

# Atrial Fibrillation Classification Using a Deep Spectral Autoencoder

Enrique Quezada-Próspero<sup>1</sup>, Dante Mújica-Vargas<sup>1,\*</sup>, Luis A. Cruz-Próspero<sup>1</sup>,  
Christian García-Aquino<sup>2</sup>, Ángel A. Rendón-Castro<sup>1</sup>

<sup>1</sup> Centro Nacional de Investigación y Desarrollo Tecnológico,  
Computer Sciences,  
Mexico

<sup>2</sup> Universidad Politécnica de Tapachula,  
Mexico

{m23ce074, dante.mv}@cenidet.tecnm.mx,  
{lhcp, angel.rendon18ce}@cenidet.edu.mx,  
cj.garcia51@gmail.com

**Abstract.** This study introduces a novel approach for atrial fibrillation classification using a deep autoencoder stacked with a fully connected softmax layer. The model is trained with spectral features extracted from ECG signals through spectral and signal analysis. The primary goal is to enhance existing algorithms in the state-of-the-art by delivering superior results with reduced computational cost. The PhysioNet Challenge 2017 database was used, which contains normal and atrial fibrillation rhythms. The signals were normalized before spectral feature extraction. These features were used to train the autoencoder, which performed additional feature extraction and dimensionality reduction. The resulting features were then used to train the fully connected layer responsible for classification. Performance was evaluated through quality metrics mentioned in the state-of-the-art using cross-validation to ensure robustness. The best median results obtained were: 99.7% Accuracy, 99.8% Precision, 99.5% Recall, 99.8% Specificity, 99.7% F1-score, 99.3% Matthews Correlation, and 99.3% Kappa.

**Keywords.** Autoencoder, spectral features, ECG signals, signal analysis, feature extraction.

## 1 Introduction

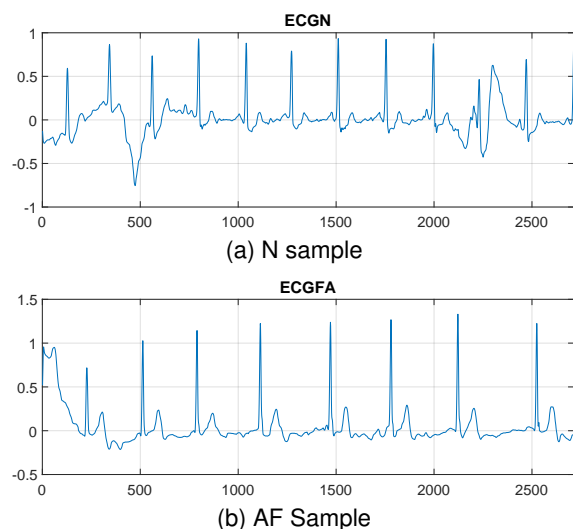
Cardiac arrhythmias (CAs) are irregularities in the heart's contraction and relaxation cycles. Among

the most common types are Atrial Fibrillation (AF), Ventricular Fibrillation (VF), Atrial Flutter (AFL), Paroxysmal Supraventricular Tachycardias (PSVT), and Atrioventricular Block (AB). According to the World Health Organization (WHO), CAs are a leading cause of death worldwide, with a prevalence affecting approximately 3% of the global population [13].

The automatic classification of cardiac arrhythmias has garnered significant interest due to the severity of this condition. In the field of machine learning, detection techniques have been developed to support specialists.

These techniques are based on convolutional neural networks (CNN) [15], recurrent neural networks (RNN) [16], and support vector machines (SVM) [8], among others. Hybrid architectures combining these algorithms have also been employed to enhance model performance [1].

However, these models require large amounts of data and numerous hidden layers to extract and analyze features, leading to significant processing time and computational cost. This issue becomes problematic for real-world applications that necessitate dedicated components like GPUs to achieve satisfactory results [11]. Training these models is time-consuming and computationally



**Fig. 1.** Samples of the physionet CinC 2017 database

expensive, limiting their implementation in embedded systems.

Additionally, preprocessing of the signals is crucial for training these models, requiring the transformation of raw signals into a domain where more significant features can be extracted and analyzed [12]. State-of-the-art preprocessing techniques include the application of signal filters to emphasize specific features and remove noise [2], time-frequency feature extraction to compress signals while preserving important features [16], and dimensionality reduction techniques like principal components analysis (PCA) [9].

The choice of technique varies depending on the approach. In this study, ECG signals were transformed into the spectral domain, as spectral representations provide more significant information than other domains, thus improving model processing effectiveness [17].

This study presents an architecture based on a deep autoencoder trained with normalized and preprocessed ECG signals using different spectral representations. The proposed architecture offers lower computational cost and better results than state-of-the-art architectures due to:

1. Fewer hidden layers, resulting in reduced processing time.
2. No need for large amounts of data.

3. Compatibility with less powerful hardware.
4. Applicability to other types of signals.

This paper is organized as follows: Section 2 explains preprocessing techniques and provides a brief description of the autoencoder architecture. Section 3 describes the proposed method and hyperparameter tuning. Section 4 presents experimental results and a comparative analysis with state-of-the-art techniques. Finally, Section 5 provides conclusions and suggestions for future work.

