Deep Learning-Based Stress Detection using Encoded Physiological Signal Representations

A D Mahit Nandan[1], Praveen K[1†], Chinta Thejdeep R[1†], Prajna Bhat[1*†], Nagamma Patil[1†]

[1]Department of Information Technology, National Institute of Technology Karnataka, Mangalore, 575025, Karnataka, India.

*Corresponding author(s). E-mail(s): prajnaudupa.217it002@nitk.edu.in;
Contributing authors: mahitnandanad.211ai001@nitk.edu.in; praveenk.211ai028@nitk.edu.in; tejdeep.211ai013@nitk.edu.in; nagammapatil@nitk.edu.in;
†These authors contributed equally to this work.

Abstract

The increasing occurrence of stress-related health issues has highlighted the need for accurate and reliable stress detection systems. This study proposes a machine learning-based framework for stress detection using two widely recognized multimodal physiological datasets: NEURO and WESAD. We apply both classical machine learning models and deep learning architectures to classify different types of stress. To effectively capture the temporal and structural patterns in physiological time-series data, we convert raw signals into image representations using Gramian Angular Summation Fields (GAFs), Gramian Angular Difference Fields (GAFd), and Markov Transition Fields (MTF). Unlike conventional methods that transform data along the time axis, our approach performs transformations along the feature axis, enabling the resulting images to capture more robust and discriminative patterns relevant to stress classification. These image representations allow convolutional neural networks (CNN) to learn spatial features that enhance classification performance. To improve interpretability, we utilize Gradient-weighted Class Activation Mapping (Grad-CAM), which visualizes the input regions most influential in the model's decision-making, offering insights into the physiological indicators of stress. Experimental results demonstrate strong generalization on both datasets, with deep learning models using feature-axis transformations achieving around 98.3% accuracy, and classical models like XGBoost and LightGBM achieving approximately 99% accuracy. The integration of Grad-CAM further strengthens the model's interpretability by validating the relevance of learned features. Overall, our approach underscores the effectiveness of combining advanced signal transformation techniques with explainable deep learning to develop robust and interpretable stress detection systems.

Keywords: Explainable AI (XAI), Mental Health Monitoring, Physiological Signals, Stress Detection, Time-Series Encoding

1 Introduction

Stress is a nonspecific physiological response to demands which maybe internal or external demands that has a strong effect on the physical health, psychological functioning, and behavioral consequences of individuals. Over 70% of individuals in the United States report facing stress in their daily lives [1]. Persistent exposure to stress has been associated with a compromised immune response [2] and a heightened likelihood of developing conditions such as cancer [3], heart disease [3, 4], depression and psychological disorders [5], diabetes [6, 7] and substance dependence [8]. Although moderate stress is adaptive, chronic exposure is responsible for an increased risk for disorders such as insomnia, obesity, cardiovascular disease, and specific forms of cancer. Furthermore, chronic stress had always been strongly linked with mental disorders like depression and anxiety.

In the modern era, chronic stress has become ubiquitous due to pressures of contemporary life. Stress had caused 50% of total work-related sickness in 2020/21—13% more than 2015/16, as indicated by the British Health and Safety Executive. Similarly, 80% of workers are currently suffering from day-to-day stress as per the American Institute of Stress, and most require intervention to manage it . These statistics underscore the urgent need for effective, real time stress detection that can support immediate intervention and the avoidance of future health impact.

Traditional methods of stress detection rely primarily on psychological self-report measures, such as the Perceived Stress Scale. These measures, however, are inherently subjective and capture merely a fleeting state of an individual and are subject to personal bias and social desirability. To address these issues, attention is increasingly being drawn to automatic, objective, and non-invasive methods of stress detection.

The availability of Internet of Things (IoT) and wearable technologies has enabled continuous monitoring of physiological signals such as blood volume pulse (BVP), electrodermal activity (EDA), heart rate (HR), heart rate variability (HRV), body temperature (Temp), and motion using accelerometers. These signals provide rich and real-time information about a person's physiological condition and have been utilized effectively in stress detection research.

Recent developments in machine learning further enhanced the effectiveness of stress recognition systems. Conventional machine learning methods have a tendency to utilize manual feature engineering, while deep learning models like CNNs offer an auto and scalable approach by learning discriminative features in a straightforward manner from raw input data. Additionally, the use of multiple physiological signals has been found to be crucial in building accurate and generalizable models for predicting stress.

One new and promising field in this research is transforming time-series physiological data into image-based forms to allow models to learn temporal and spatial correlations more effectively. Techniques like Gramian Angular Fields (GAF), MTFs, recurrence plots, grayscale encodings, spectrograms, and scalograms.

In this work, we explore the integration of encoded physiological signals with CNNs for accurate non-invasive stress detection. We use advanced image encoding and deep learning, We aim to create a strong and scalable framework capable of accurately identifying stress in real-world environments.

2 Literature Survey

Researchers had invented numerous methods to interpret physiological signals collected by body-worn sensors for the purpose of stress detection and emotion classification [9]. Previous studies have demonstrated that physiological signal analysis can consistently detect human stress [10].

Extensive research has focused on utilizing physiological signals for the identification of stress [11–13]. Most earlier approaches looked at a combination of physiological signals, such as those recorded by electrocardiogram (ECG) sensors [14], EDA sensors [15], and electromyography (EMG) sensors [16], to detect stress and related states.

STREDWES [17] is a stress detection framework that transforms physiological signals into image representations, which are then classified using CNNs. The study evaluates different encoding methods, such as GAFs, GAFd and MTF, using datasets like NEURO, SWELL, and WESAD. Results demonstrate that STREDWES outperforms traditional methods, offering a robust, non-invasive approach to stress detection through wearable sensor data.

The authors in [18] introduced two deep learning models—a one-dimensional convolutional neural network (1D-CNN) and a multilayer perceptron (MLP)—for the classification of stress and emotions from physiological signals. They were trained end-to-end from raw sensor-captured chest- and wrist-worn signals, without manual feature engineering. Such high accuracies of 99.80% binary stress detection and 99.55% three-class emotion classification are demonstrated in the study and vastly outperform conventional machine learning methods. It highlights the capability of deep learning for long-term, non-intrusive stress and emotion monitoring.

Tzirakis et al. [19] introduced a deep neural network capable of processing video segments in order to identify stress. Mirsamadi et al. [20] used a recurrent neural network for detecting stress via speech processing in another study.

