RESEARCH ARTICLE

VOL. 44 | NO. 4 | **SPECIAL ISSUE 2023** | PP 105 - 116

Revista Mexicana de Ingeniería Biomédica

https://doi.org/10.17488/RMIB.44.4.7

E-LOCATION ID: 1394

Performance Evaluation of Biomedical Time Series Transformation Methods for Classification Tasks

Evaluación del Rendimiento de Métodos de Transformación de Series Temporales Biomédicas para Tareas de Clasificación

Carlos Alejandro Ku-Maldonado[1], *Erik Molino-Minero-Re*[2]

[1]Universidad Nacional Autónoma de México, Yucatán - México

[2]Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, UNAM, Unidad Académica, Yucatán - México

ABSTRACT

The extraction of time series features is essential across various fields, yet it remains a challenging endeavor. Therefore, it's crucial to identify appropriate methods capable of extracting pertinent information that can significantly enhance classification performance. Among these methods are those that translate time series into different domains. This study investigates three distinct time series transformation approaches for addressing time series classification challenges within biomedical data. The first method involves a response vector transformation, while the other two employ image transformation techniques: RandOm Convolutional KErnel Transform (ROCKET), Gramian Angular Fields, and Markov Transition Fields. These transformation methods were applied to five biomedical datasets, exploring various format configurations to ascertain the optimal representation technique and configuration for input, which in turn improves classification performance. Evaluations were conducted on the effectiveness of these methods in conjunction with two classification algorithms. The outcomes underscore the significance of these time series transformation techniques as facilitators for enhanced classification algorithms documented in current literature.

KEYWORDS: biomedical data, classification, convolutional neural networks, time series, transformations

RESUMEN

La extracción de características de series temporales es esencial en diversos campos, pero sigue siendo un desafío. Por lo tanto, es crucial identificar métodos apropiados capaces de extraer información pertinente que pueda mejorar significativamente el rendimiento de clasificación. Entre estos métodos se encuentran aquellos que traducen las series temporales a diferentes dominios. Este estudio investiga tres enfoques distintos de transformación de series temporales para abordar los desafíos de clasificación de series temporales en datos biomédicos. El primer método implica una transformación de vector de respuesta, mientras que los otros dos emplean técnicas de transformación de imagen: RandOm Convolutional KErnel Transform (ROCKET), Gramian Angular Fields y Markov Transition Fields. Estos métodos de transformación se aplicaron a cinco conjuntos de datos biomédicos, explorando diversas configuraciones de formato para determinar la técnica y configuración de representación óptima para la entrada, lo que a su vez mejora el rendimiento de clasificación. Se realizaron evaluaciones sobre la efectividad de estos métodos en conjunción con dos algoritmos de clasificación. Los resultados subrayan la importancia de estas técnicas de transformación de series temporales como facilitadoras para mejorar los algoritmos de clasificación documentados en la literatura actual.

PALABRAS CLAVE: clasificación, datos biomédicos, redes neuronales convolucionales, series temporales, transformaciones

Corresponding author

TO: **Carlos Alejandro Ku-Maldonado**

INSTITUTION: **Universidad Nacional Autónoma de México**

ADDRESS: **Carretera Mérida-Tetiz Km. 4.5, Ucú, Yucatán, México, C. P. 97357**

EMAIL: **ku.carlos@aries.iimas.unam.mx**

Received:

31 October 2023

Accepted:

07 Dicember 2023

INTRODUCTION

The study of time series classification has gained a great deal of interest due to the large amounts of information generated from the study of various phenomena that evolve in time. This dynamic behavior is observed in the biomedical areas such as disease studies, drug efficacy assessments, treatment analyses, signal processing, and image analysis, etc. where the importance of finding patterns that can help us to segment behaviors arises.

The complexity in time series classification stems from the inadequacy of traditional similarity metrics, which fail to account for the temporal aspect inherent in such data. These datasets frequently exhibit temporal relationships, shifts over time, and variations in magnitude, posing challenges for classification methods reliant on linear comparisons. Certain conventional approaches opt for an initial feature extraction phase to derive pertinent information from the data. However, given these limitations, there's a pressing need to explore alternative representations that remain invariant to shifts, preserve magnitudes, and retain temporal dependencies.

In [1], various techniques for time series classification across different domain representations are explored. These include methods based on dictionary representation, which involves counting the frequency of specific patterns within the data. Another approach involves extracting distinctive shapes, termed as shapelets, present within each class, enabling effective differentiation between groups. Additionally, one analysis strategy involves converting signals into images, leveraging our current computational capabilities to develop more sophisticated models, often relying on neural networks [2][3][4][5][6][7].

This article's contribution lies in its exploration of three distinct methods for representing time series data, specifically applied to biomedical signals. These methods unveil various temporal relationships within the information, potentially enhancing classification performance. The study also delves into analyzing which representation proves most effective for each dataset and classifier, thereby shedding light on the optimal approach for different data contexts and classification algorithms.

