International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Integrating Wavelet Transform Detection with Convolutional Neural Networks for Intelligent Neural Spike Sorting

Helat Ahmed Hussein[1], Ahmed Khorsheed Mohammed[2]

Submitted: 25/01/2024 **Revised**: 03/03/2024 **Accepted**: 11/03/2024

Abstract: Neural spike sorting is the basic process of understanding the brain's complex operation, which identifies and classifies spikes or electrical peaks emitted by one neuron. Manual spike sorting methods usually employ manual curation and heuristic algorithms, which can be difficult to use and time-consuming. This method requires a long time and a great deal of expertise. In automatic spike sorting, the spikes generated by various neurons are first detected and then classified automatically. This approach is faster and less labor-intensive than manual spike sorting. In the detection steps the traditional method used is threshold detection but, in this method, there are many spikes can be missed if the threshold high, or many background noise can be detected as spikes if the threshold values so low therefor using wavelet transform methods in detections step more accurate detecting the spikes and make it visualized from the background noise .This article presents a new method involving the joint application of Wavelet Transform Detection and Convolutional Neural Networks (CNNs) in order to expedite the process of intelligent spike sorting. Wavelet transform improves spike detection accuracy in this regard by allowing spikes from background noise to be effectively separated. The CNNs, trained successfully to spike clustering, perform quick and precise classification of the spikes of neurons. Integration of the present techniques will be more precise and effective for the sorting spike of brain supporting the progress in neuroscience studies and brain to the machine interface.

Keywords*: Neural Spike Sorting, spikes clustering, stationary wavelet, Deep learning, Convolution Neural Networks CNN.*

1. Introduction

Examining the electrical behaviors of neurons forms the foundation for investigating brain functionalities. Communication among brain neurons occurs through the transmission of electrical impulses known as action potentials or "spikes" [1]. Understanding the connections among diverse neurons is crucial in progressing our comprehension of brain science and neural engineering [2]. The primary task of spike sorting is entails isolating the spikes originating from individual neurons identified by the extracellular electrodes before interpreting the transmitted data through these connections [2]. This process strives to support specific spikes with individual neurons, considering that each electrode records the extracellular field involving the spike activities of several neighboring neurons. Known as spike sorting [3], this technique centers on identifying unique spike waveforms. This assertion is grounded in the fact that the spike shape of each neuron is individual, influenced by factors such as its dendritic structure, spatial layout, and alignment in relation to the recording position [4].Clearly, the standard spike sorting process involves four fundamental steps, with the first step beginning with the application of a bandpass filter to the recorded raw data for the identification of spikes. The second step is Spike detection which consists of recognizing spikes using techniques such as specifying an amplitude threshold [5] or executing enhanced approaches such as wavelet transforms [6]. Following spike detection, extracting distinguishing features from these identified spikes is common, often utilizing methods such as principal component analysis (PCA) [7] and wavelet transform coefficients [8],[9]. The final step is organizing these data points within the feature space, which is vital to delineate clusters associated with individual neurons. Various techniques, encompassing both traditional and cutting-edge approaches, have been applied to achieve this objective, including [10], k-means clustering, and Gaussian mixture models [11]. Organizing spikes into separate clusters representing individual neurons is commonly conducted through manual or semi-automated methods, which can be time-intensive and susceptible to mistakes [12]. In contrast, fully automated spike sorting seeks to mechanize the entire process of spike categorization without the need for human involvement [13]. Hence, manually reassigning incorrectly sorted spikes becomes essential, demanding considerable time and effort. Furthermore, the progress in contemporary microelectronics and electrophysiology has led to the widespread adoption of high-density microelectrode arrays featuring numerous channels for recording neuronal populations [14]. Consequently, the simultaneous recording of firing activities from hundreds or even thousands of neurons result in amassing extensive

1 Department of Computer Science, Collage of Science, University of Duhok, Duhok 42001, Iraq
helat.ahmed@uod.ac
2 Department of electrical and computer engineering, College of Engineering, University of Duhok, Duhok 42001, Iraq
ahmad.khorshed@uod.ac

data volumes. Presently, methods heavily rely on manual intervention, especially in lengthy experiments, resulting in slow and unreliable processes [15]. Inevitably, these misclassifications significantly impact subsequent data analyses. Thus, an urgent need arises for an efficient method that minimizes manual workload while ensuring accuracy and reliability. This goal agrees with deep learning methods' recent advancements as they rise to overcome the challenges of manual spike classification [16]. One of the basic triggered outcomes of deep learning algorithms is CNN through which spike and spike sorting detection can be carried out successfully.

This paper presents an intelligent neural spike sorting by integrating the wavelet detection methods to detect the spikes and for the classification we proposed anew model using convolution neural network .

The remaining structure of this paper is as follows: Section 2 providing a brief overview of related work Section 3 introduce the methodology used in this paper . Section 4 detail of the proposed model using 1DCNN.In Section 5 experimental result and comparing with other studies is made. The last Section 6 is the conclusions and future works

2.Literature Review

In table (1) summarizes the CNN used in spike sorting at different stages in spike sorting .as we can see

M.Wang (2023) [17] put forward a deep learning network used in enhancing spike sorting accuracy. In this method, a convolutional neural network (CNN) is linked with the sliding window long short-term memory (LSTM) network to extract the required discriminative features from the raw extracellular recordings. The CNN captures abstract features that differentiate spike waveforms from the artifacts while the sliding window LSTM network captures temporal dependencies of such spike waveforms. Results obtained clarify [17] that the

proposed method outperforms other existing state-of-the-art algorithms, in terms of the number of hits, misses, and false positive neurons. It obtains an overall accuracy of 97.5% for simulated dataset and 95.6% for the experimental dataset. However, the method proposed has several limitations such as requiring huge amounts of training data and also having the computational complexity of the network.

In the analysis presented by Z. Li, Y. Wang, and N. Zhang (2020) [18], it was noted that the suggested 1D-CNN model displayed greater robustness in comparison to conventional methods and a deep-learning-based multilayer perceptron (MLP) model. The amplification of noise levels across all datasets led to a deterioration in classical methods, manifesting in an elevated error rate of 46.28%, while the error rate for the MLP model rooted in deep learning climbed to 48.45%. In contrast, the error amount of the proposed method was around 2% for all datasets, which was more robust.

The aim of (M. Hall 2023) [19] works is to address the spike classification of different neuronal data in a general and effective manner. The authors suggest that the Convolutional Neural Network (CNN) is dramatically more robust to waveform variations and has classification accuracy far superior to conventional techniques. Overall, the work aims to afford a further efficient and precise technique for classifying neuronal data, the CNN demonstrated a higher classification accuracy than the previous NAS (Neural Amplitude Spike) network design a simple neural network used as a baseline for comparison with the proposed CNN in this study. The CNN exhibited an average increase in accuracy of 1.99%. The CNN had an average accuracy of 86.23%, ranging from 79.62% to 89.55% across all areas.

