



# Stress detection with encoding physiological signals and convolutional neural network

Michela Quadrini<sup>1</sup> · Antonino Capuccio<sup>2</sup> · Denise Falcone<sup>1</sup> · Sebastian Daberdaku<sup>2</sup> · Alessandro Blanda<sup>2</sup> · Luca Bellanova<sup>2</sup> · Gianluca Gerard<sup>2</sup>

Received: 17 March 2023 / Revised: 18 October 2023 / Accepted: 16 December 2023 /

Published online: 15 March 2024

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2024

## Abstract

Stress is a significant and growing phenomenon in the modern world that leads to numerous health problems. Robust and non-invasive method developments for early and accurate stress detection are crucial in enhancing people's quality of life. Previous researches show that using machine learning approaches on physiological signals is a reliable stress predictor by achieving significant results. However, it requires determining features by hand. Such a selection is a challenge in this context since stress determines nonspecific human responses. This work overcomes such limitations by considering STREDWES, an approach for Stress Detection from Wearable Sensors Data. STREDWES encodes signal fragments of physiological signals into images and classifies them by a Convolutional Neural Network (CNN). This study aims to study several encoding methods, including the Gramian Angular Summation/Difference Field method and Markov Transition Field, to evaluate the best way to encode signals into images in this domain. Such a study is performed on the NEURO dataset. Moreover, we investigate the usefulness of STREDWES in real scenarios by considering the SWELL dataset and a personalized approach. Finally, we compare the proposed approach with its competitors by considering the WESAD dataset. It outperforms the others.

**Keywords** Convolutional neural network · Time series encoding · Physiological signals · non-intrusive method · Stress detection

## 1 Introduction

Stress is a nonspecific body response to any demand. Its effects and symptoms impact humans from both physical and psychological points of view, playing a significant role in overall behaviour, wellness, and potentially personal and professional successes (McEwen, 1998). Although stress is physiological and normal at some level, chronic stress increases the risk of developing health problems such as insomnia, obesity, heart disease, and cancer. Many studies have also revealed a link between stress and mental disorders, including anxiety disorders,

---

Editors: Dino Ienco, Roberto Interdonato, Pascal Poncelet.

Extended author information available on the last page of the article

depression, and cancer. Chronic stress is a common and growing phenomenon in the modern world that affects more and more individuals. According to the British Health and Safety Executive, stress accounted for 50% of all work-related ill health cases in 2020/21. This rate shows an increment of 13% since 2015/16 (Health and Safety Executive, 2021). The American Institute of Stress states that 80% of workers experience stress in their daily work and need help managing it (The American Institute of Stress, 2023). In this scenario, it is evident that the development of robust methods for prompt and accurate detection of human stress plays a significant role in people's life quality and wellness: managing stress before it turns into a more severe problem is crucial. Psychological assessment questionnaires, like the Perceived Stress Scale (Lee, 2012), are commonly used to determine human stress, although they can only capture stress at a specific moment. However, these techniques could be unreliable: asking someone how they feel is likely to obtain skewed feedback (Li & Liu, 2020). Moreover, they often contain questions that could be psychologically invasive. Therefore, the challenge is to define an approach to detect stress reliably, automatically and non-invasively. The Internet of Things (IoT) provides sensors and smart objects to capture various physiological features suitable for monitoring user's health. The recent increase in the smartphones and wearable devices use, such as smartwatches and bands, and their social acceptance permits tracking physiological parameters and monitoring the physical activities of users non-invasively. These devices have several built-in sensors able to monitor blood volume pulse (BVP), electrodermal activity (EDA), temperature (TEMP), accelerometer, heart rate (HR), heart rate variability (HRV) etc. By analyzing these signals, researchers have developed many reliable methods (Lundberg et al., 1994). Healey and Picard performed one of the first studies based on physiological signals to detect human stress (Healey & Picard, 2005). Recent studies have shown that machine learning methodologies achieve promising results (Giannakakis et al., 2019). However, the methods based on classical machine learning techniques require manual feature engineering. Finding the right features is a difficult task in this context since the physiological changes of each person are different from one another. Deep learning-based approaches overcome these limitations. Moreover, combining several biosignals and physiological signals is fundamental to defining a robust and reliable method. Recent studies demonstrated that encoding time series to images helps highlight, capture, or compress local patterns that would be dispersed over some time since they are able to capture temporal correlation between each time point. In the literature, we noticed some promising encoding algorithms: the Gramian Angular Summation Field (GAFs), the Gramian Angular Difference Field (GAFd), the Markov Transition Field (MTF) (Wang & Oates, 2015), and the recurrence plot (Marwan et al., 2007), grey-scale encoding (Xu et al., 2019), in addition to the spectrogram and the scalogram, the traditional visualization tools used in signal processing (Verstraete et al., 2017). The signal encoding into images provides competitive results in supervised setups (Sasirekha & Thangavel, 2020) and for anomaly detection tasks (Garcia et al., 2022). Nevertheless, to the best of our knowledge, the signal set encoding into a single image has not been deeply investigated yet for improving stress detection. Our preliminary result shows that the performance of our approach based on GAFs outperforms other state-of-the-art competitors (Quadrini et al., 2022).

This paper studies different encoding methods to evaluate the best way to encode signals into images in the stress detection domain. Such a study is based on STREDWES, a pipeline based on time-series images encoding and Convolutional Neural Network (CNN) as a classifier introduced in Quadrini et al. (2022). After signal preprocessing consisting of resampling, outlier removal, and normalization, STREDWES extracts signal fragments using sliding windows and encodes the signals into images. To perform the study, we apply our method to the NEURO dataset, containing 7 biophysical signals of 20 subjects. To select the best encoding method and obtain a robust performance of our approach (as independent as possible from

the dataset), we study the proposed approach with another dataset, NEURO (Birjandtalab et al., 2016), by considering the GAFs, GAFd and MTF as encoding algorithms and the leave-one-subject-out cross-validation (LOSOCV) technique. According to the results obtained in the literature, the analysis of the performances shows that the three encoding algorithms in terms of accuracy, F1 score, precision and recall are comparable. However, the performance of the GAFs encoding method show a slightly higher value of F1 score and it is more efficient than MTF. Moreover, we also investigate the performance of our method in real scenarios. To explore this direction, we propose a “personalized” approach: the data entries for training and testing are fragments of biosignals of the same subject. In this way, the model can learn the personal physiological changes under stress conditions, although such a model requires a training step for each subject. To apply the approach, we take into account the SWELL knowledge work (SWELL-KW) dataset (Koldijk et al., 2014) by evaluating only the three recorded physiological signals. Moreover, such data are noisy. The obtained performances are significant and encourage some future developments. Finally, to compare the approach with the literature in the previous research, we consider Wearable Stress and Affect Detection (WESAD) dataset (Schmidt et al., 2018). WESAD contains physiological, such as electroencephalography (EEG), electrocardiography (ECG), electrodermal activity (EDA), electromyogram (EMG), and physical measures, such as respiratory rate (RESP) and skin temperature (TEMP). The performances of the proposed approach on WESAD have been compared with other classical machine learning algorithms (Schmidt et al., 2018) and with the multimodal-multisensory sequential fusion model (MMSF), an approach based on CNN (Lin et al., 2019). In the literature, some methods have been proposed to encode signals into images to capture the spatial and temporal correlations among time series observations that have been applied in other scenarios (Garcia et al., 2022). The best accuracy on WESAD that consists of 13 signals is 91.11 whereas the best accuracy on NEURO that records 7 ones is 73.34.

The paper is organized as follows. Section 2 summarizes the works in the literature regarding the methods proposed to detect stress. Section 3 describes the three datasets, WESAD, NEURO and SWELL-KW, used in this study to evaluate our approach, STREDWES, presented in Sect. 4. The experiments will be shown in Sect. 5. The paper ends with some conclusions and a description of future directions in this scenario, Sect. 6.

## 2 Related work

In recent years, stress prediction and detection have been addressed by considering physiological and physical signals. In the literature, some works focused on extracting hand-crafted features from physiological signals like electroencephalograph, ECG, EMG, and EDA, and physical signals like respiration and temperature. Giannakakis et al. summarised such works by underling the main characteristics (Giannakakis et al., 2019). ECG and EDA are the biosignals most frequently used for stress monitoring. EDA has two main components: skin conductance level (SCL) and skin conductance response (SCR). The features extracted from ECGs include QRS curves, HR and HRV.

Thanks to the expanding of wearable devices and smartphones, new datasets containing a high number of signals have become available to the research community, including WESAD (Schmidt et al., 2018), NEURO (Birjandtalab et al., 2016), SWELL-KW (Koldijk et al., 2014), UBFC (Sabour et al., 2021), CASE (Sharma et al., 2019), and MUSE (Jaiswal et al., 2020). Physiological measures such as EDA, HR and HRV captured by wearable devices are frequently used in studies related to affection and well-being (Sano & Picard,

2013). In the literature, machine learning approaches achieved significant performances detecting stress by combining several signals (Schmidt et al., 2018; Gjoreski et al., 2017), considering Support Vector Machine, Random Forest, Decision Trees, Regression Strategies, Boost algorithms, and K-Nearest Neighbors. Gjoreski et al. proposed an approach that extracts 63 features from the considered physiological signals, i.e., ACC, BVP, electrodermal activity, heart rate, and skin temperature sensors, and the used random forest algorithm for classification (Gjoreski et al., 2017). Such features are usually hardly determined. Deep learning algorithms learn and combine relevant features from several biosignals by achieving better performance. Souza et al. propose a model that pre-processes the physiological data through a novel pipeline using a recurrent neural network (de Souza et al., 2022). Rastgoo et al. introduced a method based on CNN and long short-term memory networks (Rastgoo et al., 2019). Li and Liu proposed an approach based on 1D CNN, developed as modified 2D CNNs, and a multilayer perceptron neural network for psychophysiological stress detection (Li & Liu, 2020). Oskooei et al. proposed Deep ECGNet, a method based on optimal recurrent and CNN applied to the waveform characteristics of ECG signals (Oskooei et al., 2021).

