorithm learned features that were closer to its class to improve the intra-class compactness, which makes feature representation more discriminative.

For a single waveform input runs through the trained network, the output was a value between 0 (likely an artefact channel) and 1 (likely a neural channel), which represented the network's output probability of a channel to be classified as a neural or artefact channel. In contrast, SpikeDeeptector introduced a criterion to assign labels to entire channels by calculating the mode of the predicted outputs of all the feature vectors of the given channel.

Contrarily to background activity rejector (BAR) [28], the input to the DSD's artefact removal included data from channels with very distinct spike waveforms. These channels still included both artefact (labeled as 0) and spike (labeled as 1) waveforms. The training set was designed to emphasize relatively well-isolated single-unit action potential shapes while also exposing the network to a variety of spike and artefact waveforms.

The proposed DSD algorithm was most effective at finding events within clusters that were already labeled as artefacts in the dataset. The evaluation of DSD on both simulated and experimental data revealed a low false-positive rate and, consequently, a high level of specificity for the algorithm. The exact FPS rate results are contingent on a number of factors, including the percentage of artefacts, the number of channels, the clusters, and so on.

The CAOM is used to estimate the exact number of neural units on a recorded channel. In this study, the maximum number is fixed at 3. Some recording configurations result in a greater number of neuronal units per channel, but this is often not the case with permanently implanted electrodes. In such a circumstance, CAOM's hyperparameters must be retuned. Consequently, an algorithm (HDBSCAN or t-SNE) that provides a more generic solution for determining the number of clusters will be used in the future in place of CAOM or clustering.

Embedding the DSD algorithm in a conventional spike sorting pipeline displayed a good degree of sorting precision in both simulated and experimental datasets. In the simulated dataset, even with varying degrees of noise and similarity between the spike waveforms, the proposed DSD pipeline performed extremely well in classifying the spikes. Consequently, it is plausible to conclude that the proposed DSD pipeline demonstrated excellent detection and accuracy. The performance of the approach was based on two factors: the selection of neural channels and the removal of artefacts. Both phases had a considerable positive impact on performance and accuracy. In future work, the deep neural networks will incorporate CAE and unsupervised subspace learning [48] to extract discriminative features that will handle issues of computational complexity.

In multiple channel recordings with microelectrode arrays, the recorded activity may be the overlap of multi-neuron spikes, which will degrade the traditional spike sorting performance. The existence of this issue has been widely reported. Some new methods have been proposed to solve the problem of overlapped spikes [49]. However, the challenges associated with the overlapping of spikes are not yet well resolved. Future work will address this challenge and improve the DSD's classification accuracy further by utilizing deep neural networks.

V. CONCLUSION

In this study, a deep neural network-based framework to classify neuronal spikes was proposed. By embedding two convolutional neural network algorithms for channel selection and artefact removal, respectively, in the conventional pipeline, the spike sorting pipeline largely improves the classification accuracy with enhanced detection. It improves the sorting performance by identifying the inactive recording channels as well as discarding the artefacts before feature extraction. The channel selection successfully selects the channels that record neuronal events with an average accuracy of 99.5%, while the artefact removal accuracy is 92.3%. The combined utilization of the channel selection and artefact removal with K-means clustering is potentially promising and yields a classification accuracy of 87% and 91.53% on experimental and simulated datasets respectively, outperforming the traditional approach.

REFERENCES

[1] M. Zamani, D. Jiang, and A. Demosthenous, "An adaptive neural spike processor with embedded active learning for improved unsupervised sorting accuracy," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 3, pp. 665–676, Jun. 2018.

[2] M. Zamani, C. Okreghe, and A. Demosthenous, "Efficient approximation of action potentials with high-order shape preservation in unsupervised spike sorting," in *Proc. 44th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2022, pp. 4884–4887.

[3] R. J. Schafer and T. Moore, "Selective attention from voluntary control of neurons in prefrontal cortex," *Science*, vol. 332, no. 6037, pp. 1568–1571, Jun. 2011.

[4] M. D. Golub, B. M. Yu, and S. M. Chase, "Internal models for interpreting neural population activity during sensorimotor control," *eLife*, vol. 4, Dec. 2015, Art. no. e10015.

[5] U. Chaudhary, N. Birbaumer, and A. Ramos-murguialday, "Brain-computer interfaces for communication and rehabilitation," *Nature Rev. Neurology*, vol. 12, no. 9, pp. 513–525, 2016.

[6] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, "High-performance neuroprosthetic control by an individual with tetraplegia," *Lancet*, vol. 381, no. 9866, pp. 557–564, Feb. 2013.