C.O. Okreghe, M.Zamani (2023) [20], the improvement in spike detection precision is realized through the incorporation of two convolutional neural network algorithms into the conventional pipeline, dedicating one to channel selection and the other to artifact removal. The process of selecting channels proves to be highly effective in identifying those capturing neuronal occurrences, achieving an average precision of 99.5%, while the accuracy in eliminating artifacts stands at 92.3%. Employing K-means clustering for both channel selection and artifact removal results in a classification accuracy of 87% and 91.53% on the experimental and simulated datasets, respectively, showcasing superior performance compared to traditional methods.

The authors M.Saif-ur-Rehman. et al., (2019) [21] aim to address the limitations of existing spike sorting algorithms, which often require manual intervention, are computationally expensive, and may not generalize well to different datasets. The proposed algorithm uses a deep convolutional Siamese network and hierarchical clustering to learn the pairwise similarity of spikes and cluster them into distinct units. The proposed algorithm achieved a classification accuracy of 84.33% on the testing set of simulated data. In this paper, the stationary wavelet was employed to detect spikes in the detection step. The accuracy of the detection method used for feature extraction and classification was evaluated using a one-dimensional convolutional neural network. To evaluate the proposed system, the simulated data set has been widely used as a ground for testing [10]. Also, we compare the results of the proposed method with other classification techniques, like K nearest neighborhood (KNN) and support vector machine (SVM).

3. Methods and Materials

3.1 *Proposed Methodology*

The exploration of neural spike sorting methods is a critical pursuit in neurophysiological research, aiming to decipher intricate neuronal activities within the brain. Traditional approaches we proposed for spike sorting often incorporate stationary wavelet analysis for spike detection, principal component analysis (PCA) for feature extraction, and support vector machines (SVM) for classification. Based on the above, this study suggests a new method of using Convolutional Neural Networks (CNNs) for spike sorting which is suggested as a new scheme for feature extraction and classification. The proposed methodology is designed to compare the performances of spike sorter adopting CNN against a conventional approach using stationary wavelet analysis, PCA, and SVM. This comparison is meant to evaluate a CNN-based method's effectiveness, accuracy, and computational efficiency in spike sorting from electrophysiological recording. In comparing the approaches, one can shed light on the advantages, disadvantages, limitations, and future improvements that can significantly improve the spike sorting performance using CNN. Then, the comparison shows the possibility of CNNs in solving neural spike sorting problems and the capacity to improve classification accuracy and spike classification rate over traditional techniques. Figure (1) shows the proposed system.

3.1.1 *Generate synthetic data*

The dataset provided by Quiroga et al. [10] is a valuable resource for researchers in the field of neuroscience. It contains detailed information about neural activity recorded from brains of experimental subjects. The dataset includes recordings from various brain regions and under different experimental.

Table 1. comparison of the adapted CNN used in spike sorting

Paper Title	CNN Structure	Number of Layers	Number of Epochs	Problem Statement	Accuracy
Manqing Wang, Liangyu Zhang 2023 [17]	1D-CNN for detection and CNN-LSTM for classification	3 convolutional layers and 2 fully connected layers	Not specified	Spike sorting in extracellular recordings	Recall rate of 94.40% in low noise level dataset and accuracy of 97.5% in simulated data
Z. Li, Y. Wang, N. Zhang 2020 [18]	1D-CNN	4 convolutional layers and 2 pooling layers,1 connected layer	100 epochs	Spike sorting in extracellular recordings	99.9% noise-free, 99% for SNR=10, 94% for recorded data from rat
M. Hall 2023 [19]	1D- CNN	4 convolution layers and 1 pooling	Not specified	classification of neuronal spikes in extracellular recordings.	Average accuracy of 86.23%, ranging from 79.62% to 89.55% across all areas.
C.O. Okreghe, M.Zamani ,2023 [20]	1D-CNN detecting the spikes,2D-CNN	4 convolution layers 3 pooling 1	Not specified	Accurately detecting and classifying in high-channel-count sensing	99.5% in channel selection and 92.3 % in artefact removal Accuracy of classification 87%
M.Saif-ur-Rehman.et al, 2019 [21]	1D-CNN	4 convolutional layers and 3 pooling fully connected layers	Not specified		Batch size 20 was 97.5% and reached 99.5% with batch size 65

conditions, providing a comprehensive view of neural activity. The grouped data allows researchers to compare and contrast neural responses under different conditions, which can lead to valuable insights into brain function and behavior. In order to assess the performance of a method with certainty, simulated data is typically employed. This is because the characteristics of simulated data are known in advance, allowing for a fair comparison with the results obtained from the suggested methods. The data utilized in this study has been synthesized in the following manner: first, a triplet of spikes was singled out from the data openly available on

[https://www2.le.ac.uk/centres/csn/software]

supplied by Quiroga et al. [10]. To mimic a group of spikes containing solely these three instances, a sequence was produced based on the Poisson distribution. The temporal and cluster attributes of the spikes have been predetermined and serve as the ground truth for future comparisons. Subsequently, the noise was produced with the features of a normal distribution and a standard deviation of unity. Finally, the spikes aligned in a prior stage, which had been extracted, were polluted by the produced noise.

3.1.2 Neural Spikes Detection

Applying a threshold to the neural signal is the essence of most detection methods, either directly or after implementing an operator designed to accentuate the spikes while diminishing the noise. The presence of a spike is determined by the crossing of the threshold. If set excessively high, smaller spikes may go unnoticed. But if it is too low, noise activity might cause false positives. Therefore, advanced methods are used for detection, like Stationary wavelet transform. The Stationary Wavelet Transform (SWT) is a prominent tool in neural spike signal processing that provides a flexible framework for analyzing non-stationary signals with dynamic frequency changes over time. Unlike the Discrete Wavelet Transform (DWT), the SWT overcomes limitations by ensuring translation invariance, which is essential for applications requiring accurate time-frequency localization. Its unique ability to decompose signals into distinct frequency bands while preserving temporal information makes it valuable in various fields, including neuroscience, where it is extensively used for neural spike sorting.

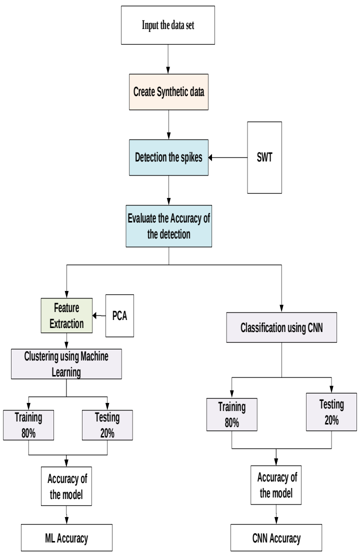

Fig.1. the proposed method for 1DCNN based spike sorting comparing it with machine learning

Wavelet-based denoising offers a multi-scale approach to noise reduction, involving signal down-sampling during decomposition and applying thresholding to wavelet coefficients, potentially targeting specific frequency bands in the frequency domain. To overcome the limitations of traditional wavelet transforms, this study employs a multi-layer stationary wavelet transform (SWT), as depicted in Figure (2),