### 3 Datasets

In this section, we describe the three public datasets, WESAD, NEURO and SWELL-KW, used in this study. All of them contain multivariate time series of signals.

#### 3.1 WESAD

WESAD is a public multimodal high-quality dataset for stress and effects detection. The participants are graduate students at the research facility where the experiments were conducted. All the subjects of the experiment are not heavy smokers, and did not suffer from mental illness, chronic and cardiovascular diseases. Furthermore, the girls were not pregnant. The dataset contains data recorded during a lab study that involves 15 subjects. This dataset, composed of 14-time series of about 2 hours each, includes physiological (ECG, EDA, EMG, RESP, and TEMP) and three-axis acceleration (ACC) modalities sampled at 700 Hz from a chest-worn device (RespiBan) and lower-resolution data from a wrist-worn device (Empatica E4). The Empatica E4 records BVP (sampled at 64 Hz), EDA (at 4 Hz), TEMP (at 4 Hz), and ACC (at 32 Hz). The experiments aimed to record three different affective states (baseline, stress, amusement) to collect data following a particular protocol, whose phases are the Transition, Baseline, Stress, Amusement, Rest, and Meditation. Experiments are based on two protocols: stress and amusement phases are interchanged between subjects to avoid the possible effects of order. Figure 1 shows the two versions. They take into account the Trier Social Stress Test (TSST), which consists of a public speaking and a mental arithmetic task since these tasks elicit stress reliably and a high psychological load on the subjects (Kirschbaum et al., 1993). The study involved participants delivering a five-minute speech about their personal traits focusing their attention on strengths and weaknesses. The participants should make a positive impression to enhance their career prospects. They had three minutes to prepare for the speech without using any notes during the presentation. After the speech, the panel instructed the participants to sequentially count backward from the year 2023 to zero, using intervals of 17. Additionally, any errors made by the participants required them to restart the counting process.

| Version | Block 1  |                       | Block 2   |                       | Block 3 |                       | Block 4 |                       | Block 5   |                       | Block 6 |                       |
|---------|----------|-----------------------|-----------|-----------------------|---------|-----------------------|---------|-----------------------|-----------|-----------------------|---------|-----------------------|
| A       | Baseline | R<br>E<br>L<br>A<br>X | Amusement | R<br>E<br>L<br>A<br>X | Medi I  | R<br>E<br>L<br>A<br>X | Stress  | R<br>E<br>L<br>A<br>X | Rest      | R<br>E<br>L<br>A<br>X | Medi II | R<br>E<br>L<br>A<br>X |
| B       | Baseline | A<br>X                | Stress    | A<br>X                | Rest    | A<br>X                | Medi I  | A<br>X                | Amusement | A<br>X                | Medi II | A<br>X                |

**Fig. 1** The two versions of the WESAD protocol

| Version |                       | Block 1            |                       | Block 2             |                       | Block 3             |                       |
|---------|-----------------------|--------------------|-----------------------|---------------------|-----------------------|---------------------|-----------------------|
|         | R<br>E<br>L<br>A<br>X | Physical<br>Stress | R<br>E<br>L<br>A<br>X | Emotional<br>Stress | R<br>E<br>L<br>A<br>X | Cognitive<br>Stress | R<br>E<br>L<br>A<br>X |

**Fig. 2** The NEURO protocol to capture signals

After TSST, participants relax for 10 min. The other study faces, i.e., Baseline, Amusement, and Meditation last 20 min, 392 s and seven minutes, respectively.

### 3.2 NEURO

NEURO is a public database of non-EEG physiological signals. It includes EDA, TEMP, ACC, HR, and arterial oxygen level (SpO2) signals of 20 healthy subjects (14 males and 6 females) for about 40 min each. The signals are recorded by two non-invasive sensors, Affectiva and Nonin 3150. The former acquires ACC and EDA with a sampling frequency of 8 Hz, while the latter registers the HR and SpO2 by sampling at 1 Hz. The data was collected at the University of Texas at Dallas following a particular experimental procedure shown in Fig. 2. Participants of the experiment performed the following tasks: First Relaxation, Physical Stress, second relaxation, Cognitive Stress, Third Relaxation, Emotional Stress and fourth relaxation. The Physical Stress was generated by a one-minute period of standing, followed by two minutes of walking on a treadmill set to a speed of one mile per hour, and concluding with two minutes of walking or jogging on the treadmill at a speed of three miles per hour. Cognitive stress was induced by performing a three-minute task of counting backwards from 2485 in intervals of seven, followed by reading colour names written in the ink of different colours and identifying the ink colour used. In both parts, the volunteers were alerted to errors by a buzzer. Emotional Stress was generated by a five-minute horror movie screening followed by a clip from a movie depicting a zombie apocalypse. The four relaxation parts last 5 min each. In these phases, the participant listened to quiet and soothing music.

### 3.3 SWELL-KW

SWELL-KW is a public dataset for stress detection. It contains data recorded during an experiment that involves 25 people performing typical office work, such as writing reports,

| Version | Block 1 |                       | Block 2                |                       | Block 3                |
|---------|---------|-----------------------|------------------------|-----------------------|------------------------|
| A       | Neutral | R<br>E<br>L<br>A<br>X | Stressor interruptions | R<br>E<br>L<br>A<br>X | Stressor time pressure |
| B       | Neutral |                       | Stressor time pressure |                       | Stressor interruptions |

**Fig. 3** The two versions of the SWELL-KW protocol

making presentations and reading e-mails, by following a particular experimental procedure shown in Fig. 3. The dataset comprises computer logging, facial expressions from camera recordings, body postures, ECG (sampled at 2048 Hz) and EDA (sampled at 2048 Hz). In this study, since we want to detect stress using information encoded into physiological signals, we consider only ECG, EDA and HR time series obtained from body sensors.

## 4 Methodology

In this section, we describe STREDWES, the methodology proposed to detect stress. After a preprocessing step outlined in Sect. 4.1, we explain the different ways to encode multi signals into images by applying some algorithms. The encoding algorithms are introduced in Sect. 4.2, while Sect. 4.4 presents the deep learning approach, Convolutional Neural Network, to classify the images by detecting the stress situations.

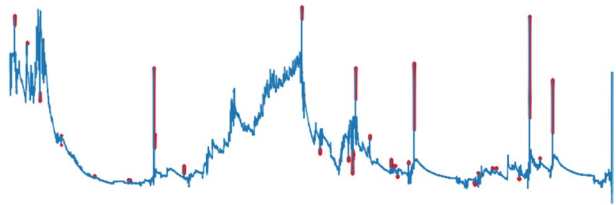
### 4.1 Preprocessing

The preprocessing step consists of signal resampling and outliers removal followed by normalization of signals, described in Sects. 4.1.1 and 4.1.2, respectively. The signal resampling aims to uniform the sampling rate of all of them, while the component of outliers removal intends to remove probable instrumental errors. Finally, signal normalization ensures that all inputs are treated equally.

#### 4.1.1 Resampling

The three datasets considered in this study contain biophysical signals sampled at different frequencies due to the structural differences of the used wearable devices. Therefore, for each dataset, we first standardize the sampling step of all signals to simplify further processing. WESAD dataset contains signals sampled at 700Hz, 32Hz and 4Hz. The highest resolution signals were resampled at 35Hz by downsampling by an integer factor, whereas the others were upsampled to 35Hz. Operationally, we reduced high-frequency signal components with a lowpass filter (finite impulse response of order 5) and decimated the filtered signal by 20. Instead, we upsampled the low-resolution using the Fourier method. NEURO collects signals of accelerations, temperature and EDA sampled at 8 Hz, and HR and SpO2 sampled at 2 Hz. For this dataset, we upsampled the low-resolution to 8Hz. Finally, the signals recorded in SWELL-KW and used in this study are ECG, EDA and HR signals sampled at 2048Hz. After the elimination of signal windows whose values are 999, a tag

**Fig. 4** A sliding window with the outlier values marked in red of electrodermal activities of subject S2 of the WESAD database



indicating the inability to determine the value for that feature, we resampled at 32Hz by downsampling by an integer factor as proposed for the WESAD dataset.

#### 4.1.2 Outliers removal and signal normalization

Some signals may show anomalous peaks due to instrumental error probably. Therefore, we eliminated those anomalies from each time series. We deleted the uneven values of WESAD signals by applying a Hampel filter. The method replaces the outliers without introducing additional observations in the time series. It takes as input sliding windows of a predetermined width of 1 min and computes the mean  $\mu$  and standard deviation  $\sigma$  of such values. The values of the fragments that exceed the threshold of  $3\sigma$  from the  $\mu$  of the corresponding window are considered outliers (Pearson's rule). They are replaced with the closest value in chronological order. For example, Fig. 4 shows a time series window with some outliers marked in red associated with electrodermal activities of subject S2 of the WESAD database.

About the dataset NEURO, we have observed that the signals are regular within their physiological domain. Therefore, no outliers needed to be removed. Finally, the outliers shown included in the signals of SWELL-KW dataset have been removed by applying the following procedures. After the calculation of the 99.8% quantile  $q_{st}$  for each subject of the dataset and for each subject's signal distribution  $t$ , we replace every signal data point equal or higher than  $q_{st}$  with a missing value (NaN). After that, we replace all NaN values in the signal timeseries with a linear interpolation method. To remove other remaning outliers, we apply robust scaling techniques to each subject's timeseries. After outliers removal, we normalize all signals in the interval  $[-1, 1]$  to treat all inputs equally.

## 4.2 Encoding methods

In this section, we describe some frameworks for encoding multivariate time series as images: GAFs, GAFd and MTF.

### 4.3 Gramian angular field

GAF consists of a process, introduced in Wang and Oates (2015), to transform a time series into a matrix. The signals are encoded to an image formalized by a matrix that encodes the time series in a polar coordinate system. To obtain the GAF matrix, we first rescale the observations of the time series. Then, the rescaled values are mapped into a polar coordinates system by encoding each data as the angular cosine and the time stamp as the radius.