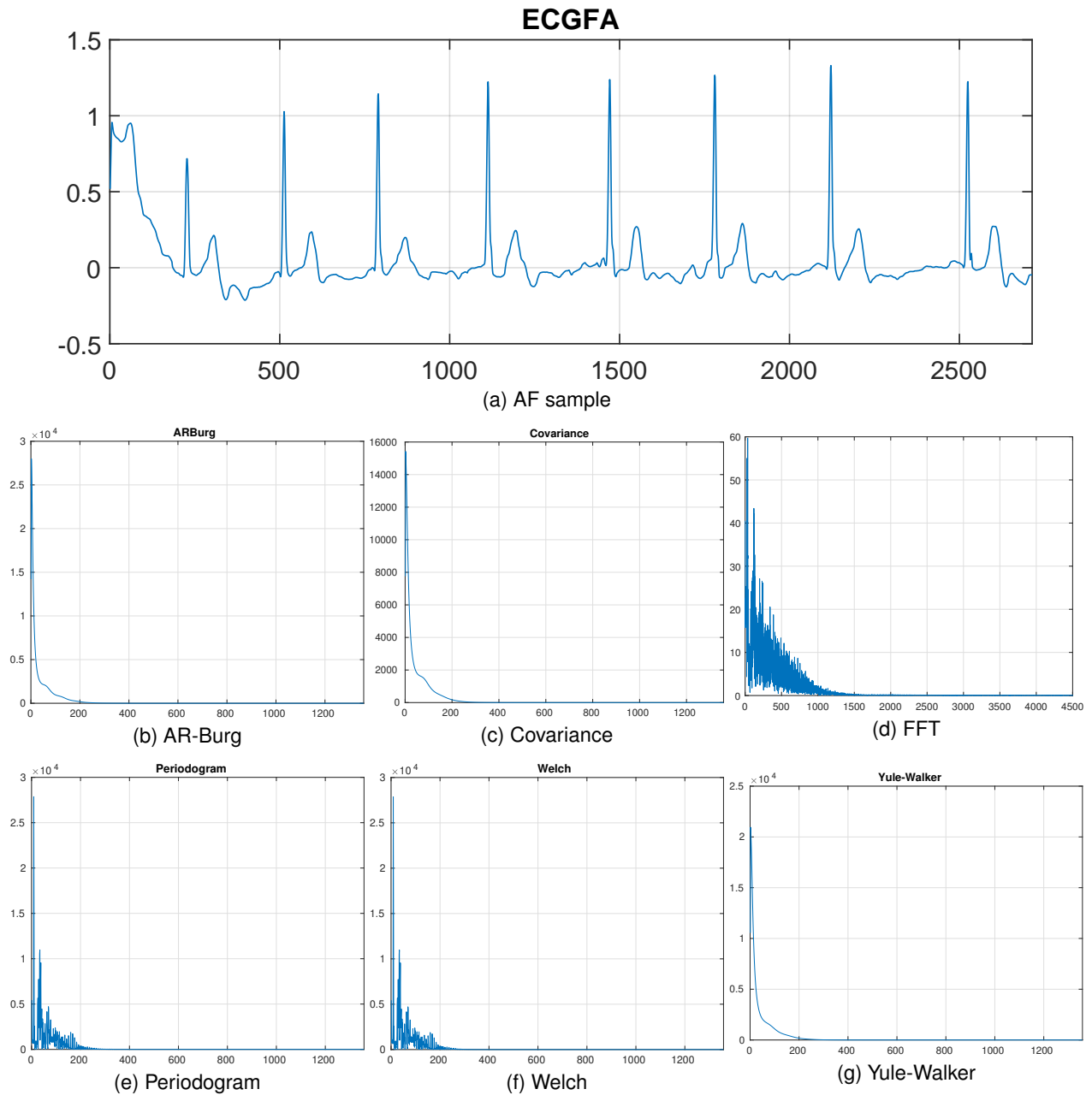
## 2 Background

This section provides a general overview of the fundamental techniques, data, and concepts used in the analysis and processing of ECG signals. It covers the database used in this work and various signal analysis methods, including the application of different spectral estimation techniques. Additionally, a brief description of the model used for ECG classification is provided.

### 2.1 Data

To evaluate the proposed model, the PhysioNet CinC 2017 database was used, containing ECG lead II samples of normal rhythms (N) and atrial fibrillation (AF) recorded at 300 Hz. The database includes 5050 instances for class N and 738 instances for class AF, with samples of varying lengths. To avoid class imbalance and preserve the maximum number of instances, the length of all signals was reduced to 2714 samples, which is the minimum length in the dataset.

The N class was reduced to 738 instances to match the AF class, instead of using any data augmentation techniques [5], thereby preserving the signals from the original database. This preprocessing step resulted in 738 instances per class, each with a length of 2714 samples. Figure 1 shows a sample of Normal Rhythm (N) and Atrial Fibrillation (AF). Note that the N class has an amplitude up to 1, whereas class AF exceeds this value. These signals can be analyzed and processed using signal analysis techniques for better representation.



**Fig. 2.** Spectral representation for an AF signal

## 2.2 Signal Analysis

Signals are entities that convey information and can propagate through a medium, whether material or vacuum.

The objective of signal analysis is to characterize a signal and interpret data in different domains such as time and frequency. Depending on the purpose and properties of the signal, one domain may be preferable over another.

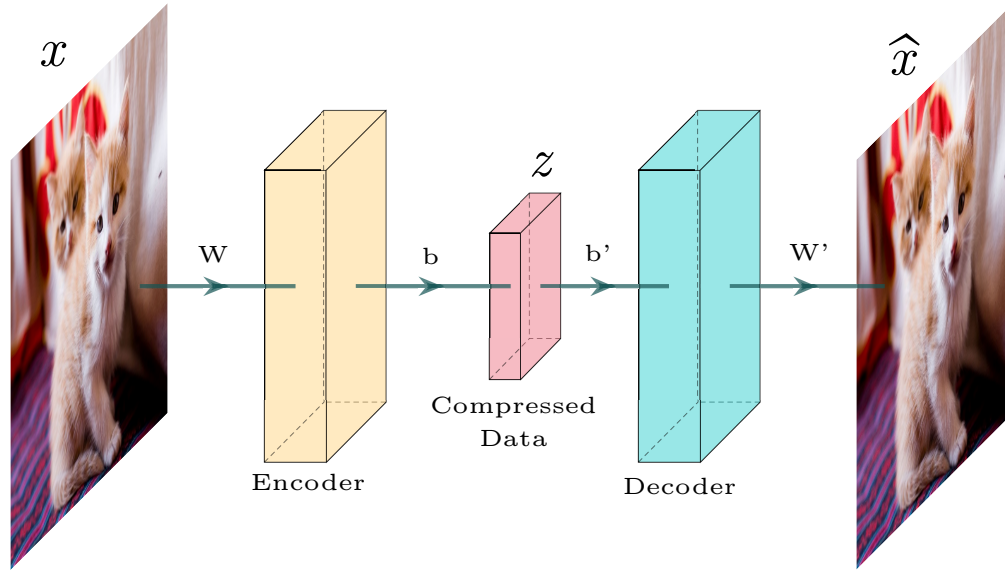


Fig. 3. Encoding and decoding process

Among the methods used in signal analysis are time-frequency methods, discrete wavelet transform, and spectral estimation. The spectral domain, in particular, offers an excellent representation of signals, providing more discriminating features [17].

