Encoding Methods Comparison for Stress Detection*

Marco Serenelli[1,2], Michela Quadrini[1], Maria Óskarsdóttir[2] and Michele Loreti[1,*,†]

[1]School of Science and Technology, University of Camerino, Via Madonna Delle Carceri 7, 62032 Camerino, MC, Italy
[2]Department of Computer Science, Reykjavik University, Iceland

Abstract

Stress is a prevalent and growing phenomenon in the modern world that could lead to significant physical issues, both physical and mental health. Analyzing physiological signals collected from wearable sensors using artificial intelligence methods has emerged as a promising approach to predicting and managing stress. However, conventional models for time series analysis are RNN architectures and encounter challenges like high computational costs and issues with vanishing or exploding gradients. Inspired by the success of deep learning methods in computer vision, several studies have proposed transforming time series into images by applying encoding time series algorithms.

This work intends to compare three time-series encoding methods: Gramian Angular Field (GAF), both summation and difference, Markovian Transition Field (MTF) and Recurrent Plot (RP) in the stress detection scenario. We employ two architectures, VGG-16 and ResNet, based on Convolutional Neural Network (CNN), to evaluate the performance of these methods on a public dataset, WESAD. Our results demonstrate that the GAF encoding method proves to be the most effective for classifying physiological signals related to stress.

Keywords

Time Series Encoding, Convolutional Neural Network, VGG-16 architecture

1. Introduction

Stress, a nonspecific body response to any demand, can affect physiological health and psychological well-being. Although stress is physiological at a moderate level, chronic stress increases the risk of developing health problems such as insomnia, obesity, heart disease, and cancer [1]. Its effects significantly influence overall behaviour and well-being, potentially impacting personal and professional success. Chronic stress is an increasingly common phenomenon in the modern world. According to the British Health and Safety Executive, work-related stress, depression, or anxiety accounted for 49% of all work-related ill health and 54% of all working days lost due to work-related ill health (19.6 days lost per case) in 2022/23 [2]. The development of robust methods for prompt and accurate detection of human stress plays a significant role in people's quality of life and wellness: managing stress before it becomes a more severe problem is crucial. The most common way exploits psychological assessment questionnaires, like the Perceived Stress Scale [3], able to detect human stress at a specific moment. Therefore, we need reliable, automatic and non-invasive methods to detect stress. Due to the nature of stress, i.e., a physiological response to stimuli triggered by the sympathetic nervous system (SNS), we can exploit physiological signals to monitor the stress responses, as proved by Plarre et al. [4] and Hovsepian et al. [5]. Such an analysis is an emergent and promising approach based on artificial intelligence methods to predict and manage stress. It is favoured by increasing wearable device usage, such as smartphones and smartwatches that permit tracking steps and monitoring other physical activities of their users non-invasively. However, deep learning models for time series analysis, such as RNN architectures, face challenges like high computational costs and issues with vanishing

3rd AIxIA Workshop on Artificial Intelligence For Healthcare and 5th Data4SmartHealth, 25-28 November 2024, Bolzano, Italy
*You can use this document as the template for preparing your publication. We recommend using the latest version of the ceurart style.
*Corresponding author.
†These authors contributed equally.
marco.serenelli@studenti.unicam.it (M. Serenelli); michela.quadrini@unicam.it (M. Quadrini); mariaoskars@ru.is (M. Óskarsdóttir); michele.loreti@unicam.it (M. Loreti)
0000-0003-0539-0290 (M. Quadrini); 0000-0001-5095-5356 (M. Óskarsdóttir); 0000-0003-3061-863X (M. Loreti)

or exploding gradients. Inspired by the success of deep learning methods in computer vision, several studies have proposed transforming time series into images using time series encoding algorithms, such as Gramian Angular Field, Markovian Transition Field and Recurrent Plot. In the scenario of stress detection, Quadrini et al. introduced STREDWES, an approach for Stress Detection that exploits time series encoding methods and convolutional neural network [6, 7]. The time series encoding method in STREDWES is Gramian Angular Field, able to formalizes the temporal correlations among time points of different signals where a measurement is taken.