Rastgoo et al. [21] introduce a multimodal deep learning approach for detecting drivers' stress levels. It combines ECG signals, vehicle dynamics, such as steering and braking behaviors, and contextual information, such as weather and road conditions. CNNs and LSTMs together detect complex temporal and spatial patterns across modalities. Experimental findings revealed that this approach outperforms traditional machine learning models in terms of classification accuracy.

The authors in [22] categorize emotional states from EEG signals transformed to spectrograms. They use a number of techniques, such as Random Forest, K-Nearest Neighbors (KNN), and deep learning models like LSTM and CNNs. According to their results, CNNs from spectrograms perform better than typical sequence models for enhancing emotion classification of six emotions—fear, anger, joy, sadness, surprise, and disgust.

[23] used a pipeline for stress classification using physiological data from wearable devices and a Recurrent Neural Network (RNN) model. The pipeline uses various preprocessing techniques like Fourier analysis, sliding window labeling, and rolling z-score normalization. The RNN model in the WESAD dataset achieves 86% accuracy for multi-class classification and 96.5% accuracy for binary stress detection, outperforming classical machine learning strategies.

The paper "Review on Psychological Stress Detection Using Biosignals" [24], offers an in-depth investigation into psychological stress detection using different biosignals. The abstract identifies the physiological and physical biosignals (EEG, ECG, EDA, EMG, speech, respiration, skin temperature, pupil size, eye activity) that react to stressors and stresses their involuntariness and reliability. The research seeks to determine reliable, efficient, and reproducible biosignal features for effective stress detection. The procedure takes the form of a survey of literature for signal-specific studies under conditions of stress, classification of features according to type and statistical significance. The effort takes the form of detailed comparison of signal patterns, multimodal analysis, and assessment of stress induction protocols. Findings indicate stable changes in signals such as EEG asymmetry, heart rate acceleration, and skin conductance levels. The evaluation leads to applied recommendations and insights into the stability of biosignal-based stress indices for psychophysiological research and real-time use.

The article "Monitoring Stress with a Wrist Device Using Context"[25] offers a new machine-learning-based approach to real-time psychological stress detection from a wearable wrist device. The article starts with the problem of stress detection in real-life based on its subjectiveness and physiological nature. The suggested method consists of a three-module system: lab-trained stress detector, activity recognizer, and context-based classifier using biosignal data (BVP, EDA, HR, ST, IBI, ACC) and context inputs (activity level, time, etc.). The experiments were performed on 21 subjects under lab settings and were further applied to 55 days of real-life data from 5 users. Results indicated the context-aware approach outperformed the lab-only model substantially, with 70% recall and 95% precision in identifying stress events. The study verifies the robustness and feasibility of the system for real-time, unobtrusive monitoring of stress.

The study "DeStress: Deep Learning for Unsupervised Identification of Mental Stress in Firefighters from Heart-rate Variability (HRV) Data" [26] introduces an unsupervised deep learning model designed to detect mental stress in firefighters using raw RR interval data. The abstract describes the application of conventional K-Means with handcrafted features, convolutional autoencoders (CAE), and LSTM autoencoders (LAE) along with DBSCAN and KNN classifiers. The process involves HRV data collection from 100 trainee firefighters during stress-inducing training, data preprocessing, feature engineering, and unsupervised model training. The study demonstrated that

K-Means was unable to retrieve useful clusters, whereas CAE performed better than LAE in identifying unique, stress-related patterns validated using HRV markers like RMSSD and LF/HF ratio. Analysis validated CAE's clusters in line with physiological stress indicators, illustrating the model's capability for real-world, label-free stress sensing in hazardous professions.

3 Dataset Description

To assess the performance of our stress detection framework, we utilize two publicly available multimodal physiological datasets: NEURO and WESAD. Both datasets are designed to capture human physiological responses under controlled stress-inducing and relaxation scenarios. They include sensor data such as heart rate, electrodermal activity, and motion signals, recorded across various experimental conditions.

3.1 NEURO Dataset

The NEURO dataset[27] was utilized to develop and evaluate various stress detection models. This subsection describes the methods used, including data partitioning strategies and the implementation of classical machine learning models, deep learning on raw time-series data, and image-based deep learning approaches. The NEURO dataset focuses on a structured stress-induction protocol combining physical, cognitive, and emotional stressors, interleaved with periods of relaxation for baseline comparison.

The NEURO data set, acquired from 20 participants, was obtained with institutional review board ethics approval from the University of Texas at Dallas Institutional Review Board (UTD IRB #12-29). The experiment design includes seven consecutive trials that are intended to elicit and observe physiological change under different types of stress:

1. **First Relaxation (5 minutes):** Initial baseline in a relaxed, calm state.
2. **Physical Stress (5 minutes):** The protocol includes standing for one minute, followed by two minutes of paced treadmill walking (1 mph) and two minutes of brisk walking or running (3 mph).
3. **Second Relaxation (5 minutes):** Recovery period to monitor recovery toward baseline following physical activity.
4. **Mini-Emotional Stress (40 seconds):** Spontaneously elicited during reading of cognitive task instructions, which unexpectedly provoked stress responses from multiple subjects.
5. **Cognitive Stress (5 minutes):** A high-order, mathematically demanding exercise on a working memory item (counting down from 2485 in sevens for three minutes) and then the Stroop task (2 minutes) with auditory error feedback in the form of a buzzer.
6. **Third Relaxation (5 minutes):** Post-cognitive task relaxation to assess recovery.
7. **Emotional Stress (5 minutes):** Comprised of a one-minute anticipation phase followed by viewing a suspenseful clip from the horror movie *The Horde* (4 minutes).
8. **Fourth Relaxation (5 minutes):** Final recovery period to conclude the session.

The NEURO dataset provides rich physiological data reflecting real-time responses to various stress-inducing stimuli, making it highly suitable for stress detection and analysis using machine learning.

3.2 WESAD Dataset

The WESAD (WEarable Stress and Affect Detection) dataset[28] is a valuable resource attempting to make a contribution to affective computing research, specifically emotional and physiological state detection. It explores the possibility of utilizing wearable sensors to monitor stress and other emotions derived from the data gathered from chest- and wrist-wearable sensors.