In this paper, we aim to review three cutting-edge time series transformation methods, each tested with diverse hyperparameter configurations. These methods will be applied across five distinct biomedical datasets, each possessing unique properties. The objective is to thoroughly evaluate their efficacy in handling classification tasks within these varied biomedical contexts.

MATERIALS AND METHODS

Datasets

The article focuses on five datasets obtained from the UCR Time Series Classification Archive [8], which are widely acknowledged and frequently employed in state-of-the-art methodologies:

A. 'SemgHandGenderCh2' (SHGC2): Captures surface electromyography signal power spectrum data depicting muscle electric activity during six hand grasp movements by five healthy subjects. These movements are categorized into two classes: Class 1 representing Female and Class 2 representing Male.

B. 'PhalangesOutlinesCorrect' (POC): Centers on hand and bone outline detection, categorizing the output label as either a correct or incorrect image outline.

C. 'CinCECGTorso' (CCECGT): Features data derived from ECG chlorine concentration readings across multiple torso-surface sites from four distinct individuals, each individual representing a single class.

D. 'ECG5000': Revolves around ECG heartbeat recordings from a patient with severe congestive heart failure, with class values determined via automated annotation.

E. 'UWaveGestureLibraryAll' (UWGLA): Comprises a collection of eight simple gestures generated from accelerometers.

Further details and characteristics of these datasets are available in Table 1. Notably, these datasets share the common characteristic of being univariate time series data, characterized by ndim = 1.

TABLE 1. **UCR Datasets [8].**

Datasets				
Type	Name	Size	Class	Length
HAR	SemgHandGenderCh2	900	2	1500
Image	PhalangesOutlinesCorrect	2658	2	80
ECG	CinCECGTorso	1420	4	1639
ECG	ECG5000	5000	5	140
HAR	UWaveGestureLibraryAll	9236	8	945

It's interesting to note that these datasets encompass three distinct types of data sources:

Human Activity Recognition (HAR): This category involves datasets like 'SemgHandGenderCh2' (SHGC2) and 'UWaveGestureLibraryAll' (UWGLA), which capture human movements and gestures, often derived from sensors or accelerometers.

Image Contours (Image): 'PhalangesOutlinesCorrect' (POC) falls under this category, focusing on the detection and assessment of image outlines, specifically related to hand and bone outlines.

Electrocardiogram Signals (ECG): Datasets such as 'CinCECGTorso' (CCECGT) and 'ECG5000' involve electrocardiogram data, primarily concerning the recording and analysis of heart-related signals, including chlorine concentration readings and heartbeat patterns.

Transformation methods

RandOm Convolutional KErnel Transform (ROCKET)

Convolutional kernels can be thought of as a matrix of values used to modify the input data by a dot product. These kernels contain certain basic parameters such as length, weights, bias, dilation, and padding. The kernels have the same 1-dimensional vector structure as the input data but are smaller in size. In the case of time series, the kernels are weighted vectors with a bias added to the result of the convolution between the input data and the kernel weights.

These kernels operate as filters, enabling the extraction of diverse shapes and patterns embedded within time series data. Through kernel dilation, these filters can extract patterns across various scales, reflecting different frequency characteristics. By leveraging a combination of multiple kernels, the system can extract complex patterns. While the weights of convolutional neural network kernels are typically learned, random convolutional kernels have demonstrated striking effectiveness.

ROCKET capitalizes on the concept of employing random convolutional kernels as a feature transformation for input in other classifiers. This transformation method offers a low computational cost owing to the random initialization of kernels instead of learned weights. As it involves a single layer, numerous kernels can be generated without significantly escalating the computational demand.

ROCKET employs a mechanism akin to max pooling by extracting maximum values from each feature map. However, it introduces a novel feature: the proportion of positive values. This addition enables ROCKET to discern the prevalence of a distinct pattern within the time series [9]. The process of this method is visually illustrated in Figure 1.

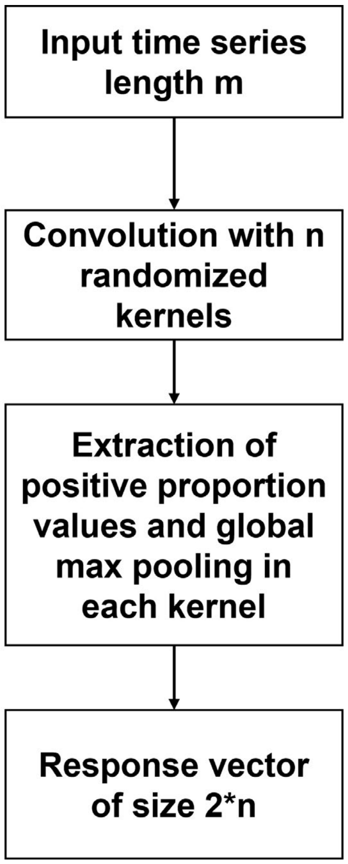
Input time series length m
Convolution with n randomized kernels
Extraction of positive proportion values and global max pooling in each kernel
Response vector of size 2*n

FIGURE 1. Random convolutional kernel transforms flow diagram.

