Multi-classification of High-Frequency Oscillations Using iEEG Signals and Deep Learning Models

Zayneb Sadek[1,*], Abir Hadriche[1,2], Rahma Maalej[1,3], and Nawel Jmail[1,3]

[1] Digital Research Center of Sfax, Tunisia
[2] Regim Lab, ENIS, Sfax University, Tunisia
[3] Miracl Lab, Sfax University, Tunisia
Email: zayneb.sadek123@gmail.com (Z.S.); Abir.hadriche.tn@ieee.org (A.H.); rahmamaalej1234@gmail.com (R.M.); naweljmail@yahoo.fr (N.J.)
*Corresponding author

***Abstract*—High-Frequency Oscillations (HFOs) have been considered as a potentially useful biomarker for localizing epileptogenic areas in drug-resistant patients requiring pre-surgical intervention, exploiting intracranial electroencephalographic iEEG. Consequently, it's important to create accurate strategies for detecting epileptic seizures. Predicting seizures requires classifying appropriate indicators, which is difficult due to their time-frequency overlap. Convolutional Neural Networks (CNN), one of the deep learning approaches, have demonstrated promising results in analyzing and classifying epilepsy-related iEEG biomarkers. In our study, we proposed to explore three global methods: multiclass of Support Vector Machine (SVM), multiple architecture CNN, and CNN-SVM, on a simulated iEEG dataset and then on a real iEEG signal. Our best results for the studied three models in the classification of HFO have yielded high accuracy rates: GoogLeNet-SVM achieves approximately 99.63% and 94.07% for simulated data (1) and real data (2), respectively, SVM multiclass achieves 98.14% and 88.51% for (1) and (2), respectively, and GoogLeNet achieves 98.52% and 91.85% for (1) and (2), respectively. Hence, we found that our proposed model performs better than other current techniques. These results suggest that deep learning models could be a successful strategy for classifying epilepsy biomarkers and may improve seizure prediction techniques, and hence can enhance epileptic patient's well-being.**

***Keywords*—High-Frequency Oscillations (HFOs), Convolutional Neural Networks (CNN), multiclass Support Vector Machine (SVM), CNN-SVM, GoogLeNet-SVM**

I. Introduction

Epilepsy is the fourth most prevalent neurological condition, featured by its unpredictable seizures. It affects individuals of all ages, including numerous patients, and is considered medically incurable and requires corresponding neurosurgery to achieve seizure freedom [1] for 30% of cases.

Preoperatively identification of epileptogenic zones is crucial for epilepsy surgery [1]. Presurgical biomarkers are therefore crucial in determining regions known as epileptogenic zones, which cause epileptic seizures. Although epileptogenic zones are difficult to localize using inconsistent or insufficient data from many tests, intracranial Electroencephalography (iEEG) technologies [2, 3] are widely adopted for these cases. iEEG recordings are considered a traumatic technique where cortical sub-zone interactions are visualized by placing electrodes directly on the brain [4, 5]. Accurate localization of EZ areas and safe ablation are major factors for successful surgical outcomes [6]. iEEG recordings are employed.to detect epileptogenic regions thanks to their ability to directly record epileptogenic discharges with high temporal and spatial accuracy and are regarded as gold standard in electrophysiology for defining SOZ, which essentially characterizes EZ [7]. One of the potential biomarkers that has been suggested is High-Frequency Oscillations (HFOs) that occur between 80 Hz and 500 Hz of the region where seizures begin and have demonstrated a high degree of accuracy in detecting interictal epileptiform discharges.

HFOs are considered an indicator of seizure build-up, and they particularly show up during the ictal period [8, 9]. It has been well demonstrated and reproduced that HFOs appear much more in SOZ than outside [10–12]. HFOs are characterized as rapid oscillations, typically elapsing from 6 ms to 30 ms with a varied morphometry and frequency range [13]. Consequently, it is well recognized that manually identifying HFOs may be incredibly difficult, time-consuming, and subject to subjective biases [14, 15].

In this context, several algorithms for automated detection have been built and implemented to help significantly minimize the amount of labor needed for HFO analysis and avoid biases induced by human

Manuscript received June 9, 2024; revised July 5, 2024; accepted September 6, 2024; published January 27, 2025.

evaluators [16]. However, the majority of HFO detection methods have only been able to detect HFOs by thresholding instantaneous frequency traces, which may be susceptible to artifact effects and irregular HFO formation [17].

In the clinical study of HFOs, detection accuracy is quite important, and yet, classifying different events is also crucial. HFOs can be categorized as ripples (80–250 Hz) and Fast ripples (250–500 Hz) based on their frequency range [18]. Fast Ripples (FR) are thought to be more focused and intimately associated with epileptogenicity than Ripples (R) [16, 19]. Consequently, a detection and classification framework was proposed to achieve high accuracy, sensitivity, and specificity in classifying HFOs and hence defining EZ.

Several studies in this field of classifying HFOs and promoting seizure detection and for the second time epilepsy diagnosis have proposed to point machine learning and deep learning models.

Lachner-Piza *et al.* [20] proposed a supervised machine learning method that classified HFOs using SVM multi-classification into four classes: FRs, Rs, IESs co-occurring during Rs, and IESs co-occurring during FRs. Sciaraffa *et al.* [21] suggested a supervised machine learning approach using SVM and LDA. Logic regression KNN Multi-classification of HFOs into three classes: Rs, FRs, and FRs co-occurring during Rs. Firpi *et al.* [22] proposed a supervised neural network to distinguish HFOs from baseline activity. Dümpelmann *et al.* [23] presented a radial basis function neural network-based for HFO detector. The detector input features were energy, line length, and instantaneous frequency. Three patients' visually indicated "ripple" HFOs (80–250 Hz) were utilized to train a neural network, and eight more patients were employed to test the detector. Chaibi *et al.* [24] proposed a Decision Tree to classify two classes (HFO and Non-HFO). It has a sensitivity of 66.96% and detects six features correlated with energy and duration. Jrad *et al.* [25] used the multiclass LDA method to classify (Ripples, Fast ripples, Ripple + Fast ripples, and artifacts). It has a median of 80.5%, and energy was computed with a discrete wavelet.

As for Krikid *et al.* [26], they proposed two approaches for the multi-classification of HFOs based on Time-Frequency (TF) analysis. The first approach was a Deep Learning (DL) based on combining certain characteristics extracted from TF representation of HFO with TF associated images binarized. It is divided into four frequency bands: gamma ([30, 80] Hz), high gamma ([80, 120] Hz), Ripples (RS [120, 150] Hz), and Fast Ripples (FRs [250, 500] Hz). This coupling aims to provide a complete characterization of HFO. A second approach focuses on providing an automatic multi-classification method for HFOs based on CNN. They increased the database using augmented reality to generate new TF images and evaluate its impact on CNN model performance.

The proposed algorithm ought to have robust classification capability in HFO subtypes and HFOs concomitant with other intercortical epileptiform discharges.

In this study, the robustness of HFO classification was evaluated using three methods: multiclass SVM, multiple CNN architectures, and a hybrid CNN-SVM model, applied to both simulated and real iEEG signals. The paper is structured as follows: Section II provides an overview of the proposed system, including data collection and the experimental methodology. Section III summarizes the experimental results, while Section IV offers a more in-depth discussion of results, highlights certain limitations, and suggests future directions. Finally, Section V concludes the paper.

II. Methods and Materials

This section presents our adopted architecture for detecting HFOs and automatic labeling of three clusters for the dataset (1) (R [80, 250] Hz, FR [250, 500]) Hz, and SR (spike-ripple) and three clusters for the dataset (2) (R, FR, and R and FR). The steps of our proposed method are illustrated in Fig. 1:

First, iEEG data are mapped into 2D time-frequency plots using Continuous Wavelet Transform (CWT). Then, we explored model development with multiclass SVM architectures, model CNN, and CNN-SVM.