[7] M. Zamani and A. Demosthenous, "Feature extraction using extrema sampling of discrete derivatives for spike sorting in implantable upper-limb neural prostheses," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 716–726, Jul. 2014.

[8] M. Pachitariu, N. Steinmetz, S. Kadir, M. Carandini, and K. D. Harris, "Kilosort: Realtime spike-sorting for extracellular electrophysiology with hundreds of channels," *bioRxiv*, p. 061481, 2016.

[9] B. B. Averbeck, P. E. Latham, and A. Pouget, "Neural correlations, population coding and computation," *Nature Rev. Neurosci.*, vol. 7, pp. 358–366, May 2006.

[10] D. Carlson and L. Carin, "Continuing progress of spike sorting in the era of big data," *Current Opinion Neurobiol.*, vol. 55, pp. 90–96, Apr. 2019.

[11] N. A. Steinmetz, C. Aydin, A. Lebedeva, and M. Okun, "Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings," *Science*, vol. 372, no. 6539, Apr. 2021.

[12] C. Klaes, S. Kellis, T. Aflalo, B. Lee, K. Pejsa, K. Shanfield, S. Hayes-Jackson, M. Aisen, C. Heck, C. Liu, and R. A. Andersen, "Hand shape representations in the human posterior parietal cortex," *J. Neurosci.*, vol. 35, no. 46, pp. 15466–15476, Nov. 2015.

[13] C. Kechris, A. Delitzas, V. Matsoukas, and P. C. Petrantonakis, "Removing noise from extracellular neural recordings using fully convolutional denoising autoencoders," in *Proc. 43rd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Nov. 2021, pp. 890–893.

[14] F. Lopes, A. Leal, J. Medeiros, and M. F. Pinto, "Automatic electroencephalogram artifact removal using deep convolutional neural networks," *IEEE Access*, vol. 9, pp. 149955–149970, 2021.

[15] J. Yi, J. Xu, E. Chen, M. Chamanzar, and V. Chen, "Multichannel many-class real-time neural spike sorting with convolutional neural networks," *IEEE Open J. Circuits Syst.*, vol. 3, pp. 168–179, 2022.

[16] S. Mirzaei, H. Hosseini-Nejad, and A. M. Sodagar, "Spike detection technique based on spike augmentation with low computational and hardware complexity," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2020, pp. 894–897.

[17] B. P. Christie, D. M. Tat, Z. T. Irwin, V. Gilja, P. Nuyujukian, J. D. Foster, S. I. Ryu, K. V. Shenoy, D. E. Thompson, and C. A. Chestek, "Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance," *J. Neural Eng.*, vol. 12, no. 1, Feb. 2015, Art. no. 016009.

[18] G. Baldazzi, G. Solinas, J. Del Valle, M. Barbaro, S. Micera, L. Raffo, and D. Pani, "Systematic analysis of wavelet denoising methods for neural signal processing," *J. Neural Eng.*, vol. 17, no. 6, Dec. 2020, Art. no. 066016.

[19] G. T. Einevoll, F. Franke, E. Hagen, C. Pouzat, and K. D. Harris, "Towards reliable spike-train recordings from thousands of neurons with multielectrodes," *Current Opinion Neurobiol.*, vol. 22, no. 1, pp. 11–17, Feb. 2012.

[20] J. E. Chung, J. F. Magland, A. H. Barnett, V. M. Tolosa, A. C. Tooker, K. Y. Lee, K. G. Shah, S. H. Felix, L. M. Frank, and L. F. Greengard, "A fully automated approach to spike sorting," *Neuron*, vol. 95, no. 6, pp. 1381–1394.e6, Sep. 2017.

[21] D. Issar, R. C. Williamson, S. B. Khanna, and M. A. Smith, "A neural network for online spike classification that improves decoding accuracy," *J. Neurophysiology*, vol. 123, no. 4, pp. 1472–1485, Apr. 2020.

[22] K. Yang, H. Wu, and Y. Zeng, "A simple deep learning method for neuronal spike sorting," in *Proc. Int. Conf. Cloud Technol. Commun. Eng.*, vol. 17, 2017, Art. no. 012062.

[23] M. Rácz, C. Liber, E. Németh, R. Fiáth, J. Rokai, I. Harmati, I. Ulbert, and G. Márton, "Spike detection and sorting with deep learning," *J. Neural Eng.*, vol. 17, no. 1, Jan. 2020, Art. no. 016038.

[24] I. Y. Park, J. Eom, H. Jang, S. Kim, S. Park, Y. Huh, and D. Hwang, "Deep learning-based template matching spike classification for extracellular recordings," *Appl. Sci.*, vol. 10, no. 1, p. 301, 2020.