Gramian Angular Field

The Gramian Angular Field (GAF) is based on the dot product operation between two vectors, which shows the similarity that exists between them. If we have two vectors u and v with a norm of 1, we get Equation 1.

$$\langle u, v\rangle = ||u|| \cdot ||v|| \cdot cos(\theta) \tag{1}$$

When working with unit vectors, the dot product is characterized only by the angle θ between *u* and *v*, since the magnitude of a unit vector is 1, which is why the result falls in the range [-1,1].

$$G = \begin{pmatrix} \langle v_1, v_1\rangle & \langle v_1, v_2\rangle & \cdots & \langle v_1, v_n\rangle \\ \langle v_2, v_1\rangle & \langle v_2, v_2\rangle & \cdots & \langle v_2, v_n\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle v_n, v_1\rangle & \langle v_n, v_2\rangle & \cdots & \langle v_n, v_n\rangle \end{pmatrix} \tag{2}$$

Considering the data as unit vectors, the Gramian matrix depicted in Equation 2 is formulated to capture linear dependencies within a vector set. This matrix serves to retain temporal dependencies, embedding the temporal dimension within the geometry of the matrix [10].

Given a time series $X=x_1, x_2, \ldots, x_n$ of *n* real observations, *X* is rescaled so that all values remain in the interval [-1,1] to preserve the unit vector property.

Once the information is scaled, the polar transformation is obtained. In this case, two values are considered:

1. The actual value of the time series (observations).
2. Their respective time labels.

From these values, the angle will be obtained, and this operation is described in Equation 3.

$$\begin{cases} \theta_i = \arccos(x_i) \\ r_i = \frac{i}{N} \end{cases} \tag{3}$$

Some advantages of this coding are that it is a composition of bijective functions and that the temporal dependence is preserved based on the variable r_i. With these new angles, the dot product operation is adapted using Equation 4 so that it can be treated as a Gram matrix.

$$x \oplus y = \cos(\theta_1 + \theta_2) \tag{4}$$

By implementing Equation 4 in G, the following Gramian matrix is obtained.

$$G = \begin{pmatrix} \cos(\theta_1 + \theta_1) & \cos(\theta_1 + \theta_2) & \cdots & \cos(\theta_1 + \theta_n) \\ \cos(\theta_2 + \theta_1) & \cos(\theta_2 + \theta_2) & \cdots & \cos(\theta_2 + \theta_n) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\theta_n + \theta_1) & \cos(\theta_n + \theta_2) & \cdots & \cos(\theta_n + \theta_n) \end{pmatrix} \tag{5}$$

where $\cos(\theta_1+\theta_2)$ is given by Equation 6 in cartesian system.

$$\cos(\theta_1 + \theta_2) = x \cdot y - \sqrt{1-x^2} \cdot \sqrt{1-y^2} \tag{6}$$

This transformation generates a density map, portraying values through a color spectrum with intensities spanning from -1 to 1. Figure 2 illustrates a visual depiction of the encoded magnitudes.

Figure 3 showcases the distinct stages of this method, delineating the process involved in achieving the intended image representation.

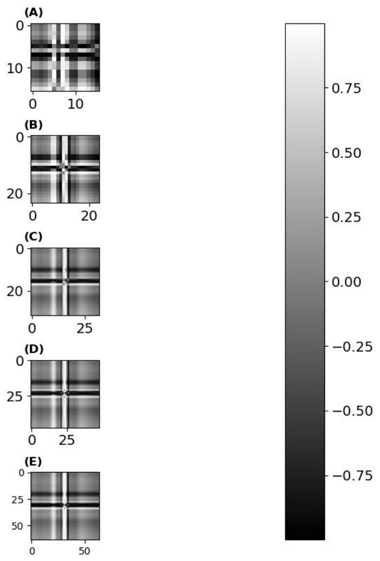

FIGURE 2. Grammian angular field encoding sizes. a) 16X16 image. b) 24X24 image. c) 32X32 image. d) 48X48 image. e) 64x64 image.

Markov transition fields

Markov Transition Fields (MTF) offer an alternative method of transforming a time series into an image, employing a probabilistic framework rooted in Markov chains. This approach provides a distinctive perspective on representing time series data as visual elements.

