

PAPER • OPEN ACCESS

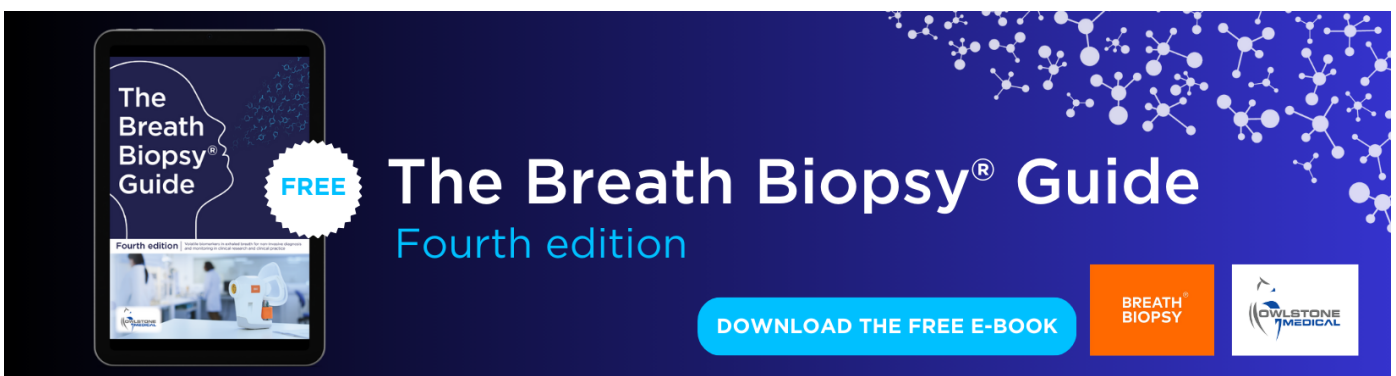
## SpikeDeep-classifier: a deep-learning based fully automatic offline spike sorting algorithm

To cite this article: Muhammad Saif-ur-Rehman *et al* 2021 *J. Neural Eng.* **18** 016009

View the [article online](#) for updates and enhancements.

You may also like

- [An unsupervised real-time spike sorting system based on optimized QSort](#)  
Yingjiang Wu, Ben-Zheng Li, Liyang Wang et al.
- [Robust spike classification based on frequency domain neural waveform features](#)  
Chenhui Yang, Yuan Yuan and Jennie Si
- [A data-driven spike sorting feature map for resolving spike overlap in the feature space](#)  
J Wouters, F Kloosterman and A Bertrand



The Breath Biopsy® Guide  
Fourth edition

FREE

DOWNLOAD THE FREE E-BOOK

BREATH BIOPSY

OWLSTONE MEDICAL



## PAPER

## OPEN ACCESS

## RECEIVED

18 September 2020

## ACCEPTED FOR PUBLICATION

9 November 2020

## PUBLISHED



5 February 2021

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



# SpikeDeep-classifier: a deep-learning based fully automatic offline spike sorting algorithm

Muhammad Saif-ur-Rehman<sup>1,2,3</sup> , Omair Ali<sup>1</sup>, Susanne Dyck<sup>1</sup>, Robin Lienkämper<sup>1</sup>, Marita Metzler<sup>1</sup>, Yaroslav Parpaley<sup>1</sup>, Jörg Wellmer<sup>1</sup>, Charles Liu<sup>4</sup>, Brian Lee<sup>4</sup>, Spencer Kellis<sup>5</sup> , Richard Andersen<sup>5</sup>, Ioannis Iossifidis<sup>3</sup>, Tobias Glasmachers<sup>6</sup> and Christian Klaes<sup>1</sup>

<sup>1</sup> Department of Neurosurgery, University Hospital, Knappschaftskrankenhaus Bochum GmbH, Ruhr-University Bochum, Bochum, Germany

<sup>2</sup> Department of Electrical Engineering and Information Technology, Ruhr-University Bochum, Bochum, Germany

<sup>3</sup> Institute of Informatics, University of Applied Sciences, Bottrop, Germany

<sup>4</sup> Neurorestoration Center and Department of Neurosurgery and Neurology, University of Southern California, Los Angeles, United States of America

<sup>5</sup> Division of Biology and Biomedical Engineering, CALTECH, Pasadena, United States of America

<sup>6</sup> Institute of Neuroinformatic, Ruhr-University Bochum, Bochum, Germany

**E-mail:** [muhammad.saif-ur-rehman@tu-dortmund.de](mailto:muhammad.saif-ur-rehman@tu-dortmund.de) and [christian.klaes@gmail.com](mailto:christian.klaes@gmail.com)

**Keywords:** tunable hyperparameters, deep-learning, supervised learning, unsupervised learning, automatic spike sorting

Supplementary material for this article is available [online](#)

## Abstract

**Objective.** Advancements in electrode design have resulted in micro-electrode arrays with hundreds of channels for single cell recordings. In the resulting electrophysiological recordings, each implanted electrode can record spike activity (SA) of one or more neurons along with background activity (BA). The aim of this study is to isolate SA of each neural source. This process is called spike sorting or spike classification. Advanced spike sorting algorithms are time consuming because of the human intervention at various stages of the pipeline. Current approaches lack generalization because the values of hyperparameters are not fixed, even for multiple recording sessions of the same subject. In this study, a fully automatic spike sorting algorithm called ‘SpikeDeep-Classifier’ is proposed. The values of hyperparameters remain fixed for all the evaluation data. **Approach.** The proposed approach is based on our previous study (SpikeDeeptector) and a novel background activity rejector (BAR), which are both supervised learning algorithms and an unsupervised learning algorithm (K-means). SpikeDeeptector and BAR are used to extract meaningful channels and remove BA from the extracted meaningful channels, respectively. The process of clustering becomes straight-forward once the BA is completely removed from the data. Then, K-means with a predefined maximum number of clusters is applied on the remaining data originating from neural sources only. Lastly, a similarity-based criterion and a threshold are used to keep distinct clusters and merge similar looking clusters. The proposed approach is called cluster accept or merge (CAOM) and it has only two hyperparameters (maximum number of clusters and similarity threshold) which are kept fixed for all the evaluation data after tuning. **Main results.** We compared the results of our algorithm with ground-truth labels. The algorithm is evaluated on data of human patients and publicly available labeled non-human primates (NHPs) datasets. The average accuracy of BAR on datasets of human patients is 92.3% which is further reduced to 88.03% after (K-means + CAOM). In addition, the average accuracy of BAR on a publicly available labeled dataset of NHPs is 95.40% which reduces to 86.95% after (K-mean + CAOM). Lastly, we compared the performance of the SpikeDeep-Classifier with two human experts, where SpikeDeep-Classifier has produced comparable results. **Significance.** The SpikeDeep-Classifier is evaluated on the datasets of multiple recording sessions of different species, different brain areas

and different electrode types without further retraining. The results demonstrate that ‘SpikeDeep-Classifier’ possesses the ability to generalize well on a versatile dataset and henceforth provides a generalized and fully automated solution to offline spike sorting.

**Clinical trial registration number** The clinical trial registration number for patients implanted with the Utah array is **NCT 01849822**. For the epilepsy patients, approval from the local ethics committee at the Ruhr-University Bochum, Germany, was obtained prior to implantation. The Clinical trial registration number for the epilepsy patients implanted with microwires is **16–5670**.

## 1. Introduction

Understanding complex behaviors and network properties of the brain requires access to the activity of a large population of neurons. One way to access the activity of single neurons in the intact brain is the implantation of small but dense micro-electrode arrays. A state-of-the-art single micro-electrode array can contain hundreds of channels which enable us to record single unit activity (SUA) of hundreds of neurons (Frey *et al* 2009, Lambacher *et al* 2011, Spira and Hai 2013, Berényi *et al* 2013, Harris *et al* 2016). Nowadays, it is a common practice to implant multiple micro-electrode arrays and record SA from more than one site, simultaneously (Aflalo *et al* 2015, Klaes *et al* 2015, Ajiboye *et al* 2017, Choi *et al* 2018). However, recorded data is usually contaminated with BA, and in addition to that it is also possible that a single channel records the activities of more than one neuron. Hence, the biggest constraint for any further analysis is to extract and isolate the activity of each single neuron in the presence of background noise. This process is called spike sorting.

The process of spike sorting is usually either manual or semi-automatic (Abeles and Goldstein 1977, Lewicki 1998, Gibson *et al* 2012). The process of manual or semi-automatic spike sorting involves human curation at various stages of a spike sorting pipeline. As a result, this process becomes labor-intensive and highly time-consuming. Therefore, these techniques could never compete with an increasing amount of data resulting from highly dense micro-electrode arrays and long duration recording sessions. Another major drawback is human subjectivity, which can lead to inconsistent results when the same data is analyzed by a different person (Wood *et al* 2004). A further limitation of manual or semi-automatic spike sorting is that the quality completely depends on the skills of the human curator. Therefore, fully automatic spike sorting has always been a major area of interest (Spacek *et al* 2009, Takekawa *et al* 2012, Bongard *et al* 2014, Carlson *et al* 2014, Pachitariu *et al* 2016, Chung *et al* 2017, Yger *et al* 2018).

A spike sorting pipeline involves at first, the pre-processing of the raw time series by applying band-pass filtering and then using a threshold to extract qualified events. It is possible that some of the extracted qualified events represent background noise

and others SUA of surrounding neurons. Finally, to assign labels to each of the extracted qualified events, clustering is used (Lewicki 1998; Einevoll *et al* 2012, Marre *et al* 2012). Mostly, at least one of these processes is performed manually. However, there exist few methods that offer an automatic solution to the spike sorting problem. A robust and automatic solution is presented in (Oliyynyk *et al* 2012). The presented solution is based on singular value decomposition (SVD) and Fuzzy C-mean (FCM) classification. Alternatively, an automatic solution for spike detection and sorting is presented in (Shalchyan *et al* 2012). This study employs an unsupervised learning method, which finds the occurrence of spike events with wavelet shrinkage denoising in combination with multiscale edge detection using wavelet-based manifestation variable. Here, a correlation-based similarity criterion was defined to update the wavelet selection during clustering. Another solution is presented in (Tiganj and Mboup 2012). In this study, spike classification is performed using an iterative independent component analysis (ICA) and a deflation-based method in two nested loops. Spiking activity of each neuron is first singled out and then deflated from the recording sessions. In another study (Pillow *et al* 2013) a model-based spike sorting algorithm is proposed, which explicitly accounts for the superposition of spike waveforms. In (Nguyen *et al* 2015), diffusion maps (DM) are used for feature extraction and K-means clustering in combination with silhouette statistics, which automatically determine the number of neural units and their activities on a channel. Similarly, in (Quiroga *et al* 2004) wavelet transformation (WT) are used for feature extraction and superparamagnetic clustering offers an automatic spike sorting solution. Recently, a solution based on density-based clustering algorithm called ISO-SPLIT is proposed in (Chung *et al* 2017).

For multiple channels recordings it has been reported that a considerable fraction of channels of implanted arrays record only BA (Lewicki 1998; Hill *et al* 2011; Klaes *et al* 2015, Rey *et al* 2015). Other channels which record SA also record a substantial amount of BA. This BA is a combination of technical artifacts and neural activity far away from the tip of the recording electrodes. The positions of recording electrodes can also be slightly perturbed by movements. Therefore, the resulting signal is of non-stationary nature (time variant). The dynamics of the

recorded signal can change from session to session. Therefore, it is a challenge to model all the resultant dynamics.

Recently, it has been reported in (Chung *et al* 2017) that the unavailability of a general solution to spike sorting is mainly because of non-stationary behavior of the background activity (BA). In this study, we show that spike sorting becomes an ordinary clustering problem upon the complete removal of BA from the source signal. We proposed a generalized solution for this problem, based on our previous study (Saif-ur-rehman *et al* 2019) in conjunction with a novel algorithm called background activity rejector (BAR). By generalized solution, we mean that the algorithm is trained once on a versatile dataset. Later, it can be applied to a dataset recorded from a different brain area, different species, different electrode type and recording hardware without any re-training. We show that it is possible to completely remove background noise with a huge amount of labeled training data and by stacking two different deep-learning methods.

Deep-learning methods, especially supervised learning algorithms in combination with a huge number of labeled training examples, have proven worthwhile in the field of computer vision (Krizhevsky *et al* 2012; Girshick *et al* 2014, Girshick 2015, Ren *et al* 2017). Recently, convolutional neural networks (CNNs) alone have become the major source of success in many computer vision applications (Girshick *et al* 2014, Girshick 2015, Guo *et al* 2017, Ren *et al* 2017). CNNs, because of their shared-weights architecture and translation in-variance characteristics can learn temporal and spatial patterns (Lecun *et al* 1998). However, one of the primary reasons behind the success story of CNNs is the availability of huge publicly available datasets (Jia *et al* 2009, Stallkamp *et al* 2011). Recently, deep-learning algorithms have gained attention in neuroscience community. In our previous study (Saif-ur-rehman *et al* 2019), we proposed a deep-learning based method to extract meaningful channels from large implanted microelectrode arrays. Based on our study (Saif-ur-rehman *et al* 2019), a deep-learning based algorithm is used to extract feature vectors for online invasive brain computer interface (BCI) applications in another study (Issar *et al* 2020). In (Rácz *et al* 2020) a deep-learning method is proposed for spike detection and sorting. We strongly believe that results of many neuroscience problems including spike sorting and online BCI decoding can be improved in the presence of large labeled datasets. Therefore, in this study, we collected and labeled a large dataset. Our dataset includes the data from our own lab and from different collaborators. Later, we also used some publicly available labeled datasets to validate our results (Shi *et al* 2013, Buneo *et al* 2016, Lawlor *et al* 2018).