[25] Z. Li, Y. Wang, N. Zhang, and X. Li, "An accurate and robust method for spike sorting based on convolutional neural networks," *Brain Sci.*, vol. 10, no. 11, pp. 1–16, 2020.

[26] C. Seong, W. Lee, and D. Jeon, "A multi-channel spike sorting processor with accurate clustering algorithm using convolutional autoencoder," *IEEE Trans. Biomed. Circuits Syst.*, vol. 15, no. 6, pp. 1441–1453, Dec. 2021.

[27] M. Saif-ur-Rehman, R. Lienkämper, Y. Parpaley, J. Wellmer, C. Liu, B. Lee, S. Kellis, R. Andersen, I. Iossifidis, T. Glasmachers, and C. Klaes, "SpikeDeeptector: A deep-learning based method for detection of neural spiking activity," *J. Neural Eng.*, vol. 16, no. 5, Oct. 2019, Art. no. 056003.

[28] M. Saif-ur-Rehman, O. Ali, S. Dyck, R. Lienkämper, M. Metzler, Y. Parpaley, J. Wellmer, C. Liu, B. Lee, S. Kellis, R. A. Andersen, I. Iossifidis, T. Glasmachers, and C. Klaes, "SpikeDeep-classifier: A deep-learning based fully automatic offline spike sorting algorithm," *J. Neural Eng.*, Nov. 2020, Art. no. 016009.

[29] J. Lecoq, M. Oliver, J. H. Siegle, N. Orlova, P. Ledochowitsch, and C. Koch, "Removing independent noise in systems neuroscience data using DeepInterpolation," *Nature Methods*, vol. 18, no. 11, pp. 1401–1408, Nov. 2021.

[30] N. M. N. Leite, E. T. Pereira, E. C. Gurjao, and L. R. Veloso, "Deep convolutional autoencoder for EEG noise filtering," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2018, pp. 2605–2612.

[31] C. Arsene, "Complex deep learning models for denoising of human heart ECG signals," 2019, *arXiv:1908.10417*.

[32] W. Sun, Y. Su, X. Wu, and X. Wu, "A novel end-to-end 1D-ResCNN model to remove artifact from EEG signals," *Neurocomputing*, vol. 404, pp. 108–121, Sep. 2020.

[33] B. Yang, K. Duan, C. Fan, C. Hu, and J. Wang, "Automatic ocular artifacts removal in EEG using deep learning," *Biomed. Signal Process. Control*, vol. 43, pp. 148–158, May 2018.

[34] M. Fabietti, M. Mahmud, A. Lotfi, M. S. Kaiser, A. Averna, D. J. Guggenmos, R. J. Nudo, M. Chiappalone, and J. Chen, "SANTIA: A MATLAB-based open-source toolbox for artifact detection and removal from extracellular neuronal signals," *Brain Informat.*, vol. 8, no. 1, p. 14, Dec. 2021.

[35] D. Mzurikwao, O. Williams Samuel, M. Grace Asogbon, X. Li, G. Li, W.-H. Yeo, C. Efstratiou, and C. Siang Ang, "A channel selection approach based on convolutional neural network for multi-channel EEG motor imagery decoding," in *Proc. IEEE 2nd Int. Conf. Artif. Intell. Knowl. Eng. (AIKE)*, Jun. 2019, pp. 195–202.

[36] R. T. Schirrmeister, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, Aug. 2017.

[37] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, no. 8, pp. 1661–1687, Aug. 2004.

[38] T. Aflalo, S. Kellis, C. Klaes, B. Lee, Y. Shi, K. Pejsa, and R. Andersen, "Decoding motor imagery from the posterior parietal cortex of a tetraplegic human Science," *Neurophysiology*, vol. 348, no. 6237, pp. 906–910, 2015.

[39] Y. Shi, G. Apker, and C. A. Buneo, "Multimodal representation of limb endpoint position in the posterior parietal cortex," *J. Neurophysiology*, vol. 109, no. 8, pp. 2097–2107, Apr. 2013.

[40] P. N. Lawlor, M. G. Perich, L. E. Miller, and K. P. Kording, "Linear-nonlinear-time-warp-Poisson models of neural activity," *J. Comput. Neurosci.*, vol. 45, no. 3, pp. 173–191, Dec. 2018.

[41] C. A. Buneo, Y. Shi, G. Apker, and P. Vangilder. (2016). *Multimodal Spike Data Recorded From Posterior Parietal Cortex of Non-Human Primates Performing a Reaction-Time Task Involving Combined Eye and Arm Movements While in a Virtual Reality Environment*. [Online]. Available: https://www.CRCNS.org and https://doi.org/10.6080/K0CZ353K

[42] M. G. Perich, P. N. Lawlor, K. P. Kording, and L. E. Miller. (2018). *Extracellular Neural Recordings From Macaque Primary and Dorsal Premotor Motor Cortex*, [Online]. Available: https://www.CRCNS.org and https://doi.org/10.6080/K0FT8J72

[43] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, Dec. 2015.