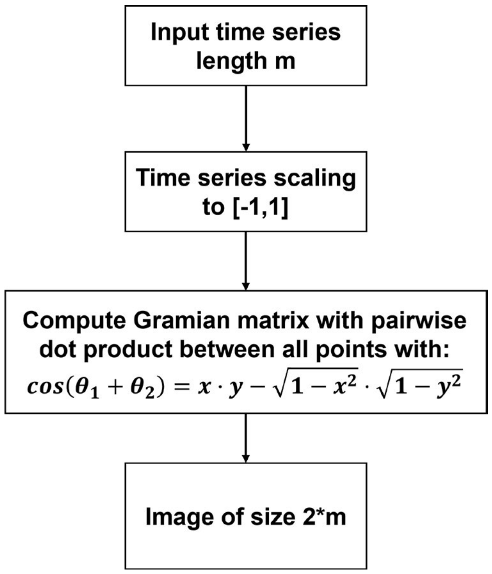
Input time series length m
Time series scaling to [-1,1]
Compute Gramian matrix with pairwise dot product between all points with: $cos(\theta_1+\theta_2) = x \cdot y - \sqrt{1-x^2} \cdot \sqrt{1-y^2}$
Image of size 2*m

FIGURE 3. Gramian angular field flow diagram.

Given a time series $X=x_1,x_2,\ldots,x_n,$ its Q quantile bins are determined and each value x_i is assigned to its corresponding bin q_j ($j\in[1,Q]$). From these bins, a weighted adjacency matrix W of size $Q \times Q$ is constructed, counting the transitions between bins in the form of a Markov chain along the time axis [10].

$$M = \begin{pmatrix} w_{ij}|x_1 \in q_i, x_1 \in q_j & \cdots & w_{ij}|x_1 \in q_i, x_n \in q_j \\ w_{ij}|x_2 \in q_i, x_1 \in q_j & \cdots & w_{ij}|x_2 \in q_i, x_n \in q_j \\ \vdots & \ddots & \vdots \\ w_{ij}|x_n \in q_i, x_1 \in q_j & \cdots & w_{ij}|x_n \in q_i, x_n \in q_j \end{pmatrix} \tag{7}$$

The calculation of w_{ij} is based on the frequency of occurrence where a point in quantile q_j is succeeded by a point in quantile q_i. After deriving this transition matrix, it undergoes normalization process to yield the matrix M. In this transition matrix, M_{ij} represents the probability of transitioning from q_i to q_j. Leveraging these probabilities, we encode the time series into an $N \times N$ matrix, where N corresponds to the length of the original signal.

The color representation of the original signal is depicted through intensities ranging from [0, 1], delineated by the probability distribution. This image illustrates the likelihood of a point transitioning to other points (states) within the time series. Figure 4 showcases the visual encoding derived from this probability distribution.

For a comprehensive view of the method's process, please refer to Figure 5, which details the sequential steps involved in this encoding procedure.

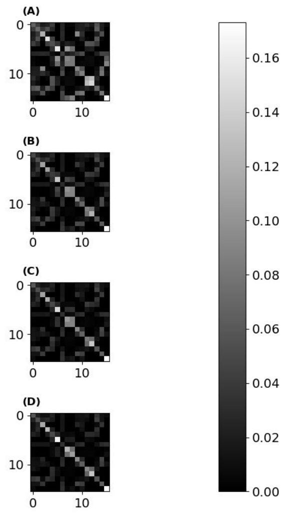

FIGURE 4. **Markov transition fields resolutions. a) 8 bins. b) 16 bins. c) 32 bins. d) 64 bins.**

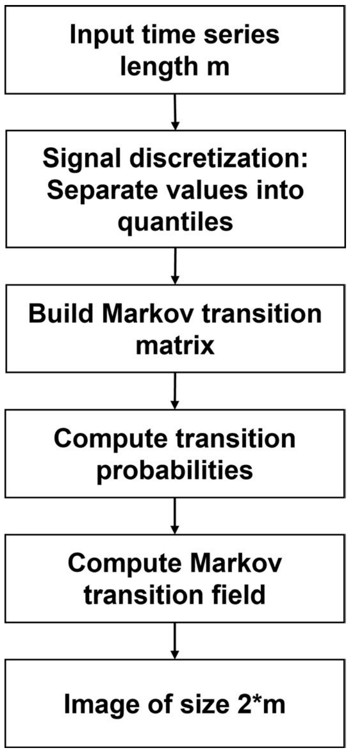
Input time series length m
Signal discretization: Separate values into quantiles
Build Markov transition matrix
Compute transition probabilities
Compute Markov transition field
Image of size 2*m

FIGURE 5. **Markov transition field flow diagram.**

All three of these transform representations are shown in Figure 6 for the same set of time series.

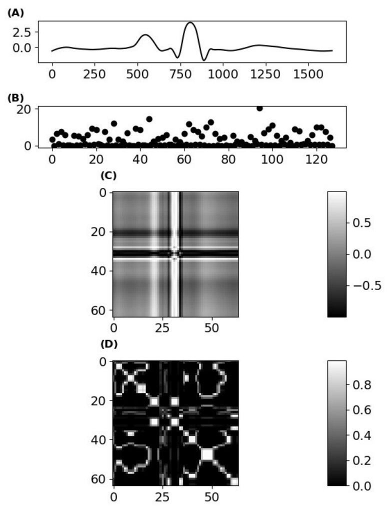

FIGURE 6. **Transformation methods in time series. a) Original time series. b) ROCKET response vector. c) Gramian angular field transformation (GAF). d) Markov transition field transformation (MTF).**

System and software specifications

This methodology was performed in a 11th Gen Intel® Core ™ i7-11800H processor with a NVidia GeForce RTX 3060 mobile GPU and 16 gb of DDR4 Sodimm.