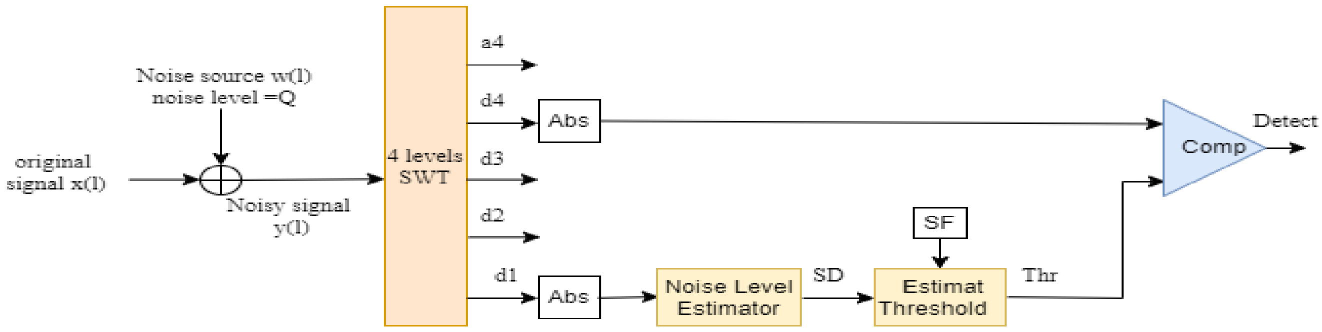

Fig 2. Architecture of the proposed SWT 4 level-based spike detection

The Stationary wavelet transform module provides the first level d1 and detail level d4 for the application of the threshold. to estimate the standard deviation SD the following equation used

$$SD = \left\{\frac{|d1|}{0.6745}\right\} \qquad (1)$$

The threshold *Thr* is estimated by using the following equation

$$Thr = SF * SD \qquad (2)$$

The scaling factor SF is an integer in the range of 3 and 5, it depends on application. In all analyses, the noise is Gaussian and white. The spike detector just outputs a pulse when the absolute level of high-pass filter detail signal $|d4|$ crosses the threshold Thr. A pulse is outputted each time the waveform crosses the threshold. coefficients represent the high- frequency component's extracts at different level of decomposition using mother filters .and (a1) is an approximation coefficient represents the low-frequency approximation obtained at the first level of decomposition using father filter. overall, the mother filter (high-pass filter) extracts high-frequency details at each level of decomposition (d1, d2, d3, d4), while the father filter (low-pass filter) extracts low-frequency approximation (a1) representing the coarse features of the signal. Together, these filters enable the SWT to decompose the signal into different frequency bands while preserving its time-domain characteristics. Therefor Stationary Wavelet Transform (SWT) offer superior frequency resolution compared to the Discrete Wavelet Transform (DWT) [22]. Furthermore, each node in the SWT tree corresponds to a unique set of SWT coefficients, which can be generated using different wavelet filters. Since each coefficient represents the magnitude of a specific frequency range within a particular time interval, additional iterations tend to generate a significant amount of redundant information [23] [24].

3.1.3 Features Extraction

In the early days of spike sorting, the feature extraction was based on the amplitude to separate spikes. This approach, although an easy and fast method, may encounter a limitation when neurons have the same magnitude of amplitude but differ in shape. Therefore, principal component analysis (PCA), a more sophisticated approach, has been used for feature extraction [26].

PCA is a highly effective and automated method for spike sorting. It involves calculating the principal components (PCs) of a group of neuronal spikes and using the first 2 or 3 PCs to create a feature space. This feature space captures the majority of the variance in the spikes. By projecting the neuronal spikes onto the feature space, the inherent characteristics of the spikes are emphasized, allowing for the observation and differentiation of several clusters that are composed of different neuronal spikes. The first two or three PCs form a low-dimensional feature space. Aligned spikes are then projected into this feature space by taking the dot product between the spikes and each PC. In the feature space, dots represent aligned neuronal spikes, while each cluster represents a potential neuron [7]. Finally, clustering algorithms are employed to differentiate the clusters in the feature space and assign the spikes to their closest cluster (neuron) [10].

3.1.4 Classification of the Neural spike using SVM

The primary objective of clustering is to differentiate and classify the detected spikes into various classes, given the recording of spikes from multiple neurons by a unique electrode, as previously indicated. In the current clustering procedures, machine learning makes the clustering intelligent [26]. One popular approach uses support vector machines (SVMs) to classify the spikes into distinct groups . SVMs are a machine learning algorithm that separates data into different classes based on labelled examples [27]. In the case of spike sorting, the training dataset will typically consist of simulations of single neurons as they fire spikes, along with the labels identifying the type of neuron (or the source) for each spike. The SVM is then trained to assign each spike in the recordings to the correct source. Once the SVM is trained, it can be used to classify new spikes in the recordings. The SVM provides a decision boundary that separates the different sources, and for each new spike, the SVM estimates the probability of its belonging to each source. The spike can then be assigned to the source with the highest probability. The Figure below shows the block diagram of the proposed SVM Model where the first 80% of the spikes were selected as training, and 80% of the labels were also selected as training labels ,and the remaining 20% of the spikesis used for testing , and labels were used for testing the model. And for nonlinearity classification a Gaussian Kernal are used ,and for the multi class SVM classifier is trained using the "one -vs-all" strategy ,where multiple binary classifiers are trained ,each distinguishing one class from the rest . The SVM classifier is configured with a Gaussian kernel and trained on the extracted features. Finally, the trained classifier is used to predict the labels of the test dataset, and the predicted labels are obtainedas showin in figure (3).

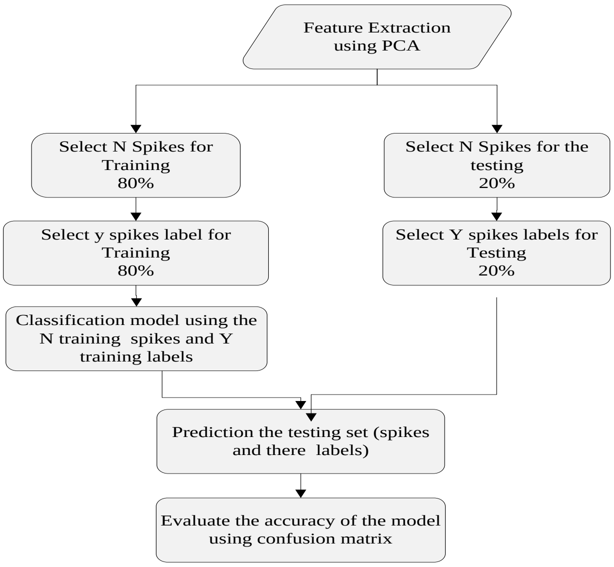

Fig 3. Block diagram for the SVM model.