In this study, we aimed to provide a universal solution to the offline spike sorting problem by

using large labeled dataset in conjunction with state-of-the-art deep learning algorithms. Our algorithm ‘SpikeDeep-Classifer’ is based on a novel pipeline, which is a set of supervised and unsupervised learning methods. First, a supervised learning method is used to select the meaningful channels as proposed in study (Saif-ur-rehman *et al* 2019). Then, we employ another supervised learning method to remove the remaining BA from the selected channels. After the complete removal of BA, we employ k-means clustering (Lloyd 1957, Macqueen 1967) with a predefined number of maximum clusters on the feature vectors extracted using principal component analysis (PCA) (Jolliffe and Cadima 2016). Lastly, a similarity-based algorithm is used to automatically accept distant clusters and merge similar clusters.

## 2. Materials & method

### 2.1. Approvals

We used a dataset collected from two tetraplegic patients implanted with two Utah arrays each and epilepsy patients implanted with depth-electrodes in preparation for surgery. Utah array patients were implanted in posterior parietal cortex (PPC). These Patients were recruited for two different BCI studies (Aflalo *et al* 2015, Klaes *et al* 2015). These studies took place after the institutional approvals held by the California Institute of Technology, and University of Southern California. Detailed approval information is available in (Aflalo *et al* 2015, Klaes *et al* 2015). Epilepsy patients were implanted with depth-electrodes/micro-wires in hippocampus in the form of bundles. These patients were implanted for medical reasons and have participated voluntarily. We obtained the approval for epilepsy patients from the Ruhr-University ethics committee. In addition, we also used publicly available datasets. Approval of each dataset is available in (Shi *et al* 2013, Lawlor *et al* 2018).

### 2.2. Demographic and implantation details

In this study, we used the data recorded from four human patients and four NHPs (male rhesus macaques). Human patients were implanted either with Utah arrays or with micro-wires using a Behnke-Fried configuration (Fried *et al* 1999). Two of the human patients were implanted with micro-wires, which were coupled in a group of eight individual platinum coated electrodes. The remaining two human patients were implanted with two Utah arrays. Each array contains 100 electrodes arranged in a grid of dimension  $10 \times 10$ . Further information about surgery and array placement is mentioned in (Aflalo *et al* 2015, Klaes *et al* 2015).

The NHP data was acquired from two different publicly available datasets provided by Collaborative Research in Computational Neuroscience (CRCNS) (Buneo *et al* 2016, Perich *et al* 2018).

**Table 1.** Subjects demographic and implantation details.

Specie	Subject ID	Sex	Age(year)	Place of implantation	Number of recordings	Number of implanted electrodes
Humans	U1	Male	32	Posterior parietal cortex	7	192 (2-Utah array)
Humans	U2	Male	63	Posterior parietal cortex	7	192 (2-Utah array)
Humans	M1	Female	49	Anterior hippocampus	1	16 (Micro-wires)
Humans	M2	Female	26	Anterior hippocampus	1	16 (Micro-wires)
NHPs	MM	Male	—	Primary motor cortex and premotor cortex	1	192 (2-Utah array)
NHPs	MT	Male	—	Primary motor cortex	1	192 (2-Utah array)
NHPs	X/B	Male	—	Superior parietal lobule	10	Single micro-electrodes

The dataset reported in (Perich *et al* 2018) recorded from two macaques using implanted Utah arrays. The second NHP dataset (Shi *et al* 2013) recorded from two rhesus macaques (X and B) using single micro-electrodes. However, this dataset was merged into one.

Further demographic and implantation details are mentioned in table 1.

### 2.3. Data collection & preprocessing

Human data was recorded using a neural signal processor (NSP) (Blackrock microsystems, Salt Lake City, UT, USA). Here, we aim for end-to-end learning (Lecun *et al* 1998, Glasmachers 2017), first for meaningful (neural) channel selection and then to discard BA. Preprocessing involves extraction of events from the raw data based on a thresholding procedure (Lewicki 1998) which is performed by the NSP hardware. Here, we used standard settings. Events are extracted using an automatic amplitude thresholding method that is applied to the high-pass filtered signal with a cut-off frequency of 250 Hz. Amplitudes that cross the threshold, which is set to be  $-4.5$  times the root-mean-square of the signal, are considered an event. We used the given settings because same setting was used previously for online BCI decoding (Klaes *et al* 2015) to extract the events corresponding to spike. For each event a waveform consisting of 48 samples, consisting of the event itself, 15 samples before the event and 32 samples after the event, is extracted and passed on for further analysis.

The first NHPs dataset (Perich *et al* 2018) was recorded using the Blackrock NSP (Blackrock microsystems Salt Lake City, UT, USA). The dataset contains 48 sampled preprocessed labeled events. Further details are available in (Perich *et al* 2018). The second NHPs dataset (Buneo *et al* 2016) was recorded with a Plexon NSP (Plexon Inc. Dallas, TX, United States). The dataset contains 32 sampled events, which were then resampled to 48 using MATLAB's resample function. The resample function performs rate conversion uniformly from one sample rate to another sample rate. This function has three input parameters: original event, desired frequency, original frequency. Original event represents 32 sampled

waveforms, desired frequency = 45 000 Hz, and original frequency = 30 000 Hz. The function outputs a uniformly resampled event (48 sampled). Further information is available in (Buneo *et al* 2016).

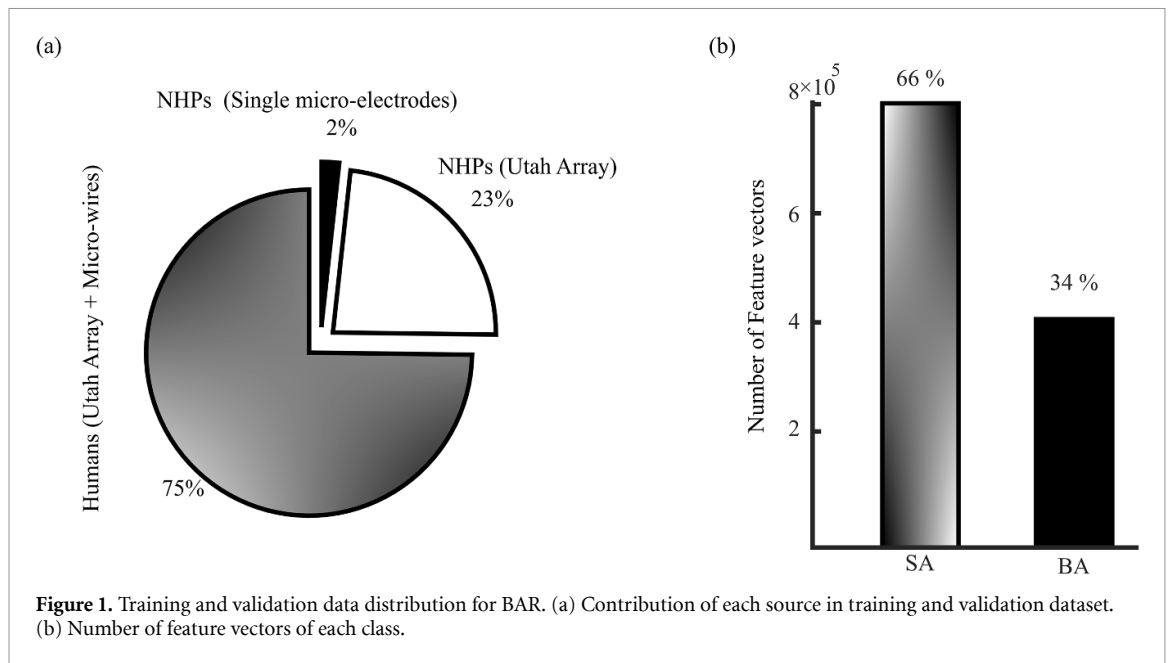
### 2.4. Data labeling

The proposed spike sorting pipeline is a combination of supervised and unsupervised learning algorithms. Supervised learning is gradient-based and minimizes the defined cost function by comparing predicted output and true output. Therefore, labeled training data is required. We labeled the given event either as 'SA' or 'BA'. Events representing action potentials (neural activity) are labeled as SA. Contrarily, events representing background activities (muscle artifacts, noise) are labeled as BA. The process of labeling is done in a semi-automatic way using a Gaussian mixture model (GMM) and careful visual inspection. A detailed explanation of the labeling process is provided in a previous study (Saif-ur-rehman *et al* 2019).

### 2.5. BAR data distribution for training and validation

We used the pretrained model of SpikeDeeptector for selecting meaningful channels. The other supervised learning method in the proposed spike sorting pipeline is a novel algorithm called BAR. It takes a single event from a meaningful channel and predicts it as SA or BA. For training the BAR, we collected data from human patients and NHPs. We considered four recording sessions from each patient (U1 and U2) implanted with Utah arrays and one recording session from a patient (M1) implanted with microwires. From NHPs, we considered data from one recording of a subject (MT) implanted with a Utah array. We also used data of six random days of subjects (X/B) implanted with single micro-electrodes. The distribution of data from all the sources is shown in figure 1(a). Figure 1(b) shows the distribution of data from each class. The total training examples of class BA are 410 584. The training examples of class SA are almost twice in number compared to the training examples of class BA. Consequently, to avoid a bias during training, we randomly chose 400 000 examples from each class. These examples are further





divided into ‘training dataset’ and ‘validation dataset’. We used 70% of total examples to compile the training dataset and the remaining 30% of the examples to compile the validation dataset. Training data is used to update the parameters of the BAR model during training process. However, validation data is not used for training but to monitor the performance of the algorithm on unseen data during training. We also used validation data to avoid overfitting by using early stopping criteria. The training process is terminated, if the validation error of six consecutive epochs increased or remained the same. Later, we evaluated the resulting trained model of BAR on the separate ‘evaluation dataset’. The evaluation dataset is gathered from both Utah array patients, microwire patient, NHP implanted with Utah array, and NHP implanted with single microelectrodes. Further detail of evaluation dataset is explained in result BAR.

## 2.6. SpikeDeep-classifier algorithm

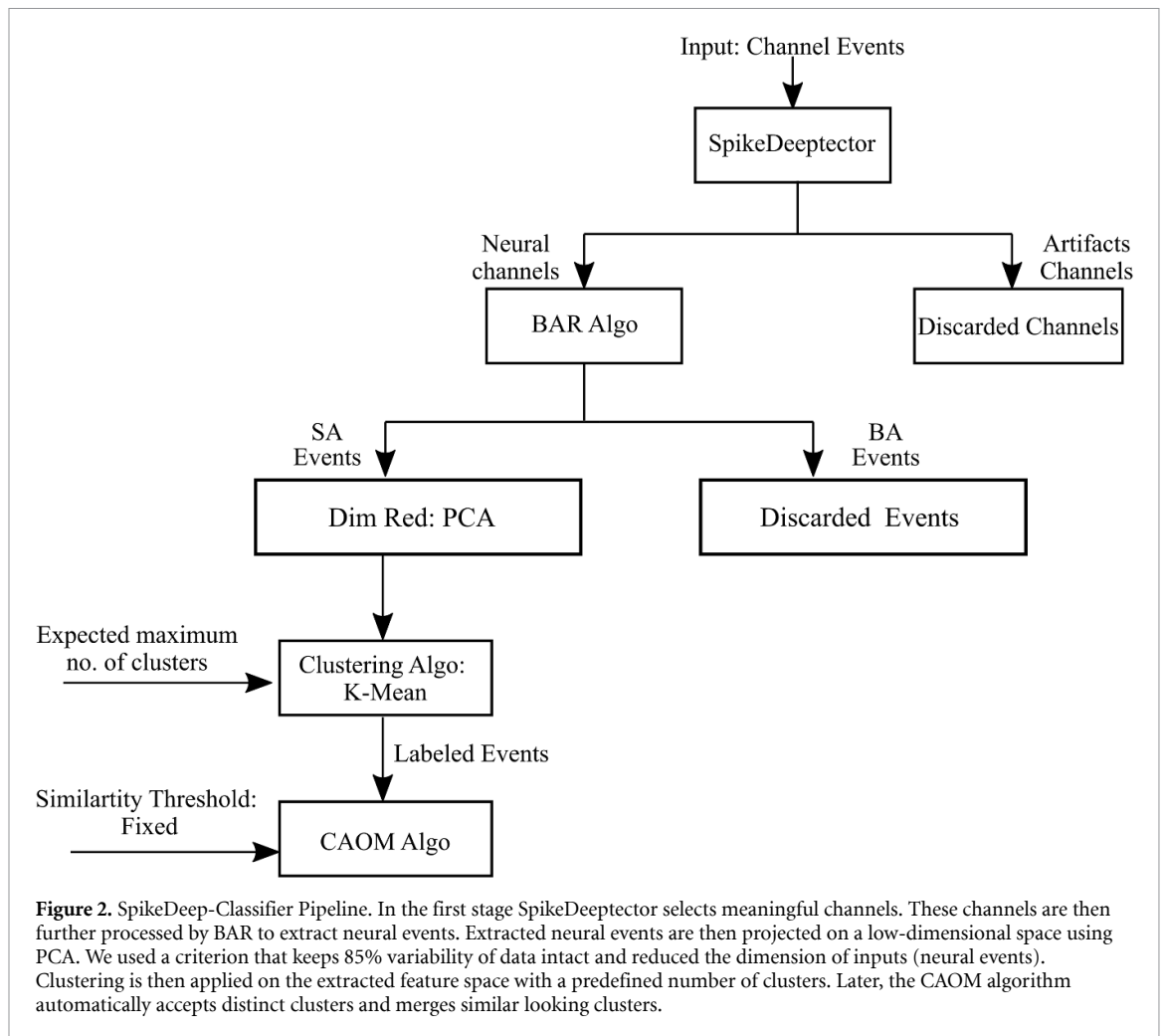
In this study, we propose an offline automatic spike sorter called SpikeDeep-Classifier. The architecture of SpikeDeep-Classifier is shown in figure 2. We completely removed BA by stacking SpikeDeeptector (Saif-ur-rehman *et al* 2019) and BAR. Both algorithms are based on supervised learning principles, therefore they require labeled training data to optimize learnable parameters iteratively. Extraction of meaningful data is very critical in many neuroscience applications including BCI applications and spike sorting. It has been shown that spike detection is the first and most pivotal step in neuro-prosthetic applications (Noc *et al* 2018). The next stage in this pipeline is dimensionality reduction of events corresponding to neural data using PCA, which is one of the standard algorithms for dimensionality reduction in spike sorting applications. Here, instead of using