The transformation methods were implemented using Pyts [11], and the classifiers were selected using the Scikit learn and tensorflow-keras packages [12][13].

Pre-processing

Given the common occurrence of artifacts like noise and baseline offsets in biomedical signal data, a preliminary filtering stage was applied to enhance signal quality. To achieve this, a Savitzky-Golay filter was employed across all five datasets. This filter, widely used in biomedical signal processing [14], utilized a window size equivalent to 10 % of the signal length with a polyorder of 2. Notably, this filter retains essential characteristics of the initial distribution, including relative maximums, minimums, and peak widths, resulting in smoother signal behavior.

Additionally, Z normalization was performed to rectify baseline offsets, ensuring a consistent reference across all signal sets. This standardization establishes a zero mean and unit variance across the dataset, a technique known to often enhance model performance.

Once the whole dataset was ready, the training and testing subsets were created. The training set was defined as 70 % of the dataset size, the remaining 30 % was used for testing the classification models. 20 % of the training set was used as validation.

For comparison purpose, all three methods allow to define the output dimensions. On the image-based transformations it is defined by an $N \times N$ image where $N \in \{16,24,32,40,48\}$. In ROCKET the response vector obtained is given by the number of kernels M generated times 2, the number of features extracted in every kernel, this equals to a $2M$ vector where $M \in \{625,1250,2500,5000,10000\}$.

Rocket

Other set of parameters that were random selected along all the kernels were, weights, bias, dilation, padding, the implementation can be seen in [9].

Gramian Angular Fields (GAF)

In GAF other variation parameters is the mode, in this case we have summation and difference mode.

Markov Transition fields (MTF)

For MTF the remaining variable parameter was the number of bins, this number gives us the number of states or resolution the data values can transition to. The number of bins was given by $nBins$ where $nBins \in \{8,16,32,64\}$.

All three transformations—ROCKET, GAF, and MTF—were applied across the five datasets, encompassing various combinations of parameters. These transformed datasets were utilized as inputs for the subsequent classification methods.

Classification methods

The classification stage aimed solely at evaluating transformation performance. The five datasets, transformed using distinct configurations, served as inputs for fixed classifiers to determine the input format that optimizes classification performance. No hyperparameter tuning was conducted on the neural network in this phase.

For the ROCKET-transformed data, a Ridge regressor classifier was utilized. This classifier incorporates a regularization parameter α within the loss function to prevent overfitting. Various values of α, selected from $\alpha \in \{0.001, 0.01, 1, 10, 100, 1000\}$, were tested to identify the best-fitting parameter that avoids overfitting the model. Classifier selection was guided by references [9][10].

In the case of image-based transformations, a convolutional neural network (CNN) with fixed parameters was employed. Categorical encoding was applied to the labels to optimize network functionality. The CNN architecture included the following modules:

1. input layer (image size)
2. convolutional layer (6 neurons, 8x8 size kernels)
3. max pool layer (3x3 pool size, padding=same)
4. convolutional layer (6 neurons, 3x3 size kernels)
5. max pool layer (3x3 pool size, padding=same)
6. flattening layer
7. dropout(0.5)
8. dense layer (categorical enconding = # labels, softmax)

The CNN utilized mean square error as the loss function, employed the adam optimizer, updated batch size weights of 50, and ran for 25 epochs. Figure 7 illustrates the various stages involved in evaluating the enhancement of time series transformations for classification.

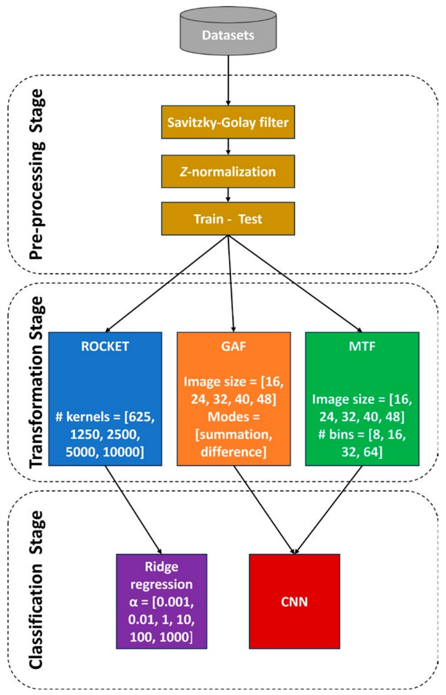
Datasets
Pre-processing Stage
Savitzky-Golay filter
Z-normalization
Train - Test
Transformation Stage
ROCKET
# kernels = [625, 1250, 2500, 5000, 10000]
GAF
Image size = [16, 24, 32, 40, 48]
Modes = [summation, difference]
MTF
Image size = [16, 24, 32, 40, 48]
# bins = [8, 16, 32, 64]
Classification Stage
Ridge regression
α = [0.001, 0.01, 1, 10, 100, 1000]
CNN

