Classification of Time Series ECG Signals using Visibility Graph[1]

Cristian B. Jetomo

Institute of Mathematical Sciences and Physics
University of the Philippines Los Baños
College, Laguna 4031, Philippines
(e-mail: cbjetomo@up.edu.ph)

Abstract

Electrocardiogram (ECG) signals are used in medicine to identify signs of cardiovascular diseases of a patient. This is done by observing and identifying possible abnormalities in the time series signal. Machine learning tools has been utilized to aid this classification problem, that is classifying normal and abnormal ECG signals. But with the introduction of Visibility Algorithm which converts time series data into a complex network, different approaches were developed to solve the classification problem. This paper presents a simple approach to perform ECG signal classification by utilizing graph properties obtained from visibility graphs. The classification is done using Logistic Regression which resulted with a promising overall accuracy of 96.9%.

Keywords: ECG Classification, Visibility Graph, Graph Properties, Logistic Regression

[1]A project in MATH 191: Complex Networks and Topological Data Analysis

1 Introduction

Electrocardiogram (ECG) is a periodic signal or time series which shows heart activities. In a long timescale, ECG signals are measured and recorded to determine presence of abnormalities. From ECG a lot of information is obtained for normal and pathological physiology of the heart [4]. However, ECG signals are non-stationary and thus, analysis of such signals is quite challenging. With this, medical practitioners utilizes computer-based techniques in ECG analysis. This is essential for identification of cardiovascular diseases which induces abnormalities and changes in the ECG signals. Hence, early detection is important to prevent such a cause of death.

Along with the different computer-based techniques in ECG analysis is the use of Machine Learning (ML). Different papers have been published that utilizes this approach [4, 5, 20]. Jambukia et.al. [9] provided a detailed survey of ECG signal classification using ML techniques, going through the necessary stages such as preprocessing, feature extraction, feature normalization, and classification. However, most of the covered studies in this paper focuses on using the signal as a time series, or a set of quantitative observations arranged in chronological order [11], in their classification.

A different approach has been utilized recently which uses network analysis to perform classification on time series data [12, 16]. This has become studied since Lacasa et.al. introduced the Visibility Algorithm [13]. Essentially, this algorithm converts a time series data into a complex network or its corresponding Visibility Graph (VG). A detailed discussion of the algorithm is presented in Section 2.2. Different methods that utilizes the Visibility Algorithm has been introduced to solve the ECG signal classification problem. This paper is a simple variety of such methods.

In this paper, ECG signals will be classified whether normal or abnormal by transforming the signals into its corresponding Horizontal Visibility Graph, a special variety of VG. The classification is done using Logistic Regression and its performance is quantified using four (4) evaluation metrics.

2 Methodology

2.1 Dataset

This paper will focus on performing classification on the PTB Diagnostic ECG Database [3]. The original database consists of 549 records of 15-lead ECG signals (12 standard leads together with Frank XYZ leads). These data are recorded from 294 subjects which include healthy patients and patients with a variety of heart diseases. From the original dataset, samples are cropped, downsampled, and padded with zeroes if necessary to obtain a fixed dimension of 188 by using a sampling frequency of 125Hz. This results with a total of 14552 samples, having two (2) categories, *normal* and *abnormal* ECG signal indicating signals obtained from a patient with normal and abnormal heart condition, respectively.

2.2 Visibility Graphs

Visibility Algorithm (VA) is an algorithm introduced by Lacasa et.al. [13]. This algorithm maps every singe time series onto a node in a complex network. The constructed network or graph, called Natural Visibility Graphs (NVG), inherits several properties of the time series. The VA maps periodic time series to regular graphs, random time series to random graphs, and fractal time series to scale-free graphs [13]. Performing network analysis on the corresponding graph provides additional properties on the time series.

Suppose $\{x(t_i)\}$ for $i = 1, 2, \ldots, N$ be a time series having N samples. Two arbitrary data values (t_1, x_1) and (t_2, x_2) will have visibility, and thus become two connected nodes in the corresponding network, if any other data value (t_3, x_3) placed between them satisfied

$$x_3 < y_2 + \frac{t_2 - t_2}{t_2 - t_1}(y_1 - y_2)$$

A graphical representation is shown in Figure 1. A different representation of the graph is given in Figure 2. Here, the hue of the green color of the node and the node size are directly proportional with the node's degree. The node labels represents the index of the time series data.