### 2.3 Spectral Analysis

Spectral analysis is a signal processing method aimed at representing the dynamic patterns of signals in the frequency domain. It offers a complementary perspective to the time domain, often providing more useful information for the characterization of biomedical signals [14]. This study utilizes spectral representations that are prevalent in the literature, including:

#### 2.3.1 AR Burg Spectrum

This method is based on forward and backward prediction errors and on direct estimation of the reflection coefficients [14]. Assuming we have data measurements  $\{y(t)\}$  for  $t = 1, 2, \dots, N$ , the

forward and backward prediction errors for a  $o_{th}$  order model are:

$$\hat{\alpha}_{f,o}(n) = x(n) + \sum_{l=1}^o \hat{\xi}_{o,l} \cdot x(n-l), \quad (1)$$

$$\forall n = o+1, \dots, N,$$

$$\hat{\alpha}_{b,o}(n) = x(n-o) + \sum_{l=1}^o \hat{\xi}_{o,l} \cdot x(n-o+l), \quad (2)$$

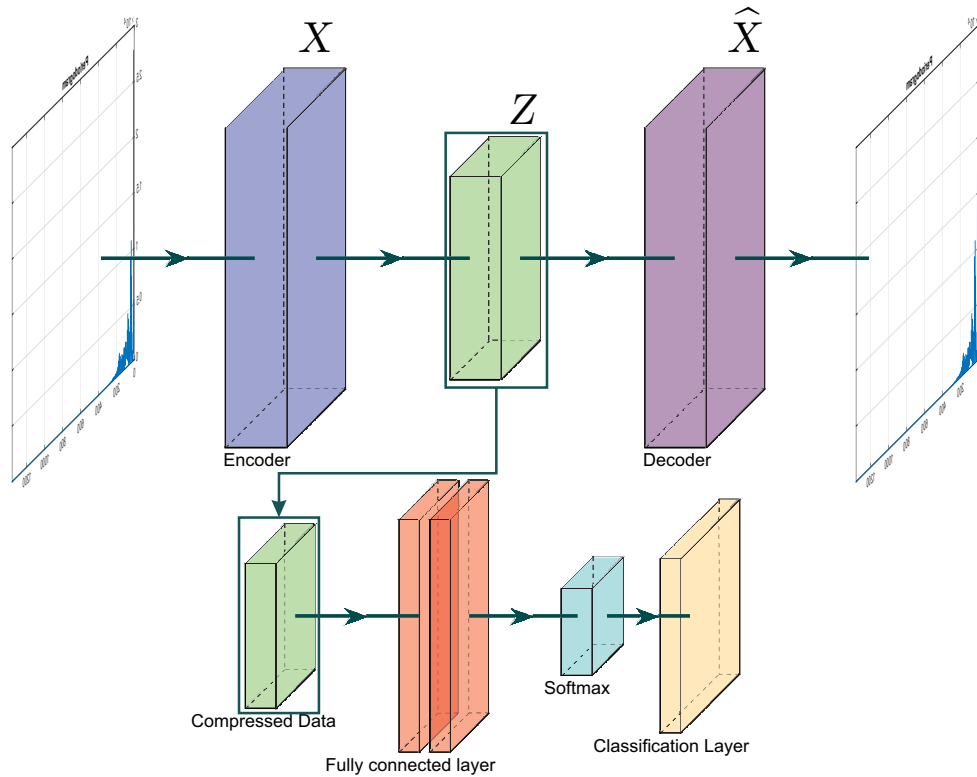
$$\forall n = o+1, \dots, N.$$

The autoregressive parameters are calculated by:

$$\hat{\xi}_{o,l} = \begin{cases} \hat{\xi}_{o-1,l} + \hat{\kappa}_o \cdot \hat{\xi}_{o-1,p-l}, & l = 1, \dots, o-1, \\ \hat{\kappa}_o, & l = o. \end{cases} \quad (3)$$

The Burg method reflects the recursive-in-order estimation of  $k_p$  given that the AR coefficients for order  $p-1$  have been computed. The reflection coefficient is estimated by:

$$\hat{\kappa}_o = \frac{-2 \sum_{n=o+1}^N \hat{\alpha}_{f,o-1}(n) \cdot \hat{\alpha}_{b,o-1}(n-1)}{\sum_{n=o+1}^N \left[ |\hat{\alpha}_{f,o-1}(n)|^2 + |\hat{\alpha}_{b,o-1}(n-1)|^2 \right]}. \quad (4)$$



**Fig. 4.** Proposed scheme for atrial fibrillation classification

The prediction errors in equations 1 and 2 satisfy the following recursive-in-order expressions:

$$\hat{\alpha}_{f,o}(n) = \hat{\alpha}_{f,o-1}(n) + \hat{\kappa}_o \cdot \hat{\alpha}_{b,o-1}(n-1), \quad (5)$$

$$\hat{\alpha}_{b,o}(n) = \hat{\alpha}_{b,o-1}(n-1) + \hat{\kappa}_o \cdot \hat{\alpha}_{f,o-1}(n). \quad (6)$$

Finally, from 5 and 6, the power spectral density (PSD) estimation is formed by:

$$\hat{\rho}_{\text{Burg}}(f) = \frac{\hat{\alpha}_o}{\left| 1 + \sum_{i=1}^o \hat{\xi}_o(i) \cdot e^{-j2\pi f i} \right|^2}, \quad (7)$$

where  $\hat{\alpha}_o = \hat{\alpha}_{f,o} + \hat{\alpha}_{b,o}$ .

### 2.3.2 Covariance Spectrum

The covariance spectrum is a representation used in signal analysis, including electrocardiogram

(ECG) signals. It is based on estimating the signal's autocovariance function, which describes how signal values at different time points are related [6]. For any complex time series, a similar estimator can be found by minimizing the estimate of the prediction error power by:

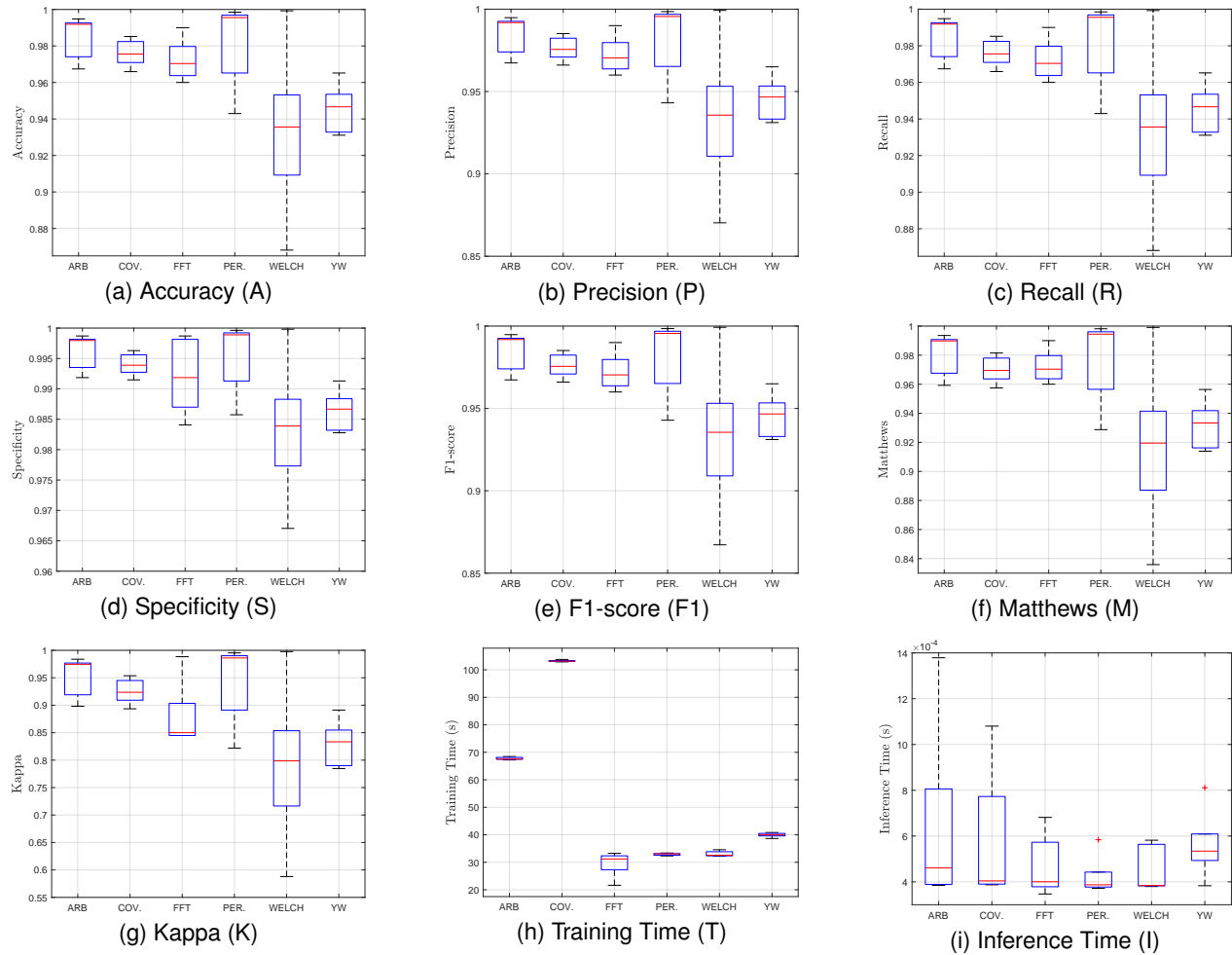
$$\hat{\lambda} = \frac{1}{N-o} \sum_{n=o}^{N-1} \left| x(n) + \sum_{l=1}^o \gamma(l) \cdot x(n-l) \right|^2. \quad (8)$$

The complex gradient can affect the minimization of 8 to achieve the AR parameter estimates as the solution of the equations:

$$c_o + C_o \cdot \hat{\gamma} = 0, \quad (9)$$

where:

$$c(i, j) = \frac{1}{N-o} \sum_{n=o}^{N-1} x(n-i) \cdot x(n-j). \quad (10)$$



**Fig. 5.** Statistical values of cross-validation for the proposed model

From 9, the AR parameters estimates are calculated by:

$$\hat{\gamma} = -c_o C_o^{-1}. \quad (11)$$

The white noise variance is estimated as:

$$\hat{\sigma}^2 = c(0, 0) + \sum_{i=1}^o \hat{\gamma}(i) \cdot c(0, i). \quad (12)$$

And then, from 11 and 12 the PSD can be estimated by:

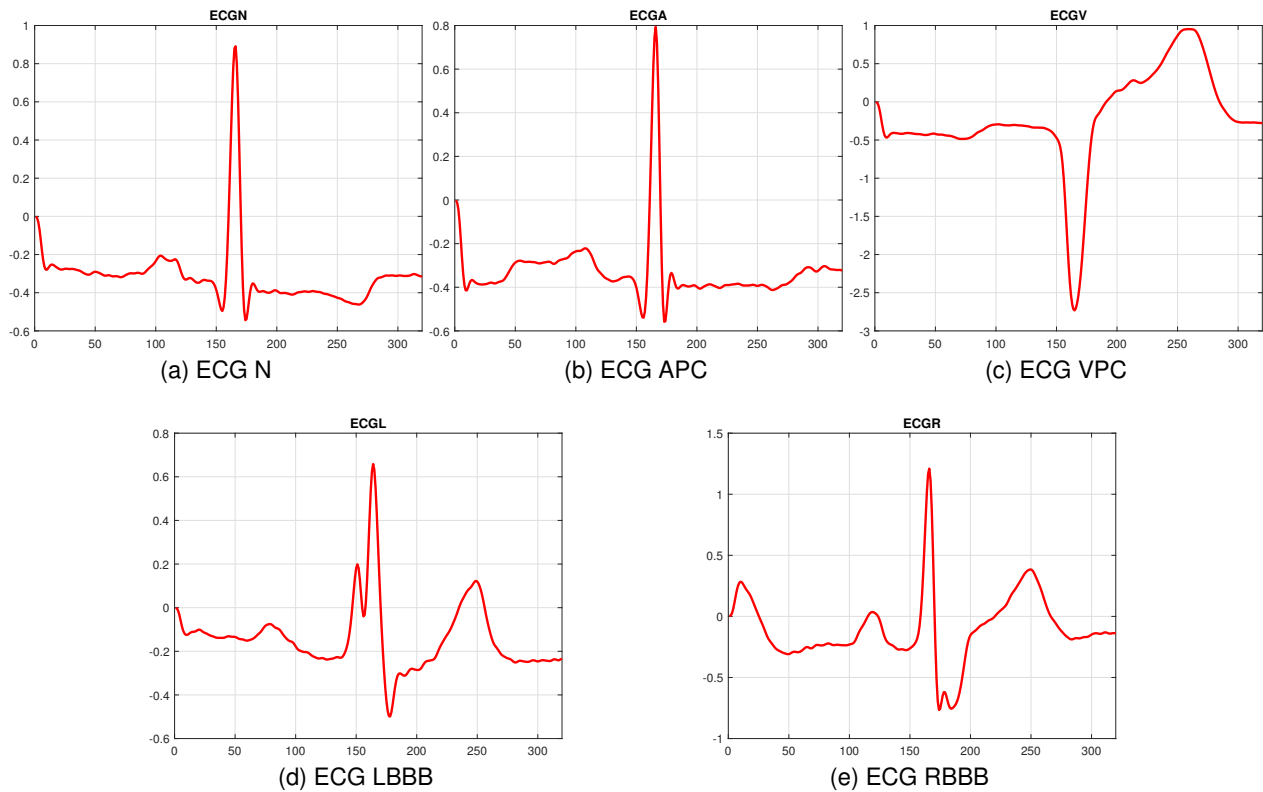
$$\hat{\varphi}_{\text{Cov}}(f) = \frac{\hat{\sigma}^2}{\left| 1 + \sum_{i=1}^o \hat{\gamma}_o(i) \cdot e^{-j2\pi f i} \right|^2}. \quad (13)$$