In this work, we intend to compare three-time series encoding methods: Gramian Angular Field (GAF), both summation and difference, Markovian Transition Field (MTF) and Recurrent Plot (RP) in the stress detection scenario. These methods encodes time series into images avoiding the some issues related to the time series analysis. The most common models for time series data exploit recurrent neural network (RNN) architecture that requires a high computational cost and shows some problems, such as vanishing and exploding gradients. The time-series data properties are defined as 2-dimensional images with colors, dots, and lines at the corresponding positions in the image in which they are transformed and differ from each other. To evaluate the effectiveness of these encoding methods, we exploit architectures based on CNN, such as VGG-16 architecture and ResNet. We conduct such an evaluation on a public dataset, Wearable Stress and Affect Detection (WESAD) [19]. WESAD is a publicly available dataset containing data recorded from a wrist-(Empatica E4) and a chest-worn (RespiBAN) device. Our results demonstrate that the GAF encoding method proves to be the most effective for classifying physiological signals related to stress. By analyzing the computational results, we observe that the GAF encoding method surpasses the other techniques in effectively classifying physiological signals. This superior performance highlights the potential of GAF as a particularly well-suited method for applications in stress detection.

The remainder of this paper is organized as follows. Sect. 2 describes the time series encoding methods, i..e, including Gramian Angular Field, Markovian Transition Field and Recurrent Plot, the neural network used as computation models (CNN and feed-forward neural network), and the metrics used to evaluate the approach. In Sect. 2.1, we describe WESAD, the public dataset in this study. Followed by 3 where we analyzed the obtained results with the introduced approach. The paper ends with some conclusions and future perspectives, Sect. 4.

2. Materials and Methods

In this work, we propose a comparison among time series encoding algorithms by exploiting a pre-trained VGG-16 and ResNet-18, two architecture based on CNN, with a custom fully connected block to detect stress as an image classification task.

2.1. Dataset

WESAD is an open-access multimodal dataset which features lab-sourced data with 15 participants [8]. It uses wearable sensors that record the following physiological signs: blood volume pulse (BVP), ECG, electrodermal activity (EDA), electromyogram (EMG), respiration (RESP), body temperature (TEMP), and three-axis acceleration (ACC). The physiological and motion data is of high quality, sampled at 700 Hz, collected through a chest-mounted (RespiBan) device (ECG, EDA, EMG, RESP, and TEMP), and the ACC signals are sampled at 35 Hz through a wrist-worn device (Empatica E4). The dataset consists of 14 time series, each about two hours long, covering the entire experiment duration. These time series depict three major stimuli: amusement, baseline and stress, collected according to the two versions of protocol shown in Figure 1. According to our aims, in this work, we consider the fragments representing baseline and stress.

Resampling The dataset contains data sampled at 700 Hz and 35 Hz. First, we standardize the sampling step of all signals. The signals sampled at 35 Hz are resampled at 700 Hz by up-sampling the

Version	Block 1		Block 2		Block 3		Block 4		Block 5		Block 6	
A	Baseline	RELAX	Amusement	RELAX	Medi I	RELAX	Stress	RELAX	Rest	RELAX	Medi II	RELAX
B	Baseline		Stress		Rest		Medi I		Amusement		Medi II	

Figure 1: The two versions of the WESAD protocol.

data using linear interpolation. After a resampling, we apply a Butterworth filter [9] to remove any potential high-frequency noise introduced during the interpolation process. We also apply an Hampel filter to eliminate some anomalous peaks that the signals exhibit [10]. This filter uses sliding windows of 1 minute as input, computes the mean μ and standard deviation σ of the values within the interval, and analyzes them using Pearson's rule. Observations that exceed the threshold of 3σ from the mean are considered outliers and replaced with the closest chronological value. This method allows us to replace outliers without introducing high frequencies. After outliers removal, we normalize all signals in the interval $[-1; 1]$ to ensure a consistent scale across all features.

2.2. Sample Construction

2.2.1. Time Series Encoding Methods

Time series encoding methods are techniques that convert the time series into image. Several approaches have been introduced in the literature, including GAF, MTF and RP.