The data comprises information from 15 participants, all contributing multimodal motion and physiology data collected during a controlled laboratory study. Two wearable devices, the RespiBAN chest sensor and Empatica E4 wrist sensor, recorded the data. The sensors recorded various signals such as ECG, EDA, EMG, breathing, body temperature along with tri-axial acceleration. Both of the data from the devices were synchronized by a double-tap gesture, with reference being drawn from the RespiBAN device.

RespiBAN device, worn around the chest, records a number of signals like ECG, EDA, EMG, breathing, body temperature, and acceleration in three axes with sampling frequency at 700 Hz. Empatica E4, worn around the wrist, records blood volume pulse (BVP), EDA, body temperature, and acceleration in three axes. Empatica E4 samples BVP and EDA at 64 Hz and body temperature at 4 Hz.

The SX.pkl files contain synchronized data from both the RespiBAN and Empatica E4 devices. These files are structured as dictionaries and include:

- **subject:** The unique participant identifier.
- **signal:** A dictionary containing raw data from both devices:
  - **chest:** Data from the RespiBAN device (ACC, ECG, EDA, EMG, RESP, TEMP).
  - **Wrist:** Data collected from the Empatica E4 device, including ACC, BVP, EDA, and TEMP signals.
- **Label:** The condition labels from the study protocol, sampled at 700 Hz.

For the classification task, we have focused on the following classes from the protocol: 1-**Baseline**, 2-**Stress**, 3-**Amusement**, and 4-**Meditation**. Labels 0, 5, 6, and 7 are excluded from the task.

4 Methodology

In this section, we present the methodology employed to detect stress from the NEURO and WESAD datasets. We outline the specific approaches and techniques used for each dataset, followed by a common strategy for stress factor interpretation using Grad-CAM. The fusion of these techniques forms the foundation of our stress detection framework.

4.1 NEURO

Training and evaluation using the NEURO dataset have been carried out for various models of stress detection. The processes employed in the current work include data split approaches and utilizing the conventional machine learning models, deep learning of raw time-series signals, as well as image-based deep learning methods.

4.1.1 Data Splitting Techniques

To allow robust testing of the models, two methods of data splitting were used:

- **Leave-One-Subject-Out Validation (LOSO):** Here, data from one subject was not included in the dataset, and training on the data of all other subjects was done. Then, the model was tested on the data of the subject that was left out. This was done for each subject so that the model was tested on unseen data. LOSO validation provided the model with information on how well it could generalize over different individuals, giving a more reliable measure of how well it was performing.
- **Cross-Validation Per Subject:** For this method, a unique model was trained for each single subject. For each subject, data was divided 80/20 training/test. This allowed model performance to be tested per subject, which gave insight into model generalizability to new data from the same individual. Accuracy was calculated per subject's model to establish performance.

4.1.2 Classical ML Models

In this subsection, we applied various classical machine learning models to the NEURO dataset for stress detection. The models evaluated include Support Vector Machines (SVM) with Radial Basis Function (RBF) and Polynomial (Poly) kernels, CatBoost, XGBoost, LightGBM, Random Forest, Multi-Layer Perceptron (MLP), K-Nearest Neighbors (KNN), Decision Tree, Naive Bayes, Fisher's Linear Discriminant Analysis (FLDA), Least Squares (LS), and a Voting Ensemble of LightGBM, CatBoost, and Random Forest classifiers.

The input for each model included features extracted from the NEURO dataset, including accelerometer axes (`ax`, `ay`, `az`), Electrodermal Activity (EDA), temperature (`temp`), oxygen saturation (`spo2`), and heart rate (`hr`). These features were structured in a format suitable for training models, where each row in the dataset was a distinct instance. Instances were created by concatenating segments of fixed lengths of the physiological signals. Specifically, for each subject and each stress category (`Relax`, `PhysicalStress`, `EmotionalStress`, `CognitiveStress`), the signals were split into 8 samples in the case of `ax`, `ay`, `az`, `EDA`, and `temp`, and 1 sample in the case of `spo2` and `hr`. These fragments were finally glued together in order to form a single feature vector for representing one example.

This systematic approach ensured that all models were supplied with uniform and well-formatted input data, enabling fair and stable comparison of performance.

4.1.3 Deep Learning on Raw Time-Series

In this, we employed deep learning models to analyze raw time-series data from the NEURO dataset for stress detection. The models being considered are CNNs, Long Short-Term Memory networks (LSTMs), and a CNN-LSTM hybrid architecture.

Data Preparation

The initial time-series data had features such as `ax`, `ay`, `az`, `temp`, `EDA`, `hr`, and `spo2`. They were scaled through standard scaling in order to achieve uniformity of features.

Sliding Window Approach

To make the data ready for deep learning models for LOSO approach ,we used a sliding window technique. Each window had a fixed number of time steps (e.g., 100) with a particular step size (e.g., 20). For every window, the target class was the most frequent label. This technique helped us transform the time-series data into a suitable format for training sequential models.

Data Input for Models

The formatted data was input into each of the deep learning models in the following way:

- Each sample (or window) of data was provided as a 2D array, with the rows as the time steps in the window and columns as the different features.

- These 2D arrays were stacked together to form a 3D array, the first dimension referring to the number of windows, the second dimension referring to the time steps, and the third dimension referring to the features.

- This 3D array was used as input to the CNN and LSTM models so that the models can process the temporal and spatial information in the time-series data.

This formal process ensured that all models were supplied with consistent and well-defined input data so that good learning and prediction of the stress levels from the raw time-series data could occur.

Model Architectures

CNN Model: The CNN architecture was designed to extract spatial features from time-series data. It consisted of Conv1D layers with ReLU activation to extract local patterns, followed by MaxPooling1D layers to down-sample, and dense layers to perform the final classification.

LSTM Model: It focuses on learning time-series signal temporal dependencies. The model used LSTM layers to represent information sequentially well and dense layers for making class predictions.

CNN-LSTM Model: It is a combined model that availed the strength of CNN as well as LSTM. Spatial features were extracted through initial Conv1D and MaxPooling1D layers, and the extracted features were passed through LSTM layers for temporal pattern extraction. Final dense layers did the classification.

