Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

A deep learning network based on CNN and sliding window LSTM for spike sorting

Manqing Wang [a,b], Liangyu Zhang [a], Haixiang Yu [a], Siyu Chen [a], Xiaomeng Zhang [c], Yongqing Zhang [a], Dongrui Gao [a,b,*]

[a] *School of Computer Science, Chengdu University of Information Technology, Chengdu, 610225, China*
[b] *School of Life Sciences and Technology, University of Electronic Science and Technology of China, Chengdu, 611731, China*
[c] *Gingko College of Hospitality Management, Chengdu, 611730, China*

ARTICLE INFO

Keywords:
Spike detection
Spike classfication
CNN
LSTM

ABSTRACT

Spike sorting plays an essential role to obtain electrophysiological activity of single neuron in the fields of neural signal decoding. With the development of electrode array, large numbers of spikes are recorded simultaneously, which rises the need for accurate automatic and generalization algorithms. Hence, this paper proposes a spike sorting model with convolutional neural network (CNN) and a spike classification model with combination of CNN and Long-Short Term Memory (LSTM). The recall rate of our detector could reach 94.40% in low noise level dataset. Although the recall declined with the increasing noise level, our model still presented higher feasibility and better robustness than other models. In addition, the results of our classification model presented an accuracy of greater than 99% in simulated data and an average accuracy of about 95% in experimental data, suggesting our classifier outperforms the current "WMsorting" and other deep learning models. Moreover, the performance of our whole algorithm was evaluated through simulated data and the results shows that the accuracy of spike sorting reached about 97%. It is noteworthy to say that, this proposed algorithm could be used to achieve accurate and robust automated spike detection and spike classification.

1. Introduction

Communication between neurons in the brain takes place by propagating action potentials (spikes), so analyzing the electrophysiological activity of neurons is fundamental to explore brain function [1–3]. The extracellular activity of neurons is recorded by placing electrodes in the brain tissue, and usually one electrode can monitor the activity of 2–4 nearby neurons. Therefore, to complete signal analysis and information decoding, it is necessary to isolate the activity of each single neuron. Due to the different shapes of the dendrites of neurons and their distance and orientation relative to the recording electrodes [4], the spike morphology differs for each neuron, so the spike waveform can be classified. This process is known as spike sorting, consisting of three main steps: filtering, spike detection and spike classification [5].

First of all, a band-pass filter is performed on the raw data to remove low-frequency local field potential and high-frequency noise. Then neural spikes are detected from the accumulated neural activity and background noise. Absolute value thresholding is the most common detection method, which is simple but sensitive to background noise [6]. Recently, studies present various approaches to overcome the influence of background activities and reduce human involvement. For instance, Saifurrehman et al. used an approach based on both supervised deep learning algorithms and k-means to extract meaningful channel, and isolate the background activity [7,8].

The process of spike classification is feature extraction followed by clustering. In feature extraction, methods such as PCA [9], wavelet transform coefficients [10], L2-normalized convolutional autoen-coder [11], and "WMsorting" [12]are introduced to map the data in lower dimension and reduce the computational complexity. In clustering, different methods are reported in literatures, such as superparamagnetic clustering (SPC) [10], k-means clustering [13], a mixture of Gaussians [14], a similarity-based K-means clustering algorithm that conditionally updates the means by observing the cosine similarity [11]. These methods are performed manually or semi-automatic, so the quality completely depends on the subjectivity of researchers and the process is high time-consuming [15]. Therefore, fully automatic spike

[*] Corresponding author. School of Computer Science, Chengdu University of Information Technology, Chengdu, 610225, China.
E-mail addresses: wangmanqing@cuit.edu.cn (M. Wang), gdr1987@cuit.edu.cn (D. Gao).

https://doi.org/10.1016/j.compbiomed.2023.106879
Received 8 November 2022; Received in revised form 8 February 2023; Accepted 30 March 2023
Available online 4 April 2023

classification has been the main area of interest, especially with the use of high-density microelectrode arrays in neuronal population recording, where in the firing activity of hundreds or even thousands of neurons are simultaneously [16]. Recently, deep learning techniques have been proven to be one solution to spike sorting. For instance, deep-learning-based multilayer perceptron model (MLP) [17] and Convolutional neural networks (CNN) model [18,19] have been tested on simulated data and significantly outperformed conventional methods at high noise level signals. Although CNN performs well in object detection [20–22], brain image analysis [23,24] and EEG decoding [25], the accuracy of spike sorting still has room for improvement.

In this paper, we proposed a detection model based on 1D_ CNN and a classification model combined CNN and Long Short Term Memory (LSTM). The proposed detection model is an automatic detection method that could detect the spike position in a segment based on the idea of image-based target detection. In the classification model, a hybrid network was introduced due to the outstanding performance of CNN and LSTM in processing one-dimensional signals [26,27]. LSTM has the advantages in solving the problem of gradient disappearance in the training process of Recurrent Neural Networks (RNN) [28] and has an inherent ability to extract the time series feature [29]. Therefore, our model combines LSTM and CNN, and aims to train the model with a small number of samples while ensuring accuracy and robustness. To verify the performance of the network, both simulated database and experimental data with different signal-to-noise ratios were employed, as detailed in Section 2. We also compared the performance of our model with WMsorting [12] and several deep learning methods include CNN, RNN, Gate Recurrent Unit (GRU) and LSTM, as shown in section 3 and section 4. Finally, discussion and conclusion on future work are included in Section 5.

2. Materials and methods

2.1. Methods

A detailed flowchart of the proposed algorithm is depicted in Fig. 1. The entire model consists of three parts: data preprocessing, spike detection, and spike classification. First, we divided the signal into segments using a sliding window with width of 200 datapoints and step of 160 datapoints. These segments were fed into the 1D_CNN to extract spike samples (SS) and remove background activity samples (BS) using the softmax classifier. Then 1D_CNN was used to detect the start point and the end point of a spike, that is a completely waveform of 40 datapoints. Finally, the spike classification based on LSTM and 2D_CNN were performed to distinguish the categories of spikes.

2.1.1. 1D_CNN model

CNN mainly includes input layer, convolutional layer, linear activation function, pooling layer and fully connected layer [30]. Fig. 2 presents the proposed 1D_CNN model, which consists of two convolutional layers, two pooling layers, and a fully connected layer.

Convolutional layer: The main block of CNN is the convolutional layer, which extracts the features of the input data. Since the number of convolutional layers is related to the computational complexity and computation time, we set two convolution layers in this study. For all convolutional layers, the stride was one to extract more comprehensive information and the convolution kernel was three to obtain more features with less computation.