the first two principal components for clustering, we defined a criterion that keeps most of the variability in the data and gives us the resulting principal components. We then employed a clustering algorithm on the extracted features (from PCA) of events representing neural data. We showed that after the removal of BA using BAR, spike sorting can be done with a very simple clustering algorithm e.g. K-mean with a pre-defined maximum number of clusters. Later, cluster accept or merge (CAOM) algorithm used a similarity-based criterion to merge similar looking clusters and accept distinct clusters as separate units.

We evaluated SpikeDeep-Classifier on different recording sessions of different subjects using the same trained model of supervised learning algorithms and with the same, fixed values of the other two parameters (expected maximum number of clusters & similarity threshold). These two hyperparameters are tuned by visual inspection and then manually changing. The process of tuning hyperparameters is done on a few recording sessions (not used for evaluation). After finding the optimized values of the hyperparameters, they are kept fixed for all the recording sessions for evaluation. The results show that SpikeDeep-Classifier provides accuracy comparable to a human expert.

## 2.7. SpikeDeeptector

The process of mapping raw signals into decision space is shown in figure 2. SpikeDeeptector is the first building block of the SpikeDeep-Classifier pipeline. The goal of SpikeDeeptector is to select the channels recording neural data and discard the channels recording only noise. In a previous study (Saif-ur-rehman *et al* 2019) it was shown that SpikeDeeptector can do such discrimination. There, we introduced a novel way to construct a feature vector by

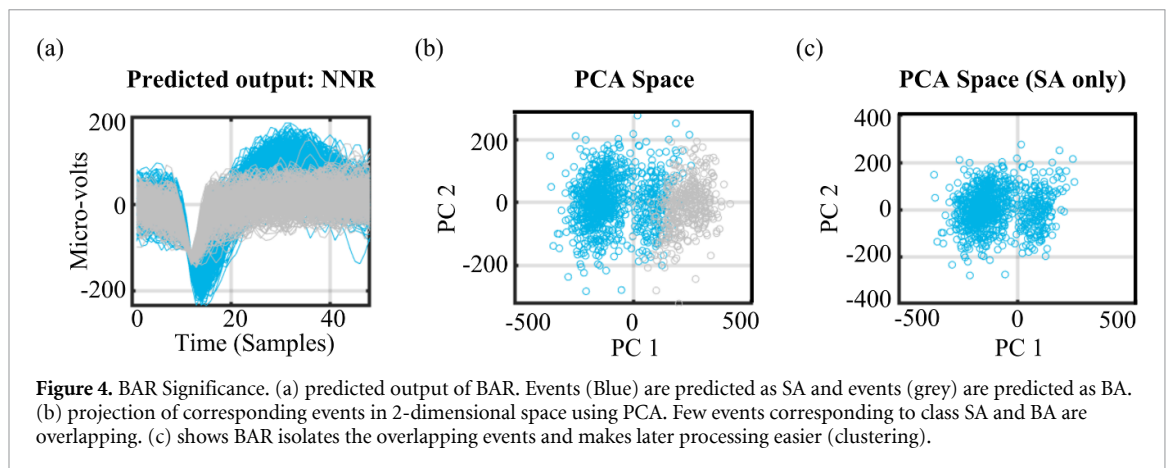
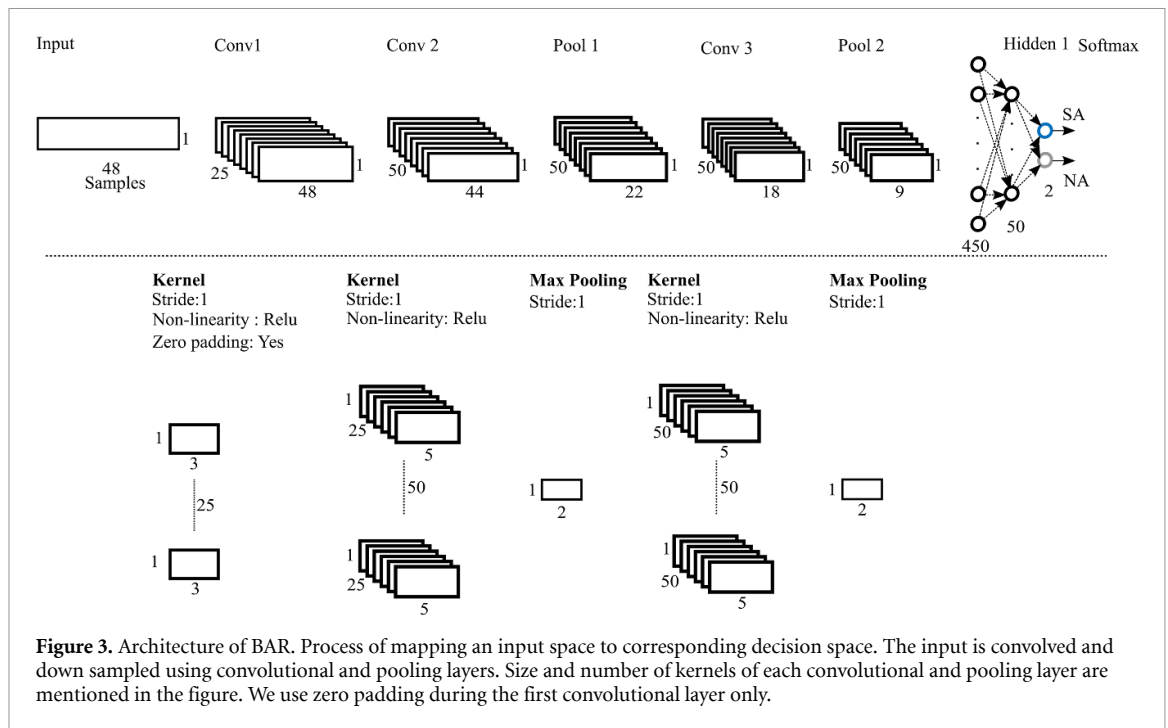


concatenating the batch of waveforms. We showed that the novel way of feature vector construction allows contextual learning and helps SpikeDeepDetector to aggregate the statistics of the inputs in a better way. SpikeDeepDetector is based on the standard architecture of CNNs. We used batch normalization and dropout as regularization techniques to avoid overfitting. Additionally, we minimized the regularized cross entropy cost by adding a L2 regularization term. The parameters of the defined architecture of SpikeDeepDetector were updated using mini-batch gradient descent with momentum. We used the data of only six recording sessions of one human patient implanted with an Utah array for training. Later, the trained model is evaluated on the data of 130 recording sessions, collected from two human patients implanted with Utah arrays and six human patients implanted with microwires. The reported classification accuracy of human patients implanted with Utah arrays is 96.7% and the human patients implanted with microwires 98.9%. The SpikeDeepDetector predicts the labels of the given feature vectors. However, the main goal of that study was to assign the label to the given channel as neural or artifact. Hence, we used a statistical criterion to assign a

label  $y(\text{channel})_{\text{pred}}$  to the given channel by calculating the mode of the predicted outputs  $y_{\text{pred}}$  of all the feature vectors of the given channel. Detailed description training and evaluation SpikeDeepDetector can be found in the **Materials and Methods** and **Results** section of our study (Saif-ur-rehman *et al* 2019). The trained model of SpikeDeepDetector is available on github and can be downloaded using the following link. <https://github.com/saifhanjra/SpikeDeepDetector/tree/master/EvaluateTrainedModel>

## 2.8. Background activity rejector (BAR)

SpikeDeepDetector provides the list of channels recording neural activities. These channels also record a considerable amount of BA. We aim to detect and discard all the events corresponding to BA from the channels SA (see figure 4). The BAR can be used as a pre-processing step for a clustering algorithm. It can isolate the overlapping events corresponding to SA and BA (see figure 4(b)). As a result, it simplifies further steps for spike sorting as shown in figure 4(c). In this case, spike sorting becomes an ordinary clustering problem after the removal of overlapped events.



To achieve this goal, we made use of available labeled training data and designed a supervised learning method based on the standard architecture of convolutional neural networks (Krizhevsky *et al* 2012; Guo *et al* 2017), as shown in figure 3. CNNs use shared weights which enable translation invariance and as a result produce more generic feature. These learned features are also robust against time delays and advancements in spike occurrence. Here, we used 1D CNNs because we are interested to learn the temporal pattern only.

BAR takes an event consisting of 48 samples as an input, and processes it through 3 convolutional layers, two pooling layers, a fully connected layer and finally classifies it as a 'BA' or as a 'SA' using a Softmax classifier, as shown in figure 3. At each convolutional layer, each kernel is convolved across the width of the input volume and then slides with stride = 1. This results in 1D convolved feature maps. Then, non-linearity is introduced using an activation layer. Here, we used

rectified linear units (ReLU)  $f(x) = \max(x, 0)$  (Nair and Hinton 2010). Except for the first convolutional layer, each convolutional layer is followed by a pooling layer. The goal of pooling is to discard unnecessary information. Here, we used max pooling. The size of each kernel and pooling layer is mentioned in figure 3.

We minimized a regularized cross-entropy cost function. To this end we added an L2 regularization term to the cross-entropy cost function. In addition, we also used batch normalization to standardize intermediate outputs of BAR to zero mean and unit variance for the training inputs in each mini batch. We used the same optimization algorithm (mini-batch gradient descent with momentum) and tuned the hyperparameters in the same way as reported in our previous study (Saif-ur-rehman *et al* 2019).

We trained a robust BAR using; the data of two species, five subjects, six brain areas, three different types of electrodes and two recording systems. The distribution of training data is shown in figure 1(b).



## 2.9. Dimensionality reduction

Dimensionality reduction is usually performed using unsupervised learning algorithms. Spike sorting algorithms try to eliminate redundant features and construct feature vectors for clustering algorithms. PCA is one of the most used algorithms by the community (Lewicki 1998, Adamos *et al* 2008; Souza *et al* 2018). PCA constructs low dimensional feature vectors by doing eigenvalue or SVD of the covariance matrix constructed from the presented data. Most spike sorting algorithms project high dimensional events onto the corresponding 2D or 3D principal component space using the eigenvectors. The PCA algorithm ensures that this 2D or 3D projection captures the highest variability in the presented data. However, it is possible that this low dimensional projection does not capture enough discriminatory power. Therefore, in this study we apply a criterion that keeps a certain amount of variability of the presented data intact and constructs low dimensional feature vectors. Here, we select the number of principal components (features) so that 85% of variability of the data remains intact. The criterion of keeping 85% variability intact resulted in at most seven or eight principal components. We used this criterion only for clustering. For visualization purposes, we considered the first and second principal components.

## 2.10. Clustering method

An important part of this study is to show that spike sorting can be casted as an ordinary clustering problem upon the removal of BA. Usually, neural data generated further away from the tips of the recording electrodes and BA overlap as shown in figures 4(a) and (b). Therefore, it is challenging for any clustering algorithm to recognize them as separate clusters. However, supervised learning algorithms are very powerful models. Particularly, deep learning algorithms can learn hidden patterns. For this reason, we trained a deep learning algorithm to isolate BA as shown figure 4(b). After the removal of BA, clustering becomes trivial, as shown in figure 4(c). Two clusters are quite distinct from each other and are easy to identify as such. Even the simplest clustering algorithm like K-mean can perform well as shown in figure 4(c). Figures 4(b) and (c) presents visualization in 2D space. However, the clustering algorithm is provided with more than two PCA features by applying the criteria explained in **Dimensionality Reduction**.