Gramian Angular Field (GAF) The GAF method leverages the polar coordinate system to capture the angular relationships between time series values, providing a unique and informative visual representation of the data. The first step is to make the time series data normalized so that every value falls between -1 and 1. Formally, let $X = \{x_1, x_2, \dots, x_n\}$ be the considered time series with n components. First, the time serie is rescaled by applying the mean normalization:

$$\tilde{x}_j = \frac{(x_i - \max(X)) + ((x_i - \min(X))}{\max(X) - \min(X)} \quad . \tag{1}$$

Then, the scaled series $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n\}$ is transformed to a polar coordinates as follows

$$\begin{cases} \theta_i = \arccos(\tilde{x}_i), & \tilde{x}_i \in \tilde{X} \\ r_i = \frac{t_i}{n}, & \text{with } i \in \{1, \dots, N\} \end{cases} \tag{2}$$

where t_i is the time stamp and n is the number of samples used to regularize the span of the polar coordinate system.

Finally, we compute the GAFs and GAFd by considering the sum and difference, respectively, between the points of the time series

$$\text{GAFs} = \begin{bmatrix} \cos(\theta_1 + \theta_1) & \dots & \cos(\theta_1 + \theta_n) \\ \vdots & \ddots & \vdots \\ \cos(\theta_n + \theta_1) & \dots & \cos(\theta_n + \theta_n) \end{bmatrix} \quad , \tag{3}$$

$$\text{GAFd} = \begin{bmatrix} \sin(\theta_1 - \theta_1) & \dots & \sin(\theta_1 - \theta_n) \\ \vdots & \ddots & \vdots \\ \sin(\theta_n - \theta_1) & \dots & \sin(\theta_n - \theta_n) \end{bmatrix} \quad , \tag{4}$$

Figure 2 shows the GAF-d and GAF-s images of a multivariate time series.

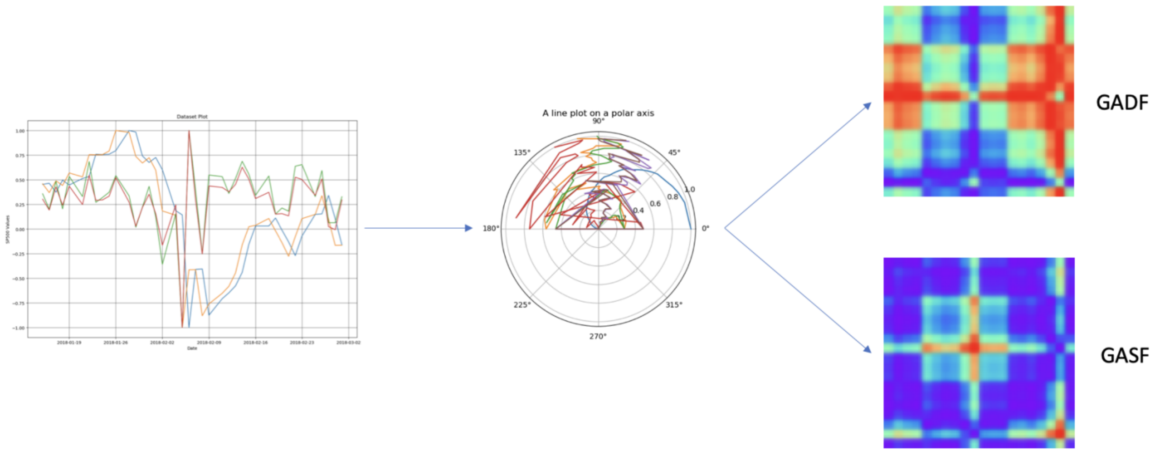

Figure 2: An example of GAF-d and GAF-s images of a multivariate time series.

Markov Transition Field (MTF) The MTF is based on the probabilistic transition between states in a time series, the dynamic behavior and patterns are captured to be transformed into spatial representations to fit an image-based analysis. The first step in the MTF method is quantizing the time series data, by dividing the range of the data into a finite number of discrete states (bins).

Given a time series $X = \{x_1, x_2, \ldots, x_N\}$, it is first discretized into Q quantile units: each value of the time series is assigned a quantile q_j, where $j \in [1, Q]$, and each x_i is mapped to its corresponding. Subsequently, each quantile is mapped into an adjacent weighted matrix W of size $Q \times Q$

$$\mathrm{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,Q} \\ \vdots & \ddots & \vdots & \\ w_{Q,1} & w_{n,2} & \cdots & w_{n,Q} \end{bmatrix} \tag{5}$$

where $w_{i,j} = P(x_t \in q_i | x_{t-1} \in q_j)$ is the frequency of quantile q_i converting to quantile q_j such that $\sum_j w_{i,j} = 1$. Since the Markov transfer matrix disregards the dependency between position and time step, matrix W is augmented with matrix M to incorporate temporal correlations between each quantile and the time step. Formally,