Pooling layer: Computational complexity is reduced by using a maximum pooling layer. In this paper, two max pooling layers were used with a kernel size of four and a stride size of four.

Fully connected layer: The output size of the fully connected layer depends on the number of labels. For both data preprocessing (spike samples v.s. background activity samples) and spike detection (start v.s. end point of spike), the dimension of output was set to two.

Define the input raw signal T = $(t_1 , t_2 , \ldots , t_N) \in R^{N*1*L}$, where N represents the batch size, L represents the length of the input data, and the input length here is 200. The signal is input to the convolution layer for learning, and the feature space is compressed through maximum pooling to obtain advanced time features. The whole process can be expressed as follows:

$$Q_t = Max\left(\delta\left(f^3\right)\right) \in R^{N*1*L} \tag{1}$$

Where Max represents maximum, δ represents the relu activation function and represents the filter with convolution kernel of 3.

In the data preprocessing, input the sample sequence into the second convolution layer, and finally classify it through the full connection layer. The results are divided into SS and BS according to the threshold value of the label. The process can be described as follows:

$$Y_e = \alpha\left(F\left(Max\left(\delta\left(f^{3*1}\right)\right)\right)\right) \tag{2}$$

where, α represents softmax activation function. F represents the full connection layer.

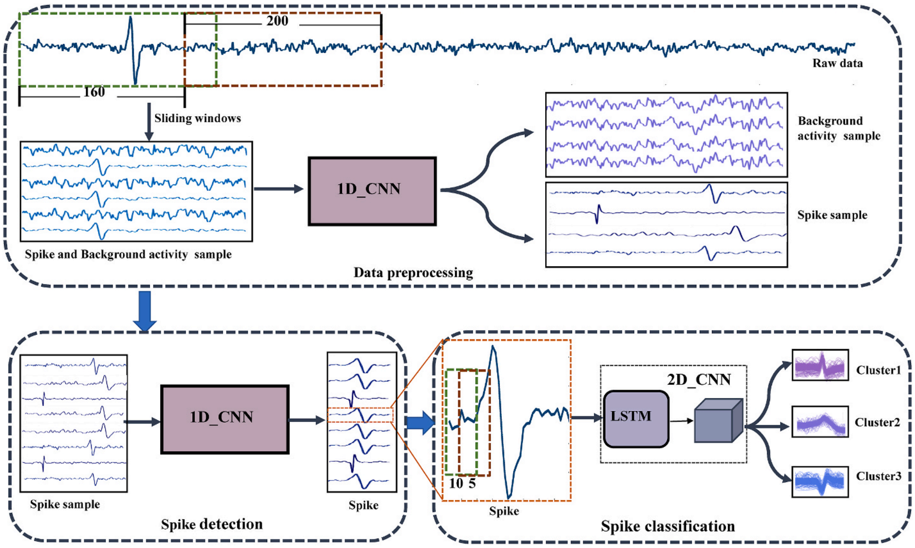

Fig. 1. Overall framework of the proposed method. The preprocessing model and the detection model use 1D_CNN, and the spike classification model uses the LSTM and 2D_CNN hybrid model.

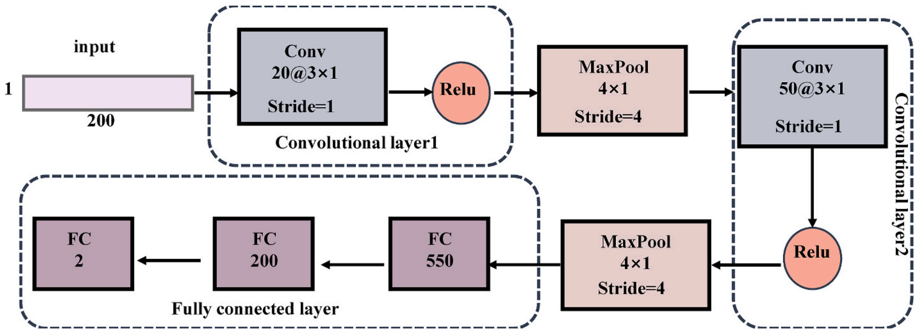

Fig. 2. Network architecture for 1D_CNN model. Both the preprocessing and the spike detection used 1D_CNN model. The sample size of input data and the parameters of models are the same for the preprocessing and spike detection.

In the spike detection, the sample sequence is input into the second convolution layer, and the last one is predicted through the full connection layer. According to the threshold value of the tag, the start position (t_s) and end position (t_e) of the peak are predicted. The process can be described as follows:

$$t_s, t_e = F\left(Max\left(\delta\left(f^{3*1}\right)\right)\right) \tag{3}$$

2.1.2. LSTM_CNN model

LSTM not only overcomes the long-distance dependence problem of RNN, but also has the characteristics of maintaining long-term memory and shows good performance in time series data processing. In this study, LSTM was used to extract the temporal features, then 2D_CNN was applied to classify the clusters, as shown in Fig. 3. First, the spike waveform was divided into series of 20-point segments by a sliding window. Then, the temporal features were obtained using LSTM and stitched into a matrix to construct more explicit feature. The feature extraction was performed by a tow-layer 2D_CNN, having a stride size of 1 and kernel size of 3 × 3 and 1 × 1, respectively. The output size of the fully connected layer depends on the number of clusters, for example there are three kinds of spikes in the simulated data, that is, the number of output dimensions is 3.

Spike signal is defined as E = $(e_1, e_2, \ldots, e_N) \in S^{N*1*L}$. A sliding window is used to segment the spike signal. The whole process can be expressed as follows:

$$W_e = \left(w^{1*l*t}(E)\right) \in S^{N*1*L} \tag{4}$$

Where w^{1*s} represents a sliding window with a length of s, a width of l, and a step of t.

Then the data segmented from the sliding window are sent into the LSTM network in turn, where the number of features of the hidden layer is 7. Seven 1-D time series features are obtained. which are spliced into 7 × 7 2-D features and then are sent to the convolution layer for enhanced learning to obtain advanced time features. The whole process can be expressed as:

$$C_e = \delta\left(f^{3*3}\beta(LSTM(W_e))\right) \in S^{N*1*s} \tag{5}$$

Where β represents the splicing of one-dimensional features output by LSTM into 2-dimensional features.