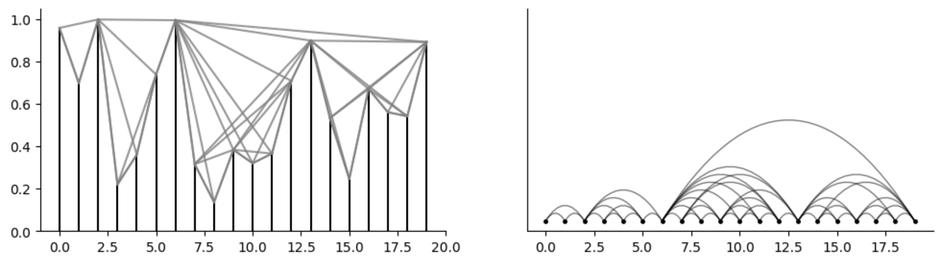

Figure 1: Example of a time series (20 data values) and its associated graph derived using NVG. Every node in the same order, corresponds to series data. Two visible data connected in the time series are also connected in the graph.

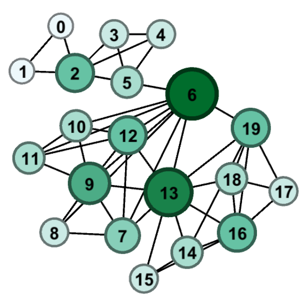

Figure 2: Another representation of corresponding NVG of sample time series data.

This proposed method allows further analysis on time series data. Through times, multiple advancements on this algorithm have being developed. One of which is the Horizontal Visibility Graph (HVG), a geometrically simpler and analytically solvable version of NVG, developed by Luque et.al. [15]. In HVG, two nodes (t_i, x_i) and (t_j, x_j) are connected if:

$$x_i > x_k \text{ and } x_j > x_k, \text{ for } i < k < j$$

Given the same time series as with the example in Figure 1, a graphical representation of its corresponding HVG is given in Figure 3. Similarly from Figure 2, the corresponding HVG of the time series data is presented in a different way in Figure 4. The visual properties, that is the node size and color, of this graph is the same as defined for Figure 2.

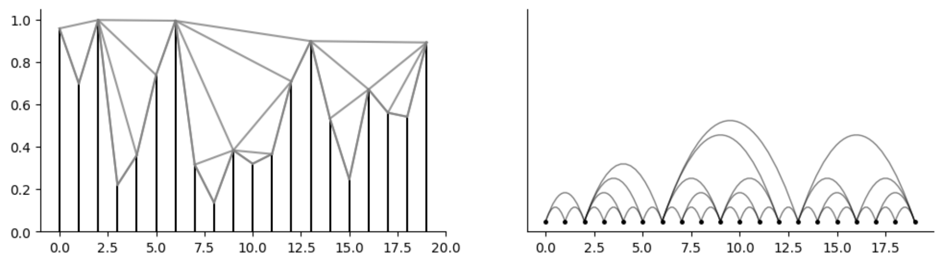

Figure 3: Example of a time series (20 data values) and its associated graph derived using HVG.

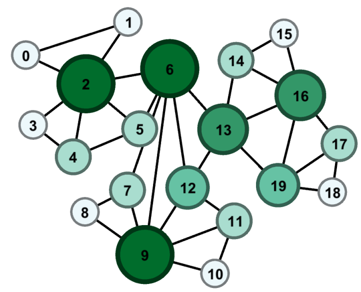

Figure 4: Another representation of corresponding HVG of sample time series data.