$$\mathrm{W} = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,N} \\ \vdots & \ddots & \vdots & \\ m_{N,1} & m_{N,2} & \cdots & m_{N,N} \end{bmatrix} \tag{6}$$

where $m_{i,j} = P(q_i \to q_j)$ is the transfer probability to move from the quantile q_i to q_j. The MTF image encoding method has the following advantages: (a) according to the relationship between quantile and time step, the temporal correlation of the original signal in different time intervals is retained. (b) The loss of one-dimensional signal information is avoided through the mapping relationship between signals. (c) The magnitude of transfer probability between quantiles is reflected by different colors, which is conducive to making full use of the advantages of CNN in image classification.

Recurrence Plot A recurrence plot (PR) is a framework that encodes a time series into an image as proposed in [11]. The approach formalizes the pairwise Euclidean distances for each value to reveal at which points some trajectories return to a previously visited state. Given a time series $X = \{x_1, x_2, \ldots, x_n\}$, the recurrence states of a point x_i are states x_j that fall into n-dimensional

Figure 3: VGG-16 Architecture

neighborhood of x_i with a given radius ϵ. The Recurrence Plot is defined as

$$\mathrm{R} = \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,N} \\ \vdots & \ddots & \vdots & \\ r_{N,1} & r_{N,2} & \dots & r_{N,N} \end{bmatrix} \tag{7}$$

where

$$r_{i,j} = \begin{cases} 1 & \text{if } \|x_i - x_j\| < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

and $\|\cdot\|$ is a norm and ϵ is the threshold.

2.3. Architectures

We have compared three time-series encoding methods of two different architectures, VGG-16 and ResNet, in the scenario of stress detection. It aims to understand which time series encoding method is most effective in detecting affect states. GAF-s, GAF-d, MTF and RP are two-dimensional matrices. Therefore, we can treat them as images and face the problem of stress detection as image classification by exploiting deep learning techniques, like convolutional neural networks (CNN), a kind of network that plays a fundamental role in various computer vision tasks. CNNs, able to analyze spatial information without requiring hand-crafted feature extraction, consist of multiple building blocks, such as an input layer, convolution layers, pooling layers, and fully connected layers. In the past decade, several CNN architectures have been introduced in the literature to improve performance across diverse applications, including the Visual Geometry Group 16 (VGG-16) [12] and layer residual nets (ResNets) [13].

VGG-16 VGG is an architecture based on the idea of reducing filter sizes and increasing the network depth. VGG-16 consist of 16 trainable layers connected in a feed-forward fashion, of which 13 layers are convolutional. The network has a small receptive field of 3×3. It has a Max pooling layer of size 2×2 and has a total of 5 such layers. These layers are organized into blocks, with each block containing multiple convolutional layers followed by a max-pooling layer for downsampling. Figure 3 shows a schema of the architecture.

ResNet-18 ResNet is a residual learning method to train deeper networks that are practically difficult to train [13]. The residual network layers, reformulated to learn residual functions relative to the layer inputs, solve the problem of accuracy degradation in deeper networks. In residual networks, the ResNet layers are stacked to learn a residual mapping, different from plain networks that stack together several layers to learn the mapping directly. The mapping function, denoted by H(x), features a few stacked layers. Residual learning exploits the idea that if several nonlinear layers can asymptotically estimate a complicated mapping function, then they can asymptotically estimate the residual function denoted