Finally, the results are classified into three categories according to the threshold value of the label through the full connection layer. The

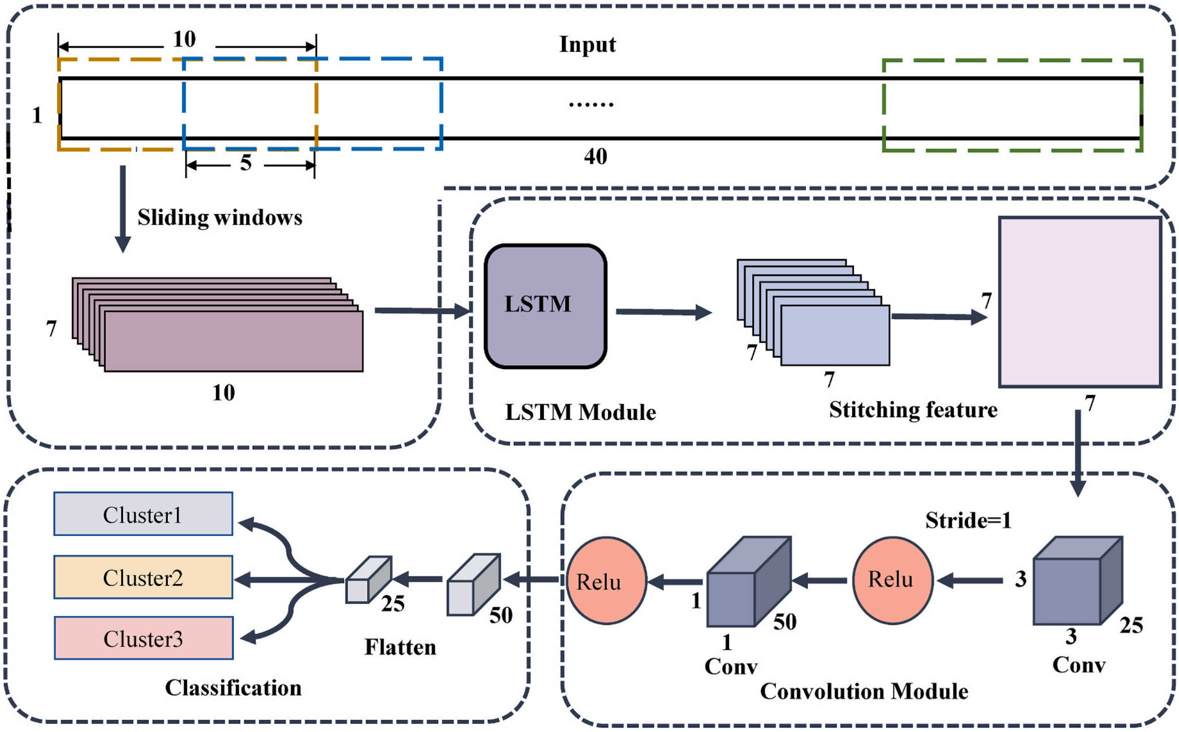

Fig. 3. Network architecture for LSTM_CNN model.

process can be described as:

$$Y_e = \alpha\left(F\left(\delta\left(f^{1*1}(C_e)\right)\right)\right) \in R^{N*3} \tag{6}$$

2.2. Database

In this study we used four database, which are represented by DS1, DS2, DS3, and DS4, as shown in Table 1 and Table 2. All database used in this paper have been filtered.

DS1 is a database from Ref. [31], which simulated spikes from three neurons with an sampling rate of 20 kHz. In this paper, four subsets were included and to mimic the effect of noise, each subset was set at different noise levels ranging from 0.5 to 2 with a step of 0.25.

DS2 is a widely used simulated database created in “Wave_Clus [12, 18,32,33]. This database is produced by 594 different averaged spike waveforms from experimental recordings with a sampling rate of 24 kHz. According to the overlapping degrees between spikes, four subsets were used in this paper, namely C_Easy1, C_Easy2, C_Difficult1 and C_Difficult2. Also, the noise level of dataset varies.from 0.05 to 0.4.

DS3 is a simulated database from Ref. [34]which is generated by adding real spike waveforms to background of varied noise levels with an sampling rate of 24 kHz. In this paper, we used the S2 dataset of 960 s length and 3 types of spikes.

DS4 is an experimental database, collected by a 64-channel multi-electrode array from the primary visual cortex (V1) of macaque monkeys with a sampling rate of 24.4 kHz [35,36]. In this paper, channels were randomly selected with different number of spike clusters. This database only contains well-labeled spikes, thus we employed it to evaluate the LSTM_CNN model.

2.3. Performance metrics

In this paper, five metrics were used to valid the performance of our model. The Intersection of Union will be referenced in subsection Detection, while the other four metrics will be mentioned in Pre-processing and Classification.

The Intersection of Union (IoU) is defined as the degree of coincidence between the detected sample and templete. The IoU ranged from 0 to 1, the higher the coincidence rate, the higher the value of IoU is. To verify the quality of the detection model, we set the initial threshold at 0.5 according to the criteria in PASCAL VOC competition. It is given analytically by equation (7).

$$IoU = \frac{A \cap B}{A \cup B} \tag{7}$$

Mathematically, recall, precision, accuracy, and F1-score were calculated by the following equations, respectively:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{9}$$

$$Accuracy = \frac{(TP + TN)}{All} \tag{10}$$

$$\text{F1\_score} = \frac{2Recall * Precision}{Precision + Recall} \tag{11}$$

where TP refers to true positives, FN refers to false negatives, FP refers to false positives, TN refers to true negatives. Notably, recall was also employed in the subsection Detection, where TP denotes the samples whose IoU is greater than the threshold, and FN denotes the samples whose IoU goes below the threshold.

Table 1
Details of simulated database.

Database	Noise level	Number of Neurons		
DS1	0.5–2	2633	2696	2744
DS2				
C_Easy1	0.05	1165	1157	1192
	0.10	1151	1134	1237
	0.15	1132	1188	1157
	0.20	1198	1128	1148
	0.25	1094	1089	1115
	0.30	1162	1164	1149
	0.35	1208	1137	1189
	0.40	1079	1158	1149
C_Eay2	0.05	1130	1113	1167
	0.10	1160	1146	1214
	0.15	1181	1098	1132
	0.20	1186	1188	1152
C_Difficult1	0.05	1115	1113	1155
	0.10	1164	1155	1129
	0.15	1159	1172	1141
	0.20	1136	1099	1179
C_Difficult2	0.05	1120	1109	1135
	0.10	1187	1136	1139
	0.15	1142	1113	1185
	0.20	1151	1195	1147
DS3				
S2		2431	3277	8139

Table 2
Details of experimental database.