2.3 Graph Properties

In this paper, nine (9) features are calculated from the NVG correspondence of a time series. These features are the following:

2.3.1 Degree Distribution

The degree of a node k_i represents the number of nodes connected to node i for $i = 1, 2, \ldots, N$. In turn, $K(i)$ represents the degree distribution of the network. The extracted features that are based on the degree distribution are the following:

$$\text{Mean } (K(i)) = \frac{\sum_{i=1}^{N} k_i}{N} \tag{1}$$

$$\text{Min}(k_i) \text{ for } i = 1, 2, \ldots, N \tag{2}$$

$$\text{Max}(k_i) \text{ for } i = 1, 2, \ldots, N \tag{3}$$

$$\text{Std}(K(i)) = \sqrt{\frac{\sum_{i=1}^{N}(k_i - \text{Mean}(K(i)))^2}{N-1}} \tag{4}$$

2.3.2 Characteristic Path Length

Characteristic path length or average minimum path length represents the average measure (across the whole network) of the minimum number of edges necessary to travel from one node to another in the network [8]. This is calculated as

$$L = \frac{\sum_i \sum_j L_{ij}}{N(N-1)} \tag{5}$$

where N denotes the number of edges and d_{ij} represents the distance between two nodes i and j or the minimum path length between nodes i and j.

2.3.3 Local & Global Efficiency

Efficiency of a network represents a measure of how efficiently it exchanges information [14]. In particular, the efficiency of a pair of nodes in a graph is the multiplicative inverse of the shortest path distance between the nodes. With this, the global efficiency of a network is obtained as

$$E_{\text{glob}} = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{d_{ij}} \tag{6}$$

while the local efficiency is given by

$$E_{\text{loc}} = \frac{1}{N} \sum_{i} E_{\text{glob}}(G_i) \tag{7}$$

where G_i is a subgraph consisting of the neighboring nodes of node i.

2.3.4 Average Clustering Coefficient

The clustering coefficient (c_i) of a node i in a graph is a measure of how connected the neighbors of node i are [7]. It is calculated using the formula

$$c_i = \frac{2e_i}{k_i(k_i - 1)}$$

where k_i is the degree of node i and e_i is the number of edges connecting their nearest neighbors to node i. Taking its average across all nodes gives us the average clustering coefficient

$$C = \frac{\sum_i c_i}{N} \tag{8}$$

2.3.5 Assortativity Coefficient

The assortativity coefficient represents the nodes of a graph that are connected to similar nodes [18]. The network classification coefficient is often estimated assortativity based on a node's degree. This property can be calculated as

$$r = \frac{M^{-1}\sum_i j_i k_i - \left(M^{-1}\sum_i \frac{1}{2}(j_i + k_i)\right)^2}{M^{-1}\sum_i \frac{1}{2}(j_i^2 + k_i^2) - \left(M^{-1}\sum_i \frac{1}{2}(j_i + k_i)\right)} \tag{9}$$

where j_i and k_i represent the degrees of two nodes at the end of the *ith* link and M is the total number of links.

2.4 Preprocessing

Feature Normalization is essential in machine learning as it addressed the issue of performing classification on features having varying degrees of magnitude, range, and units. It is used to guarantee that some machine learning models can work and also help to improve the model's training speed and performance [10]. This process rescales the input features to make the values lie between 0 and 1. One example of this is the MinMax Scaler.

Given a feature x_i of an observation where $X = \{x_i\}$ is the set containing all *ith* features of all observations. Then x_i is scaled to x_i' using

$$x_i' = \frac{x_i - \min X}{\max X - \min X}$$

2.5 Classification Method

Logistic Regression was used in the classification of the ECG signals into normal or abnormal. This classifier has low risk of overfitting, is simple to implement, its outputs have probabilistic interpretations, it can be used for both binary and multiclassication problems, and it can be updated easily after adding new data [17]. The logistic regression is done using the expression below.

$$f(x) = \frac{1}{1+e^{-x}}$$

2.6 Performance Evaluation Metrics

Four (4) performance metrics was computed to quantify the efficiency of the model used to classify the network-transformed ECG signals. These metrics and its descriptions are the following:

$$\begin{aligned}
\text{Precision} &= \frac{TP}{TP+FP} \\
\text{Recall} &= \frac{TP}{TP+FN} \\
\text{F1-score} &= \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \\
\text{Accuracy} &= \frac{TP+TN}{TP+TN+FP+FN}
\end{aligned}$$

where TP (True Positive) is the number of observations belonging class k and classified to class k, TN (True Negative) is the number of observations not belonging class k and not classified to class k, FP (False Positive) is the number of observations not belonging class k but classified to class k, and FN (False Negative) is the number of observations belonging to class k but not classified to class k.

This then infer that precision describes the exactness of model in terms of prediction, recall describes completeness, f1-score describes both precision and recall, and the overall accuracy describing total correct predictions out of all predictions.