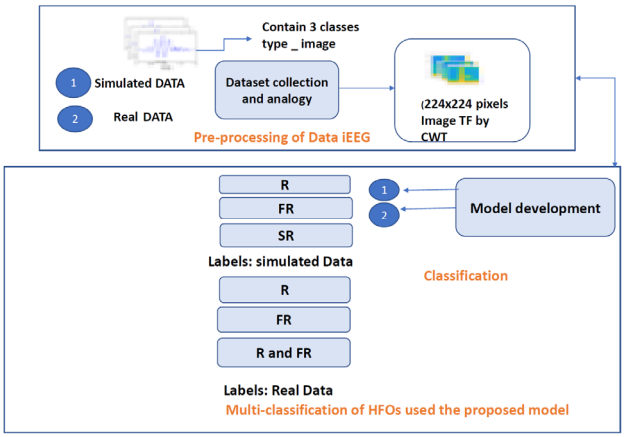


Fig. 1. Proposed model for HFO classification applied on two datasets. Three main steps: data collection, preprocessing step, and developing model.

A. Dataset Description

In this paper, two types of databases have been explored to assess the effectiveness of the suggested approaches:

Simulated data is generated using a peak and HFO shapes combination as a real iEEG signal that is sampled at 512 Hz for 2 s using 1024 samples [27].

We have three classes of signals including HFO (Ripple, Fast Ripple) and overlaid ripple and spike. Through altering various parameters, such as relative amplitudes, frequency of oscillations, Signal-to-Noise Ratio (SNR), and overlap rate.

We obtained 3000 data sets composed of HFO events (Ripple and Fast Ripple), and a mixture of spikes and fast events. We also varied the oscillation frequency in [85, 105, 200, 350, 450] Hz as (ripples and fast ripples).

Real data is obtained in combination with two sources. First, iEEG recordings for a patient with pharmacoresistant epilepsy, where the clinical neurophysiology department of Timone Hospital in Marseille was responsible for the acquisition and preparatory measures [28] and confirmed by expert. IEEG data were sampled at 1000 Hz and recorded on a Deltamed system using a low-pass filter, depicting 117 canals × 30 segments with a frame rate of 300 ms. We investigated 1000 trials of HFOs. Our second studied dataset is iEEG data with marked HFO events [29] sampled at 2000 Hz, for 20 patients, nine of them had mesial Temporal Lobe Epilepsy (TLE) and eleven had Extratemporal Epilepsy (ETE). For each patient 28 intervals, we recorded intervals with 300 ms of frames, and we explored 2220 trials of HFOs.

B. Data Preprocessing

The localized power distribution of HFOs in the frequency and time domains (2D) is the most important characteristic for visual assessment. The scalogram image is represented by the Continuous Wavelet Transform (CWT). *cwt* coefficients of iEEG data. In time-frequency analysis, wavelet analysis is a windowed Fourier analysis with a variable window width that provides details on an event's local frequency structure.

cwt is a convolution of the wavelet function $\Psi(t)$ with the signal $x(t)$ and is given by Eq. (1) [30].

$$cwt(b,k) = \frac{1}{\sqrt{|k|}} \int_{-\infty}^{+\infty} x(t) \Psi\left(\frac{t-b}{k}\right) \mathrm{d}t \qquad (1)$$

where the translation parameter is denoted by b and the scaling parameter by k. Since k and b are continuous parameters, many wavelet coefficients are produced.

The used wavelet for CWT is the Analytic Morlet (complex-valued in a time domain and has one-sided spectra), which depicts equal variance in frequency and time. Hence, we converted a 1D iEEG signal dataset into a scalogram image dataset. Our scalogram images were scaled to 224×224×3 pixels, which varied from 80 to 500 Hz vertically and 300 ms horizontally.

C. Model Development

First, we proposed here to apply machine learning techniques.

1) Multiclass SVM

To explore multi-class classifications, two basic approaches have been proposed, founded on breaking down multi-class into a collection of binary issues.

The initial strategy is known as one-versus-all, in which a group of binary classifiers has been trained to distinguish between each class. Then, based on the highest decision value, each data object is assigned to a class [31]. This method results in N-SVM (where N is the number of classes) with N decision functions. Although, this is a fast method, but it depicts errors due to the slightly unbalanced training sets. The second approach is called "one-on-one", which compares each pair of c classes using support vector machines. Select the class label that best fits the majority of one-on-one challenges after applying all SVMs to a test observation.

Next, the max-win operator is employed to determine which class the object will ultimately be assigned to. This method requires applying $N(N-1)/2$ machines. Compared to the "one-against-all" approach, the "one-on-one" requires more computation, but is found more suitable for multi-class cases.

- **Error-Correcting-Output-Codes (ECOC) for multiclass SVM classification**

The multiclass method is based essentially on Error Correction Output Coding (ECOC). It consists of applying binary (two-class) classifiers to solve multiclass classification problems.

This approach is based on converting the M class classification problem into a large number L.

ECOC represents a unique codeword to a class instead of assigning a label to each class. An error correction code (L, M, d) is L bits long, with C a single codeword with a hamming distance d. Between two codewords, the hamming distance presents a number difference of bit positions. In a classification issue, where M is the number of classes and L is a number determined by error correcting codes method.

Serval techniques are suggested, including BCH codes [32], and exhaustive codes [33] to resolve error corrections codes.

Krizhevsky *et al.* [34] proposed using maximum Hamming distance and suggested that errors $(d-1)/2$ can be corrected in codewords for a Hamming distance d between codes (which counts the different number of bits). Decomposition of a c class multiclass issue with $k_1 \dots\dots\dots\dots, k_c$ as the class labels generated a set of m binary classifiers represented by $f_1 \dots\dots\dots, f_c$ a binary classifier subdivides the input patterns into two complementary super class k_i^1 and k_i^{-1} grouping together one or more classes of multiclass problem.

Let $M = [b_{ij}]$ is a decomposition matrix of dimension $m \times c$, connecting class $k_i \dots\dots\dots\dots, k_c$ to the super classes k_i^1 and k_i^{-1}, where the definition of an element of matrix M is:

$$b_{ij} = \begin{Bmatrix} 1 \, if \; k_c \in k_i^1 \\ -1 \, if \; k_c \in k_i^{-1} \end{Bmatrix} \qquad (2)$$

Therefore, for M classes, a matrix $D \in \{\mp 1\}^{M \times C}$ is obtained.

2) Convolutional Neural Network (CNN)

Convolutional neural network is a specific type of multi-layer neural network, a simple neural network cannot learn complex features. In multiple applications [35, 36] such as image classification, object detection, and medical image analysis, CNNs have shown excellent performance.

In this part of the work, several convolutional neural networks, including GoogLeNet, ResNet18, ResNet50, and ResNet101, have been proposed on different CNNs.

As shown in Fig. 2, CNN is a multi-layer network structure made up of basically five layers: starting from the input layer, pooling layer, convolutional layer, fully connected layer, and finally output layer.

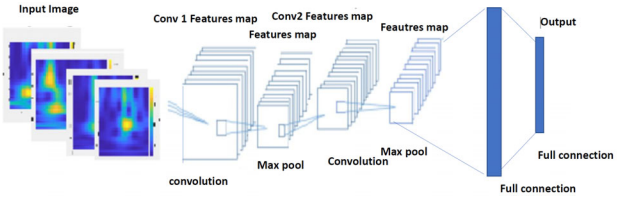


Fig. 2. The basic structure of CNN.

With CNN, we can transfer local features from inputs at higher layers to lower layers for greater complexity and functionalities. Therefore, we proposed a detailed description of these layers:

- Convolution layer: main layer of CNN, it does a linear procedure known as "convolution", that involves multiplying input-set weights. Two sets of information are combined by performing a mathematical operation between kernel Function and input image.
- Pooling layer: usually succeeding the convolutional layer, where squares of pixels (usually 2×2 or 3×3) are substituted with a single value, significantly decreasing and simplifying the size of images. Pooling can be performed using maximum, average, or sum pooling. Since maximum pooling makes it possible to identify more exact features, it was used in this study.
- Fully Connected (f_c) Layer known as a dense layer, which resembles a standard neural system.
- Last layer activation function: softmax.

Each task requires the selection of an appropriate activation function. The last fully connected layer's output real values are normalized to target class probabilities using the softmax activation function, which is employed in the multiclass classification task. All values in the function's output range from 0 to 1.