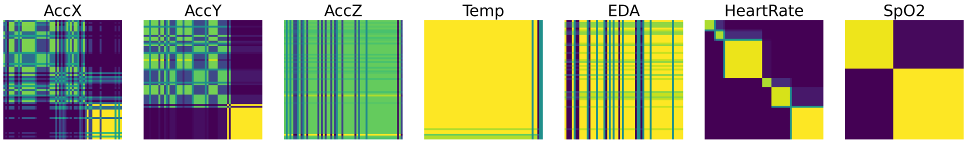

Fig. 1 **GAFd images of a single window of NEURO Dataset of Class Physical Stress**

4.1.4 Time-Series to Image-Based Deep Learning

In this sub-section, we had used deep learning models on the NEURO dataset by converting the time-series data into representations based on images. This was done using techniques like GAFs, GAFd and MTF in order to transform the time-series data into 2D images, which were fed as input to CNNs.

Data Preparation

Raw time-series data had the following attributes like `ax`, `ay`, `az`, `temp`, `EDA`, `hr`, and `spo2`. Features were standardized employing standard scaling in order to evade any variance over different features.

Sliding Window Approach

Two types of Sliding Window Approaches were used in for this task namely:

Sliding Window Approach Along Time Axis: To prepare the data for conversion into images, we employed a sliding window approach. All window consisted of a fixed number of time steps (e.g., 100) with a specified step size (e.g., 20). For each window, the most frequent label was assigned as the target class. This method enabled us to transform the time-series data into a format appropriate for converting into images.

Sliding Window Approach Along Feature Axis: The data is initially normalized using MinMaxScaler (-1 to 1), segmented with a sliding window, and reshaped by feature. Each window is transformed into 2D images using GAFs, GAFd and MTF to capture temporal and inter-feature patterns. Class imbalance is addressed via equal sampling across labels (1–4).

Time-Series to Image Conversion

The time-series data within each window was converted into images using the following techniques:

Gramian Angular Fields (GAFs): Transforms time-series data into a Gramian matrix, capturing angular relationships between time points.

Gramian Angular Fields with differences (GAFd): Similar to GAFs but incorporates differences between consecutive time points.

Markov Transition Fields (MTFs): Creates a Markov transition matrix representing the probability of transitioning between time points.

Data Input for Models

The converted image data was passed into the CNN models as 4D arrays, where dimensions corresponded to the number of windows, image height, image width, and the number of channels (features). Since we are using 7 features the input dimension to the model will be 64x64x7 as shown in image 1.

CNN Model Architecture

The CNN model architecture consisted of:

- **Input Layer:** Accepted images of shape determined by the conversion technique.
- **Convolutional Layers:** Three layers, each with 32, 64, and 128 filters, respectively, employing 3x3 kernels and ReLU activation.
- **MaxPooling Layers:** Applied after each convolutional layer with a 2x2 pool size.
- **Flatten Layer:** Converted 2D feature maps into a 1D feature vector.
- **Dense Layers:** Consisted of two fully connected layers with 128 and 64 units, respectively, utilizing ReLU activation, followed by a dropout layer for regularization.
- **Output Layer:** A dense layer with softmax activation for class probability distribution.

This architecture enabled the CNN models to effectively learn and predict stress levels from the spatial patterns and temporal dynamics captured in the image representations.

4.2 WESAD Dataset

4.2.1 Data Preparation

Sliding Windows along the Time as Axis

We transforms the physiological time-series data of the WESAD dataset into image representations of raw sensor signals by applying a sliding window approach, which is well-suited to deep learning models. The data from each subject's.pkl file is loaded in order to obtain multimodal chest sensor data (e.g., ACC, ECG, EDA, EMG, Resp, Temp) and then join it with its corresponding ground truth labels. Segments labeled from classes 1 to 4 (for various affective states) only are retained for processing.

The sliding window technique employs a constant window size of 294 time steps with a stride of 147, providing 50% overlap between adjacent windows for adequate temporal coverage. Windows are created for every physiological feature to identify short-term temporal dependencies. For the purpose of minimizing computational load and storage, only 25% of the windows are randomly sampled.

The windowed data is normalized between -1 and 1 using MinMaxScaler, ensuring compatibility with the GAF transformation. GAF, implemented via pyts.image.GramianAngularField with the 'summation' method, converts each normalized 1D time-series window into a 2D image of size 105x105, encoding temporal correlations as angular representations. These images are saved as .png files in directories organized by subject and label class.

In addition, advanced feature extraction techniques like GAFds and MTFs were implemented. GAFds extract temporal descriptors from GAF images to enhance data representation, while MTFS captures both the time and frequency domains of multivariate physiological signals. These techniques provide a richer feature set for model training, improving classification performance in affective state prediction.

Sliding Windows along the Features as Axis

The approach works on time-series data from the WESAD dataset, with physiological signals (ECG, EDA, EMG) across a number of subjects. The data is first loaded, normalized, and then converted to a structured format. Significant features include applying sliding windows to the dataset, which is reshaped along the feature dimension to analyze individual segments.

Each window, being part of time-series, undergoes GAF transformation and converts data to 2D images. Such images provide information regarding temporal as well as feature correlation. The MinMaxScaler normalizes time-series data in a consistent fashion such that every feature remains within -1 and 1.

Data is balanced by sampling an equal number of instances per label (1, 2, 3, 4). Transformed GAF images are saved in a structured directory by subject and label to provide organized access for additional analysis or machine learning tasks.

4.2.2 Deep Learning Approach

The model employs a CNN with PyTorch for image classification. The dataset contains images labeled into four labels. A 'GAFImageDataset' class is used to load and preprocess images, resizing them to 32x32 pixels and converting them to tensors. The CNN model ('CustomCNN') is made up of three convolutional layers with ReLU activation functions, followed by max-pooling and adaptive average pooling, culminating in a fully connected layer for classification.

Adam optimizer and cross-entropy loss are used to train the model, with metrics being Precision, Recall, F1-score, and Accuracy, computed via PyTorch's 'torchmetrics'. We use an 80/20 train/test split on the dataset, and the training loop is done with tqdm for tracking progress. The model is trained on a GPU if available, and the results are printed for each epoch. The model employs a learning rate of 0.001, batch size of 32, and trains for 10 epochs with the performance metrics calculated based on macro-average over 4 classes.

