Tampere University

Sami Vuolo

PHYSIOLOGICAL DATA ANALYSIS FOR STRESS DETECTION UTILIZING 1D-CNNS

Bachelor’s thesis
Faculty of Information Technology and Communication Sciences
Examiner: Dr. Fahad Sohrab
December 2025

ABSTRACT

Sami Vuolo: Physiological Data Analysis for Stress Detection utilizing 1D-CNNs
Bachelor's thesis
Tampere University
Bachelor's Degree in Computing and Electrical Engineering
December 2025

A majority of people experience stressful situations in their life. These situations differ, causing varying physiological and psychological effects on individual, many of them harmful. Because of these unhealthy responses, improving the detection of stress is crucial for efficient reacting and therefore healing these consequences of stress. Fortunately, technology has evolved to offer various different non-invasive wearable devices for measuring different biosignals, from which these stressful situations can be identified and analyzed.

This study aims to explore the capabilities of 1D-CNNs on binary classification (BC) problem of classifying wearable sensor data, and the differences in performance metrics of these data modalities. The data points are classified either stress or non-stress. For each data modality, an independent 1D-CNN is trained and tested. During testing phase, different performance metrics are calculated, and finally they are compared between different data modalities. The data used for analysis and training the networks is the publicly available Wearable Stress and Affect Detection (WESAD) dataset. The BC-problem results from WESAD are used as indicative baseline.

Based on this study, EDA and TEMP are the best data modalities for 1D-CNNs, EDA being clearly the best. Although the accuracies did not surpass 90 %, by experimenting more with the hyperparameters the results may yet be improved. Utilizing 1D-CNNs in stress detection clearly has significance, and further experimentation is needed for better results. Collecting physiological data that satisfies quality requirements is a challenging task especially for stress detection, hence improving the classification methods in speed and accuracy is crucial for benefiting from current and future stress data.

Keywords: machine learning, binary classification, stress detection, 1D-CNN

USE OF AI IN THESIS

I have utilised AI tools in my thesis:

- ☒ No
- ☐ Yes

PREFACE

I would like to express my gratitude to my supervisor, Dr. Fahad Sohrab, for providing this exciting and meaningful thesis topic, as well as for his continuous support and guidance throughout the entire process. I am also deeply thankful to everyone who supported and encouraged me along the way.

Tampere, 2nd December 2025

Sami Vuolo

CONTENTS

NOMENCLATURE

α	Learning rate
b	Bias term
f	Activation function
L	Loss function
N_n	Number of negative classes in the training data
N_p	Number of positive classes in the training data
p	Positive weight parameter in BCEWithLogitsLoss function
$\mathbf{W}$	Weight matrix
w	Old weight value
w_n	New weight value
$\mathbf{x}$	Input vector
x_{relu}	Input for ReLU function
x_{sig}	Input for sigmoid function
$\mathbf{y}$	Output activation vector
y_{relu}	Output for ReLU function
y_{sig}	Output for sigmoid function

LIST OF ABBREVIATIONS

1D	One-dimensional
1D-CNN	One-dimensional Convolutional Neural Network
BC	Binary Classification
BCE	Binary Cross-Entropy
CNN	Convolutional Neural Network
DNN	Deep Neural Network
ECG	Electrocardiogram
EDA	Electrodermal activity
EEG	Electroencephalogram
EMG	Electromyogram
FC	Fully connected
FN	False negative
FP	False positive
LDA	Linear discriminant analysis
ML	Machine Learning
NIBP	Non-Invasive Blood Pressure
PPG	Photoplethysmogram

ReLU	Rectified linear unit
RESP	respiration
RIP	Respiration inductive plethysmograph
SGD	Stochastic gradient descent
SSL	Self-supervised learning
TEMP	Body temperature
TN	True negative
TP	True positive
TSST	Trier Social Stress Test
WESAD	Wearable Stress and Affect Detection

1. INTRODUCTION

A majority of people experience stressful situations in their life. These situations differ, causing varying physiological and psychological effects on individual, many of them harmful. Because of these unhealthy responses, improving the detection of stress is crucial for efficient reacting and therefore healing these consequences of stress. Fortunately, technology has evolved to offer various different non-invasive wearable devices for measuring different biosignals, from which these stressful situations can be identified and analyzed.