3.2 Classification of neural spike using CNN

Convolutional Neural Networks are a type of neural network that excel in processing two-dimensional images.The core of a Convolutional Neural Network is the convolutional layer. This layer performs the convolution operation, which can be seen as matrix-vector multiplication[28]. Using this operation, the neural network is able to extract features from the input data. These features capture important patterns and information that are useful for classification . In the context of spike sorting, an 1D Convolutional Neural Network (1DCNN) can be used to analyze electrophysiological data and automatically determine the activities of individual

neurons[29]. Therefore, using Convolutional Neural Networks play a crucial role in neural spike sorting. They are able to process one-dimensional neural data and extract important features that can be used for classification. This allows for the automatic determination of the activities of individual neurons, reducing the chance of misclassification.1D Convolutional Neural Networks have been proven to be effective in solving the problem of neural spike sorting. They can accurately classify spikes by identifying the cluster that each spike belongs to. These networks use convolutional layers to process the encoded raw data with different filters, allowing them to find hidden features within the spike waveforms. These features capture important patterns and information that are useful for accurately classifying the spikes[30]. Determining the structure of the 1D CNN used in neural spike sorting can vary based on the specific task that the CNN is used for and the characteristics of the spike waveforms, but the basic architecture components of the 1D CNN contain the following .

1- Input layer: where the spikes waveforms are fed into the model as one dimension sequence data, typically representing the temporal characteristics of the neuronal spikes[18].

2- Convolution layers contain different components like:

a) Conv1Dlayers : These layers perform convolution operation between the input layer and the filter across the whole temporal dimension of the spike waveforms.

b) Filters: different filters are applied to select different patterns present in the spikes

c) Activation function : there are many activation functions but the most widely used in spike sorting is the ReLU which is applied to introduce non-linearity .

3- Pooling Layers: These are optional layers, the most applied of which is Max Pooling 1D , which might be used to reduce the computational complexity and downsample the learning features

4- Dense Layers:

a) Fully connected Layers: FC layers process the flattened features for the classification or clustering of the spikes.

b) Output layer : the output layer might have different nodes representing different spike clusters [31]

3.3 Architecture of the Proposed CNN Model

This section describes the proposed model for intellegent spike sorting by using new 1DCNN architecture specifically designed for spike sorting classification. Our model takes advantage of the temporal nature of spike waveforms by processing them as one-dimensional signals. The architecture consists of multiple convolutional layers followed by max-pooling layers to extract relevant features from the input spike waveforms. We also incorporate batch normalization. The final layers of the network include fully connected layers with softmax activation to perform classification into different spike units. The proposed one-dimensional CNN architecture is shown in Figure (4). In the design we carefully consider the choice of activation functions to capture the non-linearities present in spike waveforms. We experiment with activation functions such as ReLU to identify the most suitable option for spike sorting tasks.

Additionally, we employ data augmentation techniques to increase the diversity of training samples and improve the generalization capability of the model.The initial layer is a sequence input layer with 64 dimensions. Subsequently, four blocks of convolutional layers are applied, each comprising three 1D convolutions with 64 filters, using causal padding and. "causal" means that the activations computed for a particular time step cannot depend on activations from future time steps [31]. The architecture of proposed networks consists of multiple blocks, each containing a stride of 1. Layer normalization follows each convolutional layer, enhancing stability during training. Between these layers, skip connections are established through element-wise additions, aiding in gradient flow. Additionally, max pooling layers with a pool size of 1 are introduced for downsampling. ReLU activation functions are applied after normalization, contributing to non-linearity in feature extraction. The final layers include a fully connected layer with four neurons, representing output classes, followed by a softmax activation for probability distribution generation. The architecture demonstrates a systematic approach to capturing temporal features in sequential data, a crucial aspect for accurate spike sorting in neural recordings.

3.3.1 Input Sequance Layer

The process leading up to the use of the 1D convolutional neural network (1DCNN) for spike sorting involves several important steps. Initially, a synthetic dataset is created to imitate neural spike signals with known characteristics, which includes modeling spike waveforms, noise, and various spike shapes. This provides a controlled environment for evaluating spike sorting algorithms. After creating the dataset, spike detection and alignment are performed to accurately identify spike occurrences by using SWT and align their waveforms across the dataset, ensuring consistent input sequences for further analysis. Following this, the dataset is split into training and testing sets, with 80% of the data allocated for training and 20% for testing. This division helps evaluate the model's ability to generalize on unseen data. The training set is used to optimize the parameters of the 1DCNN, allowing the network to learn meaningful features and patterns from the input sequance spike data. Meanwhile, the testing set serves as an independent evaluation to assess the model's performance on new spike waveforms. This strict division between training and testing ensures a comprehensive evaluation of the spike sorting model, validating its effectiveness in accurately classifying neural spikes. The main idea of spike sorting is identifying and classifying the neural spikes recorded from electrodes implanted in the brain. using the CNN to classify different spikes patterns, which could represent different neural activities. The proposed 1DCNN Architecture .The main building block of a 1DCNN is a causal convolution layer, which operates over the time steps of each sequence. The architecture of proposed networks consists of multiple blocks, each containing . Figure (4) visually represents the CNN architecture operating on a simulated database. Consisting of four blocks , each block content four set of causal convolution layer with the same dilation factor followed by max pooling used to reduce the computational complexity and down sample the learning features , followed by normalization ReLU activation . The proposed network adds the input of each block to the output of the next block as shown in figure (4). At the end the connect the four blocks with the Fully connected layer followed by SoftMax , final steps are the classes output. For each block ,64 filters have been used and ,the size of each filter is size of 3 these filters

used for the 1_D convolution layers. For the training options used 100 epochs with minibatch size 1.Also with the learning rate of 0.001 . after training the model we need to test the classification accuracy of the model by comparing on a held-out test set with the true labels for each spike.The feature extraction process through varying abstractions and exchanging information from one layer to another establishes a receptive field, enabling the CNN model to achieve near-invariance to spatial alterations with a more economical computational expenditure. In the context of classification endeavors, this network structuring approach is termed an encoder block, and the outcomes derived from these encoders constitute extracted attributes from the input information, like the characteristics of the identified spikes. Consequently, to transform encoders into a classifier, it is typical to affix fully connected strata to the ultimate layer of the network, succeeded by non-linear operations like Sigmoid or SoftMax [32]. To solve the overfitting, incorporating batch normalization and dropout strata are among the layers. Throughout the training phase of a CNN model, the 1-dimensional kernels within the network are initially established with random initialization or diverse techniques for initialization. Using the backpropagation technique, initialized kernels should be optimized using an algorithm like Adam.