4.2.3 Grad-Cam

To visualize the key regions of the image that influenced the CNN model's predictions, Gradient-weighted Class Activation Mapping (Grad-CAM) was used to highlight the areas that played the most significant role in the model's decision-making process. Grad-CAM produces heatmaps by calculating the gradients of the target class with regard to the last convolutional layer, marking out significant regions that influence the class prediction. Then the emphasized regions are projected onto GAFs, either the GAFd or MTF matrices, which provide a mapping of spatial properties into an organized manner. The matrices are then mapped onto a window array to allow for further analysis and meaningful extraction of insights regarding the model's behavior and justification for prediction. Through this methodology, a more detailed comprehension of the decision-making process of the CNN model is gained by tracing the paths of connectivity between the image features and the predictive output.

4.3 Evaluation Metrics

Evaluating the effectiveness of classification algorithms involved the use of various performance metrics. These indicators are based on the confusion matrix, which provides a summary of the results provided by a binary classification model.

Table 1 Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

The following metrics help assess various components of the model's performance:

4.4 Accuracy

Accuracy reflects the ratio of correctly predicted outcomes to the overall number of cases evaluated.

$$\text{Accuracy} = \frac{TP + TN}{FN + TP + FP + TN} \tag{1}$$

4.4.1 Precision (Positive Predictive Value)

Precision quantifies the number of instances predicted as positive are actually positive.

$$\text{Precision} = \frac{TP}{FP + TP} \tag{2}$$

4.4.2 Recall (True Positive Rate)

Recall indicates how many of the truly positive instances were accurately predicted.

$$\text{Recall} = \frac{TP}{FN + TP} \tag{3}$$

4.4.3 F1 Score

The F1 Score denotes the harmonic average of Recall and Precision, provinding a balanced measure between the two. It is Highly beneficial in cases where the dataset has an uneven class distribution.

$$\text{F1 Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{4}$$

Table 2 Performance of MTF Encoding across Sliding Window Strategies and Validation Schemes

Metric	CV (Time)	CV (Feature)	LOSO (Time)	LOSO (Feature)
Accuracy	72.61	98.27	79.97	52.28
F1-Score	70.95	98.26	79.23	35.90
Precision	72.47	98.28	80.39	52.27
Recall	72.61	98.27	79.95	52.27
Specificity	82.16	99.12	88.77	47.73

Table 3 Performance of GAFs Encoding across Sliding Window Strategies and Validation Schemes

Metric	CV (Time)	CV (Feature)	LOSO (Time)	LOSO (Feature)
Accuracy	64.10	98.36	69.96	51.10
F1-Score	61.80	98.35	68.94	46.78
Precision	61.12	98.36	70.34	47.39
Recall	64.10	98.35	69.98	51.11
Specificity	78.47	99.20	84.37	70.83

Substituting from equations (2) and (3), we can also express it as:

$$\text{F1 Score} = \frac{2TP}{2TP + FN + FP} \tag{5}$$

4.4.4 Specificity (True Negative Rate)

Specificity shows how good the model identifies true negative cases out of all actual negative instances.

$$\text{Specificity} = \frac{TN}{FP + TN} \tag{6}$$

The above metrics collectively show a picture of the classification ability of a model, especially in scenarios where the class imbalance may skew the accuracy alone.

5 Results

This section summarizes the outcomes of our experiments on the NEURO and WESAD datasets. It includes model performance comparisons and interpretability analysis using Grad-CAM.

5.0.1 NEURO Dataset

The performance of all models on the NEURO dataset was assesed using two experimental setups: Cross-Validation Per Subject and Leave-One-Subject-Out (LOSO). The evaluation considered five metrics—F1-score, Precision, Recall, Specificity, and Accuracy. The dataset contains four classes representing different stress states: Cognitive Stress, Physical Stress, Emotional Stress, and Relaxation.

Evaluation of Sliding Window Strategies

Table 4 Performance of GAFd Encoding across Sliding Window Strategies and Validation Schemes

Metric	CV (Time)	CV (Feature)	LOSO (Time)	LOSO (Feature)
Accuracy	67.50	98.35	77.89	51.19
F1-Score	65.52	98.35	76.77	50.40
Precision	67.78	98.36	78.63	55.26
Recall	67.50	98.35	77.85	51.89
Specificity	79.04	99.21	86.45	76.99

We compared sliding window strategies along the time and feature axes using GAFs, GAFd, and MTF encodings. Under Cross-Validation (CV), the feature-axis approach consistently outperformed the time-axis across all encoding methods, achieving F1-scores around 98.3% as shown in Tables 2–4. In contrast, for Leave-One-Subject-Out (LOSO), the time-axis strategy proved more effective, especially for MTF (F1-score: 79.23%) compared to feature-axis (F1-score: 35.9%). This indicates that feature-axis windowing works best when training on subject-specific data, while time-axis is more robust for generalizing across individuals.

Overall, CV results significantly outperform LOSO across all models and encodings. For instance, MTF with feature-axis sliding in CV achieved an F1-score of 98.26% (Table 2), while the best LOSO setup (MTF with time-axis) achieved only 79.23%. This gap reflects the impact of inter-subject variability. Thus, while CV provides an upper bound on model performance, LOSO offers a more realistic evaluation of generalizability—crucial for real-world stress detection applications.

Evaluation of Models using LOSO and CV

As seen in Table 5, the MTF-based CNN outperformed all other methods showing an F1-score of 79.23% and accuracy of 79.97%, demonstrating strong generalization in subject-independent classification. Among classical models, Random Forest (77.63%), XGBoost (77.10%), and CatBoost (76.09%) were top performers, benefiting from robust feature learning. Deep learning models like LSTM and CNN+LSTM achieved lower accuracy (below 61%), indicating limited cross-subject generalizability without further adaptation.

In the cross-validation setup (Table 6), all models showed significantly higher performance due to the presence of subject-specific data during training. The Voting Ensemble achieved the highest accuracy (99.47%) and F1-score (98.48%), followed closely by CatBoost, LightGBM, and XGBoost. Deep learning models like 1D CNN and CNN+LSTM also performed well, with accuracies above 97.9%. Image-based models (GAF, GAFd, MTF) maintained high accuracy (above 98.2%), confirming their effectiveness for personalized classification.

Model Comparison with Base Paper

Table 7 presents a side-by-side evaluation of the models reported in the original study—based on Leave-One-Subject-Out (LOSO) validation—and our proposed approaches assessed using both LOSO and Cross-Validation (CV). While the base paper exclusively employed LOSO, our implementation under the same setting consistently outperformed the original results. For example, the Random Forest model showed a significant improvement in accuracy, exceeding the baseline by over 13%.