3 Results and Discussion

From the original ECG, a sample of signal was obtained for each the normal and abnormal data for visualization purposes. This is shown in Figure 5. Applying HVG on the normal and abnormal data, we obtain the networks presented in Figure 6. Evidently, we see here differences in the shape of their corresponding networks. To quantify this, the graph properties are solved as stated in Section 2.3. Doing so yields the result in Table 1. We see here some similarities in the parameters obtained from the degree distribution of the nodes of the HVGs. However, we can see huge differences in values for the other measures such as the *Characteristic Path Length*, *Local and Global Efficiencies*, *Average Clustering Coefficient*, and the *Assortativity Coefficient*.

The above process is done for each of the ECG signals in the dataset. Once all features (or graph properties) are extracted, the features were normalized according to the MinMax Scaler method from Section 2.4. The normalized features were then used as inputs for classification using Logistic Regression. The performance metrics of the model were obtained and quantified. The final results are presented in Figure 7.

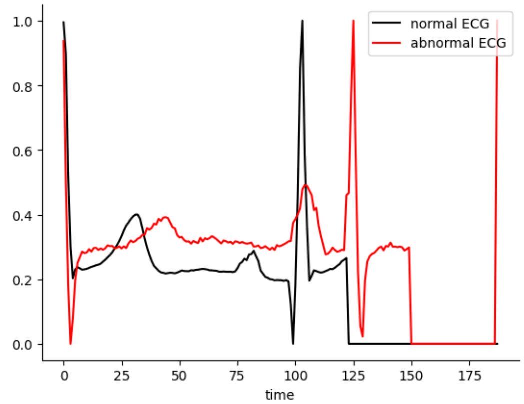

Figure 5: Sample plots of normal and abnormal ECG signals.

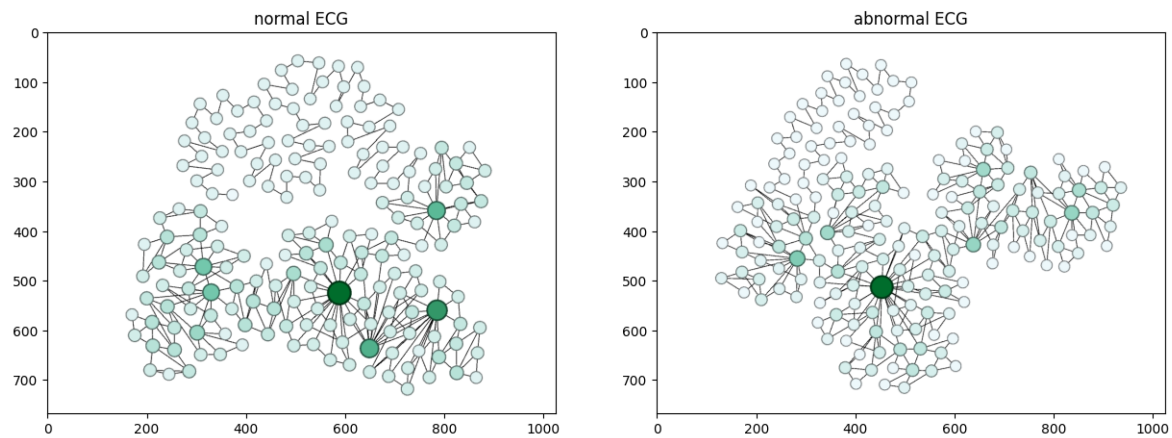

Figure 6: Corresponding HVGs of the Sample Normal and Abnormal ECG signals.

Table 1: Graph Properties of the Corresponding HVGs of the Sample Normal and Abnormal ECG signals.

Category	Mean($K(i)$)	Max($K(i)$)	Min($K(i)$)	Std($K(i)$)	L	E_{glob}	E_{loc}	C	r
Normal	3.22	2.1	1	2.27	2.38	4.89	1.26	3.89	-3.25
Abnormal	3.53	2.7	2	2.43	9.61	6.02	1.69	4.93	-2.31

From Figure 7, the trained Logistic Regression model shows that it obtained a Precision value of 90.3% and 100% for the normal and abnormal ECG signals, respectively. This means that when the model predicts an ECG signal to be abnormal, then it is correct 100% of the time. Similarly, it is correct in identifying a normal ECG signal 90.3% of the time.