Database	Channels	Number of Neurons				
DS4	66	10076	6919	3455		
	69	11719	11183	9140		
	98	103	9998	1067		
	115	679	15541	12832		
	68	889	7017	2068	2164	
	70	1672	8910	5448	1979	
	83	2233	7289	9073	3063	
	95	522	37769	13478	2980	
	82	76	9441	1216	478	494
	126	11853	14038	10046	4460	4011

3. Results

This section consists of four subsections. First, we validated the subsection Data preprocessing and Spike detection with the database DS1. Then, both the simulated database DS2 and the experimental database DS4 were used in subsection Spike classification. Finally, the whole proposed method applied to the simulated dataset DS3.

3.1. Data preprocessing

The first step of our model is dividing the raw data into segments by a sliding window and extracting spike samples by 1D_CNN model. Fig. 4 visualized segments at different noise levels ranged from 0.5 to 2. We manually labeled samples and got 8073 spike samples and 66924 background activity samples in total. In order to avoid bias in the training process, we randomly selected background activity samples with the same number of spike samples.

We trained the model on labeled samples by taking 70% of the database as the training set and 30% as the test set. Table 3 presents the precision, recall and accuracy for data with different noise levels. Overall, although the performance of model decreases with the increasing noise level, the accuracy is above 90% for data at the noise level of 1.75.

3.2. Spike detection

After that, the second 1D_CNN model was performed to detect the spike waveform from a 200-datapoint spike sample. Fig. 5 presents the

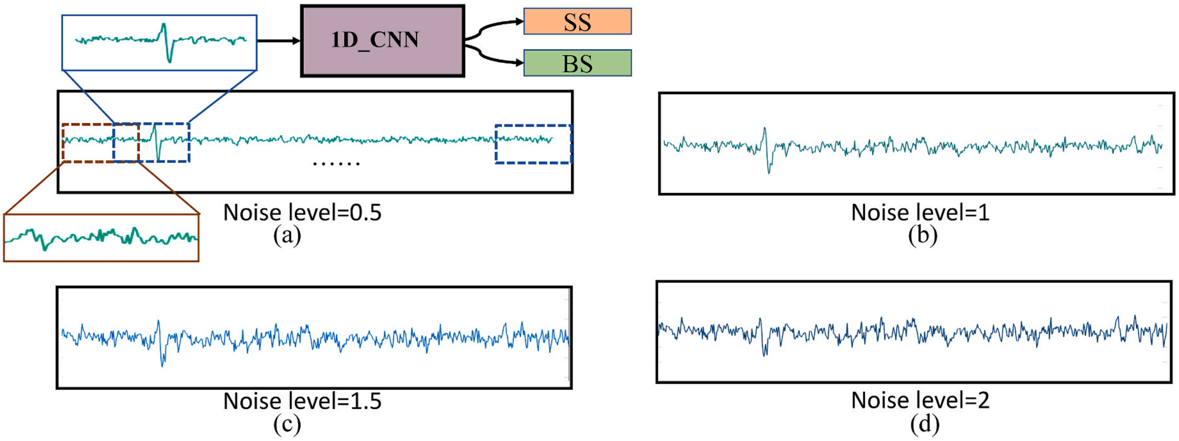

Fig. 4. Examples of segment with different noise levels of 0.5, 1, 1.5, and 2, respectively. Segments were labeled as spike sample (SS) or background activity sample (BS) according to whether it contained spike or not.

Table 3
Performance evaluation of the data preprocessing model on database DS1.

Noise level	NS (Precision)	SS (Precision)	NS (Recall)	SS (Recall)	Accuracy
0.5	99.97	99.87	99.87	99.97	99.92
0.75	99.80	99.94	99.93	99.81	99.87
1	99.40	99.62	99.60	99.43	99.51
1.25	97.05	98.96	98.88	97.24	98.02
1.5	94.62	95.81	95.66	94.81	95.22
1.75	91.31	90.19	89.52	91.87	90.72
2	87.92	83.54	83.06	88.28	85.64

examples of detection result when the IoU threshold was set at different values. The results of each indicator of the 1D_CNN model are shown in Table 4, and it can be seen that the results are relatively stable without any noise reduction treatment.

Then we defined the results as true positive when their IoU were higher than the threshold, and calculated the recall for data with different noise levels, as shown in Fig. 6(a). The recall rate of our model is 95.45% with the IoU threshold of 0.5 and the noise level of 0.5, and it decreases significantly with increasing noise level. Whereas the recall slightly decreases to 94.40% with the IoU threshold of 0.8 and the noise level of 0.5, suggesting the model predicts the spike position of the spikes accurately and has a higher recall rate.

Moreover, the recall of our model is compared with RNN, LSTM, and GRU in Fig. 6(b). Notably, the noise level is an important factor in spike detection. For low noise levels, the recalls of RNN and GRU are 92.71% and 91.91%, respectively, which are lower than the results of LSTM and our model. Although the downtrend in recall of four models are similar, best performance is still achieved using our model with recall of 77.83%, comparing with RNN 39.78%, LSTM 57.99% and GRU 50.23%.

3.3. *Spike classification*

In this subsection, the performance of the proposed classification model was compared with four traditional methods based methods metioned in Ref. [12] and deep learning methods. Simulated database DS2 and the experimental database DS4 were employed in this section.

3.3.1. *Simulated database*

Table 5 reports the accuracy of the proposed LSTM_CNN model by setting 5%, 10%, 20%, 30%, and 40% of the total data as training set, respectively. It has shown that the accuracy is above 99% in most case even only 5% of spikes were used to train the model, and a slight upward

Table 4
Performance evaluation of the 1D_CNN model on database DS1.

Noise level	Precision	F1	Recall
NL 0.5	91.35	93.41	95.57
NL 0.75	91.10	93.23	95.44
NL 1	91.19	93.29	95.49
NL 1.25	89.66	92.11	94.69
NL 1.5	84.71	88.22	92.04
NL 1.75	80.95	85.22	89.97
NL 2	70.33	76.50	83.86

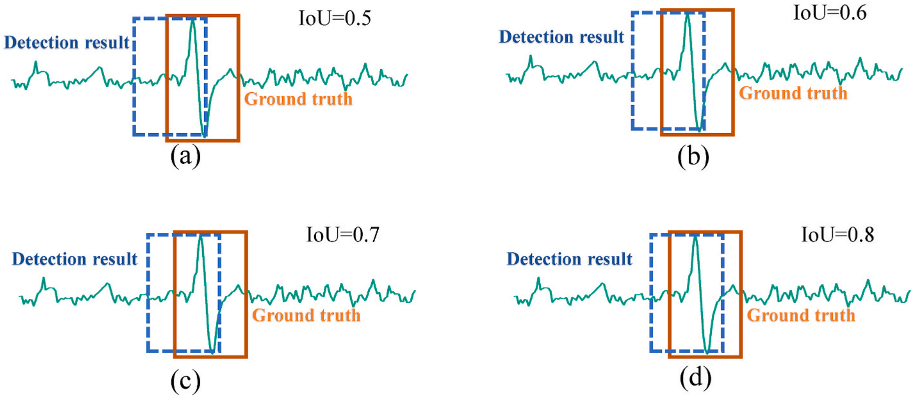

Fig. 5. Examples of spike detection result under different IoU thresholds. The IoU of detection spike waveform (blue dashed box) and ground truth (blue dashed box) was set as 0.5(a), 0.6(b), 0.7(c) and 0.8(d), respectively.