Formally, let  $X = \{x_1, x_2, \dots, x_n\}$  be the considered time series with  $n$  components. First, the time series is rescaled by

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

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

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

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

GAFs and GAFd is obtained by computing the sum and difference, respectively, between the points of the time series

$$\begin{aligned} \text{GAFs} &= \begin{bmatrix} \cos(\theta_1 + \theta_1) & \dots & \cos(\theta_1 + \theta_n) \\ \vdots & \ddots & \vdots \\ \cos(\theta_n + \theta_1) & \dots & \cos(\theta_n + \theta_n) \end{bmatrix} \\ &= \tilde{X}^T \cdot \tilde{X} - \sqrt{I - \tilde{X}^2}^T \cdot \sqrt{I - \tilde{X}^2} , \end{aligned} \quad (3)$$

$$\begin{aligned} \text{GAFd} &= \begin{bmatrix} \sin(\theta_1 - \theta_1) & \dots & \sin(\theta_1 - \theta_n) \\ \vdots & \ddots & \vdots \\ \sin(\theta_n - \theta_1) & \dots & \sin(\theta_n - \theta_n) \end{bmatrix} \\ &= \sqrt{I - \tilde{X}^2}^T \cdot \tilde{X} - \tilde{X}^T \cdot \sqrt{I - \tilde{X}^2} , \end{aligned} \quad (4)$$

where  $I$  is the unit row vector  $[1, 1, \dots, 1]$ .

Equations 3 – 4 determine a 1D matrix as output of the encoding process. Each matrix is a heatmap, whose values range from 0 (blue) to 1 (red). Then, the RGB color map applied to the image determines a three channel matrix (interested readers can refer to Wang and Oates (2015a) and Wang and Oates (2015b)).

#### 4.4 Markov transition field

MTF is a framework that maps a set of signals into an image. The image encodes dynamic transfer information mainly by calculating the Markov transfer probability to indicate the temporal correlation. Operatively, the approach first discretizes the time series and computes transition probabilities from one quantile to another. Then, it counts such transition probabilities and normalised them over every two consecutive data points.

Given a time series  $X = \{x_1, x_2, \dots, x_N\}$ , first it is discretized the time series into  $Q$  quantile units: each value of the time series is quantified by the quantiles  $q_j$ ,  $j \in [1, Q]$ , and any  $x_i$  is mapped to the corresponding  $q_i$ . Then, the quantile is mapped into an adjacent weighted matrix  $W$  of  $Q \times Q$ ,



**Table 1** Description of the employed hyperparameters

| Hyperparameter  | Description                                 |
|-----------------|---|
| Batch size      | Batch dimension                             |
| Image size      | The dimension of images                     |
| Learning rate   | Learning rate of the optimization algorithm |
| Optimizer       | Optimization algorithms                     |
| Time step       | The step of the sliding window              |
| Window size     | The dimension of the sliding window         |
| Encoding method | Methods to compute the images               |

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,Q} \\ \vdots & \ddots & \ddots & \vdots \\ w_{Q,1} & w_{Q,2} & \cdots & w_{Q,Q} \end{bmatrix} \quad (5)$$

whose element  $w_{ij} = P(x_i \in q_i | x_{i-1} \in q_j)$  is the frequency of quantile  $q_i$  converting to quantile  $q_j$  such that  $\sum_j w_{ij} = 1$ . Since the Markov transfer matrix ignores the dependence between position and time step, the matrix  $W$  is extended with matrix  $M$  by adding the temporal correlation between each quantile and the time step. Formally,

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,Q} \\ \vdots & \ddots & \ddots & \vdots \\ m_{Q,1} & m_{Q,2} & \cdots & m_{Q,Q} \end{bmatrix} \quad (6)$$

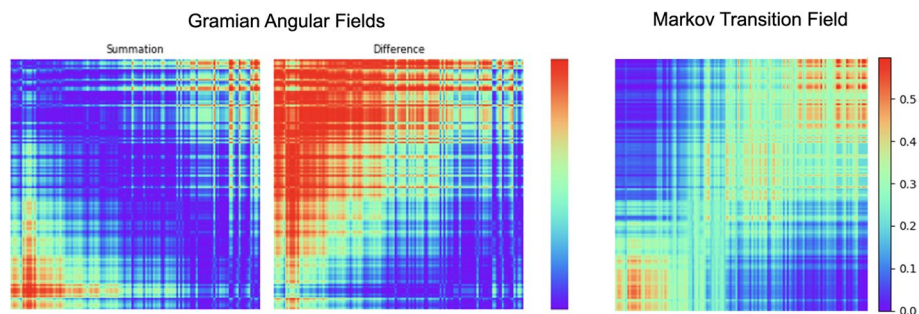
where  $m_{ij} = P(q_i \rightarrow q_j)$  is the transfer probability to move from the quantile  $q_i$  to  $q_j$ .

## 4.5 Dataset entries

The entries of each dataset (SWELL, NEURO and SWELL-KW) consist of images encoded from multivariate time series fragments. To encode the signals into images, we consider several variables, such as the time window length, the step to advance this window, and the image size. The used hyperparameters equipped with their description are listed in Table 1.

Image precomputing by combining all possible parameters requires high computational (time and space) costs. To overcome this problem, we consider a sample generator called *ImgGenerator* which determines samples online one batch at a time. The offsets of the sliding windows are precomputed, ensuring that each window does not span over more than one emotional state. Then, they are randomly shuffled at the end of each training epoch. The sliding windows and, therefore, the encoded images are extracted/calculated “on the fly” when the neural network requests a new batch from the generator.

To encode the signals samples GAFs, GAFd and MTF-like representations, *ImgGenerator* produces matrices of dimensions  $(d, d, n)$ , where  $d$  is the size of the image and  $n$  is the number of channels of the image, corresponds to the number of time series available for each subject. Therefore,  $n$  is 14, 7 and 3 for the images related to WESAD, NEURO and SWELL-KD datasets, respectively. Figure 5 shows the GAFd, GAFs, and MTF images of a fragment of a subject S1 of NEURO dataset. The samples depend on the hyperparameters’ values. We consider classification accuracy and sparse categorical cross-entropy of a

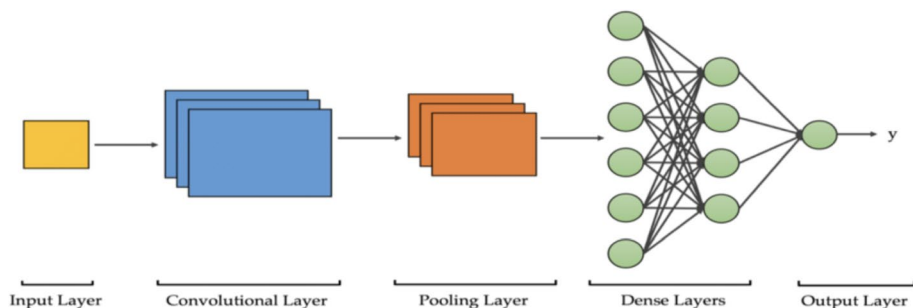


**Fig. 5** GAF and MTF like representations of a fragment of subject S1 of NEURO

validation set as the loss function to select the best-performing hyperparameters. For each configuration of hyperparameter values, a neural network has been trained on the generated samples related to these parameters for a given number of epochs. For each combination of hyperparameters, we evaluated the obtained loss function and the accuracy of the generated samples.

#### 4.6 Convolutional neural network

CNN is a class of deep learning methods which plays a fundamental role in various computer vision tasks and has been engaging the interest across different domains for their capacity to analyze spatial information without requiring hand-crafted feature extraction. CNNs consist of multiple building blocks, such as an input layer, convolution layers, pooling layers, and fully connected layers. The pipeline, shown in Fig. 6, learns spatial hierarchies of features through a backpropagation algorithm automatically and adaptively. The input (image), formalized as a tensor characterized by a shape, passes through a convolutional layer that abstracts a feature map. In other words, the convolutional layer learns to find spatial features in an input image by applying a series of many different image filters, also known as convolutional kernels. The filters are applied repeatedly across the entire dataset in order to improve the efficiency of the training process by reducing the number of parameters to learn. The outputs of such convolutions are then processed by a nonlinear activation function. After a convolutional layer, each CNN is



**Fig. 6** Basic structure of CNNs consisting of an input layer, a convolution layer, a non-linear layer, a pooling layer and an output layer

characterized by a pooling layer. Each pooling layer performs a downsampling operation by performing maximum or average subsampling of non-overlapping regions in feature maps. This non-overlapping subsampling permits the CNNs to aggregate local features to identify more complex ones.

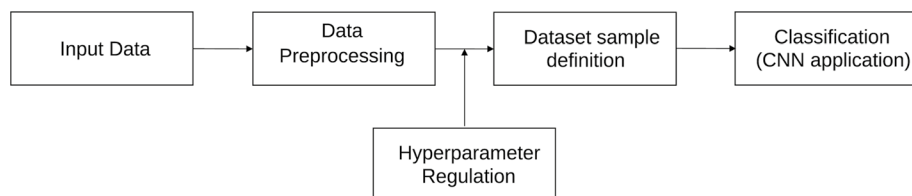
## 5 Experiments

The work aims to analyze signals from wearable sensors for stress detection. To have general results, we consider three datasets, WESAD, NEURO and SWELL. We study the proposed approach based on CNN and signals encoded into images on NEURO by considering the GAFs, GAFd and MTF as encoding algorithms. We select GAFs as the method for encoding signals and apply STREDWES on WESAD. Although the stress detection problem can be formulated as a binary classification, we classify the signals as *baseline*, *stress*, and *amusement* to compare the obtained results with the ones in the literature (Schmidt et al., 2018). Finally, we introduce a further approach in which we train a model for each subject to detect stress by considering personal physiological changes. We test STREDWES considering such an approach to SWELL-KW.