We used k-means as a clustering algorithm with a squared Euclidean distance metric and K-means++ algorithm for initializing the centers of the defined number of clusters. We defined K as the maximum number of expected clusters, which we empirically determined to be 3. In a later step we accept or reject the clusters as different neurons.

## 2.11. Cluster accept or merge algorithm (CAOM)

We used K-means clustering and defined K as a maximum number of expected clusters on one channel. Since most channels record activity of one or two neurons, we need a method to reject or merge clusters at need. Here, we introduced a very simple method to accept distinct clusters and merge similar looking clusters. We are considering the data of multiple species, recorded from different brain areas using different recording hardware and different types of implanted electrodes. It is possible that recorded data can be on different scales. Therefore, we first normalized the data using Z-normalization (Patro and Sahu 2015) as a preprocessing step. Z-normalization ensures that all the features have zero mean and standard deviation equals to one. Then, we measured the similarity between each cluster. Hence, we compared the mean Euclidean distance of each cluster. We either merge two clusters with minimal distance less than the defined threshold and keep the remaining clusters unaffected, or all the defined clusters are accepted as representing independent sources (if the mean Euclidean distance of all the clusters is greater than defined threshold). In the case of merging two clusters, the new mean of the merged clusters is calculated and compared with the remaining cluster means. This process of merging clusters is repeated unless the mean Euclidean distance of each cluster from each other is greater than a defined threshold.

The threshold distance is the hyperparameter of the CAOM algorithm. We tuned the value of 'threshold distance' empirically by visual inspection and fixed it to 5.5. Later, we used the same value for all recording sessions used for evaluation.

We used 'Deep learning' and 'Neural Networks' toolboxes of MATLAB (The MathWorks, Inc) to define and train the SpikeDeeptector and BAR models. The source code of SpikeDeep-Classifier is available online and can be downloaded using the following link. <https://github.com/saifhanjra/SpikeDeepClassifier>

## 3. Results

### 3.1. Evaluation metrics

We reported classification accuracy for SpikeDeeptector and BAR. In addition, to ensure transparency, we also reported recall for BAR. Mathematically, equations (1) and (2) represent accuracy and recall, respectively.

$$\text{Accuracy} = \left( \frac{\text{Number of correct predictions}}{\text{Total number of examples}} \right) \times 100 \quad (1)$$

$$\text{Recall} = \left( \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \right) \times 100 \quad (2)$$

**Table 2.** Cumulative Performance evaluation of SpikeDeeptector on the data recorded from human patients and NHPs during multiple recording sessions.

Subject group	Number of sessions	Neural channels (neural channels/total channels)	Artifact channels (artifacts channels/total channels)	False positives	False negatives
Utah array (Humans—U1 and U2)	6	109/576	467/576	2	0
Microwires (Humans—M2)	1	15/16	1/16	0	0
Utah array (NHP—MM)	1	95/96	1/96	0	1
Single microelectrodes (NHP—X/B)	4	4/4	0	0	0

The evaluation performance of the CAOM is represented using classification accuracy and Rand index. Rand index is the measure of the similarity between two clustering algorithms. It also represents the measure of the percentage of correct decisions made by the algorithm. Mathematically, equation (3) represents the Rand index in percentages.

$$\text{Rand index} = \left( \frac{\text{True positives} + \text{True negatives}}{\text{True positives} + \text{False positives} + \text{True negatives} + \text{False negatives}} \right) \quad (3)$$

### 3.2. Evaluation datasets

SpikeDeep-Classifier is a pipeline that presents a universal solution to the spike sorting problem. We evaluated the SpikeDeep-Classifier pipeline on the data of three human patients and two NHPs. Two human patients (U1 & U2) were implanted with Utah arrays and for each patient three recording sessions were considered for evaluation; the third human patient (M2) was implanted with microwires and only one recording session was available. Similarly, two NHPs were either implanted with two Utah arrays or single microelectrodes. One recording session of an Utah array subject (MM) and four recording sessions of another subject (X/B) are considered for evaluation.

### 3.3. SpikeDeeptector

In this study, we did not train the model of SpikeDeeptector. We used the pretrained (previously trained) model of SpikeDeeptector that we used for our previously published study (Saif-ur-rehman *et al* 2019). We evaluated the pre-trained model of SpikeDeeptector on the data of four subject groups and multiple recording sessions as shown in table 2. SpikeDeeptector has wrongly classified only 3 channels

out of 692 channels, which shows SpikeDeeptector has a good quality of generalization. In addition, we also highlighted the consistent performance of SpikeDeeptector by evaluating it on each session individually. This evaluation aspect is evident on all three different types of recording sessions with few, some and several channels recording neural activities. Evaluation performance of SpikeDeeptector on each type of recording session is reported in (supplementary table 9 (available online at [stacks.iop.org/JNE/18/016009/mmedia](https://stacks.iop.org/JNE/18/016009/mmedia))).

### 3.4. Background activity rejector (BAR)

SpikeDeeptector correctly identifies the channels with neural activities with accuracy of 99.6% (see table 2, only 3 wrong predictions out of total 692 channels). Here, we employed the trained model of BAR and evaluated on the channels which have been identified as neural channels by SpikeDeeptector.

We evaluated the trained model of BAR on the data of all seven evaluation recording sessions of three human patients. Subjects were implanted with either Utah arrays or microwires in different areas of the brain (PPC or Hippocampus). We reported the evaluation accuracy in a confusion matrix as shown in table 3. BAR provides 93.4% recall on the feature vectors of class ‘SA’ and 86.4% recall on the feature vector of class ‘BA’. Data distribution between the two classes is unbalanced with 83.6% of data representing class ‘SA’ and 16.4% of data representing class ‘BA’. The overall classification accuracy is 92.3%. In addition to cumulative performance on the data of all human subjects, we also reported the evaluation performance of BAR on each recording session individually (see section **supplementary material: Background activity rejector**). Performance of BAR during each individual recording session remains consistent as shown in supplementary tables 10 and 11 with a minimum and maximum reported accuracy of 88.9% and 95.4%.

**Table 3.** Evaluation performance of BAR on the data recorded from human patients implanted with Utah arrays and microwires. The confusion matrix reports the overall accuracy and the classification accuracy of each class.

Background activity Rejector (BAR)				
Predicted Labels	SA	384370 78.1%	11029 2.2%	97.2%
	BA	27041 5.5%	70096 14.2%	72.2%
		93.4%	86.4%	92.3%
		SA	BA	
		True Labels		

The<sup>7</sup> NHPs datasets contain the events that only correspond to class ‘spike activity’. Here, the classification accuracy of BAR on selected recording sessions of NHP (MM) implanted with a Utah array on premotor is 92.14% and on primary motor cortex is 99.03%. Similarly, the average classification accuracy of NHP (X/B) on four single electrodes is 95.04%.

#### 3.4.1. Visualization

We selected three different examples for visualization. Figure 5 shows the waveforms with associated ground truth labels and predicted labels along with the mean waveforms of each class and the projection of waveforms in 2D using PCA. For visualization, we showed the response of BAR on three different types of recording channels. Figure 5(a) shows the response of BAR on the channel where SA events and BA events are only partially overlapped in PCA space. However, in figure 5(b) SA events and BA events are almost completely overlapped in PCA space. Hence, it is a difficult task for a clustering algorithm to discriminate two clusters. Figure 5(c) shows another type of channel which records only a few events corresponding to BA. These few events hardly represent a separate cluster. In all the above explained conditions, BAR performs equally good (or even better) in comparison with (imperfect) ground truths.

#### 3.5. Significance of BAR: overlapping waveforms

Events representing spikes and BA can completely overlap as shown in figure 6(a). As a result, even humans can make mistakes during labeling (see figure 6(a)) where during labeling, a human curator missed a distinct unit by merging it with BA.

<sup>7</sup> The confusion matrix in Table 3 is not readable. Please resize it.

However, by considering BAR as a preprocessing step before clustering, the process of clustering becomes trivial, as shown in figure 6(b). Inclusion of BAR in the SpikeDeep-classifier pipeline successfully isolates the overlapped clusters. As a result, spike sorting can become an ordinary clustering problem (see figure 6).

We present a few more examples for visual insight in supplementary material (supplementary figures 11(a) and 12(a)). The presented examples show that BAR can facilitate the clustering process by removing overlapped events.

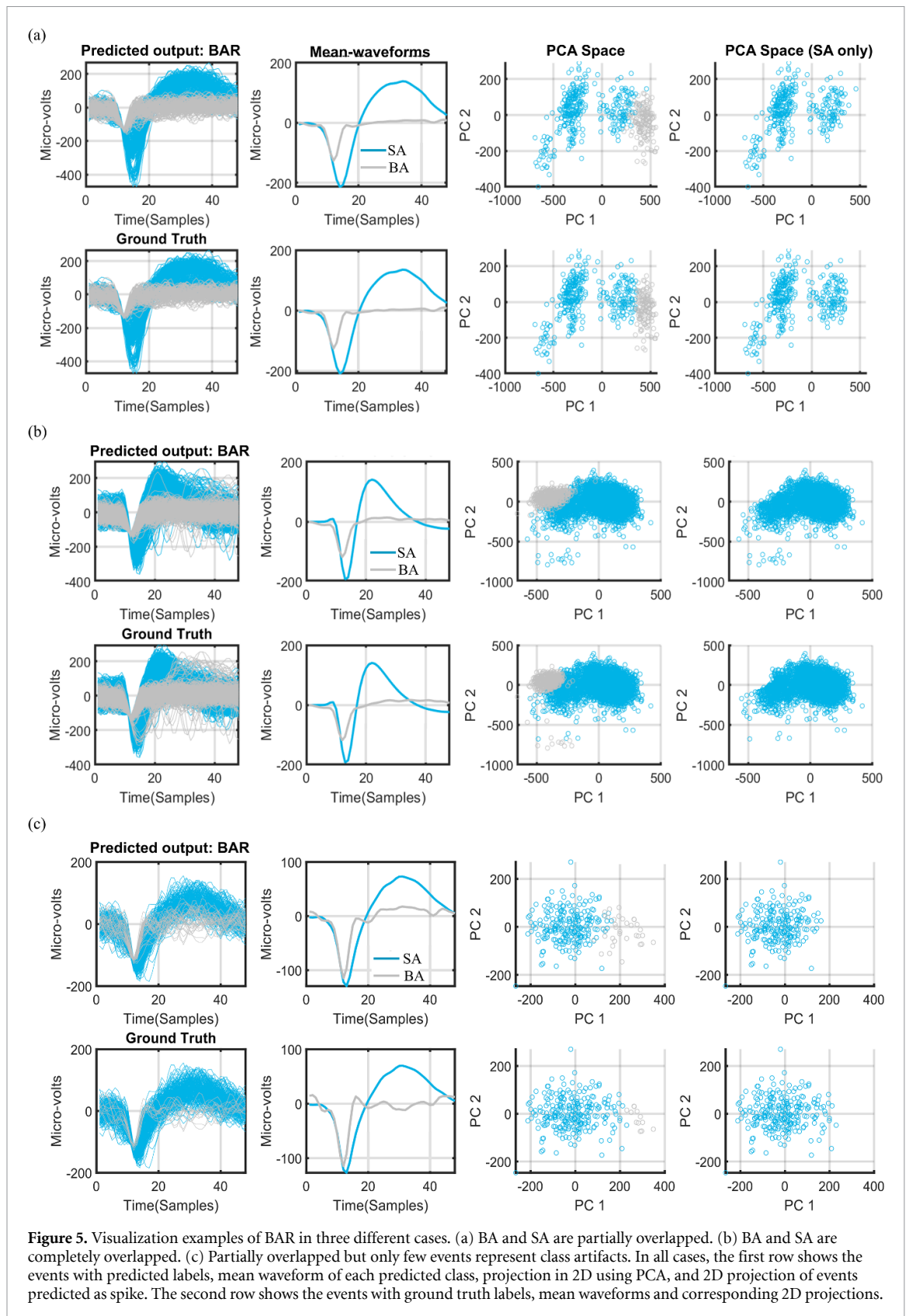
#### 3.6. Clustering & CAOM

SpikeDeeptector in conjunction with BAR nearly completely removes BA in two steps. After the removal of BA, the remaining data (SA) is used to identify the number of neural units present on a single channel. This process is taking place in two steps: the first step involves the process of clustering with a predefined maximum number of clusters and in the second step similarity between each cluster is measured as explained in section ‘CAOM’. Similar looking clusters are merged, and distinct looking clusters are treated as separate clusters (units). We have defined the maximum number of clusters as 3 and the similarity threshold as 5.5.