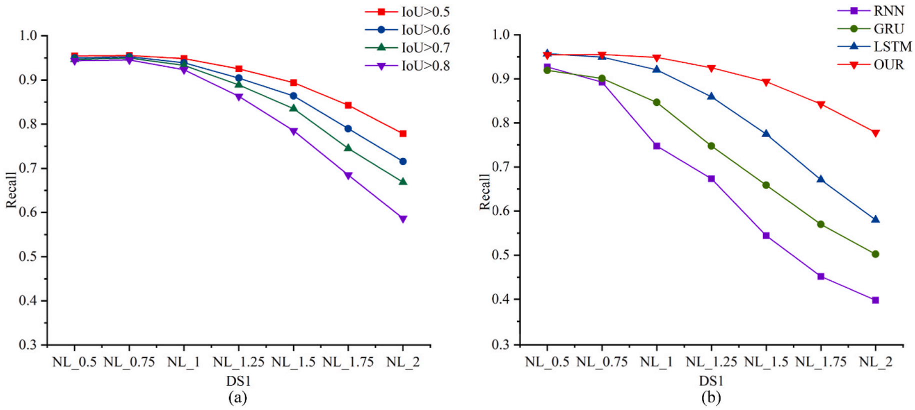

Fig. 6. Comparison of recall for data with different noise levels. The effect of IoU threshold on recall (a) and the comparison of recall between our model and LSTM, GRU, RNN models(b) with IoU threshold of 0.5.

Table 5
Results of classification experiments using different proportions of the training set. E1, E2, E3, E4, and E5 represent 5%, 10%, 20%, 30%, and 40% of the training set, respectively.

DS2	Noise level	E1	E2	E3	E4	E5
C_Easy1_	0.05	99.52	99.59	99.57	99.51	99.62
C_Easy1_	0.10	99.58	99.56	99.68	99.76	99.86
C_Easy1_	0.15	99.51	99.42	99.56	99.63	99.66
C_Easy1_	0.20	99.27	99.55	99.49	99.59	99.62
C_Easy1_	0.25	99.58	99.66	99.70	99.70	99.64
C_Easy1_	0.30	99.12	99.36	99.75	99.75	99.81
C_Easy1_	0.35	98.62	99.43	99.64	99.63	99.72
C_Easy1_	0.40	98.66	99.57	99.78	99.75	99.75
C_Easy2_	0.05	98.73	99.64	99.82	99.70	99.75
C_Easy2_	0.10	99.49	99.65	99.61	99.72	99.81
C_Easy2_	0.15	99.35	99.44	99.41	99.45	99.50
C_Easy2_	0.20	99.25	99.40	99.43	99.72	99.76
C_Difficult1_	0.05	98.31	98.59	98.59	98.69	98.71
C_Difficult1_	0.10	98.00	98.58	98.69	99.00	99.17
C_Difficult1_	0.15	95.93	97.52	98.58	98.71	98.58
C_Difficult1_	0.20	95.85	97.85	98.79	98.90	99.02
C_Difficult2_	0.05	99.59	99.70	99.74	99.83	99.75
C_Difficult2_	0.10	99.48	99.52	99.64	99.75	99.71
C_Difficult2_	0.15	99.33	99.54	99.60	99.58	99.76
C_Difficult2_	0.20	99.73	99.68	99.78	99.84	99.86

trend presents as a result of increasing training data. On the other hand, the accuracy shows no significant reduction as the noise level increased. Both factors suggest the good robustness of the proposed model.

Furthermore, Table\ref{Table 6} reports the accuracy of the proposed LSTM_CNN model with traditional methods including utilized PCA-based and correlation-based (CORR) feature extraction with FCM as the clustering method, other public method (fusion + SVM) and WMsorting [12]. Traditional methods deteriorate with increasing noise levels across all datasets, with an error rate of up to 46%, while ours is around 2%. For lower noise levels, the accuracy of the method still outperforms traditional methods. We find that "WMsorting" has a high accuracy on the Easy dataset, and our model can achieve more than 99% accuracy even though it is only trained with 5% of the data.

Table 7 reports the accuracy of the proposed LSTM_CNN model with deep learning methods including GRU, LSTM, RNN and CNN. As can be seen, the results of GRU are very unstable, with an accuracy of only 34.19% on C_Easy1_010 and an average accuracy of 50.32% on C_Difficult1. Although RNN and CNN get better performance than GRU and LSTM, the improvement of our model have obtained on most datasets, especially on dataset C_Difficult1. Taken together, our deep learning method not only achieves a high accuracy with small samples of training set (5%), but also performs excellently for datasets with varied noise level.

Table 6
Comparison of classification accuracy using four traditional methods and our 1D_CNN model on database DS2.

DS2	Noise level	PCA + FCM	CORR + FCM	Fusion + SVM	WMsorting [12]	OUR
C_Easy1_	0.05	99.37	97.50	98.66	99.60	**99.62**
C_Easy1_	0.10	99.72	96.45	98.98	99.66	**99.86**
C_Easy1_	0.15	99.28	94.94	98.22	**99.71**	99.66
C_Easy1_	0.20	99.40	92.43	97.35	99.57	**99.62**
C_Easy1_	0.25	99.24	86.84	95.45	99.45	**99.70**
C_Easy1_	0.30	98.73	81.50	88.66	99.57	**99.81**
C_Easy1_	0.35	97.76	77.02	83.22	99.43	**99.72**
C_Easy1_	0.40	96.49	75.58	78.12	99.76	**99.78**
C_Easy2_	0.05	98.68	96.04	92.23	99.50	**99.82**
C_Easy2_	0.10	98.24	86.02	92.93	99.52	**99.81**
C_Easy2_	0.15	94.49	83.05	89.80	99.44	**99.50**
C_Easy2_	0.20	88.60	79.81	86.24	99.29	**99.76**
C_Difficult1_	0.05	95.86	86.08	97.58	95.00	**98.71**
C_Difficult1_	0.10	89.56	71.55	94.81	93.76	**99.17**
C_Difficult1_	0.15	76.41	58.84	87.85	91.18	**98.58**
C_Difficult1_	0.20	63.03	53.81	78.59	86.45	**99.02**
C_Difficult2_	0.05	98.81	94.50	87.40	99.44	**99.83**
C_Difficult2_	0.10	98.76	96.33	88.07	99.51	**99.75**
C_Difficult2_	0.15	97.33	96.02	74.65	99.51	**99.76**
C_Difficult2_	0.20	84.63	95.48	67.25	99.66	**99.86**

Table 7
Comparison of classification accuracy using four deep learning methods and our proposed model on database DS2.