### 2.3.3 Yule-Walker Spectrum

This method is also called autocorrelation method, fits an autoregressive model to the windowed input data by minimizing the forward prediction error in the least squares sense [18]:

$$\hat{\lambda} = \frac{1}{N} \sum_{n=-\infty}^{\infty} \left| x(n) + \sum_{l=1}^o \gamma(l) \cdot x(n-l) \right|^2. \quad (14)$$

All the values of the  $x(n)$  process out of the range  $0 \leq n \leq N - 1$  are equal to 0. The complex gradient can help to minimize the estimated prediction error power by differentiating



**Fig. 6.** Samples of the MIT-BIH arrhythmia database

the equation with respect to the real and imaginary parts of  $\gamma(l)$ 's:

$$\frac{1}{N} \sum_{n=-\infty}^{\infty} \left( x(n) + \sum_{l=1}^o \gamma(l) \cdot x(n-l) \right) \cdot x(n-l), \quad (15)$$

$$\forall l = 0, \dots, o.$$

The matrix representation is defined by:

$$\hat{r}_o + \hat{R}_o \cdot \hat{\gamma} = 0, \quad (16)$$

where:

$$\hat{r}(k) = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-1-k} \Theta, & k = 0, 1, \dots, o, \\ \hat{r}(-k), & k = (-m+1), \dots, -1, \end{cases} \quad (17)$$

where  $\Theta = x(n) \cdot x(n+k)$ . From 16, the AR parameters are obtained by:

$$\hat{\gamma} = -\frac{\hat{r}_m}{\hat{R}_m}. \quad (18)$$

Then, the white noise variance  $\hat{\sigma}^2$  is found as  $\hat{\sigma}_{\min}$ , which is given by:

$$\hat{\sigma}^2 = \hat{\sigma}_{\min} = \frac{1}{N} \sum_{n=-\infty}^{\infty} \left| x(n) + \sum_{k=1}^o \hat{\gamma}(k) x(n-k) \right|^2. \quad (19)$$

And the final result is obtained by using 14:

$$\hat{\sigma}^2 = \hat{r}(0) + \sum_{k=1}^o \hat{\gamma}(k) \cdot \hat{r}(-k). \quad (20)$$

Finally, from 19 and 20, the PSD is estimated by:

$$\hat{\phi}_{Y^W}(f) = \frac{\hat{\sigma}^2}{\left| 1 + \sum_{l=1}^o \hat{\gamma}(l) \cdot e^{-j2\pi f l} \right|^2}. \quad (21)$$

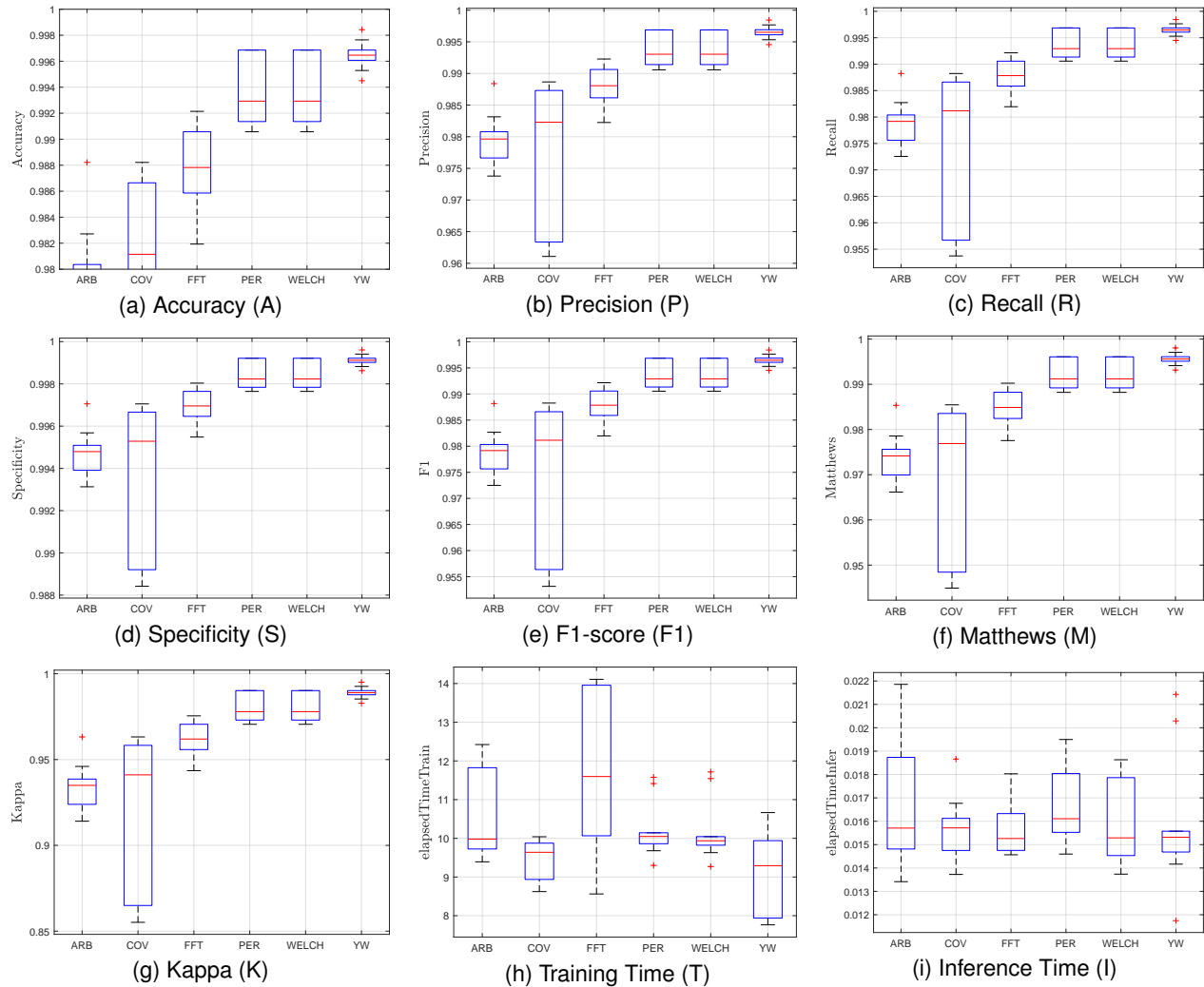


Fig. 7. Statistical values of cross-validation

### 2.3.4 Fast Fourier Transform Spectrum

Fast Fourier Transform is a very efficient algorithm in computing Discrete Fourier Transform (DFT) coefficients and can reduce a very large amount of computational complexity evaluating the DFT series [18]. The expression for the DFT is:

$$X(m) = \sum_{n=0}^{N-1} x(n) \cdot e^{(-j2\pi mn/N)}, \quad (22)$$

where  $m$  is the family member or harmonic number, and  $N$  is the length of the digital data.