[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[45] C. Pedreira, J. Martinez, M. J. Ison, and R. Q. Quiroga, "How many neurons can we see with current spike sorting algorithms?" *J. Neurosci. Methods*, vol. 211, no. 1, pp. 58–65, Oct. 2012.

[46] G. Dehnen, M. S. Kehl, A. Darcher, T. T. Müller, J. H. Macke, V. Borger, R. Surges, and F. Mormann, "Duplicate detection of spike events: A relevant problem in human single-unit recordings," *Brain Sci.*, vol. 11, no. 6, p. 761, Jun. 2021.

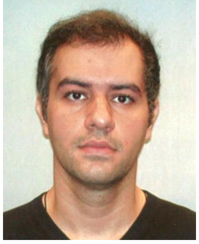
[47] R. Q. Quiroga. (2009). *Wave_Clus: Unsupervised Spike Detection and Sorting*. Accessed: Feb. 2022. [Online]. Available: https://www2.le.ac.uk/centres/csn/software/wave-clus

[48] M. Zamani, J. Sokolic, D. Jiang, F. Renna, M. R. D. Rodrigues, and A. Demosthenous, "Accurate, very low computational complexity spike sorting using unsupervised matched subspace learning," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 221–231, Apr. 2020.

[49] J. Wouters, F. Kloosterman, and A. Bertrand, "A data-driven spike sorting feature map for resolving spike overlap in the feature space," *J. Neural Eng.*, vol. 18, no. 4, Aug. 2021, Art. no. 0460a7.

CHRISTIAN O. OKREGHE (Member, IEEE) received the B.Eng. degree in electrical and electronic engineering from the University of Benin, Nigeria, in 1999, and the M.Sc. degree in electronic and electrical engineering from Obafemi Awolowo University, Nigeria, in 2010. He is currently pursuing the Ph.D. degree with the Department of Electronic and Electrical Engineering, University College London.

His research interests include machine learning, control engineering, FPGA, and integrated circuit design.



MAJID ZAMANI (Member, IEEE) received the M.Sc. degree in microelectronics from Islamic Azad University, Science and Research Branch, Tehran, in 2011, and the Ph.D. degree from University College London (UCL), London, U.K., in 2017.

He was a Research Associate with the Bioelectronics Group, UCL. His research interests include design and fabrication of advanced and energy efficient computational systems utilizing pattern recognition, machine learning, and computer vision algorithms, especially for wearable and implantable biomedical applications. He was a recipient of the Oversea Research Scholarship and the UCL Graduate Research Scholarship to pursue his Ph.D. degree. He was also a recipient of the Best Researcher M.Sc. Student Award.

ANDREAS DEMOSTHENOUS (Fellow, IEEE) received the B.Eng. degree in electrical and electronic engineering from the University of Leicester, Leicester, U.K., in 1992, the M.Sc. degree in telecommunications technology from Aston University, Birmingham, U.K., in 1994, and the Ph.D. degree in electronic and electrical engineering from University College London (UCL), London, U.K., in 1998.

He is currently a Professor with the Department of Electronic and Electrical Engineering, UCL, and also leads the Bioelectronics Group. He has made outstanding contributions to improving safety and performance in integrated circuit design for active medical devices, such as spinal cord and brain stimulators. He has numerous collaborations for cross-disciplinary research, both within the U.K. and internationally. He has authored more than 350 articles in journals and international conference proceedings, several book chapters, and holds several patents. His research interests include analog and mixed-signal integrated circuits for biomedical, sensor, and signal processing applications. He is a fellow of the Institution of Engineering and Technology and a Chartered Engineer. He serves on the International Advisory Board for Physiological Measurement. He has served on the technical committees for a number of international conferences, including ICECS, ESSCIRC, and ISCAS. He was a co-recipient of a number of best paper awards and has graduated many Ph.D. students. He was an Associate Editor and the Deputy Editor-in-Chief of the IEEE Transactions on Circuits and Systems—II: Express Briefs, from 2006 to 2007 and from 2014 to 2015, respectively; and an Associate Editor and the Editor-in-Chief of the IEEE Transactions on Circuits and Systems—I: Regular Papers, from 2008 to 2009 and from 2016 to 2019, respectively. He is an Associate Editor of the IEEE Transactions on Biomedical Circuits and Systems.