DS2	Noise level	GRU	LSTM	RNN	CNN	OUR
C_Easy1_	0.05	85.91	46.35	99.52	99.51	**99.62**
C_Easy1_	0.10	34.19	48.63	99.76	99.68	**99.86**
C_Easy1_	0.15	99.33	48.46	99.66	**99.71**	99.66
C_Easy1_	0.20	98.64	41.07	99.51	99.59	**99.62**
C_Easy1_	0.25	98.26	40.23	99.39	99.57	**99.64**
C_Easy1_	0.30	97.57	40.63	98.97	99.51	**99.81**
C_Easy1_	0.35	97.87	47.68	99.24	99.43	**99.72**
C_Easy1_	0.40	97.92	46.53	99.31	99.45	**99.75**
C_Easy2_	0.05	94.79	42.41	99.06	99.54	**99.75**
C_Easy2_	0.10	93.75	41.95	99.34	99.47	**99.81**
C_Easy2_	0.15	90.63	43.01	98.71	**99.51**	99.50
C_Easy2_	0.20	87.59	41.90	99.05	99.59	**99.76**
C_Difficult1_	0.05	52.63	34.28	98.56	98.16	**98.71**
C_Difficult1_	0.10	49.66	32.91	98.10	98.75	**99.17**
C_Difficult1_	0.15	50.05	33.08	95.58	98.54	**98.58**
C_Difficult1_	0.20	48.93	34.23	93.46	98.73	**99.02**
C_Difficult2_	0.05	93.11	37.35	98.81	99.63	**99.75**
C_Difficult2_	0.10	91.46	34.67	99.22	99.67	**99.71**
C_Difficult2_	0.15	85.40	35.35	99.36	99.54	**99.76**
C_Difficult2_	0.20	81.49	35.19	99.42	99.55	**99.86**

In order to verify the performance of LSTM_CNN model, we compared the run time of our model with the execution time of CNN, as shown in Table 8. As tested in dataset DS2, we preformed CNN and LSTM_CNN 50 times in training step and one time in testing step, and the results show that the run time of CNN model in training step is indeed shorter than that of our model, but there is no significant difference in testing step with trained LSTM_CNN and CNN. As can be seen from the variance of training and the variance of testing, the LSTM_CNN model is more stable.

3.3.2. Experimental database

Experiments were performed with different proportions of training data on the experimental database, as shown in Table 9. It can be seen that the accuracy of most channels is above 80% using 5% of data as training set, and the average accuracy improves to about 95% when the proportion of training data increases to 40%. It is also noted that the accuracy of channel 70, channel 82 and channel 83 is lower than that of other channels even with 40% of spikes as training set.

We further calculated the precision, recall and F1 function of these three channels, as shown in Table 10. Taken the results of channel 70 as an example, the precision and recall rate of cluster1 and cluster4 are significantly lower than those of the other two categories. The reason for this is that the number of spikes is 1672 in cluster1 and 1979 in cluster4, accounting for 9.2% and 10.98% of the total data, respectively (as shown in Table 1). This data unbalance problem also presents in channel 82 and channel 83, where the accuracy and F1 are affected significantly.

Table 11 shows the comparison of classification accuracy using different models on experimental database. 40% of the data are employed as the training set to ensure the best performance of all deep learning methods. Obviously, the proposed model performs better than RNN, LSTM, and GRU in all channels, and only lower than CNN in channel 70.

3.4. Overall model validation

The DS3 database was used to validate the overall performance of this study. In spike detection, 70% of the spike signals were used as the training set and 30% as the test set. Then in spike classification, 40% of the detected spikes were used as the training set and 60% as the test set.

Table 8
The running time using CNN model and our proposed model on some database DS2.

DS2	Noise level	CNN		OUR	
		Train/s	Test/s	Train/s	Test/s
C_Easy1_	0.05	46.49	2.45	52.745	2.62
C_Easy1_	0.10	47.59	2.56	54.02	2.32
C_Easy1_	0.15	48.31	2.28	52.12	2.34
C_Easy1_	0.20	43.29	2.50	52.95	2.40
C_Easy1_	0.25	40.27	2.32	50.91	2.29
C_Easy1_	0.30	42.26	2.36	52.85	2.33
C_Easy1_	0.35	42.69	2.53	54.56	2.49
C_Easy1_	0.40	40.96	2.55	51.89	2.31
C_Easy2_	0.05	41.31	2.47	51.74	2.34
C_Easy2_	0.10	42.49	2.63	54.59	2.34
C_Easy2_	0.15	40.78	2.29	51.41	2.37
C_Easy2_	0.20	42.41	2.27	53.49	2.33
C_Difficult1_	0.05	46.49	2.32	52.07	2.44
C_Difficult1_	0.10	47.59	2.58	54.94	2.43
C_Difficult1_	0.15	48.31	2.52	52.10	2.45
C_Difficult1_	0.20	43.69	2.58	51.32	2.43
C_Difficult2_	0.05	40.63	2.48	52.99	2.41
C_Difficult2_	0.10	41.13	2.41	51.28	2.43
C_Difficult2_	0.15	41.10	2.37	52.38	2.44
C_Difficult2_	0.20	41.88	2.59	53.44	2.49
Mean		**43.48**	2.45	52.69	**2.4**
Variance		**2.77**	0.11	**1.16**	**0.08**

Table 9
Classification accuracy using different proportions of the training set on database DS4. E1, E2, E3, E4, and E5 represent 5%, 10%, 20%, 30%, and 40% of the training set, respectively.