FIGURE 7. **Flow diagram of methodology.**

Evaluation metrics

In the evaluation process, both mean square error (MSE) and accuracy were utilized to assess the performance of various input configurations and different parameter settings within the classifier. These metrics were computed on both the training and validation sets. To identify the most effective input configuration, the validation set's lowest MSE was considered.

For assessing test performance, metrics including accuracy, precision, recall, and f1-score were obtained. These measurements offer a comprehensive understanding of the model's predictive capabilities and effectiveness in correctly classifying instances.

RESULTS AND DISCUSSION

The preprocessing stage involved a stratification split to ensure an equal representation of class examples within the training and test sets. However, only the CinCECGTorso and UWaveGestureLibraryAll datasets exhibited an equal number of examples in each class.

Figure 8 demonstrates that ROCKET coupled with ridge regression exhibited notable performance, particularly with datasets where all classification test accuracies surpassed 0.8. One significant advantage observed in employing this method was its low memory resource consumption, as it did not require batch execution. Additionally, it was noted that the regularization parameter tended to perform better when $\alpha \geq 1$. However, the ECG5000 dataset showed poor precision, potentially due to class imbalance issues. It's important to note that the computing time for this method increased with a higher number of kernels, although the fitting process to the classifier was relatively swift once the data was transformed.

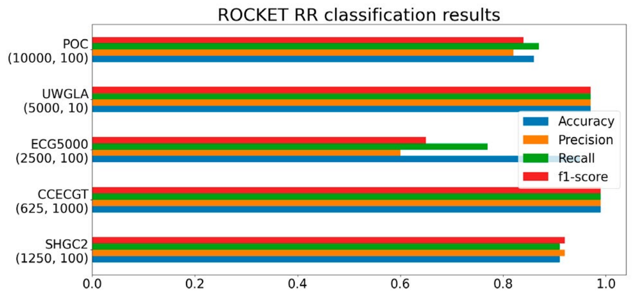

FIGURE 8. **ROCKET transformation performance on the five datasets, number below each dataset correspond to (#kernels, α).**

GAF, similar to ROCKET, showed promising performance on this type of data. However, certain limitations were observed, particularly with respect to memory consumption. Unlike ROCKET, the GAF transformation required batch processing for datasets larger than 1000 samples. Despite this limitation, the transformation process itself was rapid, as were the training times for the classifier.

Regarding the mode selection in GAF, it was noted that the difference mode yielded better results. Additionally, the optimal image size seemed to be 48.

A detailed overview of performance results can be seen in Figure 9.

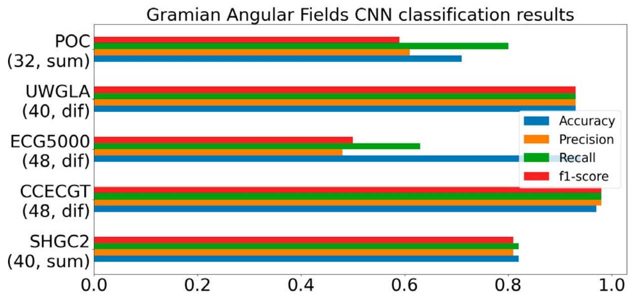

FIGURE 9. **Gramian angular field transformation performance on the five datasets, number below each dataset correspond to (image size, transformation mode).**

Finally, we conducted the MTF classification, yet this transformation didn't yield superior results compared to the preceding two methods. As illustrated in Figure 10, specifically in the precision column of ECG5000, MTF exhibited heightened sensitivity to quantile selection. Signals with low variability necessitated a smaller quantile size; larger quantile sizes led to certain bins containing only a few data points. Similar to the former method, MTF also required transformation into batches for signals exceeding a length of 1000 samples.

The optimal outcomes were achieved with an image size of 48 and # bins set between 8 to 16.

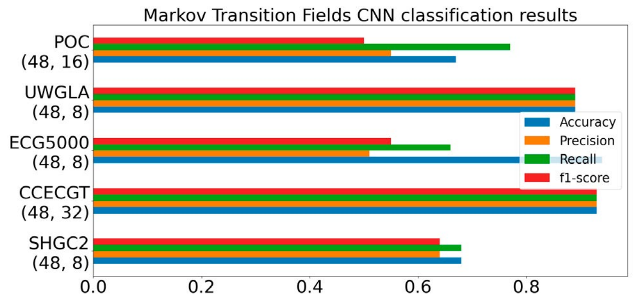

FIGURE 10. **Markov transition field transformation performance on the five datasets, number below each dataset correspond to (image size, # bins).**

In Figure 11 we summarize the F1 score on all datasets given by the optimal parameters for each transformation methods.