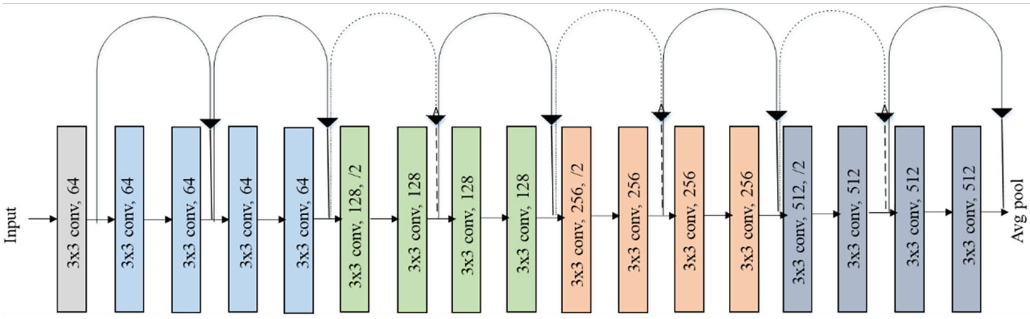

Figure 4: ResNet-18 Architecture

as F(x) that can be expressed as $F(x) = H(x) - x$, where $H(x)$ is the original function. This method assumes that the residual mapping function is easier to optimize than the original function. Figure 4 shows the ResNet-18 architecture. It consists of eighteen layers in the network, among which 17 are convolutional layers. Such convolutional layers use 3×3 filters, and the network is designed so that layers producing output feature maps of the same size have the same number of filters. However, when the output feature map size is halved, the number of filters is doubled in the layers. Convolutional layers with a stride of 2 perform the downsampling. Throughout the network, residual link connections are inserted between the layers. There are two types of connections, denoted by solid lines or dotted lines in Figure 4. The former is used when input and output have the same dimensions, while the latter is used when dimensions increase.

The two architectures, VGG-16 and ResNet-18, show a final fully connected block. In our models, we define such a block with only one layer with dropout.

2.4. Evaluation Metrics

Stess detection can be formulated as a binary classification problem where each image can represent stress or non-stress. We evaluate the performance of our approach and compare it with one of some other methods in the literature using six evaluation metrics: Accuracy (Acc), Precision (P), Recall (R), F-measure ($F1$), the area under the receiver operator characteristic curve ($AUC - ROC$), Avg. precision score (PR)

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}$$

where TP represents the number of samples representing stress identified correctly (true positive), FN denotes the number of images of stress identified incorrectly (false negative), FP represents the number of stress identified incorrectly (false positive), TN denotes the number of stress identified correctly (true negative).

Table 1
Description of the employed hyperparameters with the relative used values.

Hyperparameter	Description	Value
batch size	batch dimension	32
image size	the dimension of images	224x224
learning rate	learning rate of the optimization algorithm	0.0012
optimizer	optimization algorithms	Adam
time step	the step of the sliding window	1 second
window size	the dimension of the sliding window	60 seconds

3. Results

3.1. Dataset Entries

The entries of our model consist of images encoded from multivariate time series fragments stored in WESAD. To encode the signals into images, we consider several variables, such as the frequency to resample the time series, time window length, the step for the slide of the window, and the image size. By taking inspiration from the work proposed in [7], and after multiple iterations, we set up the values of the frequency to 700Hz, the window length to 1 minute and the step to 1 second, along with an image size of 224x224 to match the requirements of the ResNet and VGG network. We realized, that if considered as a binary classification problem, the dataset was somewhat unbalanced, 65% for Baseline and 35% for Stress, with a bias towards the baseline class. To address this imbalance, we applied Synthetic Minority Over-sampling Technique (SMOTE) [14] to upsample the minority class (stress).

3.2. Implementation

We implemented our approach in Python by exploiting the PyTorch package [15]. For obtaining the GAFd, GAFs and MTF images, we leveraged the *pyts* package [16]. The activation function used is the PreLU for all layers except the last one that uses the softmax function. Adam is the optimization algorithm with the learning rate set as a definable parameter. The used hyperparameters equipped with their description and the used values are listed in Table 1.

All experiments were performed on a machine with the following characteristics, RAM 32 GB, CPU Ryzen 7 3700X, and GPU NVIDIA RTX 3090. The code used to develop this project is available at https://github.com/marcoserenelli/Encoding-Methods-Comparison-for-Stress-Detection

3.3. Computation Results

We can now compare the effectiveness of the four time-series encoding methods in the stress detection scenario. For this reason, we consider two different architectures, ResNet-18f and VGG-16. Note that, in our approach, the last blocks of the two architectures are costumed.