### 5.1 Implementation

We propose two Python implementations of STREDWES, whose pipeline scheme is shown in Fig. 7. One uses the Tensorflow package (Girija, 2016), while the other exploits the PyTorch package (Paszke et al., 2019). However, both use the *pyts* package for obtaining the GAFd, GAFs and MTF images (Faouzi & Janati, 2020). The former implementation takes advantage of the library for implementing both the sample generator and Convolutional Neural Network and extends the class Sequence (Chollet et al. 2015). The neural network is composed of two 2D convolutional layers with 32 filters, each followed by a MaxPooling2D layer (with pool size = (2, 2)) and a layer of Dropout (with 0.25 dropout rate), and finally from two fully connected layers with a Dropout layer in between (with 0.5 dropout rate). The activation function used is the ReLU 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 during the creation of the network. The code based on the PyTorch package implements the neural network with four 2D convolutional layers with 32, 64, 128, and 256 filters; each followed by a MaxPooling2D layer (with pool size = (2, 2)) and a Dropout layer (0.2 dropout rate). 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. Table 2 summarizes the implementation details.

All experiments were performed with the Amazon SageMaker Service using one “ml.g4dn.xlarge” instance. The code used in this manuscript is available from the corresponding author upon reasonable request.



**Fig. 7** General pipeline of our approach

**Table 2** Implementation Details of the STREDWES versions

|                        | Tensorflow  | PyTorch   |
|------------------------|---|---|
| Time series encoding   | pyts Package  | pyts Package  |
| Sample generator       | Keras   | DataModule of Pythorch Lightning  |
| Neural network         | CNN with two convolutional layers 32 filter MaxPooling2D layer (pool size = (2, 2)) | CNN with four convolutional layers 32, 64, 128, and 256 filters MaxPooling2D layer (pool size = (2, 2)) |
| Dropout                | 0.25  | 0.20  |
| Activation function    | Relu and softmax  | Prelu   |
| Optimization algorithm | Adam  | AdamW   |

## 5.2 Evaluation metrics

We evaluate performance and effectiveness of the developed model and compare it with one of some other methods in the literature using five evaluation metrics. We evaluate the performance and effectiveness of the developed model and compare it with one of some other methods in the literature using five evaluation metrics. The *Accuracy* (Acc) allows you to measure the total number of predictions a model gets right, the *Precision* (P) evaluates how precise a model is in predicting positive labels, and the *Recall* (R) calculates the percentage of actual positives a model correctly identified. The F-measure (F1) score is a measure of the harmonic mean of precision and recall, and *Specificity* (Spec) measures how the model can detect positive instances. They are formally defined as follows.

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (7)$$

$$\text{P} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (8)$$

$$\text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (9)$$

$$\text{F}_1 = 2 \times \frac{\text{P} \times \text{R}}{\text{P} + \text{R}} \quad (10)$$

**Table 3** Selected hyperparameters for NEURO dataset

| Hyperparameter | Best value   |
|----------------|--------------|
| Time step      | 1            |
| Learning rate  | 0.0012       |
| Batch size     | 31           |
| Window aize    | 10           |
| Image aize     | 64           |
| GAF method     | “Difference” |

$$\text{Spec} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (11)$$

where  $TP$  represents the number of stress positive test samples identified correctly (true positive),  $FN$  denotes the number of stress negative test samples identified incorrectly (false negative),  $FP$  represents the number of stress positive test samples identified incorrectly (false positive),  $TN$  denotes the number of stress negative test samples identified correctly (true negative). As mentioned earlier, we perform a multiclass classification. Therefore, F-measure, Precision, and Recall are the three most important evaluation metrics as they can provide more comprehensive measures than other evaluation metrics (Zeng et al., 2016).

### 5.3 Experiments

**Study of encoding methods** We study the proposed approach based on CNN and signals encoded into images on NEURO by considering the GAFs, GAFd and MTF encoding algorithms. The 20 subjects of the NEURO dataset were split into two disjoint sets, i.e., training and test, stratifying according to sex. The two sets consist of 17 and 3 subjects, respectively. To train our model, we tune the hyperparameters listed in Table 1 by a Bayesian search considering the following ranges:

- *Batch size*, tested integer values from 16 to 512 in logarithmic scale;
- *Image size*, tested integer value 64;
- *Learning rate*, tested real value from  $1 \cdot e^{-6}$  to  $1 \cdot e^{-2}$  in logarithmic scale;
- *Time step (in seconds)*, tested integer values 1, 3, 9, 20<sup>1</sup>;;
- *Window size (in seconds)*, tested integer values 10, 30, 90, 270;
- *Encoding methods*, tested GAFs, GAFs, MTF

For each combination of hyperparameters, the network is trained for up to 25 epochs. To avoid overfitting, the method stops the training in advance by employing the early stopping method if the accuracy of the validation set does not improve for five consecutive epochs. The validation set consists of 2 subjects randomly collected from the training set. At the end of the training, it selects the best weights corresponding to the epoch with the best

<sup>1</sup> the tuning algorithm chooses the best scale for the hyperparameter exploration among linear, logarithmic, and reverse logarithmic.

**Table 4** Performances of approaches on NEURO dataset

| Method        | Accuracy (mean) | F1-score (mean) | Precision (mean) | Recall (mean) | Specificity (mean) |
|---------------|-----------------|-----------------|------------------|---------------|--------------------|
| MTF and CNN   | 73.34%          | 72.36%          | 76.40%           | 71.27%        | 83.84%             |
| GAFs and CNN  | 71.72%          | 69.35%          | 74.94%           | 67.41%        | 83.4%              |
| GAFd and CNN  | 73.03%          | 72.49%          | 76.32%           | 71.57%        | 84.14%             |
| Random Forest | 64.54%          | 62.34%          | 68.73%           | 64.54%        |                    |

result in accuracy. The method achieves the maximum accuracy for the hyperparameters reported in Table 3.

In Appendix A, we report the scatter plots to show how the accuracy depends on the explored hyperparameters, such as window size, time step, image size, and batch size. To visualize the relationship between the accuracy and the considered hyperparameter, we draw the relation curve estimated based on a third degree polynomial regression. We observe that the choice of hyperparameters significantly affects the final accuracy, almost doubling it in the case of optimizer, and window size. However, we note that the increase in the accuracy value corresponds to an increase in the variability. This fact means that some parameters, such as window size and time step, are correlated to each other. Consequently, their combined effect is greater than the effect of the individual components.

After the selection of hyperparameters, we test the approach by considering the three different methods to encode signals into images (GAFd, GAFs and MTF) by using a LOSOCV technique to obtain a more robust result given the low sample size. We consider 1 subject and we train the model on the remaining 19 subjects (17 for training and 2 for validation). Therefore, 20 training jobs are run individually for each encoding method. For multiclass classification, the values of some evaluation metrics are reported in Table 4. Such values are obtained as the mean of the values obtained for each subject (shown in Appendix B) since the technique by applying the LOSOCV technique, i.e. we have considered 1 subject as test, and we trained the model on the remaining 19 subjects, as described in the “Study of Encoding Methods” paragraph. According with the protocol used to capture signals into NEURO dataset, the classes to identify are three: Relax, Physical Stress and Cognitive/Emotional Stress. Moreover, we report also the values of the metrics for each subject in classification and the confusion matrix.

According to the results obtained in the literature, the analysis of the performances shows that the three encoding algorithms in terms of accuracy, F1 score, precision, recall and specificity are comparable. Moreover, since there is no benchmark related to NEURO in the literature and the results of the some experiments related to the same dataset are not comparable with the ones obtained due to the design of the experiment, we apply a machine learning algorithm, random forest, to compare our approach with a classical machine learning method by considering the LOSOCV technique. The obtained performances are reported in Table 4. To select the encoding algorithm in STREDWES, we use the F1-score as a choice parameter because it considers both precision and recall. We apply STREDWES to the WESAD dataset and compare its performance with other methods proposed in the literature.

### Comparison with other methods

To compare STREDWES with other methods in the literature, we apply it to WESAD dataset. The 15 subjects of the dataset were split into two disjoint sets, i.e., train and test,

stratifying according to sex. The two sets consist of 12 and 3 subjects, respectively. To train our model, we select the encoding method according to the previous results and tune the remaining hyperparameters listed in Table 1 by a Bayesian search considering the following ranges:

- *Batch size*, tested integer values from 4 to 512 in logarithmic scale;
- *Image size*, tested integer values from 16 to 128 in logarithmic scale;
- *Learning rate*, tested real value from  $1 \cdot e^{-7}$  to  $1 \cdot e^{-1}$  in logarithmic scale;
- *Optimizer*, tested methods: “Adam”, “AMSGrad” and “SGD”;
- *Time step*, tested integer value from 1 to 30 in auto scale<sup>2</sup>;
- *Window size (in seconds)*, tested integer value from 30 to 300 in logarithmic scale;

We train the network for up to 50 epochs, for each combination of hyperparameters. However, the method stops the training in advance by employing the early stopping method if the loss function on the validation set does not improve for five consecutive epochs to avoid overfitting. At the end of the training, the weights corresponding to the epoch with the best result on the validation loss are selected. The Bayesian hyperparameter search is repeated for 100 iterations. The method achieves the maximum accuracy, i.e., 98.79%, for the hyperparameters reported in Table 5. In Appendix A, we report the scatter plots to show how the accuracy depends on the explored hyperparameters.

Due to the limited data sample we use Cross-validation, a resampling procedure, to evaluate model. We randomly split the data into five groups composed of 3 subjects each. In each step of the procedure, a group forms the test dataset, while the remaining groups set up a training dataset. Given the results obtained in the hyperparameter search, we fixed the values of some hyperparameters, and we updated the range of the others. More specifically, the time step is set to 1 and the optimizer is set to ‘AMSGrad’. We performed a Bayesian search considering the following ranges of hyperparameters:

- *Batch size*, tested integer values from 4 to 82 by using auto scale;
- *Image size*, tested integer values from 48 to 105 by using auto scale;
- *Learning rate*, tested real values from  $9.884e^{-4}$ ,  $4.235e^{-3}$ , by using auto scale;
- *Window size*, tested integer values from 145 to 300 by using auto scale.

As in the previous hyperparameter search, we trained the network for a maximum of 50 epochs by considering the Bayesian search and the stopping method. In this procedure, we obtained an average accuracy of the 92.58% and a standard deviation of 10.09% by considering the hyperparameter values reported in Table 6.