In this research, we looked into four different convolutional neural network techniques:

- **GoogLeNet:** GoogLeNet, also called Inception, is a deep convolutional neural network architecture design created By Google researchers [37], depicting 22 layers deep. GoogLeNet stands out for its in-depth architecture and efficient use of computing resources. GoogLeNet is based on building modules, with multiple parallel convolutional operations of different kernel sizes. These parallel operations capture information at different scales, allowing the network to efficiently learn hierarchical features. Our network has an input image size of 224 by 224, 144 layers, and 6.9 million trainable parameters. Starting with the input layer, convolution layer, ReLU, and Max pooling. This structure is repeated until it reaches the fully connected layer, and then the classifier layer registers the output classification result.

  The architecture addressed the challenge of vanishing gradients in very deep networks and achieved impressive performance in image classification tasks, particularly in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014.
- **ResNet:** ResNet (Residual Network) is a deep convolutional neural network architecture designed to overcome very deep neural networks. The key innovation of ResNet is exploring residual learning blocks (residual units or residual blocks). These blocks have shortcut connections that allow the model to get around one or more layers to learn residual mappings. The main advantage of residual connections is their ability to alleviate the vanishing gradient problem. In residual connections, further information can be directly propagated to further layers, which assists the network in learning complex mappings. ResNet architectures have been widely adopted in a variety of computer vision tasks, including image classification, segmentation, and recognition of objects, due to their ability to efficiently train and scale very deep networks.
- **ResNet50**: ResNet50 is a specific variant of Residual Network (ResNet) architecture, featured by its depth, using 50 layers and 25.5 million parameters. The key innovation of ResNet50 is the use of residual learning blocks. These blocks contain shortcut connections (skipped connections), that push the model to learn residual mappings. ResNet50 includes several building blocks including residual units, which are composed of convolutional layers with batch normalization and Rectified Linear Unit (ReLU) activations. Skipped connections in these units allow the direct flow of information from one layer to another, reducing degradation that can occur in extremely deep networks.
- **ResNet 18:** Composed of 18 layers and 11,511,784, total trainable parameters [36], 3×3 CONV layers with filters, and only two pooling layers used at the beginning and end of the network. Connections are found between every two CONV layers. ResNet18 uses shortcut connections to solve the disappearance problem [36].
- **ResNet101:** Presenting101 layers, about 44.6 million trainable parameters are. The network depth is 347 layers, Resnet-101 differs from other architectures by optimizing residuals between desired input and convolution properties. The desired functionality is achieved more easily and efficiently compared to other architectures [38].

  In ResNet architecture, missing learning information will be transferred to the next layer using the ResBlock layer. The residual values are passed to the following layer in the Resnet architecture by the Resblock layer. For each two-layer activation, jumps between weight layers, are corrected by the Relu activation code.

The nonlinear ReLU function is represented by two-layered residual block structures.

$$F = w_{2\sigma}(w_1 x) \tag{3}$$

The output result y is obtained by adding a second ReLU value.

$$y = f(x, (\{w_i\})) + x \quad (4)$$

Eq. (10), x denotes the input vector, while y denotes the output vector.

3) The proposed CNN-SVM hybrid architecture

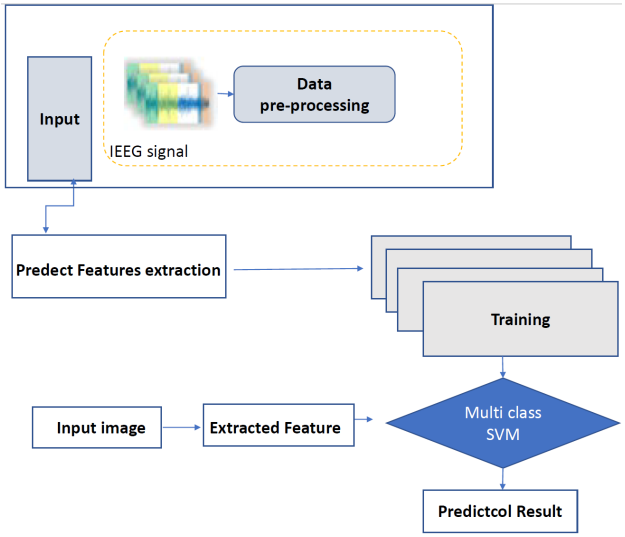


Fig. 3. CNN_SVM model architecture outlines two basic processes: CNN for feature extraction and SVM for classification.

In this study, we proposed several well-known pre-trained networks, including AlexNet, GoogLeNet, ResNet50, and VGG-19, which are trained on simulated and then real datasets.

The size of the input image (224×224×3 pixels) is defined by the first layer.

A pre-trained CNN is used for feature extraction, with a starting network layer capturing the main image features (edges, blobs). A convolutional layer contains network weight. Preparation of training and test datasets is carried out by dividing our dataset into 70% training data and 30% testing data. Both the training and testing datasets are processed by the CNN model. The features extracted from the training dataset are then used to train the SVM classifier, as illustrated in Fig. 3.

- **Classification Steps of HFO using hybrid model CNN_SVM**

Step 1: Convert 1D iEEG signal to 2D TF scalogram image.

Step 2: Exporting an image from one category folder.

Step 3: Load database: load images from the Image DataStore function that operates on image location to associate for each image category their labels. An Image DataStore allows us to store voluminous image data, it also divides data into 70% training data and 30% test data. The smallest number of images in each category is determined by each label. Then we tested pre-trained ResNet50, ResNet101, ResNet18, google NET, and VGG19 using ResNet50, ResNet101, ResNet18, GoogLeNet, and VGG19, functions respectively.

Step 4: Image pre-processing: CNN model processes both training and test Set. Image pre-processing for CNN according to used network used is carried out by resizing images according to the network (224 by 224).

Step 5: Extract features with CNN using the Activation function of each algorithm from ResNet50, ResNet18, ResNet101, VGG19, and GoogLeNet.

Step 6: The training of a multiclass SVM classifier is completed using CNN's features.

Step 7: The classifier is evaluated by taking image features from the test dataset. Subsequently, the classifier receives these features back to determine the accuracy of the trained classifier. Fig. 4 summarizes the classification steps of HFOs using the CNN_SVM models.

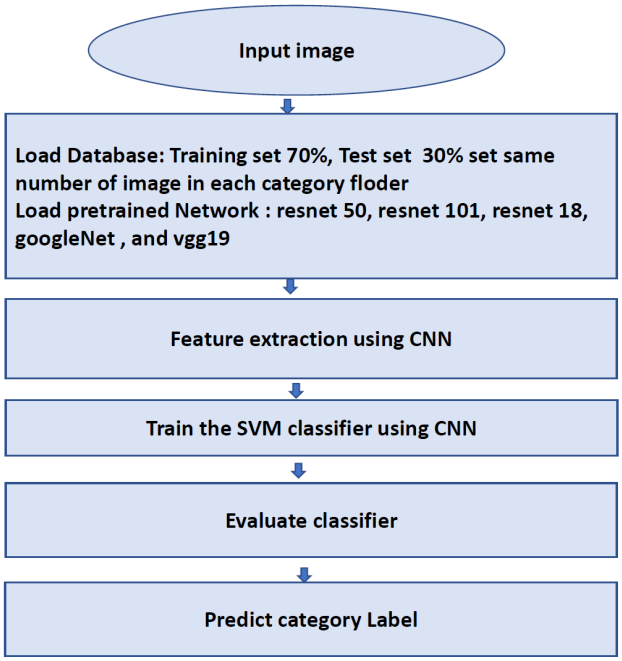


Fig. 4. Model classification steps CNN_SVM.

D. Metrics

In our study, we proposed to evaluate Classification Performance using: accuracy, precision, F1-Score, sensitivity, and specificity. Performances of classification are computed to define the proposed system's robustness.

We explored the confusion matrix to compare classification results, using:

Accuracy formulated in Eq. (5).

$$Acc = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

Precision formulated in Eq. (6).

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

F-Measure (F1-Score) presented in Eq. (7).

$$F1\ score = \frac{2\times TP}{(2\times TP+FP+FN)} \quad (7)$$

Sensitivity (SE) formulated in Eq. (8).

$$SE = \frac{TP}{TP+FN} \quad (8)$$

Specificity (SP) formulated in Eq. (9).

$$SP = 1 - \frac{FP}{FP+TN} \quad (9)$$

III. Experimental Results

Our study was performed on a PC equipped with a six-core Intel i7 processor using MATLAB (2023b). The server was equipped with an NVIDIA GEFORCE 920M with 6 GB of memory. In this section, we depicted the results of HFO classification using our models, SVM multiclass, multiple architecture CNN, and CNN-SVM. We conducted both methods on different datasets.