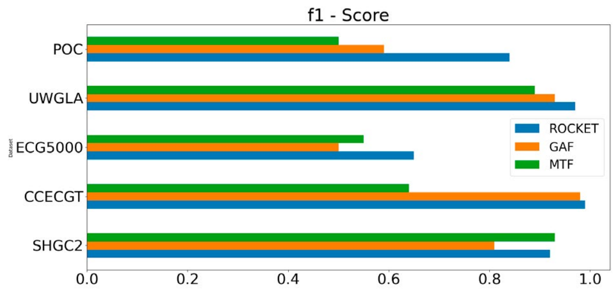

FIGURE 11. **f-1 score of all 3 transformation methods on the five datasets.**

The computation time was systematically measured to analyze how signal size and the number of examples influence the time complexity of these methods. Observations revealed that while ROCKET consistently demonstrated better accuracy across all datasets, its computation time increased with a higher number of filters.

Conversely, GAF and MTF exhibited shorter transformation times, but they tended to consume more memory, scaling the information by N^2 where N represents the signal size. Additionally, MTF encountered challenges when discretizing time series with large bins and limited variability, preferring a smaller number of bins in such cases.

The time metrics were obtained by averaging the time taken for each transformation configuration across each dataset. These computation times are detailed in Table 2.

TABLE 2. **Computation time measurements.**

Dataset mean computation time (seconds)			
Name	ROCKET	GAF	MTF
SemgHandGenderCh2	30.7	1.8	5.7
PhalangesOutlinesCorrect	5.9	2.3	2.4
CinCECGTorso	51.8	2.0	9.7
ECG5000	18.2	2.9	3.4
UWaveGestureLibraryAll	97.2	3.1	18.7

It's evident that these transformation methods serve as pivotal preprocessing steps, significantly enhancing the classification performance of these methods. The mean accuracy across all five datasets demonstrates strong performance: 0.936 for ROCKET, 0.874 for GAF, and 0.822 for MTF, all showcasing accuracies exceeding 0.8. This underscores their effectiveness in improving classification outcomes across diverse datasets.

For an overview of the key features and comparative analysis of each transformation method, please refer to Table 3.

TABLE 3. **Methods observed characteristics.**

Comparison Table of methods		
ROCKET	GAF	MTF
-Presents the highest accuracy of the three representations, with f1 scores over 0.6, even with unbalanced datasets. -The computation time increases linearly with the increase of number of kernels and the time series length. -It's not image based, can be used with more traditional methods. -Due to the random initialization of the kernels, no hyperparameter tunning is required.	-Presents the lowest computation time. -The accuracy obtained is lower than ROCKET, but comparable in some datasets. -Performance may increase with hyperparameter tunning on the NN or implementing other architectures. -Involves more hyperparameter for the transformation stage. -Memory consuming method. -Piecewise aggregate approximation is needed for high length data to avoid memory overflow.	-Presents the lowest accuracy of all three methods. Struggles with unbalanced datasets. -Presents stability issues with high number of bins. -Has a fast computing time when transformation hyperparameters are well suited. - Involves more hyperparameter for the transformation stage. -Memory consuming method. -Piecewise aggregate approximation is needed for high length data to avoid memory overflow.

Depending on the specific application and the tradeoffs you're willing to consider, you might opt for one method over another. Among all three, GAF demonstrated the most balanced performance concerning accuracy and utilization of time resources.

Additionally, these scrutinized time series transformation methods could prove beneficial in various machine learning tasks beyond classification, such as forecasting [15][16][17][18] or clustering [19]. This can be explored in future work.

CONCLUSION

In this study, our primary focus was on transforming time series data to enhance classification performance. We introduced three distinct methods—Random Convolutional Kernel Transform, Gramian Angular Fields, and Markov Transition Fields—and rigorously tested their efficacy using five diverse biomedical datasets. The results illustrate that biomedical signals, when treated as time series data, can achieve classification performances akin to state-of-the-art methodologies [20][21].

While our choice of classifiers was based on existing literature, it's crucial to note their adaptability to various other classifier types and architectures. For instance, in [10], a tiled Convolutional Neural Network was employed, with its last layer serving as input for a Support Vector Machine classifier. Different neural network architectures might yield comparable outcomes.

Notably, these imaging methods exhibit sensitivity to noise and generally perform optimally with signals less than 1000 samples in length due to memory constraints. However, they enable the extraction of diverse features from the data, significantly improving classification performance for temporal datasets.

For large-scale applications, the integration of deep learning computer vision approaches with attention mechanisms could leverage these representations, potentially offering promising outcomes in big data contexts.