As in the experiments related to NEURO dataset, we consider the LOSOCV approach and train the model for a maximum of 10 epochs. For each LOSOCV iteration, the hyperparameter search was performed on the 14 training subjects with the 5-fold CV described earlier. The model stops the training when the first local minimum of the training loss is reached and selects the weights at that minimum. By this approach, we obtain an average

<sup>2</sup> the tuning algorithm chooses the best scale for the hyperparameter exploration among linear, logarithmic, and reverse logarithmic.

**Table 5** Selected hyperparameters

| Hyperparameter | Best value |
|----------------|------------|
| time step      | 1          |
| Optimizer      | AMSGrad    |
| Learning Rate  | 0.0012     |
| Batch Size     | 82         |
| Window Size    | 293        |
| Image Size     | 105        |

**Table 6** Selected hyperparameters with 5-Fold Cross-Validation

| Hyperparameter | Best value |
|----------------|------------|
| Batch Size     | 43         |
| Image Size     | 83         |
| Learning Rate  | 0.0016     |
| Window Size    | 294        |

accuracy of the 91.16% and a standard deviation of 0.09%. The confusion matrix of the approach is shown in the Appendix B.

Finally, we compare the performance of STREDWES with other approaches based on classical machine learning algorithms (Schmidt et al., 2018) and the multimodal-multisensory sequential fusion model (MMSF) (Lin et al., 2019) defined in the literature. Table 7 shows the accuracy and F1 score of each approach.

### STREDWES on real case investigations

We propose a further approach to investing in the performance of STREDWES in real scenarios by defining a “personalized” approach. This approach includes that the data entries for training and testing consist of fragments of bio signals of the same subject, differently from the others in which the model is trained by considering a group of subjects and tested on data of other ones. Therefore, it requires a training step for each subject. The approach can learn precisely the personal physiological changes considering only a few kinds of signals that may be recorded by a smartwatch in everyday life. To apply the model in real applications using a mobile application, the user needs to use the application to give information since the personalized model is trained only on the user data. The experiment considers only the ECG, EDA and HR signals of the SWELL-KW dataset. Such dataset, as introduced in Sect. 3.3, records computer logging, facial expressions from camera recordings, body postures, ECG and EDA of 25 people that perform typical office work, such as writing reports, making presentations and reading e-mails, by following the experimental procedure shown in Fig. 3. However, we consider the data related only to 20 subjects since the ECG, EDA and HR signals of others do not conform to the protocols, maybe due to some instrumental errors. Since the approach requires a training step for each subject, we build a specific dataset for each participant. The statistical information about the entries of such datasets is reported in Appendix C. To train the model, we use the hyperparameters reported in Table 8.

In these experiments, we do not tune the hyperparameters and choose a window size and an image size smaller than in previous experiments since we assume that there are not too many prolonged stimuli in daily life as in laboratory experiments. The obtained



**Table 7** 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                 |
| <b>STREDWES</b>              | <b>91.11</b> | <b>90.40</b>          |

Bold indicates the best result obtained with methods used

**Table 8** Selected hyperparameters for simulating STREDWES on real cases (SWELL dataset)

| Hyperparameter | Best value |
|----------------|------------|
| time step      | 1          |
| Optimizer      | Adam       |
| Learning Rate  | 0.0012     |
| Batch Size     | 82         |
| Window Size    | 112        |
| Image Size     | 112        |

accuracy and F1-score are 86.9% and 86.49%. We consider the performances significant and encouraging for some future developments. The performance of the method during each experiment, evaluated via metrics, is shown in Appendix C.

## 6 Conclusions and future works

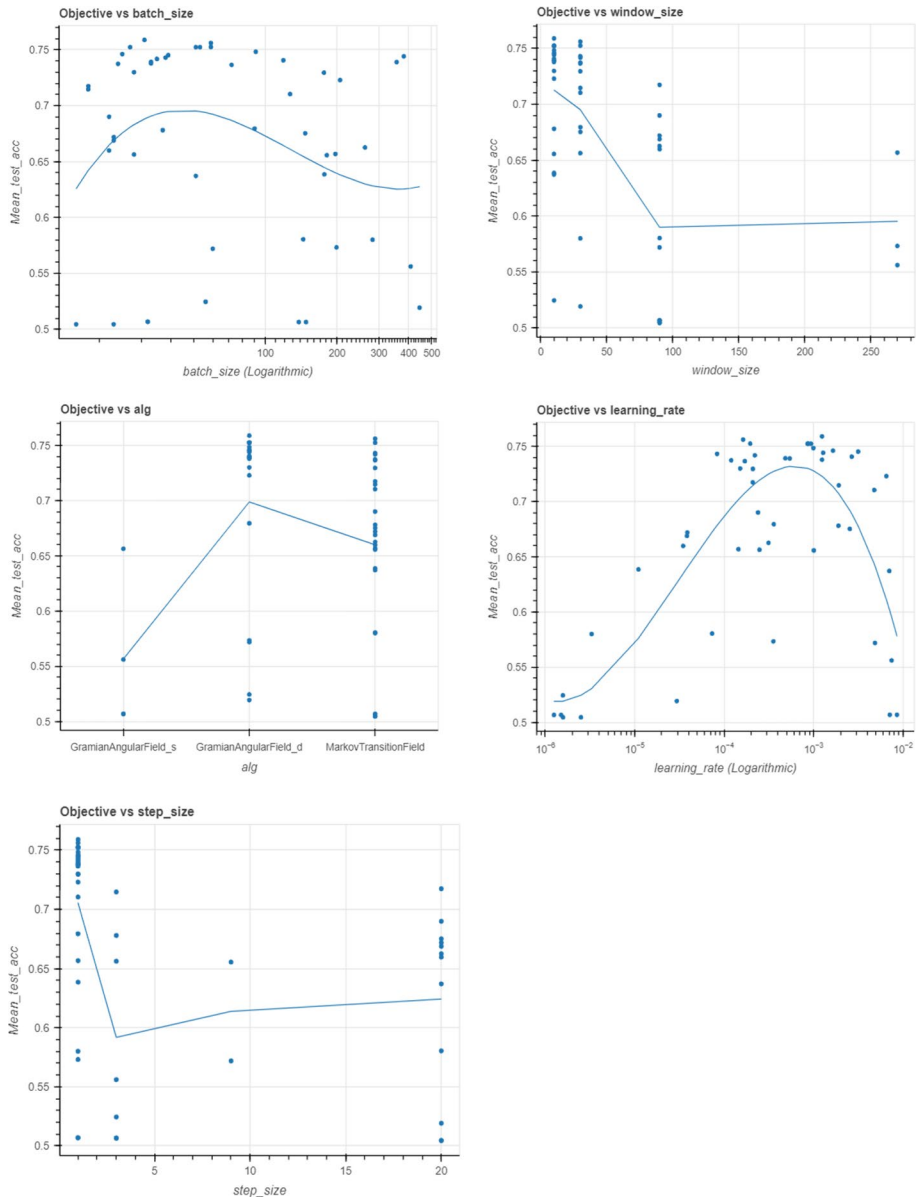
The definition and development of robust and non-invasive methods to detect stress plays a fundamental role in everyday life since stress is a significant and growing phenomenon that can cause numerous health problems. Detection and managing stress affects people's life quality and wellness. Taking advantage of the social acceptance of smartphones and wearable sensors, able to track physiological parameters and monitor the physical activities of users, we can define robust non-invasive methods to detect stress. In this work, we have considered STREDWES, a reliable and non-invasive approach for Stress Detection from Wearable Sensors Data that face the stress detection as a classification problem. The method encodes fragment of physiological signals into GAFd-like representations and classifies them by applying a CNN. This study has aimed to evaluate the best way to encode signals into images. We have considering three methods, GAFd, GAFs and MTF, to the NEURO dataset. We have selected this dataset since it records only 7 biophysical signals compared to the 13 of WESAD but, at the same time, involves 20 subjects in the experiments. According to the literature, the performances of our approach, obtained considering the three encoding methods, are comparable. However, the MTF images are more expensive to obtain in terms of computational cost and the performance associated

with the GAFs images is slightly lower. For these reasons, we have implemented the GAFd into STREDWES. To compare its performances with respect to some classical machine learning algorithms and other competitors, we have applied apply STREDWES on the WESAD dataset since such a dataset is the more used in the literature. The results show that our method outperforms others. Moreover, we have also tried to simulate STREDWES in real cases and we have trained the model on the SWELL-KW dataset since it records only three noisy physiological signals, ECG, EDA and HR, which should be recorded by a smartwatch in everyday life. In this context, we have trained a model for each subject. In other words, the model is tested on a training set (70%) and test set (30%) of the same person to consider the personal physiological changes of each user. Moreover, STREDWES does have potential limitations, most notably its indiscriminate input of various physiological signals without preliminary selection. This approach, while encompassing all data, can introduce redundancy, increment the computational cost of the model and degrade its accuracy due to the possible inclusion of non-informative and noisy signals. A strategic feature selection, especially by focusing on single-user signal data, could enhance the model's computational efficiency and predictive precision. Moving forward, we are committed to compare the performance of STREDWES obtained on NEURO and SWELL-KW datasets against diverse methods in existing literature to detect stress. The comparison will consider the two approaches used to train the model: the first one trains and tests the model considering data of distinct subjects, to emphasize generalization capabilities, while the second, closer to a real-application scenario, one trains and tests on the same subject. Each approach offers its unique insights - the former into generalization capabilities, whereas the latter into actual application viability. The application of these approaches in everyday life scenarios remains a challenge. For real-time analysis, we need to reduce the computational complexity of these methods. Identifying the most relevant biophysical signals and feature selection are crucial to reducing the dimensionality of the data and computational demands. To face the challenge, we also intend to perform other experiments to understand which physiological signals are more relevant than others. This aspect plays a fundamental role in applying our approach in daily life. In this case, we want to record the minimum signals to be non-invasive and reduce the dimensionality of the stored data. With the aim to capture the personal changes and generalize at the same time, we also intend to combine the two different approaches. Finally, we have planned to develop a mobile app based on STREDWES to detect stress by considering the user's physiological signals captured by smartphones and smartwatches.