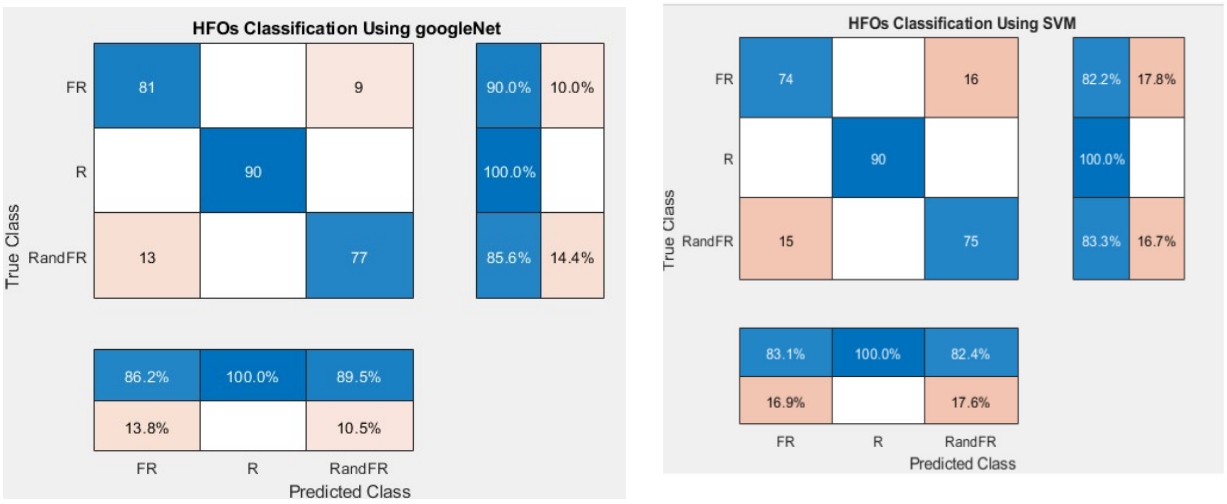
A. Results of SVM Multiclass

We evaluated SVM ECOC results on different datasets. First, we employed two real and simulated datasets. To illustrate the multiclass image classification, the experiment was carried out with three categories taken from a simulated dataset (1) and three categories taken from a real dataset (2). Each category for dataset (1) and dataset (2) is assigned an index ranging from 1 to 3. A multiclass SVM is a classifier that predicts category labels with its index. The five used performance measures are precision, F1-Score, specificity, sensitivity, and accuracy.

The confusion matrix is presented in Fig. 5 depicting performance measures for each category, respectively, for dataset (1) and dataset (2). The outcomes from the experiment of image classification using predicted image labels and indexes. Experiments were evaluated after 10-fold cross-validation. Table I presents an outline of results based on SVM test performance applied on 224×224 image size of datasets (1) and (2).

TABLE I. Precision, F1-Score, Specificity, Sensitivity, and Accuracy Macro AVG by SVM ECOC from Two Datasets

Model	Dataset	Precision %	F1-Score %	Specificity %	Sensitivity%	Accuracy %
SVM Multiclass	Simulated DATA (1)	98.19	98.15	99.07	98.14	98.14
	Real DATA (2)	88.52	88.51	94.25	88.51	88.51



(a) (b)

Fig. 5. Confusion matrix analyses based on SVM multiclass after 10-fold cross validation model obtained from (a) dataset (1), and (b) dataset (2).

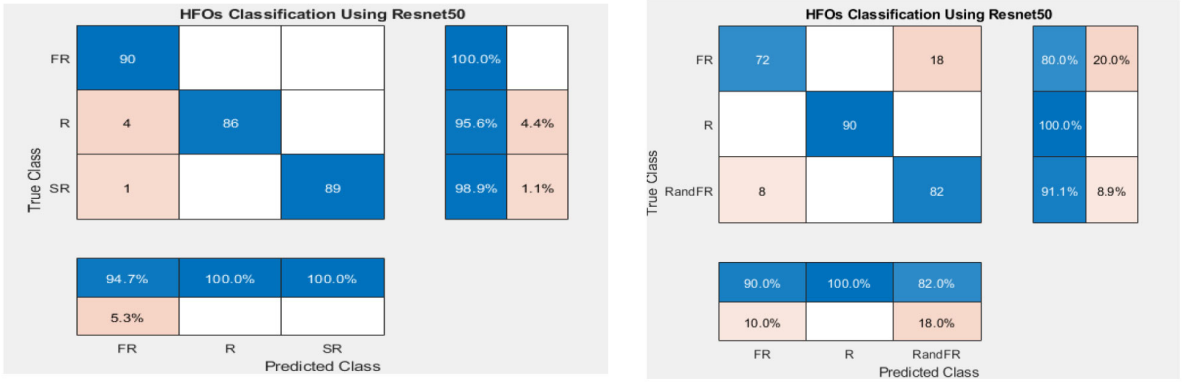
B. Results of GoogLeNet, ResNet50, ResNet101, and ResNet18

Table II presents a summary of results obtained from various test performances. Considering the same conditions of datasets (a 224×224 image). GoogLeNet attained the maximum accuracy at the same conditions of learning rate 0.001 and Adam optimization function.

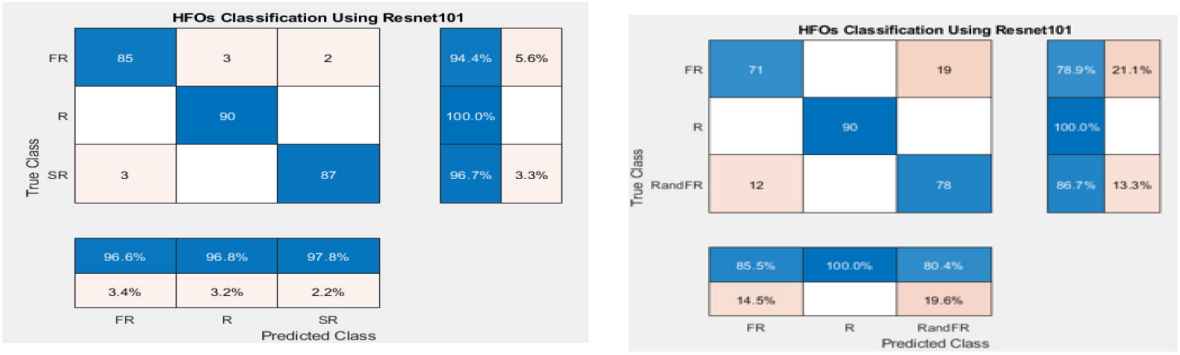
In Fig. 6, we presented the confusion matrix of the four investigated classifiers. It has information available on the overall number of segments that are correctly categorized and the total number that are misclassified. The first three rows of the confusion matrix table are related to the predicted category class (output class), and the true class (target class) is related to the first columns. Diagonal cells show performances of categories classification. Cells outside the diagonal refer to misclassified categories.

According to the confusion matrix in Fig. 6, the performance measures were evaluated for each category applied on both datasets by CNN models (GoogLeNet, ResNet50, ResNet18, and ResNet101). For simulated data, and the entire classes (FR, R, and SR), the obtained recognition accuracies are higher than 97.8% and 86.2% for real data. These results can be justified by the fact that real HFOs mainly overlap with other oscillatory activities that make the recognition a bit difficult compared to the simulated data.