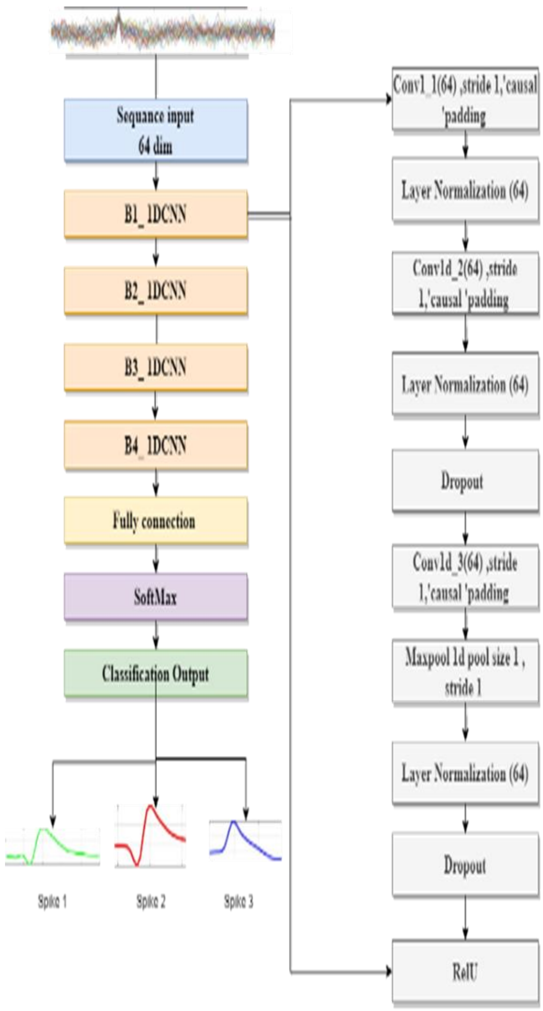

Fig 4. The architecture of the proposed 1D CNN forspike sorting classification

3.3 The simulated data

The first time the dataset was introduced, I was the main one used in several works in the spike signals sorting. The generation of simulated signals involved the utilization of a compilation of 12 varied spike configurations obtained from real neocortical and basal ganglia recordings [Quiroga et al. 2004]. Further insights into this collection can be found in Collection 2 (Easy - Difficult). The 12 spike outlines are categorized into three groups, forming four sets, with each quartet featuring four signals exhibiting diverse noise levels. These outcomes are a sum of 16 test signals. Figure (5) displays the four sets (12 spikes total) of spike configurations utilized in the simulations. A duration of 60 seconds was assigned to each signal, and the simulation was executed at a sampling rate of 24 kHz.

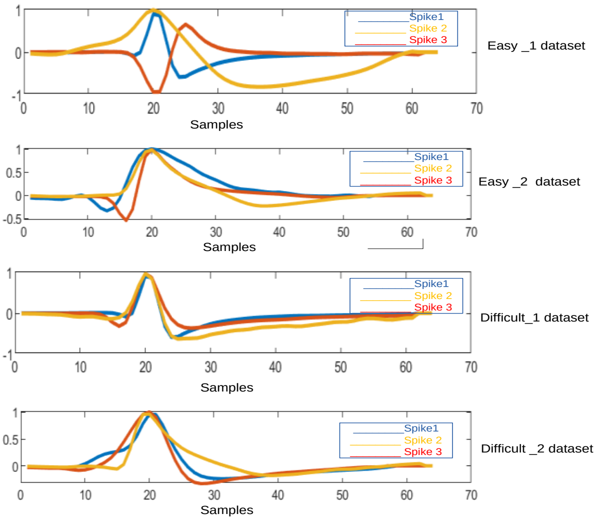

Fig 5. Overview of the four sets of spike shapes. Three spike shapes represent each set.

To distribute these three spikes, a Poisson distribution has been applied to mimic the distribution of the spikes in the brain; also, it is more accurate and scientific. Figure (6) shows how the three spikes have been distributed.

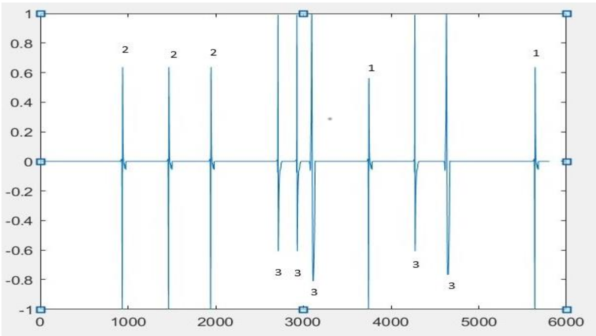

Fig 6. Sequence of spikes generated using Poisson distribution

Then, different noise levels have been added to the sequence of spikes by using gaussian noise(sigma) to estimate the noise standard deviation from the generated sequence of spikes. Also, the dataset is divided into 16 datasets built from a 594-spike waveform database. These datasets were built using an averaging method of the real recording from the neocortex and basal ganglia. In this dataset, the noise was randomly created by selecting and adding several spikes from the database with different amplitudes. The noise level in this database is varied between 0.05 and 0.4.

3.5 Assessing the Effectiveness of the Proposed System

Four Experiments were conducted per dataset. The distribution of data utilized in training and testing is depicted in Figure (7). For every dataset, the training subset encompassed 50%, 60%, 70%, and 80% of the overall data, while the corresponding testing subset encompassed the remaining 50%, 40%, 30%, and 20%, respectively. To assess the model's performance, accuracy, measured as the proportion of accurately classified samples out of the total data, was employed to determine the overall classification score, and it was instrumental in analyzing the experimental data.

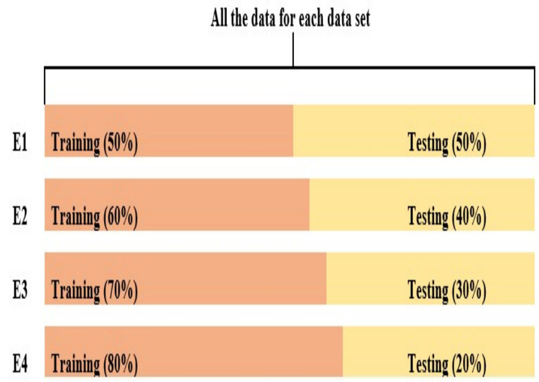

Fig 7: Ratio of data allocated to the training and testing sets across the four experiments, denoted by 'E'.

The confusion matrix is a vital tool for evaluating the performance of classification model performance. It comprehensively analyses how a model's predictions compare to the actual labels derived from the ground truth. The confusion matrix consists of three essential components:

1. True Positives (TP): In these spikes, that model successfully predicted the positive class.

2. True Negatives (TN): In each situation, the model properly predicted the negative class, which was incorrect.

3. False Negatives (FN): In these cases, the

model predicted the erroneous class, which was negative, whereas the true class was positive.

The Confusion Matrix, which calculates the proportion of correctly classified samples to all samples, as demonstrated in the example below, can be used to evaluate the classification models' accuracy.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \quad (3)$$

To determine the accuracy for the TP divided by the total number of spikes with positive

labels (TP plus FP added together), as stated in formula 4, high precision indicates that the model and categorization are producing more useful results.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

Calculating the recall by dividing the total number of components that genuinely belong to the positive class TP will allow you to determine the model's sensitivity (5).

$$Sensitivity = \frac{TP}{TP+FN} \quad (5)$$

4. Results and Discussions

The data set used in this work contains 4 groups (Easy1, Easy2, Difficult 1, and Difficult 2), where every set comprises three signals with changeable noise levels, as shown in Figure (6). From the tested set, we took the times when the spikes start firing and then took the classes to generate the synthetic data and disrepute the signal by using a Poisson distribution, which mimics the distribution of spikes in the brain, as shown in Figure (7).