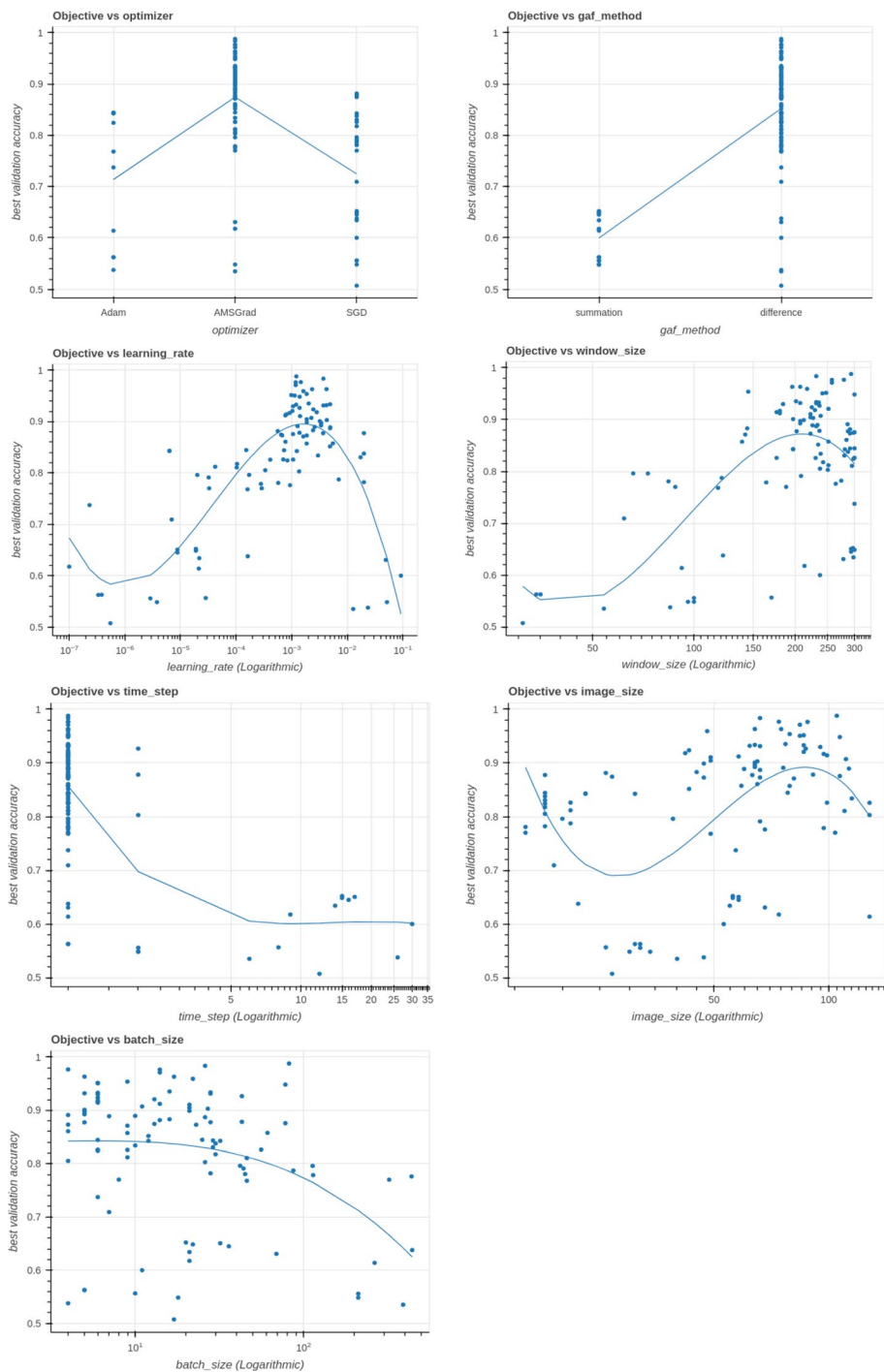
We are also considering employing more deep learning approaches, such as graph convolution networks or recurrent neural networks, motivated by the results obtained in other scenarios (Quadrini et al., 2020, 2022). We aim to study the role of the length of the sliding windows from a theoretical perspective by taking into account various entropy-based methods that have produced useful outcomes in the scenario of protein-protein interaction site prediction (Quadrini et al., 2021; Šikić et al., 2009). Understanding ways to extract and describe the correlations that define sliding windows is a crucial future direction. The topological interpretation of the temporal relations as in Piangerelli et al. (2020) or other representations like arc-annotated sequences for the analysis and comparison of signals utilizing tools like (Quadrini et al., 2020) and strings, which allow applying techniques from formal methods to identify patterns (Quadrini et al., 2019), will be explored as potential representations based on simplicial complexes.

## Appendix A: Analysis of hyperparameters

The scatter plots computed on NEURO and WESAD datasets are shown in Figs. 8 and 9, respectively, such as window size, time step, image size, and batch size.



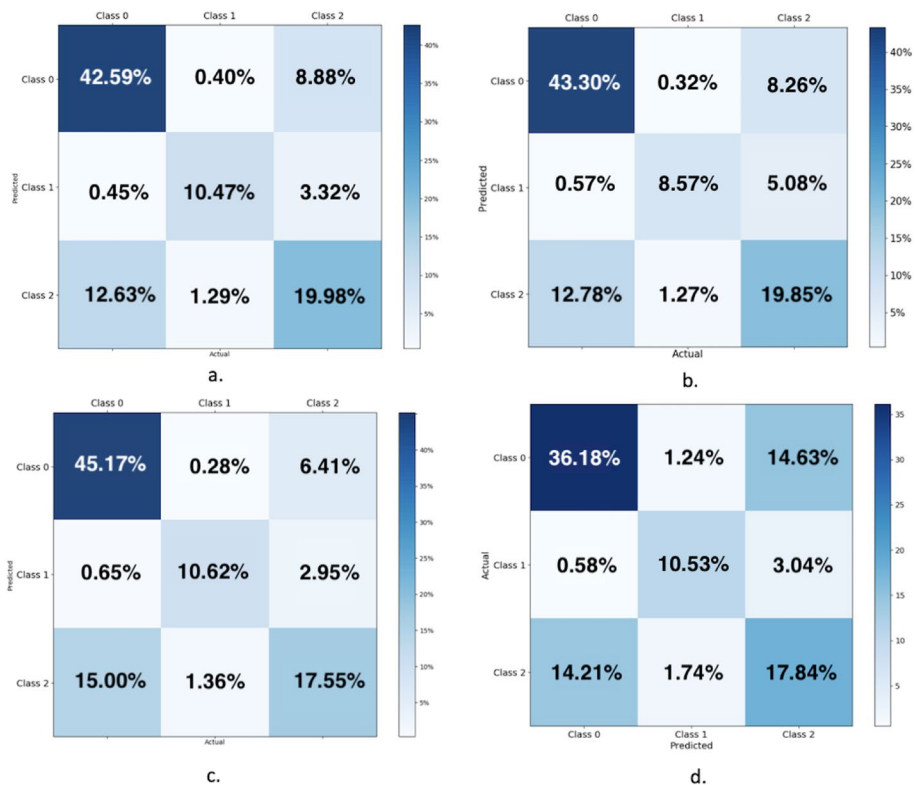
**Fig. 8** Scatter plot of the Accuracy the tested hyperparameters on NEURO dataset



**Fig. 9** Scatter plot of the Accuracy the tested hyperparameters on WESAD dataset

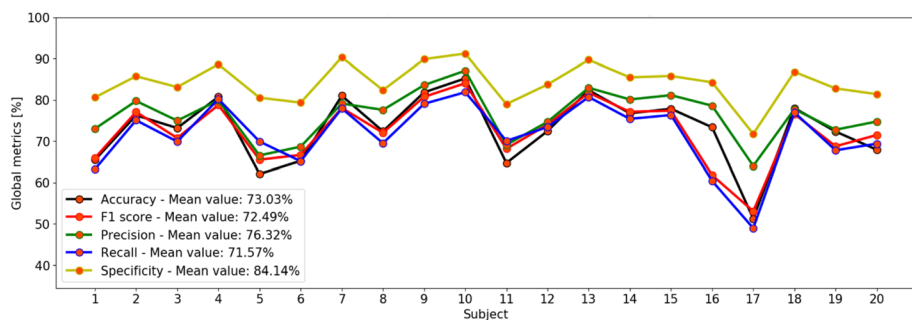
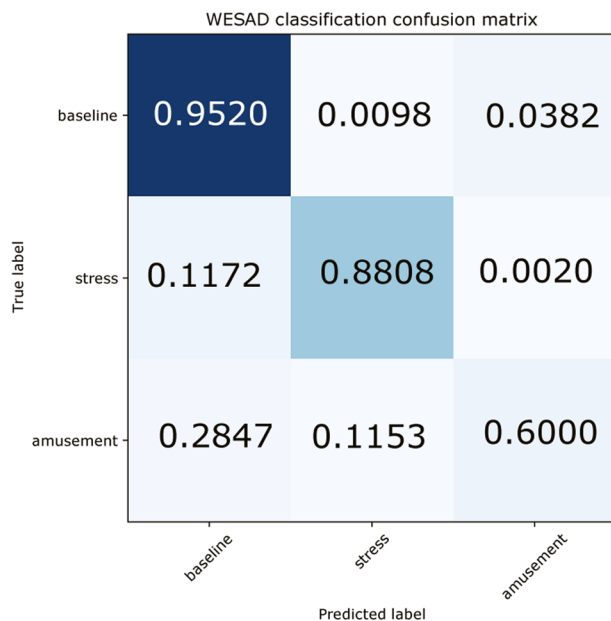
## Appendix B: Confusion matrix and global metrics

In this Section, we show the confusion matrices that evaluates the methods based with GAFd, GAFs and MTF encoding methods on NEURO dataset in Fig. 10a–c. Such images have been classified by a CNN using the LOSOCV technique. Figure 10d show the confusion matrix that evaluates the Random Forest method. Figure 11 show the confusion matrix that evaluates our approach on the WESAD dataset by considering LOSOCV technique. Finally, Fig. 12, 13, and 14 show the values of accuracy, F1 score, precision, recall and specificity calculated on each subject of NEURO dataset considering GAFd, GAFs, MTF representation and CNN method. Figure 15 show the value computed with Random Forest method on the Neuro dataset.