From a computational point of view,  $X(m)$  must be allowed to have both positive and negative values for  $m, m = -N/2 - \dots, N/2 - 1$ . Negative values correspond to negative frequencies that have no physical meaning.

Since  $X(m)$  is a function of  $m$ , this will need to be converted to the frequency domain  $f = fm = mf_T = m/NT; ifm/N = fT_s$  is substituted into 22, then the expression for the DFT becomes:

$$X(f) = \sum_{n=0}^{N-1} x(n) \cdot e^{(-j2\pi n f T_s)}. \quad (23)$$



**Table 1.** Comparison of results between traditional techniques

Model	A	P	R	S	F1	M	K	T (s)	I (s)
RanFor	0.61	0.62	0.61	0.61	0.61	0.01	0.01	32	0.01
AdaB	0.63	0.64	0.63	0.64	0.63	0.02	0.02	30	0.07
NeuNet	0.67	0.75	0.67	0.70	0.64	0.02	0.02	35	0.02
NeaNei	0.81	0.81	0.81	0.81	0.81	0.84	0.83	33	1.95
BiLSTM	0.90	0.92	0.94	0.92	0.93	0.87	0.87	43	0.35

### 2.3.5 Periodogram Spectrum

The periodogram was introduced in searching for hidden periodicities while studying sunspot data, it is called a direct method because it directly uses the time series to estimate the PSD, it provides the critical link between quantities that you can measure directly from the finite-length signal and those that you can't [10]. The original estimates of PSD  $\varphi(f)$  for discrete and continuous notations were obtained by:

$$\varphi(f) = \int_0^{\tau} \chi(\tau) \cdot e^{-j2\pi f\tau} d\tau, \quad (24)$$

where  $\chi(\tau)$  is the autocorrelation function. This is a mirror symmetric function, and from 24 it is possible to rewrite the equation in cosine terms:

$$\varphi(f) = \int_0^{\tau} \chi(\tau) \cdot \cos(2\pi f\tau) d\tau. \quad (25)$$

Since the energy of an analogue signal  $x(t)$  is equal to the magnitude of the signal, a direct method can be used using an integrated over time:

$$\mathcal{E} = \int_{-\infty}^{\infty} |x(t)|^2 dt. \quad (26)$$

Taking into account Parseval's theorem, we can assume the following:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(f)|^2 df, \quad (27)$$

where  $|X(f)|^2$  represents the energy density function over frequency which is also called PSD. Then, the power spectrum is computed as the magnitude squared of the FT of the waveform of interest  $x(t)$ :

$$\varphi_S(f) = |X(f)|^2. \quad (28)$$

### 2.3.6 Welch Spectrum

When it is necessary to average a periodogram, a popular technique for this is the Welch's method. This method is based on the same idea as Bartlett's approach of splitting the data into segments. Welch method is basically a refinement of the Bartlett method in two respects. First, the data segments in the Welch method are allowed to overlap.

Second, each data segment is windowed prior to computing the periodogram. This method uses a sequence of random data  $x[n]$ , which is divided into  $M$  frames of length  $L$  with  $D$  overlapping points. Each frame is multiplied by a window  $W_L[n]$  in the described way for averaged modified periodogram [10]. The periodograms of each frame are computed and averaged together to form the Welch periodogram, it is obtained by the means of:

$$\varphi_w(\omega, L, D, M) = \frac{\frac{1}{M} \sum_{o=0}^{M-1} \left( \left| \sum_{n=0}^{L-1} \Theta \right|^2 \right)}{\sum_{n=0}^{L-1} w_L^2[n]}, \quad (29)$$

where  $\Theta = x[n + pD] \cdot w_L[n] \cdot e^{-j\omega n}$ . For better understanding, Figure 2 shows a sample of each transformation using an Atrial Fibrillation (AF) signal of length 2714 taken at 300Hz.

As seen in Figure 2, preprocessing reduces the signal dimension from 2714 to 1358, corresponding to the frequency and features of the original signal. These spectral representations can then be used to train deep learning models for improved results.

## 2.4 Autoencoder

Autoencoders are a feedforward and non-recurrent neural network which is unsupervised learning, meaning it can learn from itself. The objective of Autoencoders is to reduce the dimension and minimize the differences between the encoded and decoded data [4]. The process of encoding and decoding is shown in Figure 3. From a mathematical point of view, the autoencoding is formalized as:

$$z = \sigma(W \cdot x + b), \quad (30)$$

$$\hat{x} = \sigma'(W' \cdot z + b'), \quad (31)$$

where  $z$  is de encoded data,  $\hat{x}$  is the decoded data,  $\sigma$  and  $\sigma'$  are activation functions,  $W$  and  $W'$  are the weights of the input and output,  $b$  and  $b'$  the biases of the encoded and decoded data. As mentioned before, the Autoencoder is an unsupervised network, in this sense, the equation 32 must be considered for minimizing the error of reconstruction:

$$\ell(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \sigma'(W' \cdot \sigma(W \cdot \mathbf{x} + \mathbf{b}) + \mathbf{b}')\|^2. \quad (32)$$

In addition to this theory, the base architecture of an Autoencoder can be modified depending on the task to be performed, other models can be stacked on their central layer to take advantage of the encoder's feature extraction.

## 3 Proposed Scheme

This section outlines the deep autoencoder architecture and the hyperparameter tuning used to optimize the model's performance. The proposed method for the classification of atrial fibrillation is divided into two sections: the encoding process and the classification process. The proposed architecture is illustrated in Figure 4.

### 3.1 Encoding

This part corresponds to the basic process of the autoencoder and to the first part of the proposed methodology. In this section, the autoencoder is trained, and the input signals are encoded in an abstract representation, extracting features and reducing dimensionality.