##### 3.6.1. Humans: Utah array subjects

We used K-means clustering (see section ‘Clustering Method’) and then CAOM (see section ‘Cluster accept or merge algorithm (CAOM)’) to accept or merge the clusters. We evaluated our methods on six recording sessions of human patients implanted with Utah arrays. Out of 576 channels only 109 channels were predicted as neural channels. Two predictions were false positives. Most of the channels either record one neural source or two neural sources on a channel. However, there were few channels, where three neural sources were recorded (see table 4). K-means clustering in conjunction with CAOM predicted the right number of clusters on most of the channels. Out of 107 channels only 8 channels were predicted with a different number of clusters than the ground truths. We use the Rand index to assess the quality of the clustering method. The Rand index is a measure of similarity between two data clustering methods. The value of the Rand index is between 0 and 1. 1 means that both clustering (ground truth, predicted) methods produced the exact same results, and 0 indicate that two data clustering methods completely disagree with each other. The achieved mean Rand index is more than 0.8 for any number of neural units on a channel (see table 4). Similarly, the achieved mean accuracy for any number of units is more than 87%.

Table 4 shows the cumulative performance of Clustering & CAOM on all the recording sessions of human patients implanted with Utah arrays. In addition to that we also show the performance of



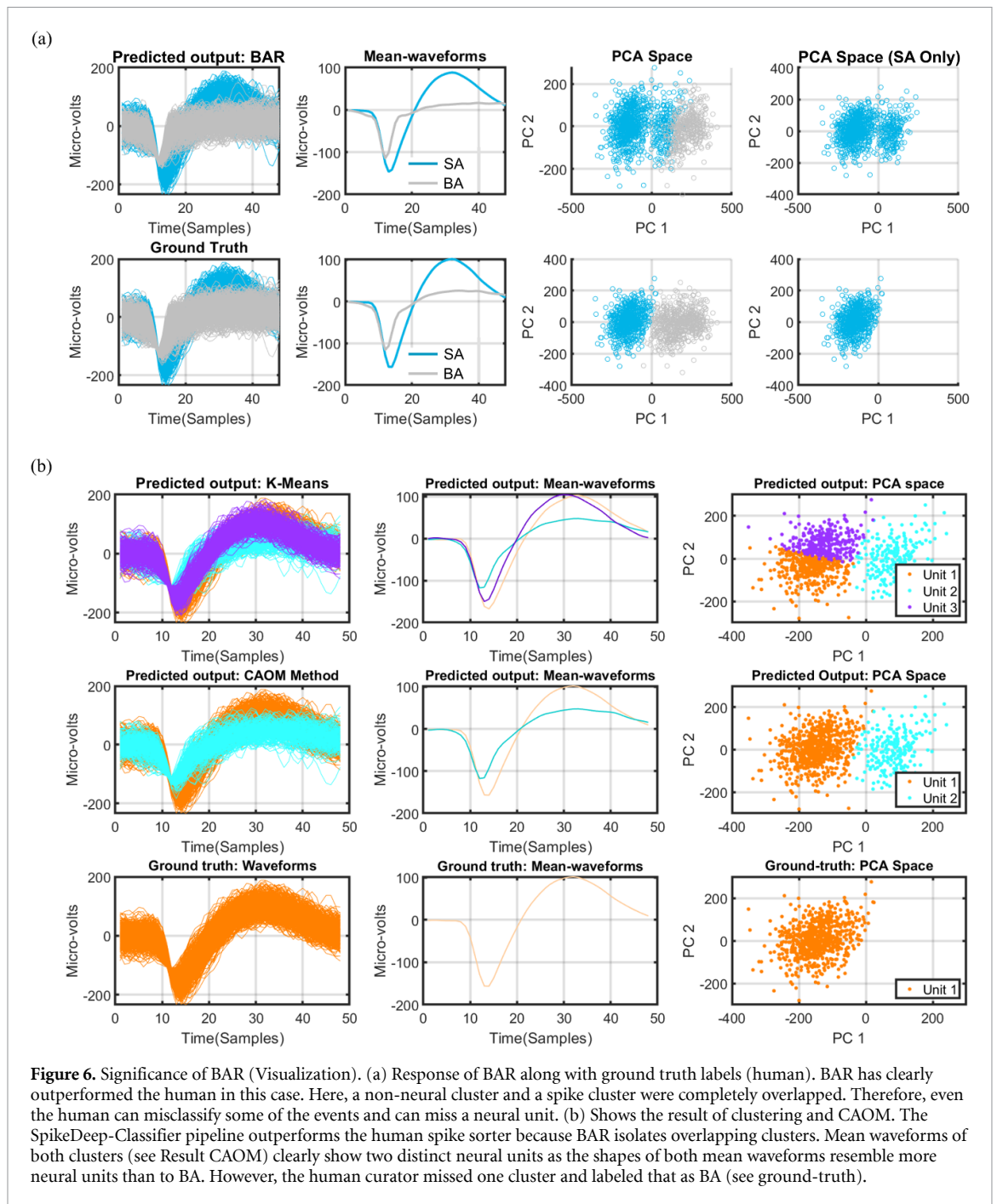
**Figure 5.** Visualization examples of BAR in three different cases. (a) BA and SA are partially overlapped. (b) BA and SA are completely overlapped. (c) Partially overlapped but only few events represent class artifacts. In all cases, the first row shows the events with predicted labels, mean waveform of each predicted class, projection in 2D using PCA, and 2D projection of events predicted as spike. The second row shows the events with ground truth labels, mean waveforms and corresponding 2D projections.

Clustering & CAOM on all individual recording sessions (see supplementary table 12). These recording sessions have different numbers of channels with different numbers of units. The performance of Clustering & CAOM remains consistent during all individual recording sessions (see supplementary table 12).

### 3.6.2. Visualization

We also present an example for visual inspection with three neural units on a channel (see figure 7). Figure 7(a) shows the predicted output of the K-means clustering algorithm with 3 clusters. The output of CAOM is shown in the second





**Figure 6.** Significance of BAR (Visualization). (a) Response of BAR along with ground truth labels (human). BAR has clearly outperformed the human in this case. Here, a non-neural cluster and a spike cluster were completely overlapped. Therefore, even the human can misclassify some of the events and can miss a neural unit. (b) Shows the result of clustering and CAOM. The SpikeDeep-Classifier pipeline outperforms the human spike sorter because BAR isolates overlapping clusters. Mean waveforms of both clusters (see Result CAOM) clearly show two distinct neural units as the shapes of both mean waveforms resemble more neural units than to BA. However, the human curator missed one cluster and labeled that as BA (see ground-truth).

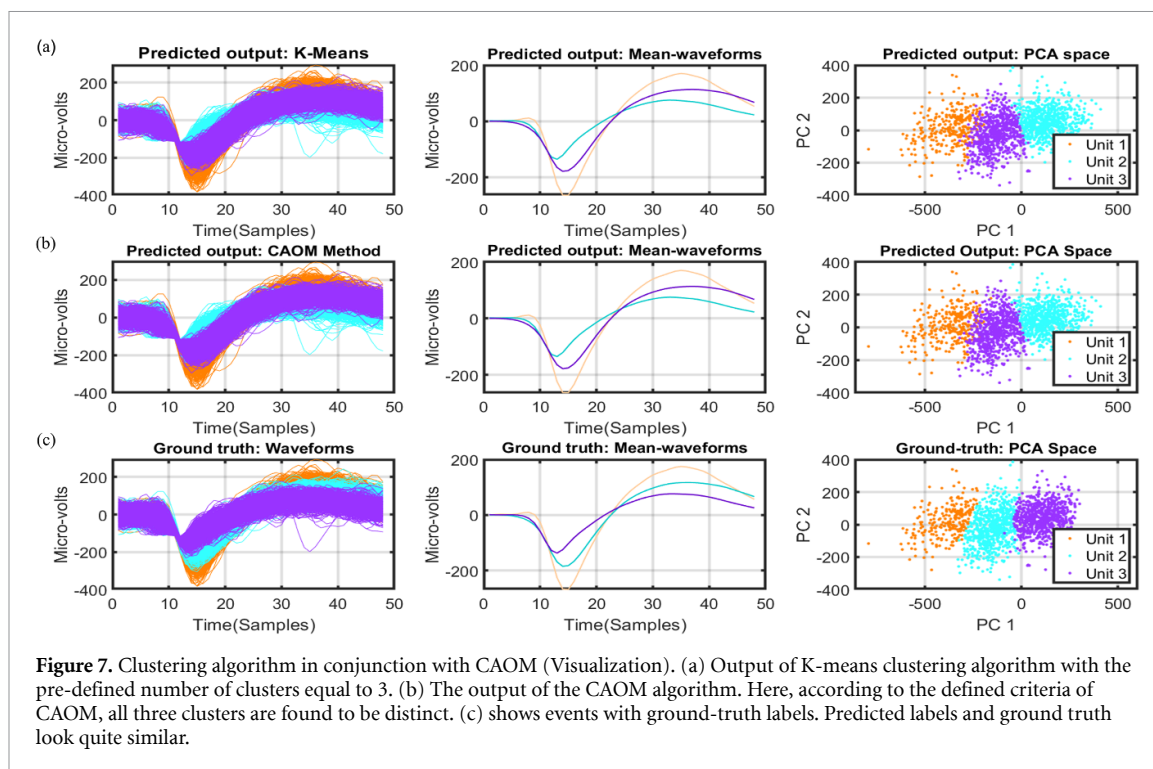
**Table 4.** Performance evaluation of clustering method & CAOM on the data of six recording sessions of human patients implanted with Utah arrays.

Number of units	No. of channels (true)	No. of channels (Pred.) (correct, wrong)	Rand index	Accuracy (%)
3	18	(18,1)	$0.84 \pm 0.08$	$87.76 \pm 5.17$
2	47	(44,5)	$0.85 \pm 0.08$	$89.09 \pm 6.69$
1	42	(37,2)	$0.81 \pm 0.12$	$87.25 \pm 11.01$

stage (figure 7(b)). Here the similarity between units is calculated based on the criteria explained in ‘CAOM’. All three clusters have been considered as distinct clusters. Figure 7(c) shows the ground truth waveforms, mean waveforms and the PCA

projection in 2D. Predicted outputs (figure 7(b)) and ground truths (figure 7(c)) look quite similar, which speaks for a high quality of the SpikeDeep-Classifier pipeline. We also present an example with two clusters (see supplementary figure 13(a)) and





another example with one unit on a channel (see supplementary figure 13(b)).

Similarly, figure 8 also provides visual insight into the performance of the SpikeDeep-Classifier pipeline. Figure 8 shows the output (PCA projection in 2D) of the clustering algorithm, CAOM and the ground truths. Figure 8(a) shows an example of three neural units on a channel, figure 8(b) shows an example with two neural units on a channel and figure 8(c) shows an example of a channel with one neural unit. In all mentioned cases the SpikeDeep-Classifier pipeline has not only been able to predict the correct number of neural units on a channel but also the predictions are very similar to the ground truth.

### 3.7. Clustering & CAOM: NHP Utah array

We aim for a universal solution to the spike sorting problem. For that reason, we first evaluated our trained models on data of multiple sessions of the same species (humans) where different kinds of electrodes were used. Additionally, we also evaluated the same trained model in multiple recording sessions of another species (NHPs) with different subjects implanted with different kinds of electrodes and different recording hardware.

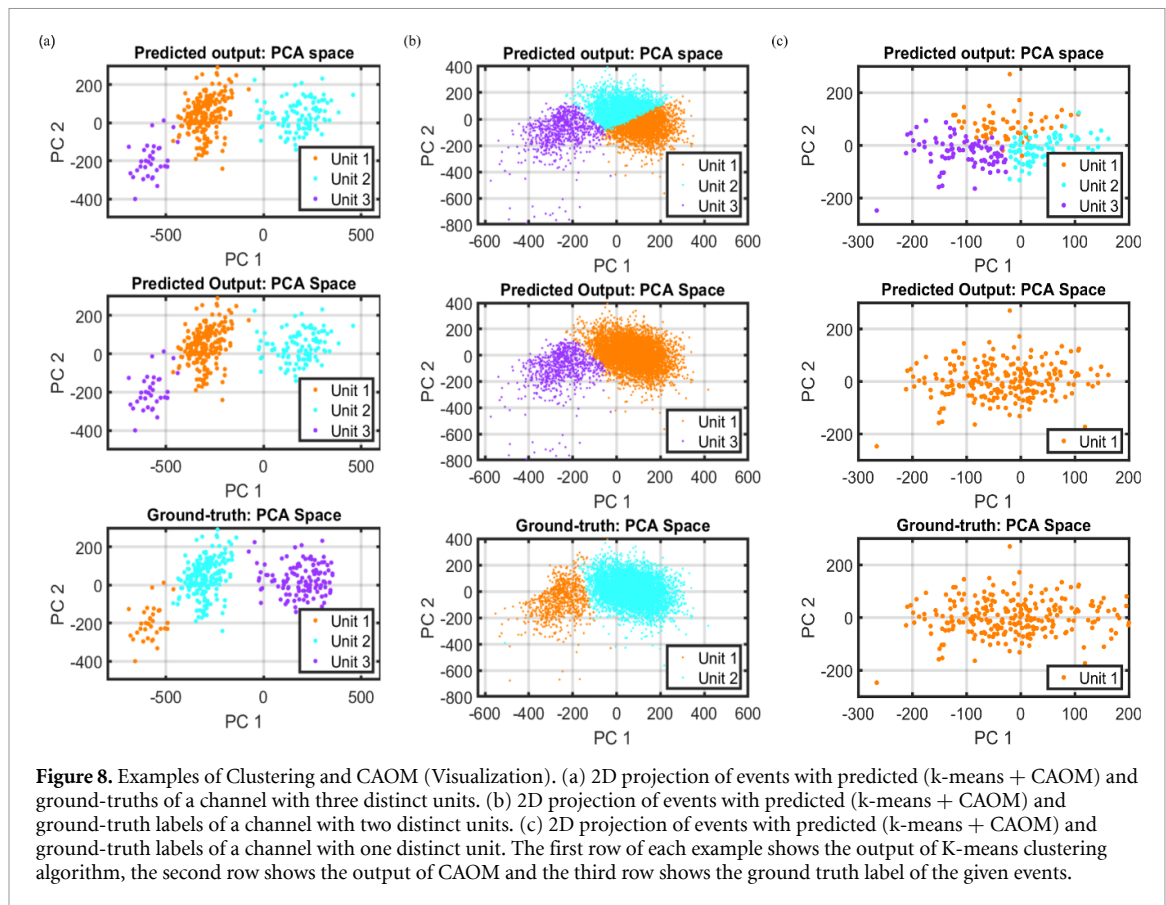
In this section, we will discuss the performance of Clustering & CAOM on the data of a NHP (MM) implanted with Utah arrays. We kept the parameters of the clustering algorithm and CAOM fixed (number of clusters = 3, similarity threshold = 5.5). In the recording session there is only one channel with four neural units (see table 5). In that case, our Clustering and CAOM method has been able to predict 3