Table 5 Performance of models using Leave-One-Subject-Out evaluation on the NEURO dataset

Model	Accuracy	Precision	Recall	F1-score	Specificity
MLP	64.32	57.67	51.08	48.23	64.32
KNN	71.62	58.63	50.85	49.11	91.40
Random Forest	77.63	65.23	59.92	58.85	93.35
Decision Tree	77.19	49.03	50.20	47.98	93.56
Naive Bayes	72.53	60.09	48.43	49.26	92.11
FLDA	77.24	55.41	52.47	49.88	87.59
LS	41.71	25.81	30.30	20.83	82.21
XGBoost	77.10	65.62	59.48	58.14	93.28
LightGBM	73.36	61.94	56.55	53.76	91.79
CatBoost	76.09	65.46	59.79	57.89	92.68
SVM (RBF)	71.07	61.06	53.32	51.55	90.91
SVM (Poly)	72.72	63.07	49.91	47.95	91.66
Voting Ensemble	75.73	64.53	58.85	56.75	92.59
1D CNN	61.36	63.78	61.36	58.51	88.53
LSTM	60.64	63.80	60.64	57.62	88.62
CNN + LSTM	58.91	63.87	58.91	56.04	88.77
GAFs	69.96	70.34	69.98	68.94	84.37
GAFd	77.89	78.63	77.85	76.77	86.45
MTF	79.97	80.39	79.95	79.23	88.77

Similarly, CNN-based models incorporating MTF and GAFd representations demonstrated enhanced performance, particularly in F1-score and specificity. These results underscore the impact of our data processing and model optimization strategies in achieving more reliable and generalizable performance.

In the cross-validation (CV) setting, our models significantly outperformed both the baseline paper and our own Leave-One-Subject-Out (LOSO) results. All of the CNN-based models achieved accuracy, precision, recall, and specificity rates over 98%, highlighting the strong performance of our proposed framework when we reduce subject-wise variability. These findings suggest that our models are not only highly effective but also offer a solid upper bound on classification performance. Overall, this demonstrates the enhanced generalizability and effectiveness of our approach, particularly when applied to both intra- and inter-subject scenarios.

5.0.2 WESAD Dataset

The classification accuracy and overall metric performance of the CNN models as show in Table 8, which were trained on various time series encodings—GAFs, GAFd, and MTF—are seen to clearly improve in the order of their progression over 10 training epochs. Out of the three, the CNN model trained on GAFd encoding performed the best overall, with an F1-score of 78.98%, with a precision of 92.89%, recall of 81.53%, and a validation accuracy of 81.53%. This suggests that GAFd captures temporal relationships well in the time series, which makes the model better at distinguishing between patterns.

Table 6 Performance of models using Cross-Validation Per Subject on the NEURO dataset

Model	Accuracy	Precision	Recall	F1-score	Specificity
MLP	98.84	98.86	98.84	98.82	99.50
KNN	93.68	76.50	73.75	72.67	98.36
Random Forest	99.26	98.53	97.45	97.91	99.82
Decision Tree	95.64	87.99	86.17	86.20	98.89
Naive Bayes	91.76	86.19	88.06	85.77	97.80
FLDA	91.89	85.55	77.50	79.18	96.10
LS	64.34	40.59	38.02	35.41	91.15
XGBoost	99.24	98.76	97.00	97.78	99.81
LightGBM	99.34	98.86	97.65	98.19	99.84
CatBoost	99.33	98.89	97.65	98.21	99.83
SVM (RBF)	96.72	93.81	88.31	90.05	99.17
SVM (Poly)	95.64	95.35	84.16	86.89	98.90
Voting Ensemble	99.47	99.14	97.96	98.48	99.87
1D CNN	98.24	98.31	98.24	98.23	97.30
LSTM	95.51	95.70	95.51	95.27	96.80
CNN + LSTM	97.98	98.08	97.98	97.98	97.60
GAFs	98.36	98.36	98.35	98.35	99.20
GAFd	98.35	98.36	98.35	98.35	99.21
MTF	98.27	98.28	98.27	98.26	99.12

Table 7 Comparison of Base Paper and Proposed Models (LOSO and CV) on the NEURO Dataset

Model	Setting	Accuracy	F1-score	Precision	Recall	Specificity
Random Forest	Base Paper[17]	64.54%	62.34%	68.73%	64.54%	–
	LOSO (Ours)	77.63%	58.85%	65.23%	59.92%	93.35%
	CV (Ours)	99.26%	97.91%	98.53%	97.45%	99.82%
MTF + CNN	Base Paper[17]	73.34%	72.36%	76.40%	71.27%	83.84%
	LOSO (Ours)	79.97%	79.23%	80.39%	79.95%	88.77%
	CV (Ours)	98.27%	98.26%	98.28%	98.27%	99.12%
GAFs + CNN	Base Paper[17]	71.72%	69.35%	74.94%	67.41%	83.40%
	LOSO (Ours)	69.96%	68.94%	70.34%	69.98%	84.37%
	CV (Ours)	98.36%	98.35%	98.36%	98.35%	99.20%
GAFd + CNN	Base Paper[17]	73.03%	72.49%	76.32%	71.57%	84.14%
	LOSO (Ours)	77.89%	76.77%	78.63%	77.85%	86.45%
	CV (Ours)	98.35%	98.35%	98.36%	98.35%	99.21%

As a comparison, the MTF-based model was also good, with a final accuracy of 79.02% and a commendable F1 score of 77.49%, suggesting that MTF encoding is able to give meaningful state transition representations. The model trained with GAF came second, resulting in a final accuracy of 77.65% along with 74.79% as F1 score, although still showing steady improvement with epochs. Overall, despite the fact that all three encodings assisted the CNNs in learning discriminative features from time series data,

Table 8 Performance of CNN Model with Different Encoding Techniques on the WESAD Dataset - Sliding Windows along the Time as Axis

Model	Accuracy	Precision	Recall	F1-score
GAFs	77.65	71.23	77.65	74.79
GAFd	81.53	92.89	81.53	78.98
MTF	79.02	79.40	79.02	77.49

Table 9 Performance of CNN Model with Different Encoding Techniques on the WESAD Dataset - Sliding Windows along the Feature as Axis

Model	Accuracy	Precision	Recall	F1-score
GAFs	98.48	98.53	98.48	98.47
GAFd	98.92	99.05	98.92	98.97
MTF	99.57	99.52	99.57	99.54

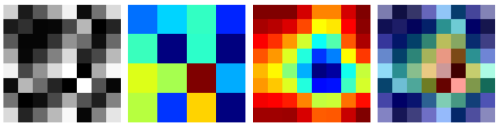

Fig. 2 Original Image, Feature Map (Conv Layer 1), Grad-CAM Heatmap, Grad-CAM Overlay for Class - 3

GAFd encoding proved to be the most effective transformation in this research, which delivered the best classification performance in all the evaluation metrics.