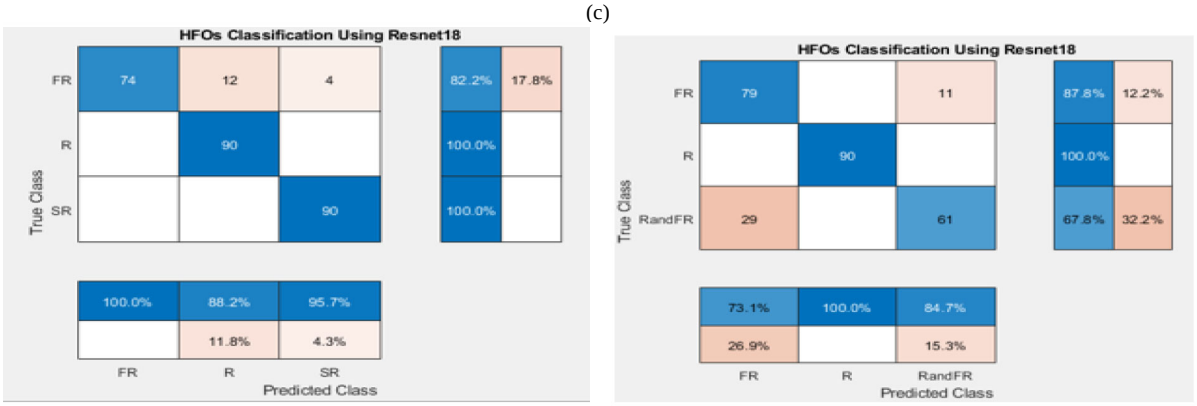
(a)



(b)



(c)



(d)

Fig.6. Confusion matrix and multigroup classification models of test analyses using the following models: (a) GoogLeNet; (b) ResNet50; (c) ResNet101; and (d) ResNet18, each sub figure presented a result above for simulated data and below for real data, respectively.

TABLE II. COMPARISON BETWEEN CLASSIFICATION PERFORMANCES IN TERMS OF SENSITIVITY, SPECIFICITY, OVERALL ACCURACY, AND PRECISION OF FOUR CNN MODELS USING TWO DATASETS

Model	Dataset	Precision %	F1-Score %	Specificity %	Sensitivity %	Accuracy %
GoogleNet	**Simulated DATA (1)**	**98.55**	**98.51**	**99.25**	**98.51**	**98.52**
	Real DATA (2)	**91.90**	**91.84**	**95.92**	**91.85**	**91.85**
ResNet 50	**Simulated DATA (1)**	98.24	98.15	99.07	98.14	98.**15**
	Real DATA (2)	90.66	90.34	95.18	90.37	90.37
ResNet 101	**Simulated DATA (1)**	97.03	97.02	98.51	97.03	97.04
	Real DATA (2)	88.65	88.50	94.25	88.51	88.52
ResNet 18	**Simulated DATA (1)**	94.66	93.94	97.03	94.07	96.04
	Real DATA (2)	85.95	85.03	92.59	85.19	87.**04**

Note: bold values represent the highest performance.

Table II gathers the classification performances in terms of sensitivity, specificity, overall accuracy, F1-Score, and precision. Maximum accuracy for datasets (1) and (2) is around 98.52%, and 91.85% respectively with maximum values of F1-Score (98.51% and 91.84%). The high value of the GoogLeNet model also indicates that the trained classifier performs significantly better than other tested classifiers.

C. *The Results Obtained by CNN-SVM Hybrid Models*

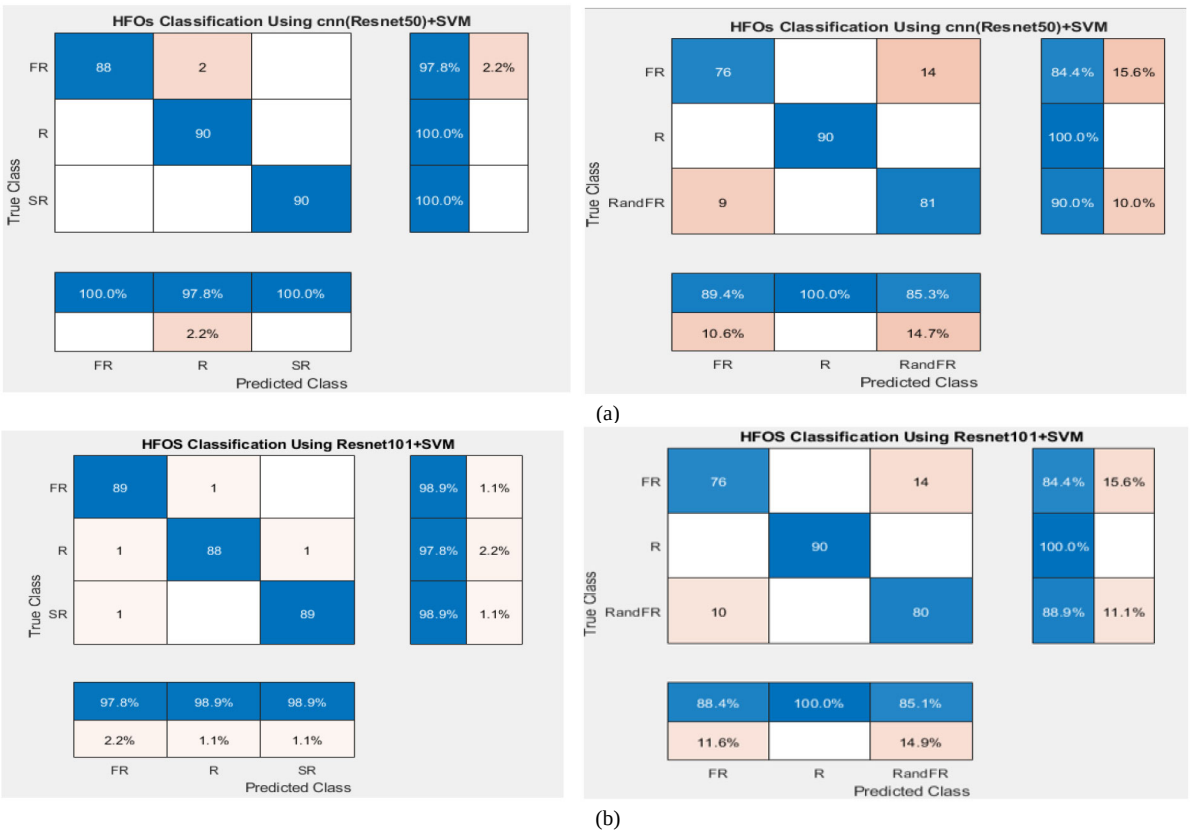
The model combines two methods for supervised classification, CNN and Support Vector Machine (SVM). The models used batch size 128 and optimized loss function using the Adam optimizer.

To take temporal information into account, five hybrid models based on CNN and SVM classifier models were built, and their classification performances were tested both on simulated data and real data.

The confusion matrices of the CNN_SVM model are depicted in Fig. 7, where three classes for data set (1) representing Fast Ripple (FR), Ripple (R), and Spike Ripple (SR) respectively, and for data set (2) representing Fast ripple (FR), Ripple (R), and Ripple with Fast Ripple (R and FR), respectively. However, the HFO detector was compared to machine learning, including deep learning models, respectively.

The confusion matrix of recognition results in Fig. 7 presented the classification result of five proposed hybrid models: ResNet50_SVM, ResNet101_SVM, ResNet 18-SVM, VGG19_SVM, and GoogLeNet_SVM.

Table III presents the classification performance achieved by CNN that has been trained with SVM network architecture. Different hybrid models classify our dataset within convenient performances, where GoogLeNet combined with SVM depicts the highest one.