In table (2) show the distributed of spikes generated by using synthetic data set the total number of spikes for each group of (Easy _1, Easy_2, Difficult_1, Difficult_2) in different noise level (σ) from 0.05 to 0.4 for data set Easy_1 and from 0.5 to 0.2 for the rest datasets. The total number (sc1) of sequences of spike type 1 is noted as Spike Class 1, and total number of spikes class type 2 is (sc2) and the same thing for spike Class 3 (sc3). The last column is the total spikes input to the system at each level of noise and for each data set. The second step of the proposed methodology is applying a stationary wavelet transform (SWT) to detect the spikes. In this work, we used the Coif_1 filter bank to detect the spikes. The results of the evaluation of the accuracy of applying the SWT _4 layer _ Coif1 method, are shown in table (3) where TSI is the total number of spikes and the output spikes of the detection area (true spike output) (TSO). The TPn is the number of spikes detected and classified in the same class, while FPn is the number of spikes put in another class. FZn is the number of the signals detected, but not spikes; it is the background noise. At lower noise levels (0.05 and 0.1), the accuracy of spike detection remains consistently high across all datasets. For instance, in Data Set 1, at noise levels 0.05 and 0.1, the accuracy remains consistently at 98.5%.

Table 2. The total number of spikes input and different noise level (σ)

Data set	σ	sc1	sc2	sc3	Total
Easy_1	0.05	962	930	973	2865
	0.1	957	915	1006	2878
	0.15	937	966	935	2838
	0.2	955	929	937	2821
	0.25	910	898	879	2687
	0.3	918	949	919	2786
	0.4	891	947	950	2788
Easy_2	0.05	923	902	945	2770
	0.1	940	913	990	2843
	0.15	978	893	911	2782
	0.2	968	968	912	2848
Difficult _1	0.05	905	912	950	2767
	0.1	962	922	905	2789
	0.15	909	967	919	2795
	0.2	927	871	974	2772
Difficult_2	0.05	897	914	920	2731
	0.1	991	954	901	2846
	0.15	917	914	961	2792
	0.2	924	972	944	2840

in Data Set 1, there is a noticeable decline in accuracy from 98.5% at 0.1 noise level to 98.6% at 0.15, but a subsequent drop to 98.5% at 0.2. Similar trends are visible in Easy-2 and Difficult-1 datasets. Also, accuracy considerably decreased across all datasets at noise levels of 0.25, 0.3, and 0.4. For instance, in Data Set 1, the accuracy drops to 93.9% at 0.25 noise level, further decreasing to 80.6% at 0.3 and notably down to 55.1% at 0.4.

Table 3. The success of spike detection in terms of false positive and false zero by using SWT _4 Layer _cofi1

Data set	σ	TSI	TSO	TPn	FPn	FZn	AD
Data set 1:	**0.05**	**2865**	**2908**	**2865**	**43**	**0**	**98.5**
Easy-1	**0.1**	**2878**	**2921**	**2878**	**43**	**0**	**98.5**
	0.15	**2838**	**2877**	**2838**	**39**	**0**	**98.6**
	0.2	**2821**	**2847**	**2813**	**34**	**8**	**98.5**
	0.25	**2687**	**2601**	**2561**	**40**	**126**	**93.9**
	0.3	**2786**	**2317**	**2278**	**39**	**508**	**80.6**
	0.4	**2788**	**1594**	**1557**	**37**	**1231**	**55.1**
Data set 2:	**0.05**	**2770**	**2810**	**2770**	**40**	**0**	**98.5**
Easy -2	**0.1**	**2843**	**2887**	**2843**	**44**	**0**	**98.4**

	0.15	**2782**	**2819**	**2782**	**37**	**0**	**98.6**
	0.2	**2848**	**2884**	**2848**	**36**	**0**	**98.7**
Data set 3:	**0.05**	**2767**	**2810**	**2767**	**43**	**0**	**98.4**
Difficult -1	**0.1**	**2789**	**2830**	**2789**	**41**	**0**	**98.5**
	0.15	**2795**	**2835**	**2795**	**40**	**0**	**98.5**
	0.2	**2772**	**2803**	**2761**	**42**	**11**	**98.11**
Data set 4:	**0.05**	**2731**	**2770**	**2731**	**39**	**0**	**98.5**
Difficult -2	**0.1**	**2846**	**2887**	**2846**	**41**	**0**	**98.5**
	0.15	**2792**	**2833**	**2792**	**41**	**0**	**98.5**
	0.2	**2840**	**2878**	**2840**	**38**	**0**	**98.68**

The third step of the proposed method, as shown in Figure (2), is applying PCA for dimension reduction and extracting the most important features of the spikes from the spikes that have been detected. The next step is the classification by training a model using SVM; the data set are separated into training and testing, as shown in Figure (3). Table (6) presents the results of spike sorting classification using (SVM) across different noise levels for four datasets. The "Noise Level" column denotes the amount of noise added to the data .The subsequent columns, such as "TPn1," "TPn2," "TPn3," and "TPn4," indicate the true positives for each class. The "Accuracy of classification" column reflects the overall accuracy of the SVM model in classifying spikes. In dataset 1 (Easy-1), as the noise level increases, the accuracy gradually decreases from 99.8% to 88.7%. This suggests that the SVM model performs exceptionally well with low noise but faces challenges as the noise level intensifies. Similar trends are observed in dataset 2 (Easy-2), with accuracy ranging from 99.6% to 97.4%. Dataset 3 (Difficult-1) and dataset 4 (Difficult-2) also exhibit a decline in accuracy with higher noise levels, indicating the impact of noise on SVM classification performance. In the second section, the 1D-CNN model classification with different proportions of data in training and testing, as shown in Figure (7), with four experiments were noted by E as shown in table (5) and compared with the two most widely used methods (SVM, KNN) as shown in table (6). The model used for KNN is the same as the one proposed for the SVM, where the data set is suppurated to 80 % for training and 20 % for testing. The data results with different noise levels are shown in Table(6).

Table 4. Accuracy of classification SVM

Data set	σ	**TPn1**	**TPn2**	**TPn3**	**TPn4**	**AC**
Data set 1:	0.05	198	188	187	9	99.8
Easy-1	0.1	190	172	212	11	99.4
	0.15	175	199	193	9	98
	0.2	184	184	191	11	97.3
	0.25	165	166	178	12	96.1
	0.3	101	161	183	9	94.1
	0.4	43	79	187	10	88.7
Data set 2:	0.05	182	176	195	9	99.6

Easy -2	0.1	193	176	200	9	99.5
	0.15	187	175	192	10	99.5
	0.2	198	191	181	7	97.4
Data set 3:	0.05	178	190	183	11	99.8
Difficult -1	0.1	184	195	176	11	99.6
	0.15	175	203	181	8	99.6
	0.2	167	175	208	11	98
Data set 4:	0.05	173	183	187	11	99.7
Difficult -2	0.1	200	201	168	9	98.3
	0.15	183	183	192	9	95.4
	0.2	194	198	175	9	87.88

Table (5)shows that experiment (E4) with an 80% training dataset allocation consistently achieves accuracy rates across noise levels and training set sizes. This experiment shows reliable performance when faced with varying degrees of noise. For example, at a noise level of 0.05, Experiment 4 consistently achieves an

accuracy of 100%, indicating its reliability in relatively cleaner signal conditions. Compared

to other experiments, (E4) exhibits a stable and higher accuracy performance across different noise levels and training set sizes, making it a more dependable choice. Therefore, we select (E4) to compare it with other classification methods such as SVM and KNN. Looking at Table (6), we can observe that the proposed method of sorting classification using the 1D_CNN consistently outperforms both KNN and SVM across noise levels and SNRs. The CNN tends to achieve accuracy percentages compared to KNN and SVM at different noise levels. For instance, at

noise levels like 0.05 with SNR 20, CNN consistently demonstrates accuracy percentages close to or reaching 100%, while

KNN and SVM slightly fall below these values. In situations with levels of noise and signal-to-noise ratios, the CNN algorithm consistently demonstrates higher accuracy.