### 3.2 Classification

Once the autoencoder is trained and the signals are encoded, the abstract representations are used to feed the classification model. This model consists of training a simple fully connected layer, which analyzes the features and makes the classification through a softmax activation function.

### 3.3 Hyperparameters Tuning

Hyperparameter tuning plays a crucial role in optimizing the performance of the proposed model for ECG classification [3]. This process includes the careful selection of the learning rate, the number of training epochs, the mini-batch size, and L2 weight regularization.

These parameters are fine-tuned to find the optimal balance between model accuracy and training time, ensuring that the model neither overfits nor underfits the training data. Extensive experimentation was conducted with different hyperparameter configurations to achieve the best test accuracy. From the above, the best tuning for classifying ECG classes using the autoencoder is as follows:

Encoding and decoding layer with 1358 neurons, corresponding to the input length of 1358 samples, this is represented as:  $x = \{x_i | i = 0, \dots, 1358\}$  and  $\hat{x} = \{\hat{x}_j | j = 0, \dots, 1358\}$ ; a hidden layer of 10 neurons, represented as:  $z = \{z_\nu | \nu = 1, \dots, 10\}$ ; 1000 epochs for training; L2 weight regularization of 0.001; sparsity regularization of 4, sparsity proportion of 0.04 and a linear transfer function for encode and decode.

Due to the nature of the autoencoder, the features extracted in the encoding layer are used to reconstruct the signal. However, this reconstruction process is not directly relevant to the classification task, as the classification occurs in the central layer.

For the classification model with a softmax function, the following hyperparameters were selected: learning rate of 0.01, minibatch of 256, L2 weight regularization of 0.001 and 200 epochs for training. To infer the classes and perform the classification, a softmax function was used.

For binary classification, the output of the softmax function is defined as [7]:

$$\sigma(z) = \frac{e^x}{e^x + e^0}, \quad (33)$$

where  $\sigma(z)$  is the output vector of the softmax function,  $z$  is the input vector, it contains 2 values 0 and 1,  $e$  is the mathematical constant and  $x$  is the index of the class, in this case, 0 or 1.

## 4 Experimental Results

### 4.1 Proposed Model

Once preprocessing was completed, the spectral representations of the signals were obtained. Due to the limited size of the dataset, a cross-validation approach was employed, dividing the dataset into 10 folds to ensure variability and an 90/10 split between training and testing data.

A training session was conducted for each of the six spectra obtained to compare results. All trainings were performed using MATLAB 2023b, and the results were validated using standard metrics from the literature, including Accuracy, Recall, Specificity, Precision, F1 score, Matthews Correlation Coefficient, Kappa, and processing and inference time.

Figure 5 shows statistical results of cross validation for all spectral representations. The results indicate that the Welch and Periodogram methods are the most effective and consistent for atrial fibrillation classification using a deep autoencoder.

These methods exhibit high medians in all quality metrics, with low variance and no significant outliers. This ensures superior and stable performance, essential for real-time applications or critical environments where precision is paramount.

Although their training times are not the shortest, they remain reasonable compared to Covariance, which, despite achieving good results in several metrics, presents high variance and prolonged training times, rendering it less practical for implementation. Conversely, the FFT method, with significantly lower training times and competitive results in key metrics, emerges as an

excellent alternative when computational efficiency is prioritized. For real applications, the FFT model is suggested since it has the best balance between quality metrics and processing time.

### 4.2 Generalization of the Model

To ensure the robustness and generalization of the proposed method, it was evaluated with another database, corresponding to the MIT-BIH database, which contains 2546 instances for the classes Normal Rhythm (N), Atrial Premature Contraction (APC), Ventricular Premature Contraction (VPC), Left Bundle Branch Block (LBBB), and Right Bundle Branch Block (RBBB). All the signals are the ECG lead II and they were taken at 360Hz. Figure 6 shows a sample of each class.

For the training, the same hyperparameter tuning and configuration for 10-fold cross-validation were used. Figure 7 shows statistical values for the MIT-BIH training. The results demonstrate the effectiveness of the proposed model across different datasets, maintaining high performance in key metrics. The Welch and Periodogram methods exhibit high accuracy and specificity, close to 99%, and excellent precision and recall, surpassing 98%.

These results indicate the model's strong ability to distinguish between normal heart rhythms and various arrhythmias, with high sensitivity and precision. Evaluating the model with the MIT-BIH database confirms its generalization capacity and robustness. The Welch and Periodogram methods consistently deliver high accuracy and reliability, while FFT offers an efficient training solution. These findings suggest the model is well-suited for clinical and real-time monitoring applications, providing an effective tool for automatic cardiac arrhythmia classification.

### 4.3 State of the Art Techniques

Additionally, Table 1 shows average test results of other models used in the state of the art. On the other hand, Table 1 highlights the comparative performance of various state-of-the-art models trained with spectral features.

The proposed model surpasses these traditional models in quality metrics, showcasing the effectiveness of spectral features for ECG signal classification. Models like Random Forest and AdaBoost exhibit lower accuracy, indicating their struggles with spectral features.

While BiLSTM achieves higher accuracy, it comes with significant computational cost and longer training times, making it less suitable for real-time applications.

This comparison underscores the advantage of using spectral features with deep autoencoders, which offer a robust balance of high performance and computational efficiency, essential for practical and clinical deployments.

## 5 Conclusions and Future Work

This study presents the design and implementation of a deep spectral autoencoder combined with a fully connected softmax layer, trained using spectral representations of ECG signals for atrial fibrillation classification.

The proposed model achieves high performance across various metrics due to effective preprocessing, which normalizes ECG signals to a consistent length and transforms them into the spectral domain. Spectral representations significantly enhance feature extraction and dimensionality reduction, capturing critical frequency-domain information, thereby improving the model's effectiveness.

Among the spectral methods used, Welch and Periodogram provide the best results, exhibiting high statistical values in quality metrics with low variance and no significant outliers. Despite longer training times, they are efficient compared to other methods like Covariance.

The FFT method also performs well, offering lower training times and competitive results, making it suitable for computationally efficient implementations, such as low-medium cost embedded systems. The proposed model outperforms traditional state-of-the-art models, demonstrating the advantages of using spectral features with deep autoencoders.

This combination of high performance and computational efficiency makes the model highly suitable for real-time applications and clinical use.

The evaluation with the MIT-BIH database confirms the robustness and generalization capacity of the proposed model. The Welch and Periodogram methods consistently deliver high accuracy and reliability, while the FFT method provides an efficient solution in terms of training time.

These findings suggest that the model is highly applicable in clinical and real-time monitoring environments, providing an effective and efficient tool for the automatic classification of cardiac arrhythmias.