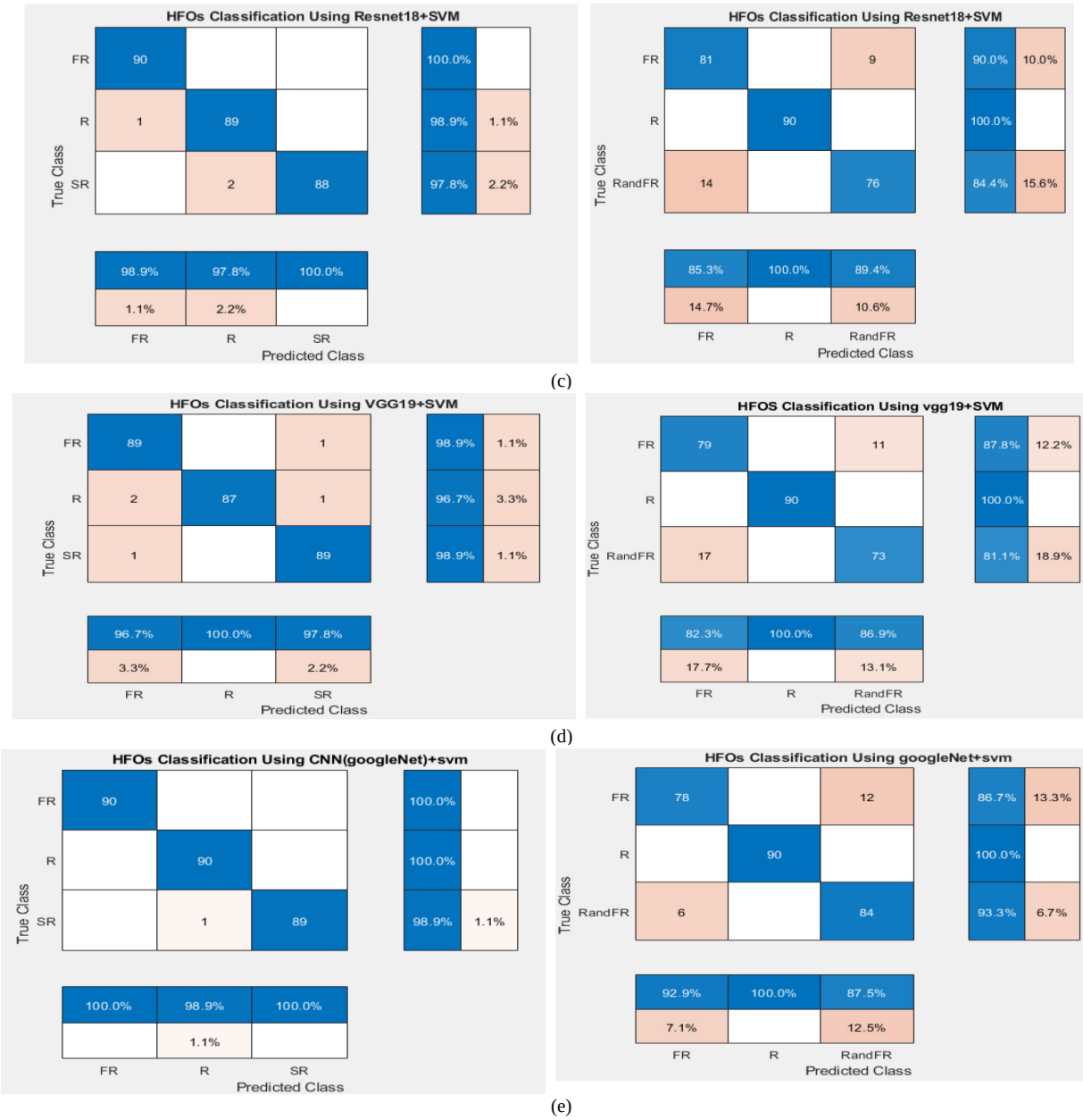


Fig.7. Confusion matrix of five CNN-SVM models classification test effects: (a) ResNet50+svm for dataset (1), and dataset (2) respectively, (b) ResNet 101+SVM for dataset (1), and dataset (2) respectively, (c) ResNet 18+SVM for dataset (1), and dataset (2) respectively, (d) VGG19+SVM for dataset (1), and dataset (2) respectively, and (e) GoogLeNet for dataset (1), and dataset (2) respectively.

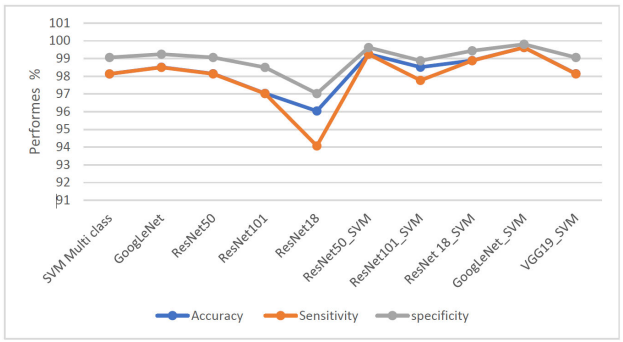
TABLE III. RESULT OF CLASSIFICATION OF DIFFERENT HYBRID MODEL CNN_SVM FOR TWO DATASETS

Model	Dataset	Precision %	F1-Score%	Specificity %	Sensitivity%	Accuracy %
Resnet50_SVM	**Simulated DATA [1]**	99.27	99.25	99.63	99.25	99.26
	Real DATA [2]	91.55	91.47	95.74	91.48	91.48
ResNet 18_SVM	**Simulated DATA [1]**	98.90	98.99	99.44	98.88	98.89
	Real DATA [2]	91.55	91.47	95.74	91.48	91.48
ResNet 101_SVM	**Simulated DATA [1]**	97.87	97.77	98.88	97.77	98.52
	Real DATA [2]	91.12	91.11	95.55	91.11	91.11
GoogleNet _SVM	**Simulated DATA [1]**	**99.64**	**99.63**	**99.81**	**99.63**	**99.63**
	Real DATA [2]	**94.08**	**94.07**	**96.66**	**94.07**	**94.07**
VGG19_SVM	**Simulated DATA [1]**	98.18	98.15	99.07	98.14	98.15
	Real DATA [2]	89.73	89.61	94.81	89.63	89.63

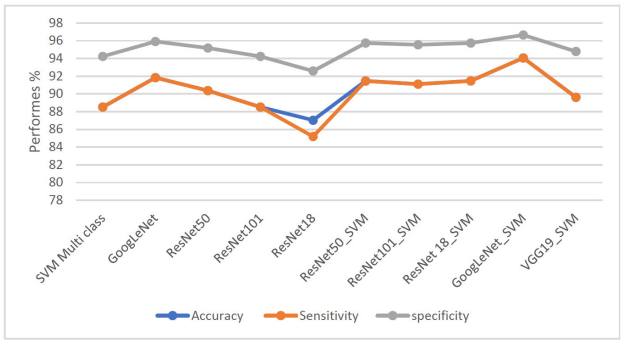
Note: bold values represent the highest performance

Fig. 8 shows the overall accuracy, sensitivity, and specificity of the tested algorithms. A higher classification performance for the proposed GoogLeNet _SVM is obtained (an average: 99.63% of sensitivity, 99.63% of accuracy and 99.81% of specificity for dataset (1), and 94.07%, 94.07%, 96.66% for dataset (2), respectively).

However, the lower values are respectively equal to 96.07%, 94.07%, and 97.03% for dataset (1), 87.04%, 85.15%, and 92.59% for dataset (2) which are obtained using ResNet18. Similarly, methods SVM, ResNet18, and ResNet18_SVM show lower multiclassification performance than the suggested method.



(a)



(b)

Fig. 8. Comparison graph of accuracy, sensitivity, and specificity between various test approaches on (a) simulated data above, and (b) real data below.

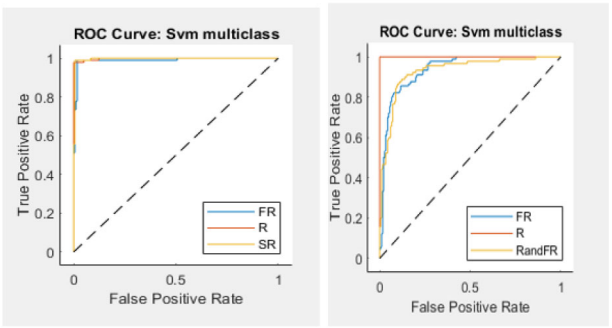
Table IV summarizes the best performances of ML and DL methods: SVM model, GoogLeNet, and hybrid GoogLeNet_ SVM.