Channels	Neurons	E1	E2	E3	E4	E5
66	3	86.82	88.05	90.33	92.86	95.48
69	3	78.90	83.24	89.60	95.32	96.03
98	3	89.60	89.93	91.96	94.30	96.17
115	3	87.53	94.92	94.81	95.90	96.55
68	4	86.77	87.66	90.46	91.82	93.44
70	4	85.61	87.66	88.81	89.83	91.49
83	4	78.52	81.04	84.33	86.91	90.17
95	4	94.14	97.36	97.32	96.87	98.61
82	5	82.65	84.52	86.45	87.57	92.17
126	5	86.21	92.78	96.19	96.46	97.63
Mean		**85.68**	**88.72**	**91.03**	**92.78**	**94.77**

3.4.1. Spike detection

The results of DS3 detection are shown in Table 12. When the threshold value of IoU is 0.5, the spike recall rate of our model is 94.54%, which is much higher than that of LSTM, RNN and GRU. What's more, the accuracy of our proposed model decreases much more slowly with the increase of the IoU threshold than that of other models, indicating the better stability of the model. Since the IOU describes the overlap of detection waveform and template, the larger the IoU threshold is set, the better the quality of the detection spike will be. Thus, we set the IoU threshold to 0.8 to bring more information for the next classification section.

3.4.2. Spike classification

The detected spikes were labeled with the corresponding categories, then deep learning models were applied for classification, as shown in Table 13. It is obviously that RNN, CNN and the proposed model obtain good results, and among them, the accuracy of our model exhibits a slightly increase to 97.17%. The results of spike classification on DS3 illustrate that our detection and classification model is an accurate and robust method for spike sorting.

4. Discussion

This paper designed a spike sorting model including three parts, i.e., data preprocessing, spike detection, and spike classification. We evaluated the method with both simulated database and experimental database which are widely employed in the spike sorting literatures.

For the preprocessing part, a sliding window of 200 datapoints was set to segment the data, and 1D_CNN model was preformed to detect whether the segment contained spike or not. This process is similar to the single-target detection task, thus we try to ensure that only one spike or no spike for each template. The results of this part show that the preprocessing model in this study performs well for both SS and NS categories, indicating that this part could reduce the background noise interference effectively.

The spike detection part also used 1D_CNN in the simulated database DS1 with various noise levels, and the performance of this section was evaluated by recall. IoU was set to measure the coincidence rate between the detection result and the target spikes. Although the recalls decrease with high noise levels, our model still outperforms the other deep learning models, i.e., GRU, RNN, LSTM, suggesting 1D_CNN can detect the location of spikes more accurately and more robust.

The accuracy and robustness of spike classification was improved by the model combined LSTM and CNN. There are four factors may affect the method: labels, the similarity between clusters, the datapoints of spike waveforms and the uneven distribution of clusters [18]. First, spikes in this study were manually labeled, thus the quality of the labels will affect the results of the classification. Second, in database DS2,

Table 10
Precision, Recall, F1_score of each category for channel 70, channel82 and channel 83. P/R stands for Precision and Recall.

Channels	Cluster1 (P/R)	Cluster2 (P/R)	Cluster3 (P/R)	Cluster4 (P/R)	Cluster5 (P/R)	F1
70	69.57/85.78	98.32/98.19	93.91/87.13	71.92/76.93	–	84.96
82	0/0	98.89/92.75	73.86/87.95	72/90.91	42.41/87.90	71.90
83	70.14/70.02	92.73/91.16	92.37/94.91	91.19/87.81	–	86.27

Table 11
The classification results of RNN, LSTM, GRU, CNN and our model. The bold part represents the best results.

Channels	cluster	RNN	LSTM	GRU	CNN	OUR
66	3	86.98	50.56	82.63	95.36	**95.48**
69	3	79.58	58.70	73.85	95.45	**96.03**
98	3	89.83	89.80	89.74	95.17	**96.17**
115	3	87.78	88	87.49	92.39	**95.47**
68	4	86.71	57.56	73.48	91.22	**93.44**
70	4	87.84	73.35	81.12	**92.78**	91.49
83	4	83.81	51.04	73.51	89.93	**90.17**
95	4	94.46	88.83	92.87	98.50	**98.61**
82	5	84.99	80.63	80.79	90.88	**92.17**
126	5	88.39	61.06	84.70	97.36	**97.63**

Table 12
The detection accuracy of different models at different IoU thresholds on database DS3.

IoU	RNN	LSTM	GRU	OUR
0.5	50.21	91.17	45.76	**94.54**
0.6	45.83	89.29	41.51	**93.28**
0.7	43.04	87.36	39.14	**91.7**
0.8	38.14	80.5	34.47	**84.1**

Table 13
Precision, recall, F1_score of spike classification using different models, where P/R stands for precision and recall, respectively.

Model	Cluster1(P/R)	Cluster2(P/R)	Cluster3(P/R)	F1	Accuracy
RNN	93.12/94.42	93.25/94.19	97.01/96.34	94.72	95.60
LSTM	0/0	0/0	1/63.26	25.83	63.26
GRU	38.26/73.48	86.93/84.91	92.41/81.58	74.30	81.43
CNN	92.95/94.25	95.83/97.38	97.75/96.92	95.84	96.51
Our	96.27/93.87	95.23/98.76	97.99/97.71	**96.62**	**97.17**

although the accuracy of most subsets is above 99%, the classification accuracy of C_Difficult1 is relatively lower than that of others. This phenomena also has been described in Ref. [18], which plotted the waveforms of C_Easy1 and C_Difficult1 and noted the similarity between clusters as an important factor for classification. What's more, the accuracy of the experimental database is lower than that of the simulated database, especially for channel 70, channel 82 and channel 83. One reason is that the proposed model learns less features from 12-point experimental spikes than that from 40-point simulated spikes. On the other hand, as with other deep learning methods, the imbalanced data adds challenges to the classification accuracy. Nonetheless, the model proposed in this study performs well in terms of both accuracy and robustness, and the average accuracy of our model is improved comparing with WMsorting and several deep learning models. From Table 8, it can be seen that the time of LSTM_CNN training 50 times is indeed longer than that of CNN, but it is also acceptable. The LSTM_CNN test time and accuracy are better than CNN, although the improvement is not obvious. The most important is that the classification effect of LSTM_CNN combined with our detection model is significantly better than that of CNN, as shown in Table 13.

The overall model was validated using DS3, and the accuracy reported in Table 10 reached 96.51% using CNN and 97.17% using our model. The good performance indicated that our algorithm not only provides a good solution for automatic spike detection, but also proves that LSTM_CNN has strong generalization ability.

5. Conclusion

This paper proposed a new 1D_CNN-based automatic spike detection method and a sliding window feature extraction combined with LSTM_CNN spike classification method. The results suggest that our proposed model can be applied in automatic detection and classification of spikes. However, the influence of noise on the detection of spikes by 1D_CNN is undeniable. A significant decreasing trend of accuracy presents with the increase of noise level in the simulated data. Therefore, a more robust network needs be constructed to deal with the low signal-to-noise data in future work. In addition, for the experimental data, one electrode usually can record the firing of multiple neurons and the number of each cluster is uneven distribution, thus the imbalanced classification would be another area of interest.