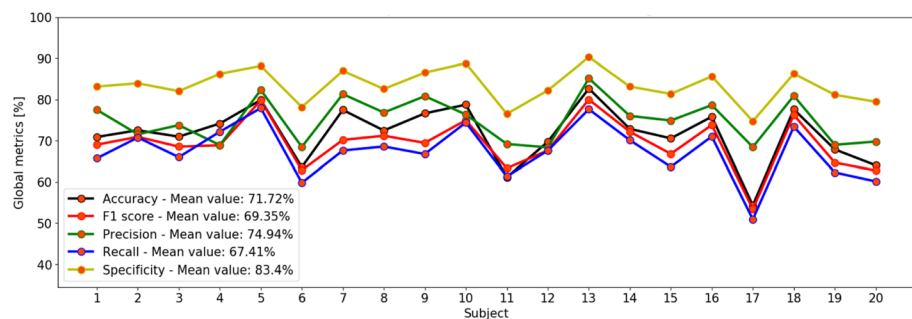


**Fig. 10** Confusion matrix that evaluates the CNN method with **a** GAFd, **b** GAFs, **c** MTF images **d** Random Forest method applied to NEURO dataset

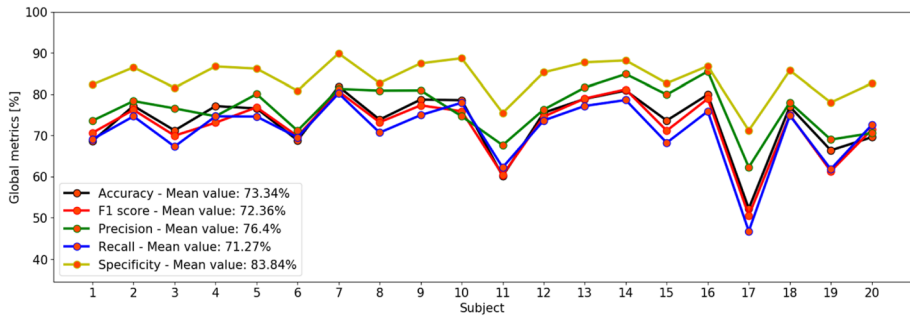
**Fig. 11** Confusion Matrices that evaluates our approach on the WESAD dataset



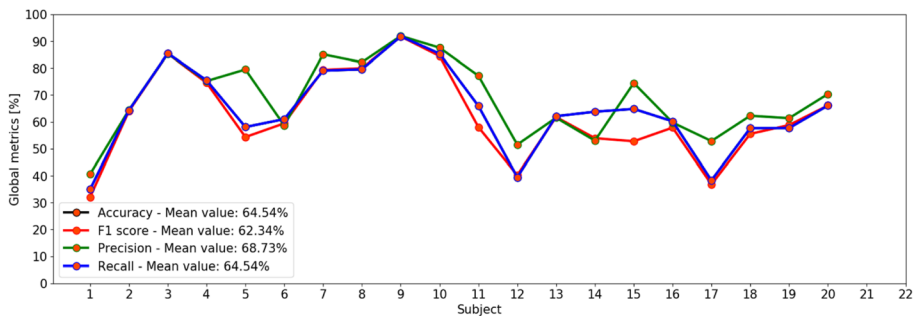
**Fig. 12** Values of Accuracy, F1 score, Prediction, Recall and Specificity calculated on the single subject using GAFd encoding method and LOSOCV technique



**Fig. 13** Values of Accuracy, F1 score, Prediction, Recall and Specificity calculated on the single subject using GAFs encoding method and LOSOCV technique



**Fig. 14** Values of Accuracy, F1 score, Prediction, Recall and Specificity calculated on the single subject using MTF encoding method and LOSOCV technique



**Fig. 15** Values of Accuracy, F1 score, Prediction and Recall calculated on the single subject using Random Forest encoding method and LOSOCV technique on NEURO dataset

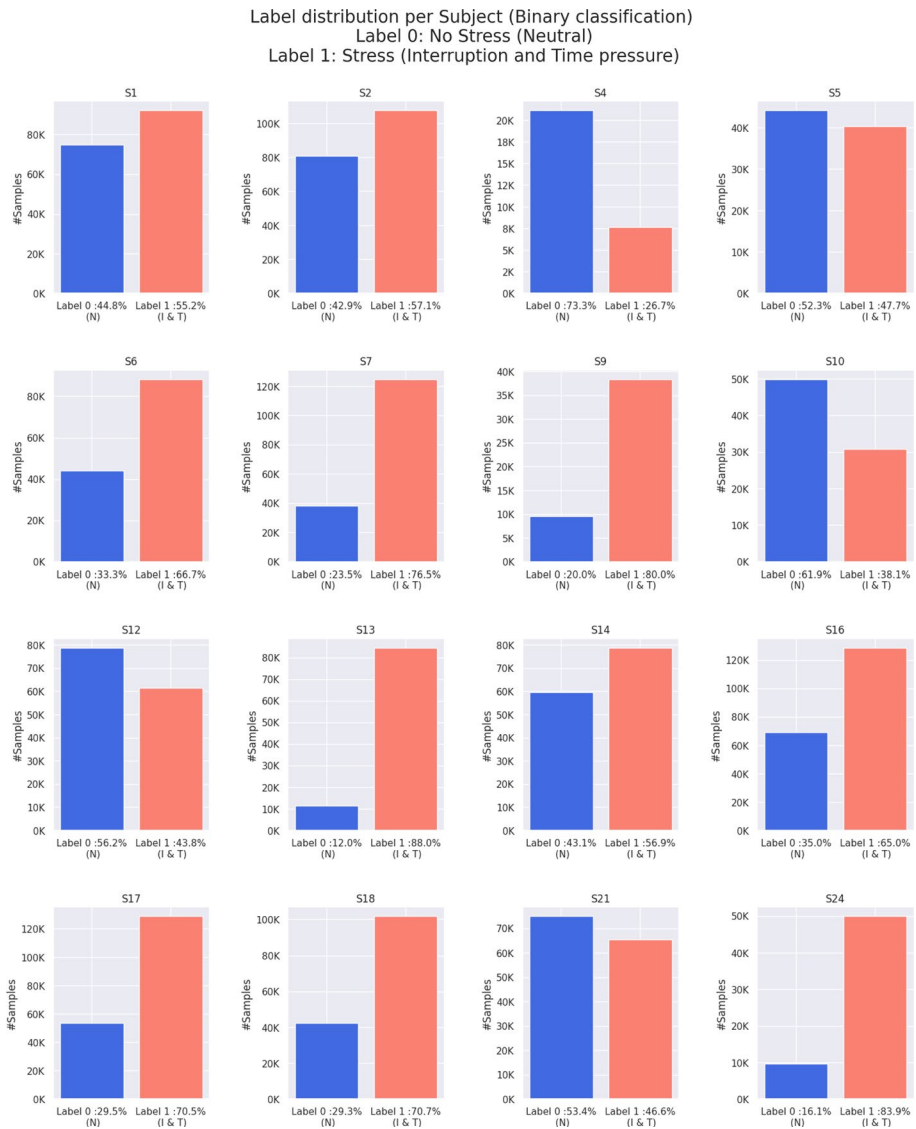
## Appendix C: SWELL-KW

In this Section, we show the data entries built based on the SWELL-WB database for the "personalized" approach. Each data entry consists of images obtained by encoding signal fragments of a particular subject. Therefore, we have a data entry for each subject of the SWELL-KW database. Moreover, we did not consider subjects whose signals did not capture both stress and non-stress states. Such fragments are obtained with a sliding approach, long 112 s, and step equals 1 s. Table 9 show the number of images for each data entries, whereas Fig. 16 show the distribution of labels (stress/no stress) for each data entry. Figure 17 show the values of accuracy, F1 score, precision, recall and specificity calculated on the data entry related to each subject of SWELL-KW dataset.

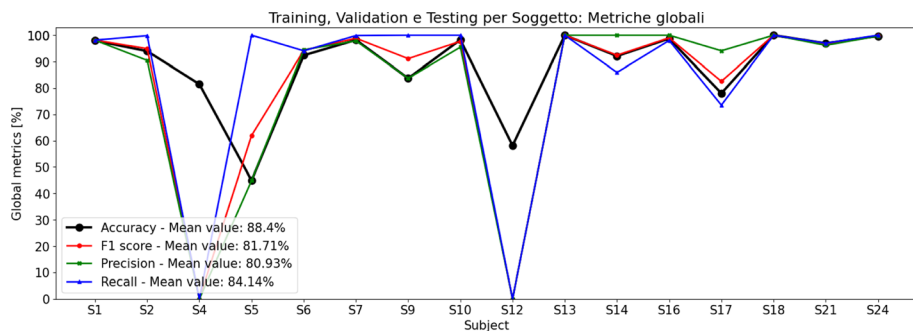
**Table 9** The quantity of images per data entry that is linked to the subject of the SWELL-KW database

| Subject | Number of training entries | Number of validation entries | Number of test entries |
|---------|----------------------------|------------------------------|------------------------|
| S1      | 3416                       | 489                          | 976                    |
| S2      | 3880                       | 555                          | 1109                   |
| S4      | 473                        | 68                           | 135                    |
| S5      | 1612                       | 231                          | 461                    |
| S6      | 2662                       | 381                          | 761                    |
| S7      | 3334                       | 477                          | 953                    |
| S9      | 814                        | 117                          | 233                    |
| S10     | 1607                       | 230                          | 459                    |
| S12     | 2830                       | 405                          | 809                    |
| S13     | 1864                       | 267                          | 533                    |
| S14     | 2788                       | 399                          | 797                    |
| S16     | 4090                       | 585                          | 1169                   |
| S17     | 3754                       | 537                          | 1073                   |
| S18     | 2914                       | 417                          | 833                    |
| S21     | 2830                       | 405                          | 809                    |
| S24     | 1145                       | 164                          | 327                    |





**Fig. 16** Distribution of labels for each data entry associated with each subject of the SWELL-KW database. Label 0 indicates no stress, whereas Label 1 identifies stress situations



**Fig. 17** Values of Accuracy, F1 score, Prediction, Recall and Specificity calculated on the data entries of each subject of SWELL-KW database

**Author contributions** All authors conceived the approach, STREDWES. MQ wrote the manuscript. AC, SD, AB and LB have implemented the draft of code. AC finalized the implementation. DF and AC performed the experiments. MQ and GG supervised the definition of approach and the experiments. MQ and GG reviewed the final version of the manuscript. All authors have read and approved the final manuscript.