The rapidly evolving field of machine learning (ML) has found numerous applications in several different areas, one of them being physiological stress detection. Many different ML algorithms have been used for stress detection in studies, such as support vector machines, random forest and k-nearest neighbors. Deep learning methods have also been used for this purpose in studies, utilizing Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) and Transformers. [1]

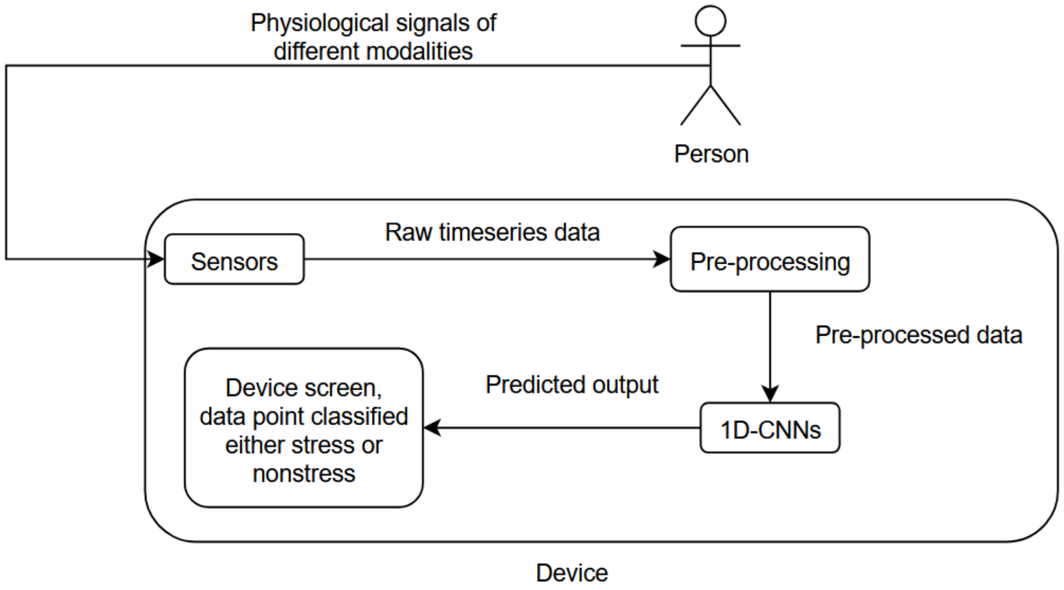

Figure 1.1. *Example of a possible application for this study*

This study aims to explore the capabilities of 1D-CNNs on binary classification (BC) problem of classifying wearable sensor data, and the differences in performance metrics of these data modalities. The data points are classified either stress or non-stress. For each data modality, an independent 1D-CNN is trained and tested. During testing phase, different performance metrics are calculated, and finally they are compared between different data modalities. The data used for analysis and training the networks is the publicly available Wearable Stress and Affect Detection (WESAD) dataset [2]. The BC-problem results from WESAD are used as indicative baseline. In figure 1.1 one possible application example for this study is depicted.

2. BACKGROUND

2.1 Stress and physiological signals

The first renowned stress researcher, Hans Hugo Bruno Seyle, initially defined stress in a generic way as "the nonspecific response of the body to any demand" [3]. Harvard Health Publishing defines stress as a feeling of emotional or physical tension. It can help face a challenge, but when the amount or duration of stress increases too much, it can be harmful to health. [4] In the book "Neuroendocrinology of stress", stress is defined with respect to homeostasis, which is the body's ability to maintain and regulate vital physiological parameters necessary for survival. The definition of stress is "a sensed threat to homeostasis (elicited by a stressor) where the stress has a degree of specificity, depending, among other things, on the particular challenge to homeostasis, the organism's perception of the stressor and its ability to cope with it". [5] As can be perceived from these definitions, defining stress concisely is challenging.

According to Harvard Health Publishing, stress can be categorized into two main types. The first is acute stress, which is stress that is rapidly-passing and short-term. If acute stress occurs repeatedly, it is called episodic acute stress. The second type is chronic stress, which is stress with greater duration, from days to months. [4] Acute stress can be beneficial by temporarily increasing performance, but it can also have negative effects. One such possible negative effect could be anhedonia [6]. Chronic stress is known to be harmfully linked to many diseases such as depression, cancer and cardiovascular disease [7].

The stress hormones cortisol and adrenaline are secreted by adrenal glands when a stressful situation is encountered. These hormones induce various physiological responses, including increased heart rate and blood pressure, dilation of airways, release of glucose into the bloodstream and diversion of blood flow away from non-essential functions. [1] These responses affect various physiological signals in human body.