Declaration of competing interest

The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript and there is no financial interest to report. We certify that the submission is original work and is not under review at any other publication.

Acknowledgement

This work is supported by the foundation of Science and Technology Innovation 2030 major Project (No.2022ZD0208500) and the Scientifific Research Foundation of CUIT (No. KYQN202241).

References

[1] I. Arel, D.C. Rose, T.P. Karnowski, Deep machine learning - a new Frontier in artificial intelligence research [research frontier], IEEE Comput. Intell. Mag. 5 (4) (2010) 13–18.
[2] H. Azami, et al., Extracellular spike detection from multiple electrode array using novel intelligent filter and ensemble fuzzy decision making, J. Neurosci. Methods 239 (2015) 129–138.
[3] Antal, et al., Large-scale, high-density (up to 512 channels) recording of local circuits in behaving animals, J. Neurophysiol. 111 (3) (2014) 1132–1149.
[4] C. Gold, On the origin of the extracellular action potential waveform: a modeling study, J. Neurophysiol. 95 (5) (2006) 3113–3128.
[5] M.S. Lewicki, A review of methods for spike sorting: the detection and classification of neural action potentials, Network 9 (4) (1998) 53–78.
[6] I. Obeid, P.D. Wolf, Evaluation of spike-detection algorithms for a brain-machine interface application, Biomed. Eng. IEEE Trans. 51 (6) (2004) 905–911.
[7] M. Saif-Ur-Rehman, et al., SpikeDeeptector: a deep-learning based method for detection of neural spiking activity, J. Neural. Eng. 16 (5) (2019), 056003.
[8] M. Saif-ur-Rehman, et al., SpikeDeep-Classifier: a deep-learning based fully automatic offline spike sorting algorithm, J. Neural. Eng. 18 (1) (2021), 016009.
[9] S. Gibson, J.W. Judy, D. Markovic, Spike sorting: the first step in decoding the brain: the first step in decoding the brain, IEEE Signal Process. Mag. 29 (1) (2012) 124–143.
[10] R.Q. Quiroga, Z. Nadasdy, Y. Ben-Shaul, Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering, Neural Comput. 16 (8) (2004) 1661–1687.
[11] C. Seong, W. Lee, D. Jeon, A multi-channel spike sorting processor with accurate clustering algorithm using convolutional autoencoder, IEEE Trans. Biomed. Circ. Syst. 15 (6) (2021) 1441–1453.
[12] L. Huang, et al., WMsorting: wavelet packets decomposition and mutual information based spike sorting method, IEEE Trans. NanoBio. (2019) 283–295.
[13] Chah, et al., Automated spike sorting algorithm based on Laplacian eigenmaps and *k*-means clustering, J. Neural. Eng. 8 (1) (2011), 016006.
[14] S.N. Kadir, et al., Spike sorting for large dense arrays, Front. Neuroinf. 7 (2013).

[15] F. Wood, et al., On the variability of manual spike sorting, IEEE (Inst. Electr. Electron. Eng.) Trans. Biomed. Eng. 51 (6) (2004) 912–918.

[16] K.D. Harris, et al., Improving data quality in neuronal population recordings, Nat. Neurosci. 19 (9) (2016) 1165.

[17] I.Y. Park, et al., Deep learning-based template matching spike classification for extracellular recordings, Appl. Sci. 10 (1) (2019) 301.

[18] Z. Li, et al., An accurate and robust method for spike sorting based on convolutional neural networks, Brain Sci. 10 (11) (2020) 835.

[19] M. Rácz, et al., Spike detection and sorting with deep learning, J. Neural. Eng. 17 (1) (2019).

[20] J. Dai, , et al.R-fcn, Object detection via region-based fully convolutional networks, Adv. Neural Inf. Process. Syst. 29 (2016).

[21] J. Redmon, et al., You Only Look once: Unified, Real-Time Object Detection, IEEE, 2016.

[22] R. Girshick, et al., Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[23] M.W. Nadeem, et al., Brain tumor analysis empowered with deep learning: a review, taxonomy, and future challenges, Brain Sci. 10 (2) (2020) 1–33.

[24] Ali, M.B., et al., Domain mapping and deep learning from multiple MRI clinical datasets for prediction of molecular subtypes in low grade gliomas. Brain Sci.. 10 (7).

[25] Akshay, et al., Comparison of logistic regression, support vector machines, and deep learning classifiers for predicting memory encoding success using human intracranial EEG recordings, J. Neural. Eng. 15 (6) (2018) 066028.

[26] A. Craik, Y. He, J. Contreras-Vidal, Deep learning for Electroencephalogram (EEG) classification tasks: a review, J. Neural. Eng. 16 (3) (2019).

[27] Hochreiter, et al., Long short-term memory, Neural Comput. (1997).

[28] J. Chen, D. Liu, W.U. Dashuo, Stock index forecasting method based on feature selection and LSTM Model, Comput. Eng. Appl. (2019).

[29] K. Kumar, M.T.U. Haider, Enhanced prediction of intra-day stock market using metaheuristic optimization on RNN–LSTM network, New Generat. Comput. 39 (10) (2020) 1–42.

[30] Y. Hou, et al., A novel approach of decoding EEG four-class motor imagery tasks via scout ESI and CNN, J. Neural. Eng. 17 (1) (2020), 016048.

[31] M. Bernert, B. Yvert, An attention-based spiking neural network for unsupervised spike-sorting, Int. J. Neural Syst. 29 (8) (2018) 1850059.

[32] K. Yang, H. Wu, Y. Zeng, A simple deep learning method for neuronal spike sorting, in: Journal of Physics: Conference Series, IOP Publishing, 2017.

[33] T. Wu, et al., Deep compressive autoencoder for action potential compression in large-scale neural recording, J. Neural. Eng. 15 (6) (2018), 066019.

[34] J. Wild, et al., Performance comparison of extracellular spike sorting algorithms for single-channel recordings, J. Neurosci. Methods 203 (2) (2012) 369–376.

[35] C.C.J. Chu, P.F. Chien, C.P. Hung, Multi-electrode recordings of ongoing activity and responses to parametric stimuli in macaque V1, CRCNS. org 10 (2014) K0J1012K.

[36] C.C.J. Chu, P.F. Chien, C.P. Hung, Tuning dissimilarity explains short distance decline of spontaneous spike correlation in macaque V1, Vis. Res. 96 (2014) 113–132.