Table5. comparing the classification accuracy based on different training and testing data allocations in the 1D_CNN model

Data set	σ	**E1 50 %**	**E2 60 %**	**E3 70%**	**E4 80 %**
Data set 1:	**0.05**	**99.9**	**99.8**	**100**	**100**
Easy-1	**0.1**	**99.9**	**99.3**	**99.6**	**100**
	0.15	**99.2**	**98.3**	**99.4**	**99.7**
	0.2	**97.9**	**96.8**	**97.8**	**98.9**

Table 6 . Comparing the accuracy of the proposed 1D_CNN with the traditional method used machine learning in classification

Data set	σ	**SNR**	**KNN**	**SVM**	**1D-CNN**
Data set 1:	**0.05**	**20**	**99.6**	**99.8**	**100**
Easy-1	**0.1**	**10**	**99.5**	**99.4**	**100**
	0.15	**6.6**	**99.5**	**98**	**99.7**
	0.2	**5**	**98.9**	**97.3**	**98.9**
	0.2	**4**	**96.7**	**96.1**	**96.75**

	0.25	94.9	95.3	96.7	96.75
	0.3	94.4	94.4	94.4	94.9
	0.4	89.6	89.6	89.8	91.2
Data set 2:	0.05	99.9	100	100	100
Easy -2	0.1	99.7	99.8	99.8	99.98
	0.15	98.7	99.1	98.9	99.6
	0.2	97.8	96.3	98.2	98.6
Data set 3:	0.05	99.9	100	100	100
Difficult -1	0.1	99.7	100	100	100
	0.15	96.7	99.3	99.7	99.8
	0.2	96.4	96.6	97.4	98.2
Data set 4:	0.05	100	100	99.9	100
Difficult -2	0.1	98.1	98.7	99	99.8
	0.15	93.3	94.2	94.4	96.2
	0.2	84.4	84.8	86.7	89.97

	5				
	0.3	3.3	94.4	94.1	94.9
	0.4	2.5	88.5	88.7	91.2
Data set 2:	0.05	20	99.7	99.6	100
Easy -2	0.1	10	99.7	99.5	99.98
	0.15	6.6	99.2	99.5	99.6
	0.2	5	97.3	97.4	98.6
Data set 3:	0.05	20	99.7	99.8	100
Difficult -1	0.1	10	99.5	99.6	100
	0.15	6.6	98.5	99.6	99.8
	0.2	5	97.3	98	98.2
Data set 4:	0.05	20	99.5	99.7	100
Difficult -2	0.1	10	98.9	98.3	99.8
	0.15	6.6	94.7	95.4	96.2
	0.2	5	86.5	87.88	89.97

across various noise levels, with the lowest accuracy at 91.2%, still outperforming their reported rate. While Li, Wang, and Zhang (2020), in their study, reported 99.9% accuracy in noise-free scenarios and varying accuracy rates for different SNR levels. The result of our model accuracy closely matches their figures, particularly in noise-free environments (99.8% and above). Also, Hall (2023) reported an average accuracy of 86.23%, ranging from 79.62% to 89.55% across all areas. Our results consistently surpass this range, showcasing higher accuracy even at the lowest end of highly noise-recorded data (91.2%). In summary, the proposed 1D-CNN-based model consistently demonstrates competitive or improved performance compared to these studies, especially in noise-free scenarios and across various noise levels, highlighting its robustness and effectiveness in spike sorting tasks.

Fig 10 . Confusion matrix (a) for data set C_Difficut1 _noise 02 (b) for data set Easy_1_noise_02

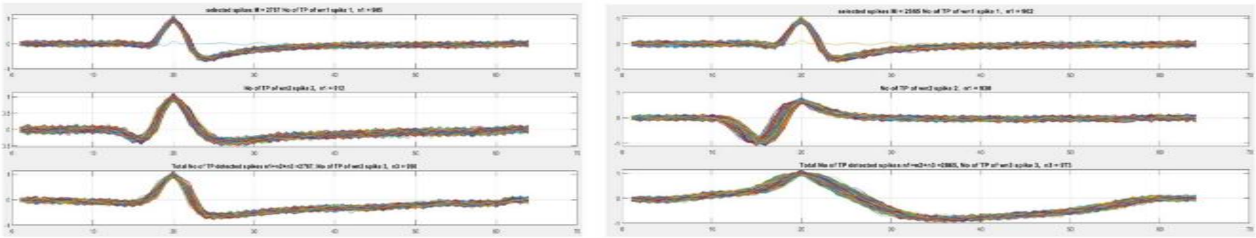

Fig 11. Shape of the three spikes in (a) C_Difficult1 with noise 0.05, (b) C_Easy1_noise 0.05

5.Conclusion

The 1D-CNN model gave a good performance accuracy compared to the classical methods like SVM and KNN in their ability to do spike sorting tasks. Compared to SVM and KNN, 1D-CNN performed better, increasing accuracy in all the considered noise conditions and SNRs. Additionally, compared to previous studies in spike sorting, the 1D-CNN runs a high level of performance. In noise-free environments, it closely approaches or surpasses the best accuracy of that reached at in the previous research. Indeed, as described earlier, even in reduced SNR value, the proposed model does not lose competitiveness in achieving accuracy with the intensification of noise to display robustness regarding the detection and classification of spikes. The accuracy rates obtained by this 1D-CNN model exhibited effectiveness in dealing with complex spike sorting tasks, either outperforming or coming very close to the state-of-the-art methods mentioned recently in the literature. This indicates its potential as an advanced, reliable tool for neuronal spike analysis, especially in challenging the noisy recording conditions. In future studies, this CNN model will be improved to cluster overlapped spikes efficiently and, at the same time, improve its ability to classify multichannel recordings simultaneously

References

[1] T. T. Pham, "Spike Sorting: Application to Motor Unit Action Potential Discrimination," pp. 83–95, 2019, doi: 10.1007/978-3-319-98675-3_6.

[2] R. Toosi, M. A. Akhaee, and M. R. A. Dehaqani, "An automatic spike sorting algorithm based on adaptive spike detection and a mixture of skew-t distributions," Sci Rep, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-93088-w.

[3] P. Buccino et al., "SpikeInterface, a unified framework for spike sorting," bioRxiv, pp. 1–24, 2019, doi: 10.1101/796599.