REFERENCES

[1] M. Middlehurst, P. Schäfer, and A. Bagnall, "Bake off redux: a review and experimental evaluation of recent time series classification algorithms," 2023, arXiv:2304.13029, doi: https://doi.org/10.48550/arXiv.2304.13029

[2] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," Data Min. Knowl. Discov., vol. 33, no. 4, pp. 917-963, 2019, doi: https://doi.org/10.1007/s10618-019-00619-1

[3] C. Li, J. Xiong, X. Zhu, Q. Zhang, and S. Wang, "Fault Diagnosis Method Based on Encoding Time Series and Convolutional Neural Network," IEEE Access, vol. 8, pp. 165232-165246, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3021007

[4] G. R. Garcia, G. Michau, M. Ducoffe, J. Sen Gupta, and O. Fink, "Temporal signals to images: Monitoring the condition of industrial assets with deep learning image processing algorithms," Proc. Inst. Mech. Eng. O J. Risk Reliab., vol. 236, no. 4, pp. 617-627, 2022, doi: https://doi.org/10.1177/1748006X21994446

[5] J. Lines, S. Taylor, and A. Bagnall, "Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification," in 2016 IEEE 16th international conference on data mining (ICDM), Barcelona, Spain, 2016, pp. 1041-1046, doi: https://doi.org/10.1109/ICDM.2016.0133

[6] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 2017, pp. 1578-1585, doi: https://doi.org/10.1109/IJCNN.2017.7966039

[7] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, et al., "Inceptiontime: Finding alexnet for time series classification," Data Min. Knowl. Discov., vol. 34, no. 6, pp. 1936-1962, 2020, doi: https://doi.org/10.1007/s10618-020-00710-y

[8] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, E. Keogh, "The UCR time series archive," IEEE/CAA J. Autom. Sin., vol. 6, no. 6, pp. 1293-1305, 2019, doi: https://doi.org/10.1109/JAS.2019.1911747

[9] A. Dempster, F. Petitjean, and G. I. Webb, "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels," Data Min. Knowl. Discov., vol. 34, no. 5, pp. 1454-1495, 2020, doi: https://doi.org/10.1007/s10618-020-00701-z

[10] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," 2015, arXiv:1506.00327, 2015, doi: https://doi.org/10.48550/arXiv.1506.00327

[11] J. Faouzi and H. Janati, "pyts: A Python Package for Time Series Classification," J. Mach. Learn. Res., vol. 21, no. 46, pp. 1-6, 2020. [Online]. Available: http://jmlr.org/papers/v21/19-763.html

[12] F. Chollet, "Keras." 2015, GitHub. [Online]. Available: https://github.com/fchollet/keras

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, no. 85, 2825-2830. [Online]. Available: http://www.jmlr.org/papers/v12/pedregosa11a.html

[14] Tanu and D. Kakkar, "Accounting for order-frame length tradeoff of Savitzky-Golay smoothing filters," in 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2018, pp. 805-810, doi: https://doi.org/10.1109/SPIN.2018.8474194

[15] X. Li, Y. Kang, and F. Li, "Forecasting with time series imaging," Expert. Syst. Appl., vol. 160, art. no. 113680, 2020, doi: https://doi.org/10.1016/j.eswa.2020.113680

[16] A. M. Gonzalez-Zapata, L. G. de la Fraga, B. Ovilla-Martinez, E. Tlelo-Cuautle, and I. Cruz-Vega, "Enhanced FPGA implementation of Echo State Networks for chaotic time series prediction," Integration, vol. 92, pp. 48-57, 2023, doi: https://doi.org/10.1016/j.vlsi.2023.05.002

[17] A. Ben Said and A. Erradi, "Deep-Gap: A deep learning framework for forecasting crowdsourcing supply-demand gap based on imaging time series and residual learning," 2019, arXiv:1911.07625, doi: https://doi.org/10.48550/arXiv.1911.07625

[18] W. Ni, C. Zhang, T. Liu, Q. Zeng, L. Xu, and H. Wang, "An efficient astronomical seeing forecasting method by random convolutional Kernel transformation," Eng. Appl. Artif. Intell., vol. 127, art. no. 107259, 2024, doi: https://doi.org/10.1016/j.engappai.2023.107259

[19] J. Marco-Blanco and R. Cuevas, "Time Series Clustering With Random Convolutional Kernels," 2023, arXiv:2305.10457, doi: https://doi.org/10.48550/arXiv.2305.10457

[20] B. Dhariyal, T. Le Nguyen, and G. Ifrim, "Back to Basics: A Sanity Check on Modern Time Series Classification Algorithms," 2023, arXiv:2308.07886, doi: https://doi.org/10.48550/arXiv.2308.07886

[21] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, "HIVE-COTE 2.0: a new meta ensemble for time series classification," Mach. Learn., vol. 110, no. 11-12, pp. 3211-3243, 2021, doi: https://doi.org/10.1007/s10994-021-06057-9