Future work should explore integrating additional models in the central layer to enable deeper analysis and potentially enhance performance. It is crucial to validate the proposed approach with larger and more diverse datasets to ensure the robustness and generalization of the model. These efforts can improve model performance and contribute to developing more efficient and accessible tools for automatic cardiac arrhythmia classification.

## Acknowledgments

The authors would like to thank the CONAHCyT, TecNM/CENIDET for their support of this research through the project "Classifier to detect atrial fibrillation in electrocardiographic signals using a deep recurrent network trained with time-frequency moments".

## References

1. Akkuş, M., Karabatak, M., Tekin, R. (2024). Classification of ECG signals encrypted with CNN based autoencoder with LSTM. Proceedings of the 12th International Symposium on Digital Forensics and Security, pp. 1–6. DOI: 10.1109/ISDFS60797.2024.10527229.

2. **Asif, M. S., Faisal, M. S., Dar, M. N., Hamdi, M., Elmannai, H., Rizwan, A., Abbas, M. (2023).** Hybrid deep learning and discrete wavelet transform-based ECG biometric recognition for arrhythmic patients and healthy controls. *Sensors*, Vol. 23, No. 10, pp. 4635. DOI: 10.3390/s23104635.
3. **Bacanin, N., Stoean, C., Zivkovic, M., Rakic, M., Strulak-Wójcikiewicz, R., Stoean, R. (2023).** On the benefits of using metaheuristics in the hyperparameter tuning of deep learning models for energy load forecasting. *Energies*, Vol. 16, No. 3, pp. 1434. DOI: 10.3390/en16031434.
4. **Enttsel, A., Marchioni, A., Setti, G., Mangia, M., Rovatti, R. (2024).** Enhancing anomaly detection with entropy regularization in autoencoder-based lightweight compression. *Proceedings of the IEEE 6th International Conference on AI Circuits and Systems*, pp. 273–277. DOI: 10.1109/AICAS59952.2024.10595983.
5. **García-Aquino, C., Mújica-Vargas, D., Matuz-Cruz, M., Gonzalez-Franco, N., González-Serna, G. (2022).** Implementation of time-frequency moments for the classification of atrial fibrillation sequences through a bidirectional long-short term memory network. *Telematics and Computing*, pp. 201–214. DOI: 10.1007/978-3-031-18082-8\_13.
6. **Hashim, B. T., Ziboon, H. T., Abdulsatar, S. M. (2023).** Covariance absolute values spectrum sensing method based on two adaptive thresholds. *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 30, No. 2, pp. 1029. DOI: 10.11591/ijeecs.v30.i2.pp1029-1037.
7. **Karyemsetty, N., Syamsundarao, T., Venkata-Kishore-Babu, S., Suresh-Kumar, D., Goddu, J., Samatha, B. (2023).** Intrusion detection system in vehicular network using deep learning approach. *Proceedings of the 5th International Conference on Smart Systems and Inventive Technology*, pp. 999–1003. DOI: 10.1109/ICSSIT55814.2023.10060872.
8. **Kumar, R., Jyoti (2023).** A hybrid approach using SVM, kNN and random forest for ECG classification. *Proceedings of the 11th International Conference on Intelligent Systems and Embedded Design*, pp. 1–6. DOI: 10.1109/ISED59382.2023.10444570.
9. **Li, Y., Han, L., Liu, X. (2023).** Accuracy enhancement and feature extraction for GNSS daily time series using adaptive CEEMD-multi-PCA-based filter. *Remote Sensing*, Vol. 15, No. 7, pp. 1902. DOI: 10.3390/rs15071902.
10. **Melinda, M., Enriko, I. K. A., Furqan, M., Irhamsyah, M., Yunidar, Y., Basir, N. (2023).** The effect of power spectral density on the electroencephalography of autistic children based on the welch periodogram method. *Jurnal Infotel*, Vol. 15, No. 1, pp. 111–120. DOI: 10.20895/infotel.v15i1.874.
11. **Nai, Y., He, D., Fang, Z., Yu, Z. (2023).** SoC design of ECG intelligent classification. *Proceedings of the 10th International Symposium on Next-Generation Electronics*, pp. 1–3. DOI: 10.1109/ISNE56211.2023.10221624.
12. **Sharma, P., Dinkar, S. K. (2024).** An intelligent deep neural network with opposition based laplacian equilibrium optimizer to improve feature extraction using ECG signals. *Biomedical Signal Processing and Control*, Vol. 87, pp. 105415. DOI: 10.1016/j.bspc.2023.105415.
13. **Soo, K. M., Liew, T. S., Heath, P. (2024).** A cross sectional study of atrial fibrillation prevalence and associated cardiac arrhythmias in stroke patients: Insights from commercial data archives. *MedRxiv*, Vol. 2024, No. 4, pp. 1–10. DOI: 10.1101/2024.04.21.24306150.
14. **Subasi, A. (2019).** Practical guide for biomedical signals analysis using machine learning techniques: A

MATLAB based approach. Elsevier.  
DOI: 10.1016/c2018-0-02414-7.

15. **Subhash, C. S., Mylarappa, N. K., Sharan, M. G. (2024).** Optimized ECG classification using wavelet decomposition and CNN's. *International Advanced Research Journal in Science, Engineering and Technology*, Vol. 11, No. 6. DOI: 10.17148/iarjset.2024.11637.
16. **Tahsin, K., Hossain, M. F., Asadur-Rahman, M. (2023).** Atrial fibrillation diagnosis from ECG signal using statistical and time-frequency attributes-based LSTM network. *Proceedings of the 26th International Conference on Computer and Information Technology*, pp. 1–6. DOI: 10.1109/ICCIT60459.2023.10441524.
17. **Tang, H., Xie, S., Xie, X., Cui, Y., Li, B., Zheng, D., Hao, Y., Wang, X., Jiang, Y., Tian, Z. (2024).** Multi-domain based dynamic graph representation learning for EEG emotion recognition. *IEEE Journal of Biomedical and Health Informatics*, Vol. 28, No. 9, pp. 5227–5238. DOI: 10.1109/jbhi.2024.3415163.
18. **Terán, M., Osorio, J., Garcia, J., Marin, J., Aranda, J. (2023).** A methodology for GNSS signals acquisition for non-conventional spectral estimation techniques evaluation using software-defined radio. *Proceedings of the IEEE Colombian Conference on Communications and Computing*, pp. 1–6. DOI: 10.1109/COLCOM59909.2023.10334282.

*Article received on 28/05/2024; accepted on 04/07/2024.*

*\*Corresponding author is Dante Mujica-Vargas.*