Meanwhile, the model obtained a Recall value of 100% and 95.7% for the normal and abnormal ECG signals, respectively. This means that out of a pool of normal ECGs and another pool of abnormal ECGs, then we can correctly identify 100% of the normal pool and 95.7% of the abnormal pool.

In general, the F1-scores obtained by the model respectively for the normal and abnormal ECG signals are 94.9% and 97.8%. Finally , the model obtained an overall accuracy of 96.9% which is relatively high as a result of a classification problem.

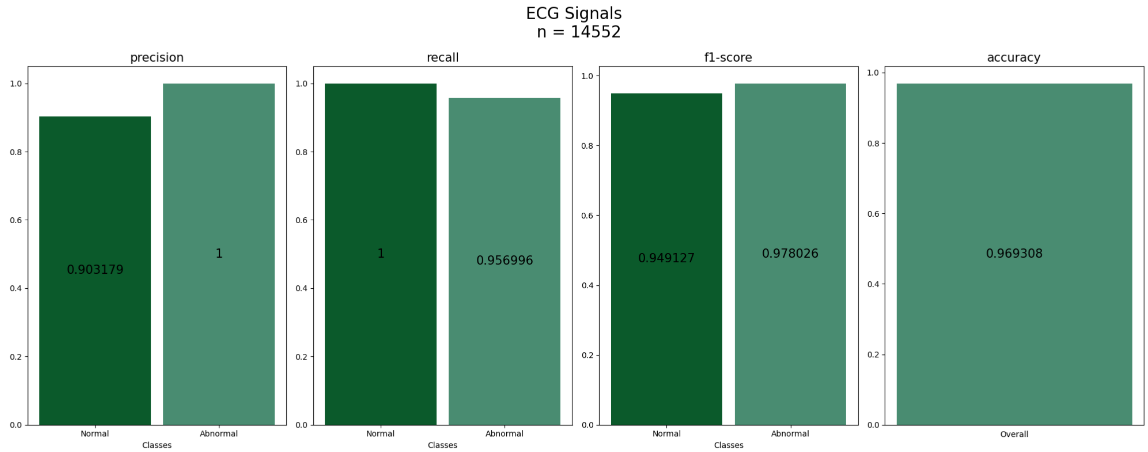

Figure 7: Classification Report of HVG-based features using Logistic Regression.

4 Conclusion

This paper introduces an alternative way of classifying normal and abnormal ECG signals using Visibility Graphs. ECG signals, being time series in nature, often requires complex preprocessing tools needed before a classification algorithm can be performed. To address this, the ECG signals are mapped into their corresponding complex networks using the HVG algorithm. The advantage of the transformed networks or graphs are their intrinsic graph properties that highlight also the features from the original time series. Hence, classification can instead be done on selected graph properties of the corresponding HVGs.

Performing classification using Logistic Regression on the HVGs of the ECG signals from the PTB Diagnostic ECG Database [3] shows great results. The performance metrics specified, Precision, Recall, F1-score, and Overall Accuracy, obtained values not less than 90%. In conclusion, the classification report of the performance metrics of the model shows that the introduced method can be done and can provide alternative ways for classification of ECG signals in the form of a time series data.

It can be noted that the performance result in this paper is limited only to classification of the dataset used. Different classification problems can be done using this approach. Moreover, the original ECG signals can also be classified using the conventional approach, or using the data as its time series form, and compared with the results of this paper to verify further the effectiveness of this approach. Some improvements can be done such as utilizing network analysis to perform classification on a different set of graph properties of the HVGs. Additional categories can also be included in the classification as there are different varieties of heart diseases which in turn require different medical processes to cure.

The entire methodology in this paper is done in Python using libraries such as ts2vg [2] for computing HVGs, NetworkX [6] for computing graph properties, scikit-learn [19] for classification tools, Gephi [1] for plotting networks, etc. The Python program can be found at `https://github.com/ji-chani/ClassificationECGwithHVG`.

References

[1] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks," in *Third international AAAI conference on weblogs and social media*, 2009.