[4] P. Buccino et al., "SpikeInterface, a unified framework for spike sorting," bioRxiv, pp. 1–24, 2019, doi: 10.1101/796599.

[5] J. Wouters, F. Kloosterman, and A. Bertrand, "SHYBRID: A graphical tool for generating hybrid ground-truth spiking data for evaluating spike sorting performance," bioRxiv, 2019, doi: 10.1101/734061.

[6] K. H. Kim and S. J. Kim, "A wavelet-based method for action potential detection from extracellular neural signal recording with low signal-to-noise ratio," IEEE Trans Biomed Eng, vol. 50, no. 8, pp. 999–1011, 2003, doi: 10.1109/TBME.2003.814523.

[7] T. Takekawa, Y. Isomura, and T. Fukai, "Spike sorting of heterogeneous neuron types by multimodality-weighted PCA and explicit robust variational bayes," Front Neuroinform, vol. 6, no. MARCH, pp. 1–13, 2012, doi: 10.3389/fninf.2012.00005.

[8] J. Røislien and B. Winje, “Feature extraction across individual time series observations with spikes using wavelet principal component analysis,” Stat Med, vol. 32, no. 21, pp. 3660–3669, 2013, doi: 10.1002/sim.5797.

[9] Sharmila and P. Mahalakshmi, “Wavelet-based feature extraction for classification of epileptic seizure EEG signal,” J Med Eng Technol, vol. 41, no. 8, pp. 670–680, 2017, doi: 10.1080/03091902.2017.1394388.

[10] R. Quian Quiroga and Z. Nadasdy, “Communicated by Maneesh Sahani Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering,” vol. 1687, pp. 1661–1687, 2004.

[11] C. Souza, V. Lopes-dos-Santos, J. Bacelo, and A. B. L. Tort, “Spike sorting with Gaussian mixture models,” Sci Rep, vol. 9, no. 1, pp. 1–14, 2019, doi: 10.1038/s41598-019-39986-6.

[12] X. R. Diggelmann, M. Fiscella, A. Hierlemann, and X. F. Franke, “Automatic spike sorting for high-density microelectrode arrays,” no. Bishop 2007, pp. 3155–3171, 2021, doi: 10.1152/jn.00803.2017.

[13] M. Moghaddasi, M. Aliyari Shoorehdeli, Z. Fatahi, and A. Haghparast, “Unsupervised automatic online spike sorting using reward-based online clustering,” Biomed Signal Process Control, vol. 56, Feb. 2020, doi: 10.1016/j.bspc.2019.101701.

[14] M. Pachitariu, N. Steinmetz, S. Kadir, M. Carandini, and H. Kenneth D., “Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels,” bioRxiv, p. 061481, 2016, doi: 10.1101/061481.

[15] R. Diggelmann, M. Fiscella, A. Hierlemann, and F. Franke, “Automatic spike sorting for high-density microelectrode arrays,” J Neurophysiol, vol. 120, no. 6, pp. 3155–3171, 2018, doi: 10.1152/jn.00803.2017.

[16] M. Saif-Ur-Rehman et al., “SpikeDeep-classifier: A deep-learning based fully automatic offline spike sorting algorithm,” J Neural Eng, vol. 18, no. 1, 2021, doi: 10.1088/1741-2552/abc8d4.

[17] H. Jahangir, H. Tayarani, S. Sadeghi Gougheri, M. Aliakbar Golkar, A. Ahmadian, and A. Elkamel, “Deep Learning-based Forecasting Approach in Smart Grids with Micro-Clustering and Bi-directional LSTM Network,” IEEE Transactions on Industrial Electronics, vol. 0046, no. c, pp. 1–1, 2020, doi: 10.1109/tie.2020.3009604.

[18] 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, Nov. 2020, doi: 10.3390/brainsci10110835.

[19] M. J. Hall, E. Salazar-Gatzimas, J. Soldado-Magraner, and M. A. Smith, “P a g e | 1 A convolutional neural network for generalized and efficient spike classification Abstract.”

[20] O. Okreghe, M. Zamani, and A. Demosthenous, “A Deep Neural Network-Based Spike Sorting With Improved Channel Selection and Artefact Removal,” IEEE Access, vol. 11, pp. 15131–15143, 2023,

[21] doi: 10.1109/ACCESS.2023.3242643.

[22] M. Saif-Ur-Rehman et al., “SpikeDeeptector: A deep-learning based method for detection of neural spiking activity,” J Neural Eng, vol. 16, no. 5, Jul. 2019, doi: 10.1088/1741-2552/ab1e63.

[23] L. Huang, B. W. K. Ling, R. Cai, Y. Zeng, J. He, and Y. Chen, “WMsorting: Wavelet Packets’ Decomposition and Mutual Information-Based Spike Sorting Method,” IEEE Trans Nanobioscience, vol. 18, no. 3, pp. 283–295, 2019, doi: 10.1109/TNB.2019.2909010.

[24] F. Lieb, H. G. Stark, and C. Thielemann, “A stationary wavelet transform and a time-frequency based spike detection algorithm for extracellular recorded data,” J Neural Eng, vol. 14, no. 3, Mar. 2017, doi: 10.1088/1741-2552/aa654b.

[25] Soleymankhani and V. Shalchyan, “A New Spike Sorting Algorithm Based on Continuous Wavelet Transform and Investigating Its Effect on Improving Neural Decoding Accuracy,” Neuroscience, vol. 468, pp. 139–148, Aug. 2021, doi: 10.1016/j.neuroscience.2021.05.036.

[26] S. Gibson, J. W. Judy, and D. Marković, “Spike sorting: The first step in decoding the brain: The first step in decoding the brain,” IEEE Signal Process Mag, vol. 29, no. 1, pp. 124–143, 2012, doi: 10.1109/MSP.2011.941880.

[27] H. Bˆ and R. C. Mures, “Machine Learning-Assisted Detection of Action Potentials in Extracellular Multi-Unit Recordings”.

[28] M. Kubat, An Introduction to Machine Learning. 2017. doi: 10.1007/978-3-319-63913-0.

[29] M. J. Hall, E. Salazar-Gatzimas, J. Soldado-Magraner, and M. A. Smith, “P a g e | 1 A convolutional neural network for generalized and efficient spike classification Abstract.”

[30] H. Song, F. Flores, and D. Ba, “Spike sorting by convolutional dictionary learning,” ArXiv, 2018.

[31] Kiskin, D. Zilli, Y. Li, M. Sinka, K. Willis, and S. Roberts, “Bioacoustic detection with wavelet-conditioned convolutional neural networks,” Neural Comput Appl, vol. 32, no. 4, pp. 915–927, Feb. 2020, doi: 10.1007/s00521-018-3626-7.

[32] M. H. Mozaffari and L.-L. Tay, “A Review of 1D Convolutional Neural Networks toward Unknown Substance Identification in Portable Raman Spectrometer.”

[33] M. J. Hall, E. Salazar-Gatzimas, J. Soldado-Magraner, and M. A. Smith, “P a g e | 1 A convolutional neural network for generalized and efficient spike classification Abstract.”