TABLE IV. PRECISION, F1-SCORE, SPECIFICITY, SENSITIVITY, AND ACCURACY OF THE BEST RESULTS ACHIEVED FOR THREE MODELS

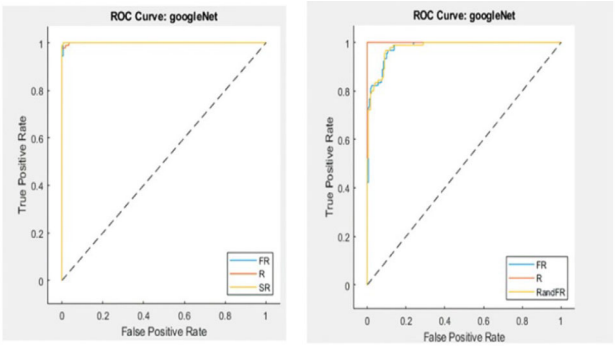
Model	Dataset	Precision %	F1-Score %	Specificity %	Sensitivity %	Accuracy %
SVM	**Simulated DATA (1)**	98.19	98.15	99.07	98.14	98.14
	Real DATA (2)	88.52	88.51	94.25	88.51	88.51
GoogLeNet	**Simulated DATA (1)**	98.55	98.51	99.25	98.51	98.52
	Real DATA (2)	91.90	91.84	95.92	91.85	91.85
GoogLeNet_SVM	**Simulated DATA (1)**	**99.64**	**99.63**	**99.81**	**99.63**	**99.63**
	Real DATA (2)	**94.08**	**94.07**	**96.66**	**94.07**	**94.07**

Note: bold values represent the highest performance.

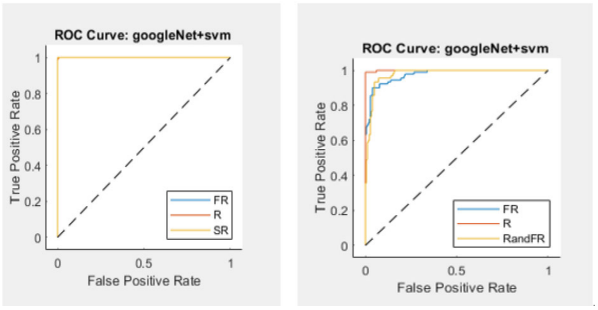
Fig. 9, depicts the ROC curve. The larger value under the ROC curve indicates a better performance of the classifier). As can be seen in Fig. 9(c), the highest AUC is obtained by the GoogleNet-SVM classifier on two sets.



(a)



(b)



(c)

Fig. 9. ROC Curve of the best results achieved for three test models; (a) SVM multiclass, (b) GoogleNet, and (c) GoogleNet_SVM presented on the left for dataset (1), and on the right for dataset (2).

IV. DISCUSSION

The main objective of this study was to develop and evaluate methods for classifying high-frequency oscillation (HFO) signals into distinct categories. By transforming HFO signals into images using Continuous Wavelet Transform (CWT), a substantial dataset consisting of 3000 images was generated from simulated data and 3220 images from real data. CNN and hybrid CNN_SVM models have separated data into training and testing sets. Then, data was classified into three classes: Fast Ripple (FR), Ripple (R), and Spike Ripple (SR) for dataset (1), R, FR, and FR and R for dataset (2), using mean accuracy of training and test networks over 100 training iterations. Important findings in our research proved that combined GoogLeNet-SVM based on TF images constitutes a potentially useful for multi-

classification of HFOs and spikes. Deep GoogLeNet features of iEEG time-frequency maps, combined with an SVM classifier depicted a competitive performance in terms of precision, specificity, sensitivity, F1-Score, and overall accuracy. These convenient results can be attributed to several crucial aspects.

First, ECOC SVM networks are efficient for large-scale classification, but still couldn't reach neural network performances. CNNs have been proven robust in recognizing detailed, multi-scale features from iEEG data, including advanced architectures such as ResNet and GoogLeNet. However, ResNet incorporates residual connections, which enables the creation of very deep networks while avoiding problems such as the disappearance of gradients. This allows networks such as ResNet18, ResNet50, and ResNet101 to maintain high performance even at high depths. While The GoogLeNet Inception module can handle functions at different scales simultaneously.

According to our studies, hybrid models, including GoogLeNet-SVM, were able to outperform traditional CNN architectures as well as hybrid methods such as ResNet-SVM and VGG19-SVM. ResNet-SVM and VGG19-SVM showed notable improvements over non-hybrid counterparts, however, their performance did not reach GoogLeNet-SVM conducting in classification. in fact, ResNet 50-SVM could achieve 99.26% accuracy on simulated data and 91.48% on real data, while VGG19-SVM achieved 98.15% on simulated data and 89.63% on real data. These findings suggest that combining CNNs with SVMs generally enhances performances compared to CNNs. Typically, The GoogLeNet-SVM model has the peculiarity of combining the extraction of advanced features by GoogLeNet with the classification power of SVM. GoogLeNet is proven efficient in generating rich and relevant features, while its efficient design minimizes overfitting, especially with limited data sets. On the other hand, SVM excels in maximizing the margin between classes in a large functionality space, enabling precise classification by effectively managing complex, nonlinear relationships in the data. The combination of GoogLeNet's advanced capability extraction and SVM's robust classification capabilities creates powerful synergy, enabling optimal class separation. This is advantageous for the classification of High-Frequency Oscillations (HFO), where subtle and dispersed characteristics are essential.

However, some limitations of this study must be taken into account. Although its effectiveness is undeniable, converting HFO signals into images increases data dimensionality, and hence classification complexity too. Although this problem has been reduced by our deep learning method that automatically selects the most relevant features, future research could explore ways of reducing dimensionality or more advanced deep learning architectures to further improve performance. Moreover, our data set size, including real data, has been able to restrict the generalization of our results. Even if the model has shown strong performance, it would be advantageous to have a wider and more varied data set to validate later and ensure that the model performance is consistent for different types of iEEG signals. It is recommended that future research should be focused on expanding the dataset, exploring other hybrid architectures, and analyzing the model's applicability to other types of signals to further improve its clinical usefulness.

V. Conclusion

Our study focuses on the recognition of epileptic biomarkers HFO considered as a hallmark of pharmaco-resistant epilepsy. We investigated several applications of multiclass SVM, CNN, and CNN-SVM models. CNN is reviewed as a key element in our classification system explored to distinguish HFO classes: three classes composed of HFO events (R, FR) and a mixture of spikes with Fast events (SR) for dataset (1), and three classes composed of R, FR, and Rand FR for the dataset (2).

Our proposed approach's effectiveness in terms of precision, sensitivity, specificity, accuracy, and F1- score has been determined by numerical tests with both real and simulated iEEG data**.** We address this issue in our proposed approach by introducing three global models: multiclass SVM, model CNN, and CNN-SVM. Our results obtained from these models demonstrate a particularly high level of accuracy; GoogLeNet-SVM achieves the highest accuracy rate 99.63% and 94.07% for dataset (1) and dataset (2), respectively, SVM multiclass achieves 98.14% and 88.51% for (1) and (2), respectively, and GoogLeNet achieves 98.52% and 91.85% for (1) and (2), respectively.

These results may open up new avenues, especially for the automatic detection of HFO signals and epileptic seizure buildup using hybrid algorithms.

Conflict of interest

The authors declare no conflict of interest.

Authors Contributions

Z.S. performed the software and data analysis; A.H. conducted statistical analysis, and drafted sections of the manuscript; R.M. helped interpret the results; N.J. Provided critical feedback on the experimental design and contributed to the final editing of the manuscript. All authors had approved the final version.

Acknowledgment

The authors are grateful to the investigators who shared iEEG data.

References

[1] A. J. Durnford, W. Rodgers, and F. J. Kirkham *et al.*, "Very good inter-rater reliability of engel and ILAE epilepsy surgery outcome classifications in a series of 76 patients," *Seizure*, vol. 20, no. 10, pp. 809–812, Dec. 2011. doi: 10.1016/j.seizure.2011.08.004

[2] M. Cossu, F. Cardinale, and L. Castana *et al.*, "Stereoelectroencephalography in the presurgical evaluation of focal epilepsy: A retrospective analysis of 215 procedures," *Neurosurgery*, vol. 57, no. 4, pp. 706–718, Oct. 2005. doi: 10.1227/01.NEU.0000176656.33523.1e

[3] M. Cossu, G. L. Russo, and S. Francione *et al.*, "Epilepsy surgery in children: Results and predictors of outcome on seizures,"

Epilepsia, vol. 49, no. 1, pp. 65–72, Jan. 2008. doi: 10.1111/j.1528-1167.2007.01207.x

[4] A. Hadriche, I. Abehy, and A. I. necibi *et al.*, "Assessment of effective network connectivity among MEG none contaminated epileptic transitory events," *Computational and Mathematical Methods in Medicine*, vol. 2021, no. 1, 6406362, 2021. doi: 10.1155/2021/6406362