[2] C. Bergillos, "Ts2vg: Time series to visibility graphs." 2020. [Online]. Available: https://github.com/CarlosBergillos/ts2vg

[3] R.-D. Bousseljot, D. Kreiseler, and A. Schnabel, "The ptb diagnostic ecg database," 2004. [Online]. Available: https://physionet.org/content/ptbdb/

[4] S. Celin and K. Vasanth, "ECG signal classification using various machine learning techniques," *Journal of Medical Systems*, vol. 42, no. 12, Oct. 2018. [Online]. Available: https://doi.org/10.1007/s10916-018-1083-6

[5] S. H. El-Khafif and M. A. El-Brawany, "Artificial neural network-based automated ecg signal classifier," *International Scholarly Research Notices*, vol. 2013, 2013.

[6] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11 – 15.

[7] D. L. Hansen, B. Shneiderman, M. A. Smith, and I. Himelboim, "Chapter 6 - calculating and visualizing network metrics," in *Analyzing Social Media Networks with NodeXL (Second Edition)*, second edition ed., D. L. Hansen, B. Shneiderman, M. A. Smith, and I. Himelboim, Eds. Morgan Kaufmann, 2020, pp. 79–94. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128177563000066

[8] N. Jahanshad, G. Prasad, A. W. Toga, K. L. McMahon, G. I. de Zubicaray, N. G. Martin, M. J. Wright, and P. M. Thompson, "Genetics of path lengths in brain connectivity networks: HARDI-based maps in 457 adults," in *Multimodal Brain Image Analysis.* Springer Berlin Heidelberg, 2012, pp. 29–40. [Online]. Available: https://doi.org/10.1007/978-3-642-33530-3_3

[9] S. H. Jambukia, V. K. Dabhi, and H. B. Prajapati, "Classification of ECG signals using machine learning techniques: A survey," in *2015 International Conference on Advances in Computer Engineering and Applications.* IEEE, Mar. 2015. [Online]. Available: https://doi.org/10.1109/icacea.2015.7164783

[10] W. Jiang, "Applications of deep learning in stock market prediction: Recent progress," *Expert Systems with Applications*, vol. 184, p. 115537, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421009441

[11] G. Kirchgässner, J. Wolters, and U. Hassler, *Introduction to Modern Time Series Analysis.* Springer Berlin Heidelberg, 2013. [Online]. Available: https://doi.org/10.1007/978-3-642-33436-8

[12] G. Kutluana and İ. Türker, "Classification of cardiac disorders using weighted visibility graph features from ecg signals," *Biomedical Signal Processing and Control*, vol. 87, p. 105420, 2024.

[13] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuño, "From time series to complex networks: The visibility graph," *Proceedings of the National*

Academy of Sciences, vol. 105, no. 13, pp. 4972–4975, Apr. 2008. [Online]. Available: https://doi.org/10.1073/pnas.0709247105

[14] V. Latora and M. Marchiori, "Efficient behavior of small-world networks," *Physical Review Letters*, vol. 87, no. 19, Oct. 2001. [Online]. Available: https://doi.org/10.1103/physrevlett.87.198701

[15] B. Luque, L. Lacasa, F. Ballesteros, and J. Luque, "Horizontal visibility graphs: Exact results for random time series," *Phys. Rev. E*, vol. 80, p. 046103, Oct 2009. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.80.046103

[16] T. Madl, "Network analysis of heart beat intervals using horizontal visibility graphs," in *2016 Computing in Cardiology Conference (CinC)*. IEEE, 2016, pp. 733–736.

[17] S. Mirjalili, P. Powell, J. Strunk, T. James, and A. Duarte, "Evaluation of classification approaches for distinguishing brain states predictive of episodic memory performance from electroencephalography: Abbreviated title: Evaluating methods of classifying memory states from eeg," *NeuroImage*, vol. 247, p. 118851, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811921011228

[18] Z. Mohammadpoory, M. Nasrolahzadeh, and S. Amiri, "Classification of healthy and epileptic seizure eeg signals based on different visibility graph algorithms and eeg time series," *Multimedia Tools and Applications*, pp. 1–22, 05 2023.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] Q. Zhao and L. Zhang, "Ecg feature extraction and classification using wavelet transform and support vector machines," in *2005 International Conference on Neural Networks and Brain*, vol. 2. IEEE, 2005, pp. 1089–1092.