**Funding** This work has been funded by the European Union - NextGenerationEU under the Italian Ministry of University and Research (MUR) National Innovation Ecosystem grant ECS\_00000041 VITALITY - CUP J13C22000430001.

**Data availability and materials** The data used in the publication is publicly available.

**Code availability** The developed code for experiments related to SWELL-WK dataset are available for download at: <https://github.com/ggerardlatek/STREDWES-SWELL>, whereas the developed code for experiments related to WESAD and NEURO datasets can be provided by the Corresponding Author on reasonable request.

## Declarations

**Conflict of interest** The authors declare that they have no competing interests.

**Consent to participate** The authors provide the consent to publish the images in the manuscript. The data used in the publication is publicly available. We provide respective citations for each of the data sources.

## References

- Birjandtalab, J., Cogan, D., Pouyan, M.B., Nourani, M. (2016). A non-ecg biosignals dataset for assessment and visualization of neurological status. In *2016 IEEE International Workshop on Signal Processing Systems (SiPS)* (pp. 110–114). IEEE.
- Chollet, F., et al. (2015). Keras. <https://keras.io>.
- de Souza, A., Melchiades, M.B., Rigo, S.J., & Ramos, G.d.O. (2022). Mostress: A sequence model for stress classification. In *2022 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). IEEE.
- Fauzi, J., & Janati, H. (2020). pyts: A python package for time series classification. *The Journal of Machine Learning Research*, 21, 1720–1725.
- Garcia, G. R., Michau, G., Ducoffe, M., Gupta, J. S., & Fink, O. (2022). Temporal signals to images: Monitoring the condition of industrial assets with deep learning image processing algorithms. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 236(4), 617–627.


- Giannakakis, G., Grigoriadis, D., Giannakaki, K., Simantiraki, O., Roniotis, A., & Tsiknakis, M. (2019). Review on psychological stress detection using biosignals. *IEEE Transactions on Affective Computing*, 13, 440–460.
- Girija, S.S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Software available from [https://tensorflow.org/39\(9\)](https://tensorflow.org/39(9)).
- Gjoreski, M., Luštrek, M., Gams, M., & Gjoreski, H. (2017). Monitoring stress with a wrist device using context. *Journal of Biomedical Informatics*, 73, 159–170.
- Healey, J. A., & Picard, R. W. (2005). Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on Intelligent Transportation Systems*, 6(2), 156–166.
- Health and Safety Executive (2021). HSE on work-related stress. <http://www.hse.gov.uk/statistics/causdis/-ffstress/index.htm>. Accessed on March 7, 2022.
- Jaiswal, M., Bara, C.P., Luo, Y., Burzo, M., Mihalcea, R., & Provost, E.M. (2020). Muse: a multimodal dataset of stressed emotion. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 1499–1510).
- Kirschbaum, C., Pirke, K. M., & Hellhammer, D. H. (1993). The trier social stress test A tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology*, 28(1–2), 76–81.
- Koldijk, S., Sappelli, M., Verberne, S., Neerinx, M.A., & Kraaij, W. (2014). The swell knowledge work dataset for stress and user modeling research. In *Proceedings of the 16th international conference on multimodal interaction* (pp. 291–298).
- Lee, E. H. (2012). Review of the psychometric evidence of the perceived stress scale. *Asian Nursing Research*, 6(4), 121–127.
- Li, R., & Liu, Z. (2020). Stress detection using deep neural networks. *BMC Medical Informatics and Decision Making*, 20, 1–10.
- Lin, J., Pan, S., Lee, C.S., & Oviatt, S. (2019). An explainable deep fusion network for affect recognition using physiological signals. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 2069–2072).
- Lundberg, U., Kadefors, R., Melin, B., Palmerud, G., Hassmén, P., Engström, M., & Elfsberg Dohns, I. (1994). Psychophysiological stress and EMG activity of the trapezius muscle. *International Journal of Behavioral Medicine*, 1(4), 354–370.
- Marwan, N., Romano, M. C., Thiel, M., & Kurths, J. (2007). Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(5–6), 237–329.
- McEwen, B. S. (1998). Protective and damaging effects of stress mediators. *New England Journal of Medicine*, 338(3), 171–179.
- Oskooei, A., Chau, S.M., Weiss, J., Sridhar, A., Martínez, M.R., & Michel, B. (2021). Destress: deep learning for unsupervised identification of mental stress in firefighters from heart-rate variability (HRV) data. *Explainable AI in Healthcare and Medicine: Building a Culture of Transparency and Accountability* (pp. 93–105).
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 8026.
- Piangerelli, M., Maestri, S., & Merelli, E. (2020). Visualising 2-simplex formation in metabolic reactions. *Journal of Molecular Graphics and Modelling*, 97, 107576.
- Quadrini, M., Cavallin, M., Daberdaku, S., & Ferrari, C. (2021). Prosps: Protein sites prediction based on sequence fragments. In *International Conference on Machine Learning, Optimization, and Data Science* (pp. 568–580). Springer.
- Quadrini, M., Daberdaku, S., & Ferrari, C. (2020). Hierarchical representation and graph convolutional networks for the prediction of protein–protein interaction sites. In *International conference on machine learning, optimization, and data science* (pp. 409–420). Springer.
- Quadrini, M., Daberdaku, S., Blanda, A., Capuccio, A., Bellanova, L., & Gerard, G. (2022). Stress detection from wearable sensor data using gramian angular fields and CNN. In *International Conference on Discovery Science* (pp. 173–183). Springer.
- Quadrini, M., Merelli, E., & Piergallini, R. (2019). Loop grammars to identify RNA structural patterns. In *10th international conference on bioinformatics models, methods and algorithms, BIOINFORMATICS 2019 - Part of 12th international joint conference on biomedical engineering systems and technologies, BIOSTEC 2019* (pp. 302–309).
- Quadrini, M., Daberdaku, S., & Ferrari, C. (2022). Hierarchical representation for PPI sites prediction. *BMC Bioinformatics*, 23(1), 1–34.
- Quadrini, M., Tesei, L., & Merelli, E. (2020). Aspralign: a tool for the alignment of RNA secondary structures with arbitrary pseudoknots. *Bioinformatics*, 36(11), 3578–3579.

- Rastgoo, M. N., Nakisa, B., Maire, F., Rakotonirainy, A., & Chandran, V. (2019). Automatic driver stress level classification using multimodal deep learning. *Expert Systems with Applications*, 138, 112793.
- Sabour, R.M., Benezeth, Y., De Oliveira, P., Chappe, J., & Yang, F. (2021). Ubfc-phys: A multimodal database for psychophysiological studies of social stress. *IEEE Transactions on Affective Computing*.
- Sano, A., & Picard, R.W. (2013). Stress recognition using wearable sensors and mobile phones. In *2013 Humaine association conference on affective computing and intelligent interaction* (pp. 671–676). IEEE.
- Sasirekha, K., & Thangavel, K. (2020). A novel biometric image enhancement approach with the hybridization of undecimated wavelet transform and deep autoencoder. In *Handbook of research on machine and deep learning applications for cyber security* (pp. 245–269). IGI Global.
- Schmidt, P., Reiss, A., Duerichen, R., Marberger, C., & Van Laerhoven, K. (2018). Introducing WESAD, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM international conference on multimodal interaction* (pp. 400–408).
- Sharma, K., Castellini, C., van den Broek, E. L., Albu-Schaeffer, A., & Schwenker, F. (2019). A dataset of continuous affect annotations and physiological signals for emotion analysis. *Scientific Data*, 6(1), 196.
- Šikić, M., Tomić, S., & Vlahoviček, K. (2009). Prediction of protein-protein interaction sites in sequences and 3D structures by random forests. *PLoS Computational Biology*, 5(1), e1000278.
- The American Institute of Stress. <https://www.stress.org/daily-life>. Accessed: 2023-02-15.
- Verstraete, D., Ferrada, A., Droguett, E.L., Meruane, V., & Modarres, M. (2017). Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings. *Shock and Vibration* 2017.
- Wang, Z., & Oates, T. (2015a). Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In *Workshops at the twenty-ninth AAAI conference on artificial intelligence*.
- Wang, Z., & Oates, T. (2015b). Imaging time-series to improve classification and imputation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- Xu, G., Liu, M., Jiang, Z., Shen, W., & Huang, C. (2019). Online fault diagnosis method based on transfer convolutional neural networks. *IEEE Transactions on Instrumentation and Measurement*, 69(2), 509–520.
- Zeng, M., Zou, B., Wei, F., Liu, X., & Wang, L. (2016). Effective prediction of three common diseases by combining smote with tometk links technique for imbalanced medical data. In *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)* (pp. 225–228). IEEE.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

Michela Quadrini<sup>1</sup>  · Antonino Capuccio<sup>2</sup> · Denise Falcone<sup>1</sup> · Sebastian Daberdaku<sup>2</sup> · Alessandro Blanda<sup>2</sup> · Luca Bellanova<sup>2</sup> · Gianluca Gerard<sup>2</sup>

✉ Michela Quadrini  
michela.quadrini@unicam.it

Antonino Capuccio  
antonino.capuccio@latek.it

Denise Falcone  
denise.falcone@studenti.unicam.it

Sebastian Daberdaku  
sebastian.daberdaku@gmail.com

Alessandro Blanda  
alessandro.blanda@latek.it

Luca Bellanova  
luca.bellanova@latek.it

Gianluca Gerard  
gianluca.gerard@latek.it

<sup>1</sup> School of Science and Technology, University of Camerino, Via Madonna Delle Carceri 9,  
62032 Camerino, MC, Italy

<sup>2</sup> Sorint.Tek, Via Zanica 17, 24050 Grassobbio, BG, Italy