The CNN models trained on GAFs, GAFd, and MTF encodings along features as axis as shown in 9 all had good performance with significant differences in convergence speed and ultimate accuracy. The model trained on GAFs encoding had consistent improvement throughout epochs, attaining the most validation accuracy of 99.23% and F1 measure of 0.9925 at epoch 9. What is interesting is that though its performance started lower, it showed consistent improvement, indicating GAFs encoding contains deep temporal-spatial features the model successfully learned during the learning process.

In contrast, the CNN using GAFd encoding as training data for the model resulted in improved convergence, with over 98% accuracy reached by epoch 3 and 99.46% accuracy and a 0.9948 F1 score by epoch 8. This shows that the derivative-based GAFd is more representative of dynamic temporal changes, thus exhibiting improved early-stage learning. Likewise, the MTF-based model also ranked competitively, with the top final accuracy of 99.57% and F1 score of 0.9954 at epoch 10. This reflects MTF's ability to maintain transition probabilities in time series data. Generally, all three encodings perform well, with MTF just beating on final metrics and GAFd with better early convergence, so both are good options for time-series classification based on task needs.

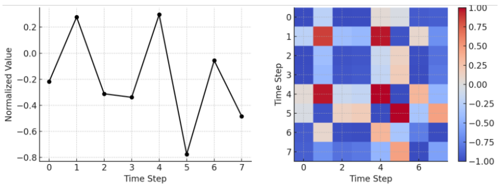

Fig. 3 **Normalised Feature Series, GAFs Matrix**

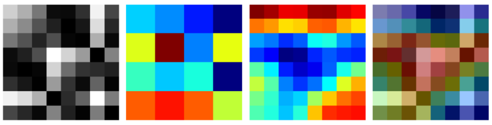

Fig. 4 **Original Image, Feature Map (Conv Layer 1), Grad-CAM Heatmap, Grad-CAM Overlay for Class - 2**

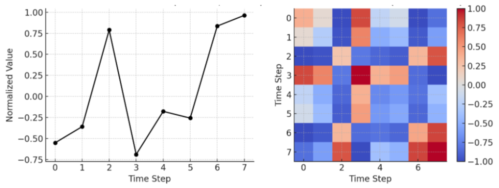

Fig. 5 **Normalised Feature Series, GAFs Matrix**

5.0.3 GRAD-CAM

The resulting GAF matrix from the normalized feature as shown in Figure 2 series retains angular cosine-based pair-wise temporal relationships. The Grad-CAM heatmap identifies hotspots of high activation within the central GAF area at time steps 3–5. These locations have abrupt contrast — a dip-peak-dip shape — shown in 3 that causes drastic angular changes and extreme GAF values (e.g., ±1), reflecting dramatic temporal changes.

These mid-sequence points are the primary contributors to class 3 prediction as shown in Figure 3. Convolutional feature maps highlight these areas, and the Grad-CAM overlay verifies that the model attention is concentrated in the vicinity of rows/columns 3–5. This indicates that local contrast and angular diversity in the time series are important discriminative features for CNN-based time series classification using GAF representations.

The Grad-CAM examination of Label 2 as shown in Figure 4, Row 0 GAF matrix demonstrates that the CNN classifier is centered on angular high-contrast regions, especially GAF[3,6], GAF[3,7], GAF[6,3], and GAF[7,3] entries. These correspond to interactions between extreme normalized coordinates—step 3 (-0.6885) and steps 6–7 (0.8353, 0.9624). High angular difference between trough and peak positions creates high cosine-based activations, which propel fundamental discriminative features in the GAF encoding.

The time series has a strong upward trend from steps 2 to 7 as show in Figure 5, resulting in angular coherence in the bottom-right quadrant of the GAF. Grad-CAM picks out this area, particularly rows/columns 6 and 7, affirming the model's dependence on angular transitions and high-magnitude contrasts. These features support the CNN's correct classification of the sequence as class 2.

6 Conclusion

From the research that has been done, it is clear that the combination of encoded physiological signals with CNNs greatly improves the accuracy of stress detection systems. Through the conversion of time-series physiological data into image representations by using methods like GAFs, GAFd, and MTF, we made it possible for CNNs to learn spatial patterns effectively, thus resulting in better classification performance. The experimental evidence on the WESAD and NEURO datasets showed consistent generalization with high accuracy, especially with the GAF-based representations. The additional usage of Grad-CAM delivered informative insights on the physiological states of stress relevance, which provided valuable information for interpreting the model's decision. This work highlights the power of integrating signal transformation techniques with interpretable deep learning for constructing effective and understandable stress detection systems toward future directions of mental health surveillance and intervention programs.

7 Future Works

Subsequent research must build upon multimodal expansion of input through fusion and context-sensitive analysis. Although the present study is based on physiological signals, combination of facial expressions, speech, and contextual metadata (e.g., activity, location, time) may lead to a richer measure of affective states. Such modalities have the ability to complement physiological measures by encoding environmental and behavioral context that affect stress levels. Coupling short-term signal processing with longitudinal analysis can also reveal early signs of chronic stress or mental health deterioration, imperative for ongoing monitoring.

In addition, applying Grad-CAM to identify high-activation regions in signal-image representations enables direct statistical analysis (e.g., peak variance, rate of change, feature correlation) to be performed on raw segments—potentially reducing dependence on advanced models while enhancing interpretability and robustness.

Improving model explainability and individualization is critical to user trust and clinical utility. New avenues include employing model-agnostic interpretation methods (e.g., SHAP, LIME) and attention mechanisms to facilitate high-accuracy temporal and spatial information. Highlighted regions can guide expert domain interpreters. Furthermore, adaptive, individualized models that learn subject-specific physiological bases and stress response—by semi-supervised or user-assisted learning—can improve classification accuracy, responsiveness, and transparency for useful stress surveillance systems.