clusters with a Rand index of 0.78 and a classification accuracy of 84.24%. Even though the results based on the provided ground truths are suboptimal, there are 12 out of 51 channels, where SpikeDeep-Classifier has predicted a different number of neural units as compared to the provided ground truths (See table 5). However, it is difficult to conclude about the correct number of neural units on most of these channels. Visual insights of these channels are provided in figures 9 and supplementary figure 14. Some of these channels are wrongly labeled (see figures 9(a), (b), (e) and supplementary figure 14(a)) and for some channels it not clear about the number of distinct units (see figures 9(d), (f) and supplementary 14(b), (d), (e), (f)), and the only two channels shown in figures 9(c) and supplementary 14(c) were misclassified by SpikeDeep-Classifier.

NHP subject MM was implanted with two Utah arrays in two different brain areas (premotor cortex & primary motor cortex). Here, we have reported the performance of SpikeDeep-Classifier of one array implanted in premotor cortex (see table 5). SpikeDeep-classifier performs equally well on the data of another brain area (primary motor cortex) (see Supplementary Material: Clustering and CAOM: Utah array NHP: supplementary table 13). The reason of reporting the performance of SpikeDeep-Classifier on both arrays separately is to evaluate the consistency of the algorithm.

#### 3.7.1. Visualization

We provided visual insight of a few correctly classified examples (number of clusters on a channel)



**Figure 8.** Examples of Clustering and CAOM (Visualization). (a) 2D projection of events with predicted (k-means + CAOM) and ground-truths of a channel with three distinct units. (b) 2D projection of events with predicted (k-means + CAOM) and ground-truth labels of a channel with two distinct units. (c) 2D projection of events with predicted (k-means + CAOM) and ground-truth labels of a channel with one distinct unit. The first row of each example shows the output of K-means clustering algorithm, the second row shows the output of CAOM and the third row shows the ground truth label of the given events.

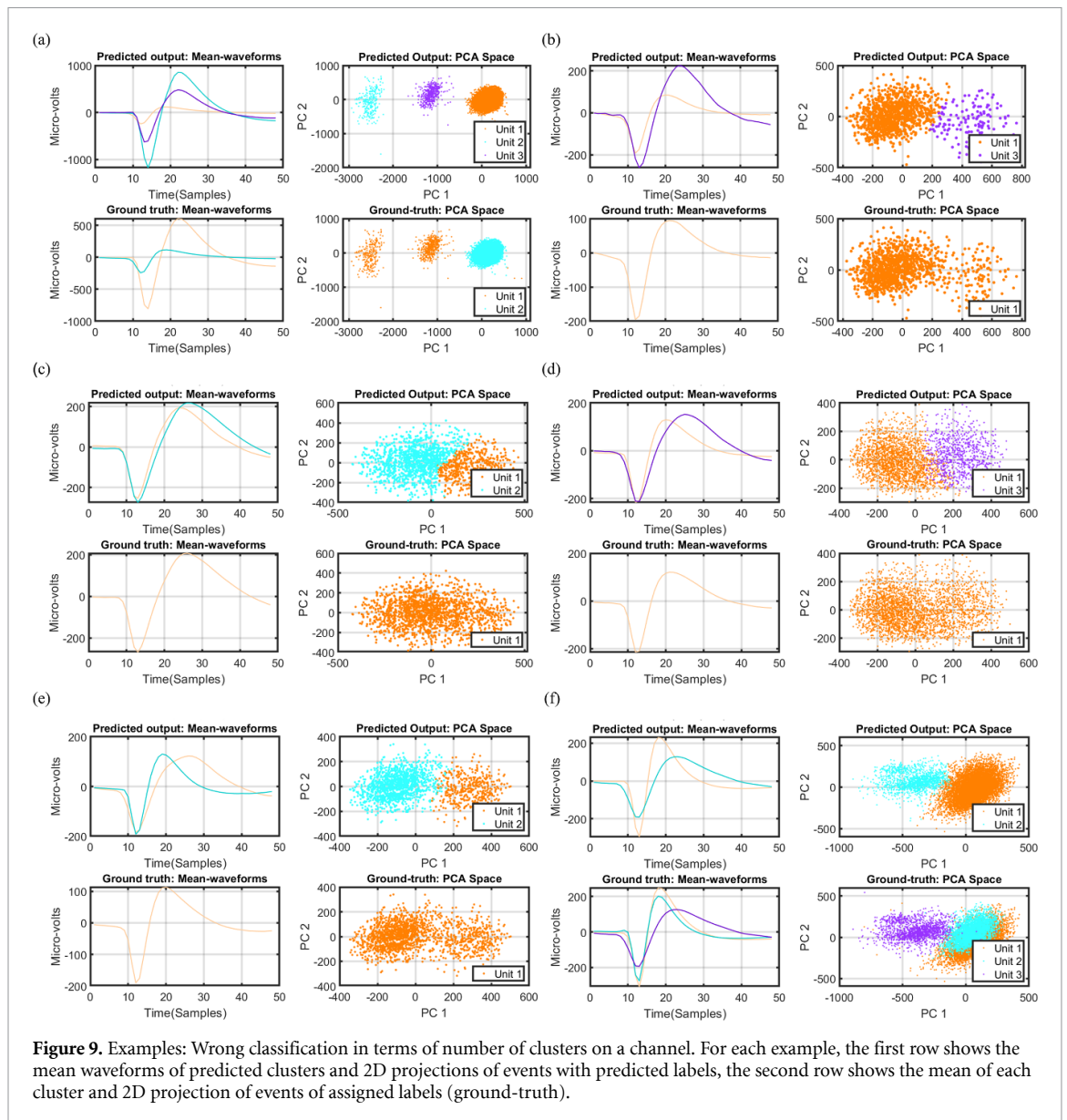
**Table 5.** Performance evaluation of the clustering method & CAOM on the recording session of a NHP (MM) implanted with a Utah array in Premotor cortex.

Number of Units	No. of Channels (True)	No. of Channels (Pred.) (Correct, wrong)	Rand index	Accuracy (%)
4	1	(0, 1)	0.78	84.24
3	7	(5, 3)	0.86 ± 0.10	84.70 ± 11.79
2	25	(22, 8)	0.88 ± 0.11	90.06 ± 10.20
1	18	(12, 0)	0.84 ± 0.18	89.02 ± 13.20

in section ‘**Results: Clustering & CAOM: Visualization**’. Here, we show a few wrongly classified examples for visual inspection (see figure 9). Even though the number of distinct units (clusters) in most of these examples are either debatable or wrongly labeled, the Rand index which is a measure of similarity between SpikeDeep-Classifier and the provided ground truths remains consistent. The average Rand index for different numbers of units on the channels is given in table 5. The Rand index in table 5 shows that the SpikeDeep-Classifier and the ground truth on average coincide with each other more than 84% in terms of the predicting class labels. However, at some instants SpikeDeep-Classifier can output debatable splits between clusters, as shown in figure 9(c). As both the predicted mean waveforms in this example look similar so it is hard to make a definitive split between them. Human curators labeled them as one class whereas the SpikeDeep-Classifier predicted two separate clusters.

### 3.8. Clustering & CAOM: NHP single micro-electrode

We showed that SpikeDeep-classifier provides a reliable solution to the spike sorting problem. We evaluated it on data of humans in multiple recording sessions using different kinds of electrodes (see tables 4 and supplementary table 12). Additionally, we evaluated it on data from an NHP implanted with Utah arrays in two different areas of the brain (tables 5 and supplementary table 13). Furthermore, we evaluated it on data of a second NHP implanted (X/B) with a single micro electrode. In this case, we did not have ground truth labels. Therefore, we show the performance of our SpikeDeep-classifier pipeline by presenting examples. We selected one example of each case. Figure 10(a) shows an example of a channel with one neural unit, figure 10(b) shows an example of a channel with two neural units, and figure 10(c) shows an example of a channel with three units.

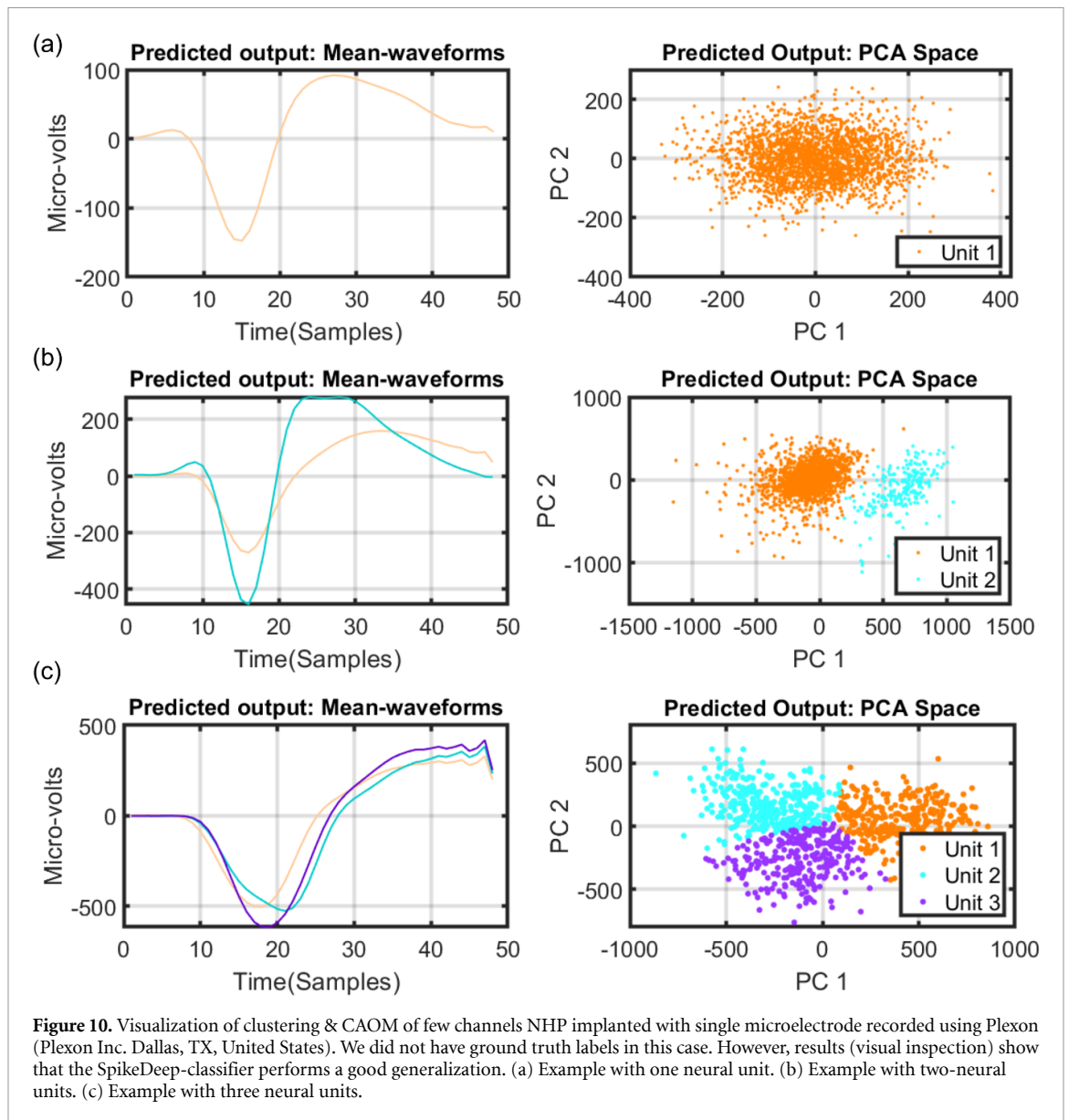


### 3.9. Performance comparison of spikedeep-classifier with human experts

We asked two members of our lab to annotate data for the comparison. We have randomly selected a recording session of subject (U1) and asked our lab members to annotate the data with the help of another in-house spike sorting algorithm. This in-house software provides visualization of all the detected events and their projection on a 2D PCA space of a channel. Hence, after the visual inspection of a channel, a curator defines the number of units along with their initial points for the clustering algorithm. Lastly, GMM was used for the clustering. In case the GMM does not provide satisfactory results, the curator has the possibility of manual spike sorting. All these steps were repeated for each channel of an implanted array.