The Table 2 show the computational results, evaluated by considering the standard metrics. The

Table 2
Performance of ResNet-18 and VGG-16 using different encoding methods

CNN Architecture	Encoding Methods	Accuracy	Precision	Recall	Weighted F1-score
ResNe-18t	GAF-s	0.9039	0.9033	0.9039	0.904
	GAF-d	**0.9392**	**0.9210**	**0.920**	**0.920**
	MTF	0.890	0.884	0.92	0.904
	RP	0.590	0.461	1.0	0.631
VGG-16	**GAF-s**	**0.958**	**0.932**	**0.933**	**0.9317**
	GAF-d	0.944	0.932	0.920	0.9262
	MTF	0.75	0.885	0.701	0.782
	RP	0.72	0.857	0.660	0.746

ResNet behavior changes variably for different encoding methods, achieving maximum accuracy and F1 when using GAF-d at 93.92% and 92.0%, respectively. The RP approach performs much worse, its accuracy is only 59.0%, while the recall is 100%, meaning that it manages to detect every single relevant case correctly but produces many false positives Recall and F1-measure are a bit lower with VGG-16, maintaining an accuracy of 95.8% using the encoding GAF-s. Both RP and MTF encodings of VGG-16 show reduced performance in accuracy and F1-measure compared to the GAF encodings. This information indicates that the choice of the encoding method plays a vital role in obtaining improved performance from CNN architectures, with GAF-s and GAF-d generally performing better.

Finally, we also compare the performance of our approach with others based on classical machine learning algorithms [8] and the multimodal-multisensory sequential fusion model (MMSF) [17] proposed in the literature. MMSF is a model based on a multimodal fusion scheme for merging heterogeneous signals [18]. The used fusion scheme is 'late fusion'. It trains multiple submodels individually and combines the obtained outputs as input for a final supervised classifier, different from the approaches based on time series encoding methods that first aggregate the heterogeneous signals and map them into an image, avoiding the overfitting problem related to the signal length. Table 3 shows the accuracy and F1 score of each approach.

Table 3
Accuracy and F1-score of the considered approaches.

Method	**Accuracy** (%)	**Weighted F1-score** (%)
k-nearest	56.14	48.70
Decision Tree	63.56	58.05
Random Forest	74.97	64.08
AdaBoost	79.57	68.85
Linear discriminant analysis	75.80	71.56
MMSF	85.00	86.00
STREDWES	91.11	**90.40**

By analyzing the results, we can observe that both VGG-16 with GAF-s and ResNet with GAF-d outperform all the classical baseline methods and the MMSF model, in terms of accuracy and F1-score. This highlights the effectiveness of advanced CNN architectures in capturing complex patterns in the data, likely benefiting from deeper layers and more advanced feature extraction capabilities. In many cases, particularly with the GAF-s and GAF-d encodings, it is shown that the CNN can reach higher metric values, indicating that these methods are a solid way to preprocess and present data to networks. Even if the MMSF model outperforms the classical machine learning models, it still cannot catch up with the results achieved by the CNN architectures indicating that while the multimodal approaches are formidable, further improvement can be achieved in the specific designs of the CNNs with optimal encoding techniques. The comparison shows that CNN architectures, particularly when paired with effective encoding methods like GAF-s and GAF-d, offer superior performance over both classical machine learning methods and newer models like MMSF, underlining the importance of choosing the right architectural and preprocessing strategies to maximize the efficacy of machine learning models in complex classification tasks.

4. Conclusion and Future Work

Developing robust and non-invasive methods to detect stress plays a fundamental role in everyday life since stress is a significant and growing phenomenon that can lead to numerous health problems. Detection and managing stress can greatly improve people's quality of life and overall well-being. The social acceptance of smartphones and wearable sensors, able to track physiological parameters and monitor the physical activities of users, allows us to establish reliable non-invasive methods for stress detection. In this work, we compared the GAF-s, GAF-d, MTF and RP time series encoding techniques to determine which one is most effective for stress detection. Since GAF-s, GAF-d, MTF and RP are two-dimensional matrices, we can treat them as images and face the problem of stress detection as image classification by two architectures based on CNN, i.e., VGG-16 and ResNet-18. By analyzing the results, we observe that the GAF (Gramian Angular Field) encoding method consistently performs best for encoding physiological signals as images across both networks. Specifically, for ResNet-18, the GAF-s method excels, while for VGG-16, the GAF-d method shows better performance. The approach showed promising results in controlled laboratory environments. However, their application in real-time scenarios poses several challenges, such as high computational costs on mobile devices or the variability in environmental factors that may reduce the accuracy of the results.