References

[1] The American Institute of Stress: Daily Life. Accessed: 2020-02-29 (2020). https://www.stress.org/daily-life

[2] Segerstrom, S., Miller, G.: Psychological stress and the human immune system: a meta-analytic study of 30 years of inquiry. Psychological Bulletin **130**, 601–630 (2004)

[3] Cohen, S., Janicki-Deverts, D., Miller, G.: Psychological stress and disease. JAMA **298**, 1685–1687 (2007)

[4] Steptoe, A., Kivimaki, M.: Stress and cardiovascular disease: an update on current knowledge. Annual Review of Public Health **34**, 337–354 (2013)

[5] Van Praag, H.: Can stress cause depression? Progress in Neuro-Psychopharmacology and Biological Psychiatry **28**, 891–907 (2004)

[6] Heraclides, A.: Work stress, obesity and the risk of type 2 diabetes: gender specific bidirectional effect in the whitehall ii study. Obesity **20**, 428–433 (2012)

[7] Mitra, A.: Diabetes and stress: a review. Studies on Ethno-Medicine **2**, 131–135 (2008)

[8] Al'Absi, M.: Stress and Addiction: Biological and Psychological Mechanisms, 1st edn. Academic Press, Cambridge (2006)

[9] Sioni, R., Chittaro, L.: Stress detection using physiological sensors. IEEE Computer **48**, 26–33 (2015)

[10] Lundberg, U., Kadefors, R., Melin, B., Palmerud, G., Hassmen, P., Engstrom, M., Dohns, I.: Psychophysiological stress and emg activity of the trapezius muscle. International Journal of Behavioral Medicine **1**, 354–370 (1994)

[11] De Santos Sierra, A., Sanchez-Avila, C., Guerra-Casanova, J., Pozo, G.: A stress-detection system based on physiological signals and fuzzy logic. IEEE Transactions on Industrial Electronics **58**, 4857–4865 (2011)

[12] Smets, E., Rios-Velazquez, E., Schiavone, G., Chakroun, I., D'Hondt, E., De Raedt, W., Cornelis, J., Janssens, O., Van Hoecke, S., Claes, S., *et al.*: Large-scale wearable data reveal digital phenotypes for daily-life stress detection. npj Digital Medicine (2018) https://doi.org/10.1038/s41746-018-0074-9

[13] Wang, J., Lin, C., Yang, Y.: A k-nearest-neighbor classifier with heart rate variability feature-based transformation algorithm for driving stress recognition. Neurocomputing **116**, 136–143 (2013)

[14] Kyriakou, K., Resch, B., Sagl, G., Petutschnig, A., Werner, C., Niederseer, D.,

Liedlgruber, M., Wilhelm, F., Osborne, T., Pykett, J.: Detecting moments of stress from measurements of wearable physiological sensors. Sensors **19**(17), 3805 (2019) https://doi.org/10.3390/s19173805

[15] Karthikeyan, P., Murugappan, M., Yaacob, S.: Detection of human stress using short-term ecg and hrv signals. Journal of Mechanics in Medicine and Biology (2013) https://doi.org/10.1142/S0219519413500383

[16] Setz, C., Arnrich, B., Schumm, J., La Marca, R., Tröster, G., Ehlert, U.: Discriminating stress from cognitive load using a wearable eda device. IEEE Transactions on Information Technology in Biomedicine **14**, 410–417 (2010)

[17] Quadrini, M., Capuccio, A., Falcone, D., Daberdaku, S., Blanda, A., Bellanova, L., Gerard, G.: Stress detection with encoding physiological signals and convolutional neural network. Machine Learning **113**(8), 5655–5683 (2024)

[18] Li, R., Liu, Z.: Stress detection using deep neural networks. BMC Medical Informatics and Decision Making **20**, 1–10 (2020)

[19] Tzirakis, P., Trigeorgis, G., Nicolaou, M., Schuller, B., Zafeiriou, S.: End-to-end multimodal emotion recognition using deep neural networks. IEEE Journal of Selected Topics in Signal Processing **11**, 1301–1309 (2017)

[20] Mirsamadi, S., Barsoum, E., Zhang, C.: Automatic speech emotion recognition using recurrent neural networks with local attention. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 2017–2021. IEEE, New Orleans, USA (2017)

[21] Rastgoo, M.N., Nakisa, B., Maire, F., Rakotonirainy, A., Chandran, V.: Automatic driver stress level classification using multimodal deep learning. Expert Systems with Applications **138**, 112793 (2019)

[22] Nandan, A.M., Choudhary, D., Godbole, I., *et al.*: Classifying emotional states through eeg-derived spectrograms. In: 2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG), pp. 1–5 (2024). IEEE

[23] Souza, A., Melchiades, M.B., Rigo, S.J., Ramos, G.d.O.: Mostress: A sequence model for stress classification. In: 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2022). IEEE

[24] Giannakakis, G., Grigoriadis, D., Giannakaki, K., Simantiraki, O., Roniotis, A., Tsiknakis, M.: Review on psychological stress detection using biosignals. IEEE Transactions on Affective Computing **13**, 440–460 (2022) https://doi.org/10.1109/TAFFC.2019.2927337

[25] Gjoreski, M., Luštrek, M., Gams, M., Gjoreski, H.: Monitoring stress with a wrist device using context. Journal of Biomedical Informatics **73**, 159–170 (2017)

https://doi.org/10.1016/j.jbi.2017.08.006

[26] Oskooei, A., Mai Chau, S., Weiss, J., Sridhar, A., Rodríguez Martínez, M., Michel, B.: Destress: Deep learning for unsupervised identification of mental stress in firefighters from heart-rate variability (hrv) data. arXiv preprint arXiv:1911.13213 (2019)

[27] Birjandtalab, J., Cogan, D., Pouyan, M.B., Nourani, M.: A non-eeg biosignals dataset for assessment and visualization of neurological status. In: 2016 IEEE International Workshop on Signal Processing Systems (SiPS), pp. 110–114 (2016). IEEE

[28] Schmidt, P., Reiss, A., Duerichen, R., Marberger, C., Van Laerhoven, K.: 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 (2018)