For the selected recording session, both human expert 1 (H1) and human expert 2 (H2) predicted

19 out of 96 channels with one neural unit (see table 6). Both (H1 $\cap$ H2) agreed on 18 of those channels to have only one neural unit. Whereas SpikeDeep-Classifier predicted 14 out of 96 channels with one neural unit, which were also predicted (H1 $\cap$ H2 $\cap$ SpikeDeep-Classifier) as the channels with one neural unit by H1 and H2. The achieved mean Rand index with one neural unit for H1 and H2 is 0.8 and 0.81 respectively. Similarly, H1 assigned 7 channels and H2 assigned 8 channels with two neural units. Both (H1 $\cap$ H2) agreed on 5 of those channels. In this case SpikeDeep-Classifier predicted 12 channels with two neural units on it. Out of these 12 channels H1 and SpikeDeep-Classifier (H1 $\cap$ SpikeDeep-Classifier) agreed on 6 channels whereas H2 and SpikeDeep-Classifier (H2 $\cap$ SpikeDeep-Classifier) agreed on 7 channels. There were 5 channels common in all three (H1 $\cap$ H2 $\cap$ SpikeDeep-Classifier). The achieved mean



**Table 6. Performance comparison of SpikeDeep-Classifier with human experts.** First column shows the number of units predicted on a channel by human experts and SpikeDeep-Classifier. Second column shows the ID of human experts. Third column shows the number of channels predicted by the human corresponding to the number of neural units. Fourth column shows the total number of channels predicted by SpikeDeep-Classifier corresponding to the number of neural units. It also shows the right and wrongly predicted channels. Here the ground truth is taken as the predictions made by the human experts. Last column shows the Rand index corresponding to the H1 and H2 for the number of neural units on the channels.

No. of units on a channel	(Human expert Id)	No. of channels (annotated by human experts)	No. of channels (predicted) (correct, wrong)	Rand index (%)
1	H1	19	14 (14, 0)	$0.80 \pm 0.15$
	H2	19	14 (14, 0)	$0.81 \pm 0.13$
2	H1	7	12 (6, 6)	$0.84 \pm 0.10$
	H2	8	12 (7, 5)	$0.83 \pm 0.09$
3	H1	1	1 (0, 1)	0.74
	H2	0	1 (0, 1)	–

Rand index with two neural units for H1 and H2 is 0.84 and 0.83, respectively. Lastly, H1 assigned one channel with three units, which was predicted with 2 neural units by both H2 and SpikeDeep-Classifier.

#### 4. Discussion

In this study, we proposed a novel spike sorting pipeline called SpikeDeep-Classifier, which identifies



the number of neural units, along with their activities on a channel. We claimed that SpikeDeep-Classifier presents a generalized solution to the spike sorting problem. By generalized solution, we mean that the SpikeDeep-Classifier model (pre-trained and pre-tuned) has the ability to successfully perform spike sorting on the data of multiple recording sessions of multiple species, data recorded from different brain areas using different types of implanted electrodes with different recording hardware. To the best of our knowledge, there is no other method that presents such a generalized solution to the spike sorting problem. We validated our claim of providing a generalized solution to the spike sorting problem by evaluating SpikeDeep-Classifier model (pre-trained and pre-tuned) on versatile labeled datasets which include two self-labeled and two publicly available labeled datasets. The labeled datasets include the data from four human patients and four NHPs subjects (macaques), which were recorded from five different brain areas including PPC, anterior hippocampus, primary motor cortex, premotor cortex, and superior parietal lobule.

In addition, we used a publicly available simulated labeled dataset (Quiroga *et al* 2004) to discuss the performance comparison of SpikeDeep-classifier with few existing solutions. The dataset is available with Spike times and associated labels. The dataset contains a total of four examples. Each example is recorded at four different noise levels (0.05, 0.1, 0.15, 0.2). Further details of the dataset are available in (Quiroga *et al* 2004). Here, we compared our algorithm with the eight existing algorithms which present an automatic solution to the spike sorting problem. These algorithms are presented in (Nguyen *et al* 2015). The process of automatic spike sorting in the provided solution is taking place in two steps: feature vectors construction and automatic clustering. The feature vectors construction is taken place by either employing the wavelet transform (WT) or diffusion maps (DM). Then the three different solutions to automatic clustering are presented using superparamagnetic clustering (SPC), mean shift clustering algorithms, and K-means clustering. SPC is an automatic clustering algorithm based on the simulated interaction between each data point and its K nearest neighbors. Here, a range of temperature (hyperparameter) is required pre-specified to automatically determine the number of distinct clusters. Mean shift is an alternative algorithm to the SPC as it automatically selects the number of distinct clusters. However, it has band width as a hyperparameter. Temperature and bandwidth are required to be carefully tuned to achieve the optimal solution to the problem. As SPC and mean shift do not require prior determination of the number of clusters, the threshold of three clusters, which is equal to the real number (ground truth) of clusters is assigned. Once a temperature is fixed in SPC (or a band width is

selected in mean shift), if the number of automatically selected clusters is greater than 3, then the 3 largest overlapping with most overlapping are considered to calculate the accuracy. In mean shift, a range of bandwidth values are nominated, and the greatest accuracy is reported for the comparison. Lastly, K-means requires a predefined number of clusters for clustering. The number of predefined clusters are estimated and provided by either Silhouette statistics (SH) or gap statistic (GS). The numbers in the parentheses adjacent to values in the k-mean columns indicate the number of clusters, if distinct from 3 (see table 7). We used PCA for feature vectors extraction and CAOM in conjunction with K-means for automatic clustering. Table 7 presents the results of the different combination of the discussed feature extraction algorithms and automatic clustering solutions. Table 7 shows that our proposed solution has comprehensively outperformed 7 out of 8 algorithms in terms of mean classification accuracy. However, it provides comparable performance to (DM, SH+K-means). Nonetheless, SpikeDeep-Classifier performs better than (DM, SH+K-means) in terms of predicting the correct number of neural units on a channel. DM, SH+K-means makes 7 mistakes in predicting the correct number of neural units, however, SpikeDeep-Classifier made only 2 mistakes as shown in table 7.

This simulated dataset also contains an additional example which records the activity of bursting neurons. Here, the classification accuracy of CAOM + K-means is 92.18%.

We used BAR to discard the detected events corresponding to BA. Furthermore, BAR was compared with another method presented in (Quiroga *et al* 2004) on the above-mentioned simulated dataset. In (Quiroga *et al* 2004) spike detection was performed using an automatic amplitude thresholding after bandpass filtering the signal (300–6000 Hz, four pole Butterworth filter). The following setting of threshold (Thr) was used.

$$\text{Thr} = 4\sigma_n; \sigma_n = \text{median} \left\{ \frac{|x|}{0.6745} \right\} \quad (4)$$

Here,  $x$  is the band pass-filtered signal and  $\sigma_n$  is an estimate of the standard deviation of the background noise (Donoho and Johnstone 1994). The performance of spike detection algorithm and BAR is shown in table 8. The column 'No. of Spikes' shows the total number of events labeled as spikes where the overlapping spikes are mentioned in parenthesis. The brackets in misses and false positives columns show the number of events that were misclassified by the original method proposed in (Quiroga *et al* 2004) and BAR. The column 'Misses' shows the performance comparison of two methods in total as well as overlapping spikes. The dataset is available with ground truth, therefore we also employed BAR



Table 7. Performance Comparison of PCA, K-means and CAOM with existing automatic clustering algorithm for Spike sorting application.

Examples	Noise levels	No. Spikes	WT		WT		DM		DM		DM		PCA	
			SPC	Mean Shift	GS + K-means	WT	SPC	Mean shift	GS + K-means	DM	SH + K-means	CAOM + K-means		
Example 1	0.05	3514	58.56	99.22	77.96	80.72	64.48	93.33	79.69 (5)	85.66 (5)	99.23			
	0.10	3522	62.68	99.16	77.24	78.56	60.48	94.17	93.14	89.60 (4)	99.31			
	0.15	3477	71.19	99.20	72.81	90.03	89.14	94.14	75.43	86.12 (4)	99.13			
Example 2	0.20	3474	60.52	78.89	69.57	84.80	75.60	84.93	90.58	99.42	98.93			
	0.05	3410	67.79	65.88	73.37	67.07	79.85	89.20	93.92	90.81 (4)	97.44			
	0.10	3520	81.10	67.10	60.10	67.22	54.69	88.09	72.56	91.60 (4)	97.40			
Example 3	0.15	3411	70.74	67.63	59.80	67.25	48.97	85.03	68.86 (5)	92.48	92.23			
	0.20	3526	60.13	66.17	50.55	65.80	58.32	47.62	59.79 (5)	84.98	84.62			
	0.05	3383	63.17	34.14	73.71	65.89	66.04	87.99	73.16	87.47 (4)	97.22			
Example 4	0.10	3448	53.64	33.76	51.59	61.46	36.63	65.88	88.40	92.79	93.24			
	0.15	3472	66.26	33.76	47.40	41.66	33.96	33.76	88.13	86.17	57.68(2)			
	0.20	3414	52.86	34.53	47.89	48.62	34.61	34.53	75.24	75.98	67.70(2)			
Mean accuracy	0.05	3364	58.69	66.51	73.23	66.62	88.88	88.92	92.93	83.95 (5)	98.57			
	0.1	3462	55.36	66.98	73.42	66.96	90.70	89.52	94.09	94.22	98.65			
	0.15	3440	68.11	66.66	96.96	66.63	77.03	77.83	99.01	99.01	91.86			
Mean accuracy	0.2	3493	77.75	66.94	72.36	66.93	46.02	63.10	98.68	98.68	78.83(2)			
			64.28	65.41	67.37	67.89	62.83	76.12	83.97	89.94	90.75			

**Table 8.** Performance Comparison of BAR with another method of extracting neural events from the raw data.

Examples	Noise levels	No. of spikes	Misses (original, BAR)	False positives (original, BAR)
Example 1	0.05	3514 (785)	[17(193), 0(20)]	[711, 190]
	0.10	3522 (769)	[2(177), 2 (33)]	[57, 11]
	0.15	3477 (784)	[145(215), 43(57)]	[14, 5]
	0.20	3474 (796)	[714(275), 118(95)]	[10, 2]
Example 2	0.05	3410 (791)	[0(174), 0(0)]	[0, 0]
	0.10	3520 (826)	[0(191), 0(1)]	[2, 0]
	0.15	3411(763)	[10(173), 14(3)]	[1, 0]
	0.20	3526 (811)	[376(256), 44(18)]	[5, 1]
Example 3	0.05	3383 (767)	[1 (210), 3(133)]	[63, 18]
	0.10	3448 (810)	[0 (191),0(175)]	[10, 7]
	0.15	3472 (812)	[8 (203), 10 (190)]	[6, 16]
	0.20	3414 (790)	[184(219), 170(171)]	[2, 5]
Example 4	0.05	3364 (829)	[0(182), 0(2)]	[1, 0]
	0.1	3462 (720)	[0(152), 0(10)]	[5, 3]
	0.15	3440 (809)	[3(186), 5(23)]	[4, 0]
	0.2	3493 (777)	[262(228), 121(41)]	[2, 0]

on all the events labeled as spike and the results in table 8 confirm that false negatives (misses) can also be reduced. Table 8 also shows that performance of BAR remains consistent at different noise levels. Similarly, classification accuracy of BAR on the additional example which records the activity of bursting neurons is 99.53% with 13 wrong predictions of overlapped spikes and 3 wrong predictions of non-overlapped spikes. Thus, this confirms that adding BAR as a preprocessing step is quite logical and can help to improve the performance of the clustering algorithms.

The result of this study supports our hypothesis of providing a generalized solution to the spike sorting problem. Although SpikeDeep-classifier successfully provides a favorable solution to the spike sorting problem, there is still room of improvement in the CAOM algorithm, which automatically determines the number of neural units on a channel. In this study, we fixed the hyperparameters of CAOM to find the exact number of neural units on a recorded channel e.g. the maximum possible number of neural units on a channel is fixed to be 3. Some recording setups result in higher neural units per channel, but these cases typically do not occur in the intended chronically implanted electrode scenario. In such a case, the hyperparameters of CAOM are required to be re-tuned. Hence, CAOM can be replaced with an algorithm that provides a more general solution to determine the number of clusters. Here, we can propose Silhouette statistics as an alternative solution to CAOM because it provides comparable results to CAOM on the simulated labeled data set, as shown in table 7 in terms of classification accuracy. Silhouette is a more general solution for automatic clustering. However, CAOM performs better than Silhouette statistics in determining the number of neural units because it requires an educated guess about the maximum number of neural units.