In the future, we plan to include other time-series encoding methods and to study the limitations of such approaches. Moreover, we intend to investigate the approach for obtaining the sliding. An approach to implement sliding windows could follow the methods proposed in [6, 7] or exploit techniques based on entropy like [19]. Another interesting feature direction is to convert time series into a graph under a geometric principle of visibility or a simplicial complex. These conversions allow us to exploit other deep-learning architectures, like Graph Neural Networks, to develop approaches similar to the ones proposed for other fields [20], or methods based on logic frameworks to verify propertiess [21]. These data structures enables the application of innovative approaches, exploiting the concept that graph neural networks can be programmed to achieve explainable results [22, 23].

References

[1] G. S. Everly, Jr, J. M. Lating, G. S. Everly, J. M. Lating, The anatomy and physiology of the human stress response, A clinical guide to the treatment of the human stress response (2019) 19–56.

[2] Health and Safety Executive, HSE on work-related stress. 2021, http://www.hse.gov.uk/statistics/causdis/-ffstress/index.htm, ???? Accessed on March 7, 2022.

[3] E.-H. Lee, Review of the psychometric evidence of the perceived stress scale, Asian nursing research 6 (2012) 121–127.

[4] K. Plarre, A. Raij, S. M. Hossain, A. A. Ali, M. Nakajima, M. Al'Absi, E. Ertin, T. Kamarck, S. Kumar, M. Scott, et al., Continuous inference of psychological stress from sensory measurements collected in the natural environment, in: Proceedings of the 10th ACM/IEEE international conference on information processing in sensor networks, IEEE, 2011, pp. 97–108.

[5] K. Hovsepian, M. Al'Absi, E. Ertin, T. Kamarck, M. Nakajima, S. Kumar, cstress: towards a gold standard for continuous stress assessment in the mobile environment, in: Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing, 2015, pp. 493–504.

[6] M. Quadrini, S. Daberdaku, A. Blanda, A. Capuccio, L. Bellanova, G. Gerard, Stress detection from wearable sensor data using gramian angular fields and cnn, in: International Conference on Discovery Science, Springer, 2022, pp. 173–183.

[7] M. Quadrini, A. Capuccio, D. Falcone, S. Daberdaku, A. Blanda, L. Bellanova, G. Gerard, Stress detection with encoding physiological signals and convolutional neural network, Machine Learning (2024) 1–29.

[8] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, K. Van Laerhoven, Introducing WESAD, a multimodal dataset for wearable stress and affect detection, in: Proceedings of the 20th ACM international conference on multimodal interaction, 2018, pp. 400–408.

[9] I. W. Selesnick, C. S. Burrus, Generalized digital butterworth filter design, IEEE Transactions on signal processing 46 (1998) 1688–1694.

[10] R. K. Pearson, Y. Neuvo, J. Astola, M. Gabbouj, Generalized hampel filters, EURASIP Journal on Advances in Signal Processing 2016 (2016) 1–18.

[11] E. Jp, Recurrence plots of dynamical systems, Europhysics Ltters 5 (1987) 973–977.

[12] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[13] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.
[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).
[16] J. Faouzi, H. Janati, pyts: A python package for time series classification., J. Mach. Learn. Res. 21 (2020) 46–1.
[17] J. Lin, S. Pan, C. S. Lee, S. Oviatt, An explainable deep fusion network for affect recognition using physiological signals, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2069–2072.
[18] S. Oviatt, B. Schuller, P. R. Cohen, D. Sonntag, G. Potamianos, A. Krüger, The handbook of multimodal-multisensor interfaces: Signal processing, architectures, and detection of cognition and emotion, 2018.
[19] M. Quadrini, M. Cavallin, S. Daberdaku, C. Ferrari, Prosps: protein sites prediction based on sequence fragments, in: International Conference on Machine Learning, Optimization, and Data Science, Springer, 2021, pp. 568–580.
[20] M. Quadrini, S. Daberdaku, C. Ferrari, Hierarchical representation for ppi sites prediction, BMC bioinformatics 23 (2022) 1–34.
[21] M. Loreti, M. Quadrini, A spatial logic for simplicial models, Logical Methods in Computer Science 19 (2023).
[22] M. Belenchia, F. Corradini, M. Quadrini, M. Loreti, Implementing a ctl model checker with μ g, a language for programming graph neural networks, in: International Conference on Formal Techniques for Distributed Objects, Components, and Systems, Springer, 2023, pp. 37–54.
[23] M. Belenchia, F. Corradini, M. Quadrini, M. Loreti, libmg: a python library for programming graph neural networks in μg, Science of Computer Programming (2024) 103165.