[5] N. Jmail, A. Hadriche, and I. Abehy *et al.*, "A comparison of inverse problem methods for source localization of epileptic meg spikes," in *Proc. 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2019, pp. 867–870. doi: 10.1109/BIBE.2019.00161

[6] V. N. Vakharia, J. S. Duncan, and J. Witt *et al.*, "Getting the best outcomes from epilepsy surgery," *Ann. Neurol.*, vol. 83, pp. 676–690, 2018. doi: 10.1002/ana.25205

[7] F. Rosenow and H. Lüders, "Presurgical evaluation of epilepsy," *Brain*, vol. 124, no. 9, pp. 1683–1700, 2001. doi: 10.1093/brain/124.9.1683

[8] Z. Sadek, A. Hadriche, and N. Jmail, "Clustering of High Frequency Oscillations (HFO) in epilepsy using pretrained neural networks," in *Proc. International Conference on Intelligent Systems Design and Applications*, 2023, pp. 100–107. doi: 10.1007/978-3-031-35501-1_10

[9] T. Guesmi, A. Hadriche, N. Jmail, and B. A. Chokri," Evaluation of stationary wavelet transforms in reconstruction of pure High Frequency Oscillations (HFOs)," in *Proc. International Conference on Smart Homes and Health Telematics*, 2020.

[10] J. Jacobs, P. LeVan, and R. Chander *et al.*, "Interictal high-frequency oscillations (80–500 Hz) are an indicator of seizure onset areas independent of spikes in the human epileptic brain," *Epilepsia*, vol. 49, pp. 1893–1907, 2008. doi: 10.1111/j.1528-1167.2008.01656.x

[11] G. A. Worrell, L. Parish, and S. D. Cranstoun *et al.*, "High-frequency oscillations and seizure generation in neocortical epilepsy," *Brain*, vol. 127, no. 7, pp. 1496–1506, 2004. doi: 10.1093/brain/awh149

[12] E. Urrestarazu, R. Chander, and F. Dubeau *et al.*, "Interictal high-frequency oscillations (100–500 Hz) in the intracerebral EEG of epileptic patients," *Brain*, vol. 130, no. 9, pp. 2354–2366, 2007. doi: 10.1093/brain/awm149

[13] M. Zijlmans, G. A. Worrell, and M. Dümpelmann *et al.*, "How to record high-frequency oscillations in epilepsy: A practical guideline," *Epilepsia*, vol. 58, pp. 1305–1315, 2017. doi:10.1111/epi.13814

[14] A. Hadriche and N. Jmail, "A build up of seizure prediction and detection Software," *J. Clin. Images Med. Case Reports*, vol. 2, no. 2, pp. 1–2, 2021. doi: 10.26420/austinneurosurgopenaccess. 2021.1064

[15] A. M. Spring, D. J. Pittman, and Y. Aghakhani *et al.*, "Generalizability of high frequency oscillation evaluations in the ripple band," *Frontiers in Neurology*, vol. 9, p. 510, 2018. doi: 10.3389/fneur.2018.00510

[16] J. Engel Jr, A. Bragin, and R. Staba *et al.*," High-frequency oscillations: What is normal and what is not?" *Epilepsia*, vol. 50, pp. 598–604, 2009. doi: 10.1111/j.1528-1167.2008.01917.x

[17] S. Chaibi, Z. Sakka, and T. Lajnef *et al.*, "Automated detection and classification of High Frequency Oscillations (HFOs) in human intracereberal EEG," *Biomed. Signal Process. Control*, vol. 8, pp. 927–934, 2013. https://doi.org/10.1016/j.bspc.2013.08.009

[18] J. Jacobs, R. Staba, E. Asano, and H. Otsubo *et al.*, "High-frequency oscillations (HFOs) in clinical epilepsy," *Progress in Neurobiology*, vol. 98, 302315, 2012. doi:10.1016/j.pneurobio.2012.03.001

[19] T. Akiyama, B. McCoy, C. Y. Go, and A. Ochi *et al.*, "Focal resection of fast ripples on extraoperative intracranial EEG improves seizure outcome in pediatric epilepsy," *Epilepsia*, vol. 52, pp. 1802–1811, 2011. doi: 10.1111/j.1528-1167.2011.03199.x

[20] N. Sciaraffa, M. A. Klados, and G. Borghini *et al.*, "Double-step machine learning based procedure for HFOs detection and classification," *Brain Sciences*, vol. 10, no. 4, p. 220, 2020. doi: 10.3390/brainsci10040220

[21] J. A. Blanco, M. Stead, A. Krieger, and W. Stacey *et al.*, "Data mining neocortical high-frequency oscillations in epilepsy and controls," *Brain*, vol. 134, no. 10, pp. 2948–2959, 2011. doi: 10.1093/brain/awr212

[22] H. Firpi, O. Smart, G. Worrell, and E. Marsh *et al.*, "High-frequency oscillations detected in epileptic networks using swarmed neural-network features," *Annals of Biomedical Engineering*, vol. 35, no. 9, pp. 1573–84, September 2007. doi: 10.1007/s10439-007-9333-7

[23] M. Dümpelmann, J. Jacobs, K. Kerber, and A. Schulze-Bonhage "Automatic 80–250 Hz 'ripple' high frequency oscillation detection in invasive subdural grid and strip recordings in epilepsy by a radial basis function neural network," *Clin. Neurophysiol.*, vol. 123, pp. 1721–1731, 2012. doi: 10.1016/j.clinph.2012.02.072

[24] S. Chaibi, T. Lajnef, and M. Samet *et al*., "Detection of High Frequency Oscillations (HFOs), in the 80–500 Hz range in epilepsy recordings using decision tree analysis," in *Proc. Int. Image Process, Appl. Syst. Conf. IPAS*, 2014. doi: 10.1109/IPAS.2014.7043321

[25] N. Jrad, A. Kachenoura, and I. Merlet *et al.*, "Classification of high frequency oscillations in epileptic intracerebral EEG," in *Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy, 2015, pp. 5775-74. doi: 10.1109/EMBC. 2015.7318427

[26] F. krikid, A. Karfoul, and S. Chaibi *et al*.," Multi-classification of high-frequency oscillations in intracranial EEG signals based on CNN and data augmentation," *Signal Image and Video Processing*, 2023. doi:10.1007/s11760-023-02808-4

[27] N. Jmail, R. Jarray, A. Hadrich, T. Frikha, and C. Benar, "Separation between spikes and oscillation by stationary wavelet transform implemented on an embedded architecture," *Journal of the Neurological Sciences*, vol. 381, p. 542, 2017.

[28] A. Hadriche, N. Jmail, and J.-L. Blanc *et al.*, "Using centrality measures to extract core pattern of brain dynamics during the resting state," *Computer Methods and Programs in Biomedicine*, vol. 179, 104985, 2019. http://dx.doi.org/10.6080/K06Q1VD5

[29] T. Akiyama, B. McCoy, and C. Y. Go *et al*.," Focal resection of fast ripples on extraoperative intracranial EEG improves seizure outcome in pediatric epilepsy," *Epilepsia*, vol. 52, pp. 1802–1811, 2011. doi: 10.1111/j.1528-1167.2011.03199.x

[30] C. W. Hsu and C. J. Li "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415–425, Mar. 2002. doi: 10.1109/72.991427

[31] W. W. Peterson and E. J. Weldon Jr., *Error-Correcting Codes*, MIT Press, Cambridge USA, 1972.

[32] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, 1995. doi: https://doi.org/10.1613/jair.105

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process Syst*., vol. 25, 2012. https://doi.org/10.1145/306538

[34] O. Russakovsky, J. Deng, H. Su, and J. Krause *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015. doi: 10.1007/s11263-015-0816-y

[35] J. Zhu and J. Song, "An intelligent classification model for surface defects on cement concrete bridges," *Appl. Sci.*, vol. 10, no. 3, p. 972, 2020. doi: 10.3390/app10030972

[36] C. Szegedy, W. Liu, and Y. Jia *et al*., "Going deeper with convolutions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[37] X. Zhang and D. Zhai *et al.*, "Image inpainting based on deep learning: A review," *Information Fusion*, 2023. https://doi.org/10.1016/j.inffus.2022.08.033

[38] K. He., X. Zhang, S. Ren, and J. Sun, " Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90