An educated guess about the number of neural units on a channel provides a good initial point to CAOM. As a result, it not only determines the number of neural units more accurately but also the computational cost of the algorithm is reduced. The computational cost of the algorithm is an important factor for online applications with microelectrode arrays with hundreds of channels. It is also possible that several microelectrode arrays are implanted, simultaneously. Therefore, less computational cost of CAOM makes it more suitable as a feature extractor for BCI decoding applications. Silhouette on the other hand can be used as a replacement of CAOM if the spike sorting is required only on a few channels or if no time constraints are given.

Here, we proposed SpikeDeeptector in conjunction with BAR to first remove the unwanted channels and then BA from the selected channels. SpikeDeeptector enable contextual learning by concatenating the batch of events to construct feature vectors. The SpikeDeeptector was trained on the data of six recording sessions of only one human patient, as mentioned in our previous study (Saif-ur-rehman *et al* 2019). Here, we used the resultant pretrained model of SpikeDeeptector and evaluate it on the evaluation dataset. The SpikeDeeptector successfully select the channels recording neural activity with the average accuracy of 99.6%. Contrarily, BAR constructs the feature vector with one waveform. When it is trained on the data of one patient and evaluated on the data of multiple patients the accuracy drops to 85.8%. Therefore, for BAR we considered the fine tuning by retraining it on the dataset compiled from multiple subjects. By doing so, we raised the classification accuracy from 85.8% to accuracy of 92.3%. Henceforth, BAR alone should not be used to first select the meaningful channels and then to discard BA from the selected channels.

For the future, we aim to extend the SpikeDeepClassifier algorithm by proposing an online version

of it. In the online version of SpikeDeep-Classifer, we aim to replace the clustering algorithm with a supervised learning algorithm by using a large labeled dataset making the whole pipeline fully automatic and online.

## Acknowledgments

This study was funded by the Deutsche Forschungsgemeinschafts (DFG, German Research Foundation) under projects number KL 2990/1-1 – Emmy Noether Program, and 122679504 – SFB 874. We would also like to acknowledge Nina Misselwithz for providing clinical support during the recording sessions of the epilepsy patients.

## ORCID iDs

Muhammad Saif-ur-Rehman 

<https://orcid.org/0000-0003-1774-7330>

Spencer Kellis  <https://orcid.org/0000-0002-5158-1058>

## References

- Abeles M and Goldstein M 1977 Multispike train analysis *Proc. IEEE* **65** 762–73
- Adamos D A, Kosmidis E K and Theophilidis G 2008 Performance evaluation of PCA-based spike sorting algorithms *Comput. Methods Programs Biomed.* **91** 232–44
- Aflalo T, Kellis S, Klaes C, Lee B, Shi Y, Pejsa K and Andersen R 2015 Neurophysiology. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human *Science* **348** 906–10
- Ajiboye A, Willett F, Young D, Memberg W, Murphy B, Miller J and Kirsch R 2017 Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration *Lancet* **389** 1821–30
- Ballini M, Müller J, Livi P, Chen Y, Frey U, Stettler A and Hierlem A 2014 A 1024-channel cmos microelectrode array with 26,400 electrodes for recording and stimulation of electrogenic cells *in vitro IEEE J. Solid-State Circuits* **49** 2705–19
- Berényi A, Somogyvári Z, Nagy A, Roux L, Long J, Fujisawa S and Buzsáki G 2013 Large-scale, high-density (up to 512 channels) recording of local circuits in behaving animals *J. Neurophysiol.* **111** 1132–49
- Bongard M, Micol D and Fernández E 2014 NEV2lkit: a new open source tool for handling neuronal event files from multi-electrode recordings *Int. J. Neural Syst.* **24** 1450009
- Buneo C A, Shi Y, Apker G and Vangilder P 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 *CRCNS.org* (<https://doi.org/10.6080/K0CZ353K>)
- Carlson D E, Vogelstein J T, Wu Q, Wenzhao L, Zhou M, Stoetznner C R and Carin L 2014 Multichannel electrophysiological spike sorting via joint dictionary learning and mixture modeling *IEEE Trans. Biomed. Eng.* **61** 41–54
- Choi J, Kim S, Ryu R, Kim S and Sohn J 2018 Implantable neural probes for brain-machine interfaces? current developments and future prospects *Exp. Neurobiol.* **27** 453–71
- Chung J, Magland J, Barnett A, Tolosa V, Tooker A, Lee K and Greengard L 2017 A fully automated approach to spike sorting *Neuron* **95** 1381–94
- Donoho D L and Johnstone I M 1994 Ideal spatial adaptation by wavelet shrinkage *Biometrika* **81** 425–55
- Einevoll G T, Franke F, Hagen E, Pouzat C and D Harris K 2012 Towards reliable spike-train recordings from thousands of neurons with multielectrodes *Curr. Opin. Neurobiol.* **22** 11–17
- Frey U, Egert U, Heer F, Hafizovic S and Hierlemann A 2009 Microelectronic system for high-resolution mapping of extracellular electric fields applied to brain slices *Biosensors and Bioelectronics* **24** 2191–98
- Fried I, Wilson C, Maidment N, Engel J Jr, Behnke E, Fields T and Ackerson L 1999 Cerebral microdialysis combined with single-neuron and electroencephalographic recording in neurosurgical patients. Technical note. *J. Neurosurg.* 697–705
- Gibson S, Judy J and Marković D 2012 Spike sorting: the first step in decoding the brain: the first step in decoding the brain *IEEE Signal Process. Mag.* **29** 124–43
- Girshick R 2015 Fast R-CNN *IEEE Int. Conf. on Computer Vision (ICCV)* (Santiago: IEEE) S. 1440–1448
- Girshick R, Donahue J, Darrell T and Malik J 2014 Rich feature hierarchies for accurate object detection and semantic segmentation *IEEE Conf. on Computer Vision and Pattern Recognition* (Columbus, Ohio: IEEE) S. 580–587
- Glasmachers T 2017 Limits of End-to-End Learning (*arXiv: 1704.08305*)
- Guo T, Dong J, Li H and Gao Y 2017 Simple convolutional neural network on image classification *IEEE 2nd Int. Conf. on Big Data Analysis (ICBDA)* (Beijing: IEEE) (<https://doi.org/10.1109/ICBDA.2017.8078730>)
- Harris K, Quiroga R, Freeman J and Smith S 2016 Improving data quality in neuronal population recordings *Nat. Neurosci.* **19** 1165–74
- Hill D, Mehta S and Kleinfeld D 2011 Quality metrics to accompany spike sorting of extracellular signals *J. Neurosci.* **31** 8699–705
- Issar D, Williamson R, Khanna S and Smith M 2020 A neural network for online spike classification that improves decoding accuracy *J. Neurophysiol.* **123** 1472–85
- Jia D, Wei D, Richard S, Li-Jia L, Kai L and Li F 2009 ImageNet: A large-scale hierarchical image database *Computer Vision and Pattern Recognition, 2009* (Miami, FL, USA: IEEE) (<https://doi.org/10.1109/CVPR.2009.5206848>)
- Jolliffe I and Cadima J 2016 Principal component analysis: a review and recent developments *Phil. Trans. R. Soc. A* **374** 2065
- Klaes C, Kellis S, Aflalo T, Lee B, Pejas K, Shanfield K and Anderson R 2015 Hand shape representations in the human posterior parietal cortex *J. Neurosci.* **35** 15466–76
- Krizhevsky A, Sutskever I and Hinton G 2012 ImageNet Classification with Deep Convolutional Neural Networks *Commun. ACM* **60** 84–90
- Lambacher A, Vitzthum V, Zeitler R, Eickenscheidt M, Eversmann B, Thewes R and Fromherz P 2011 Identifying firing mammalian neurons in networks with high-resolution multi-transistor array (MTA) *Appl. Phys. A* **102** 1–11
- Lawlor P, Perich M, Miller L and Kording K 2018 Linear-nonlinear-time-warp-poisson models of neural activity *J. Comput. Neurosci.* **45** 173–91
- Lecun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition. *IEEE (IEEE) S.* 2278–2324
- Lewicki M 1998 A review of methods for spike sorting: the detection and classification of neural action potentials *Netw., Comput. Neural Syst.* **9** R53–R78
- Lloyd S 1957 Least squares quantization in PCM *Technical Report, Bell Lab RR-5497*
- Macqueen J 1967 *Some Methods for Classification and Analysis of Multivariate Observations* (Berkeley, CA: University of California Press)

- Marre O, Amodei D, Deshmukh N, Sadeghi K, Soo F, Holy T and Berry M 2012 Mapping a complete neural population in the Retina *J. Neurosci.* **32** 14859–73
- Nair V and Hinton G 2010 Rectified linear units improve restricted Boltzmann machines *27th, Int. Conf. on Machine Learning (Haifa, Israel)* S. 807–814
- Nguyen T, Bhatti A, Khosravi A, Haggag S, Creighton D and Nahavandi S 2015 Automatic spike sorting by unsupervised clustering with diffusion maps and silhouettes *Neurocomputing* **153** 199–210
- Noc E, Ciancio A L and Zollo L 2018 Spike detection: the first step towards an ENG-based neuroprotheses *J. Neurosci. Methods* **308** 294–308
- Oliynyk A, Bonifazzi C, Montani F and Fadiga L 2012 Automatic online spike sorting with singular value decomposition and fuzzy C-mean clustering *BMC Neurosci.* **13** 96
- Pachitariu M, Steinmetz N, Kadir S, Carandini M and Harris K 2016 Fast and accurate spike sorting of high-channel count probes with KiloSort *Advances in Neural Information Processing Systems 29 (NIPS 2016)* (Barcelona, Spain: NIPS) pp S. 4455–4463
- Patro S K and Sahu K K 2015 Normalization: a preprocessing stage (*arXiv: 1503/1503.06462*)
- Perich M G, Lawlor P N, Kording K P and Miller L E 2018 Extracellular neural recordings from macaque primary and dorsal premotor motor cortex *CRCNS.org.* (<https://doi.org/10.6080/K0FT8J72>)
- Pillow J, Shlens J, Chichilnisky E and Simoncelli E 2013 A model-based spike sorting algorithm for removing correlation artifacts in multi-neuron recordings *Plos One* **8** 5
- Quiroga R Q, Nadasdy Z and Ben-Shaul Y 2004 Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering *Neural Comput.* **16** 1661–87
- Rácz M, Liber C, Németh R, Fiáth R F, Rokai J, Harmati I and Márton G 2020 Spike detection and sorting with deep learning *J. Neural Eng.* **17** 1
- Ren R, He K, Girshick R and Sun J 2017 Faster R-CNN: towards real-time object detection with region proposal networks *IEEE Trans. Pattern Anal. Mach. Intell.* **39** 1137–49
- Rey H, Pedreira C and Quiroga Quian R 2015 Past, present and future of spike sorting techniques *Brain Res. Bull.* **119** 106–17
- Saif-ur-rehman M, Lienkämper R, Parpaley Y, Wellmer J, Liu C, Lee B and Klaes C 2019 SpikeDeeptector: a deep-learning based method for detection of neural spiking activity *J. Neural Eng.* **16** 5
- Sanchez J, Carmena J, Lebedev M, Nicolelis M, Harris J and Principe J 2004 Ascertaining the importance of neurons to develop better brain-machine interfaces *IEEE Trans. Biomed. Eng.* **51** 943–53
- Shalchyan V, Jensen W and Farina D 2012 Spike detection and clustering with unsupervised wavelet optimization in extracellular neural recordings *IEEE Trans. Biomed. Eng.* **2576–85**
- Shi Y, Apker G and Buneo C 2013 Multimodal representation of limb endpoint position in the posterior parietal cortex *J. Neurophysiol.* **109** 2097–107
- Souza B C, Lopes-dos-santos V, Babelo J and Tort A B 2018 Spike sorting with Gaussian mixture models (*bioRxiv*) (<https://doi.org/10.1101/248864>)
- Spacek M, Blanche T and Swindale N 2009 Python for Large-Scale Electrophysiology *Front. Neuroinf.* **2** 9
- Spira M and Hai A 2013 Multi-electrode array technologies for neuroscience and cardiology *Nat. Nanotechnol.* **8** 83–94
- Stallkamp J, Schlipsing M, Salmen J and Igel C 2011 The German traffic sign recognition benchmark: a multi-class classification competition *Neural Networks (IJCNN), The 2011 Int. Joint Conf. on Neural Networks* (San Jose, CA, USA: IEEE) (<https://doi.org/S.10.1109/IJCNN.2011.6033395>)
- Takekawa T, Isomura Y and Fukai T 2012 Spike sorting of heterogeneous neuron types by multimodality-weighted PCA and explicit robust variational Bayes *Front. Neuroinf.* **6** 5
- Tiganj Z and Mboup M 2012 Neural spike sorting using iterative ica and a deflation-based approach *J. Neural Eng.* **9** 6
- Wood F, Black M, Vargas-Irwin C, Fellows M and Donoghue J 2004 On the variability of manual spike sorting *IEEE Trans. Biomed. Eng.* **51** 912–9
- Yger P, Spampinato G, Esposito E, Lefebvre B, Deny S, Gardella C and Marre O 2018 A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings *in vitro* and *in vivo* *eLIFE* **7** e34518