The physiological signals examined in this study are electrocardiogram (ECG), electrodermal activity (EDA), electromyogram (EMG), respiration (RESP) and body temperature (TEMP). The electrocardiogram is an electrocardiographic recording in which the electrical activity of the heart is recorded with electrodes connected to the recording apparatus

[8]. As stated earlier, during stress heart rate increases, therefore measuring ECG makes these changes detectable. Electrodermal activity expresses variations in skin conductance [9]. Stress causes the sympathetic nervous system to activate, and this stimulates eccrine sweat glands to secrete sweat, which consecutively leads to changes in skin conductance [1]. The electromyogram is defined as the continuous recording of the electrical activity of a muscle produced by electrodes inserted into muscle fibers [8]. The previously mentioned stress hormones adrenaline and cortisol cause muscle tension, which can be measured with EMG [1]. Altering the respiration patterns and dilating the airways are some of the physiological responses to stress [1]. The respiration signal data used in this study were recorded by a respiration inductive plethysmograph (RIP) sensor, which can be used to calculate chest and abdominal expansion, respiratory rate, respiratory pattern, and tidal volume by using elastic bands around the chest and abdomen [10]. The body temperature data was measured from the skin, and this temperature is greatly affected by blood flow and the narrowing of blood vessels induced by stress hormones cortisol and adrenaline [1].

2.2 Stress detection methods

In order to even detect and measure stress, a stress inducing situation, stressor, must be constructed first. A study has classified these stressors into two types [1]. The first type attempts to simulate daily life situations an individual may encounter. Examples of this type of stressor used in studies include gaming, computer work, driving, video/image stimuli, email interruptions, time pressure and public speaking. The second type of stressor aims to elicit stress states by artificial procedures. These types of stressors include Stroop test, Trier Social Stress Test (TSST), Mental Arithmetic Test, IQ-test, Kraepelin Test, Rotation Letter Test, Hyperventilation Provocation Test. [1] The stress condition in data analyzed in this study was established according to the TSST [2]. This test consists of a public speaking and a mental arithmetic task. TSST is well-studied and widely used, and a reliable test for eliciting stress [1, 11].

Because of the various physiological responses induced by stress, measuring and analyzing different biosignals is a logical approach for stress detection. Many different biosignals have been used for stress detection in studies, some of them include the physiological signals defined in the previous section. Additional biosignals include Photoplethysmogram (PPG), Electroencephalogram (EEG), and Non-Invasive Blood Pressure (NIBP) [1]. PPG measures changes in the volume of blood flow in tissues like arteries and capillaries during cardiac cycle, EEG measures the electrical activity of the brain and NIBP measures blood pressure non-invasively. [1]

Whatever method is used to construct the stressor, individual test subjects may experience the stress situation differently. In order to assign ground-truth labels for collected data, different tools are used. Some of these tools are the NASA Task Load Index (NASA-TLX), the State Trait Anxiety Inventory (STAI), the Self-Assessment Manikin (SAM) and Ecological Momentary Assessments (EMAs) [1]. The WESAD-dataset used in this study included ground truth obtained by different self-questionnaires and tools, but for the purpose of this study, only the study protocol labels were used as ground truth [2].

2.3 Baseline and related research

This study analyzes WESAD dataset, which collected various biosignal data corresponding to the baseline, amusement and stress condition. The stressor was constructed using aforementioned TSST. Different classification methods were used in the related study on this data for three-class and binary classification tasks. [2] For the indicative baseline in this study, the best evaluation metrics of BC problem in data modalities measured from chest in the WESAD-dataset were used. These modalities include ECG, EDA, EMG, RESP and TEMP. For each modality, feature vectors were calculated and different classification algorithms were applied to these vectors. Accuracy and F1-score were used as evaluation metrics for the BC problem in the WESAD-paper, and the best results were from method termed linear discriminant analysis (LDA). These results are used as the indicative baseline [2].

Many researches have been conducted for stress detection with various methods and different datasets [1], some of which have used this same WESAD dataset. One such study applied 1D-CNN in the encoder part of their model for three-class classification problem [12]. Another similar research focused on the EDA modality of WESAD and pre-trained one 1D-CNN using self-supervised learning (SSL). This pre-trained netowrk was then fine-tuned to the stress prediction task, and the resulting network was compared to another 1D-CNN with the same architecture employed for stress detection. [13] Similarly, Stressalyzer-study [14] concentrated on the EDA data and trained 1D-CNN for BC task.

3. METHODOLOGY

3.1 1D-CNNs

As the name suggests, 1D-CNNs process one-dimensional data. Next sections examine deep neural networks and especially CNNs, their structure and hyperparameters. Finally, the model architecture and training setup used in this study are introduced.

3.1.1 DNNs and CNNs

The book "Accelerators for convolutional neural networks" defines deep neural networks (DNNs) as neural networks that aim to extract meaningful information and make decisions by transforming raw input into a high-level abstraction with the help of many layers between the input and output layers. The structure of DNN is the input layer, a various number of hidden layers, and the output layer. These hidden layers consist of neurons, which receive a weighted sum of input values, a bias term is added to them, and the result is then fed into a nonlinear activation function. The mathematical representation for one such neuron could be

$$\mathbf{y} = f(\mathbf{W}\mathbf{x} + b), \tag{3.1}$$

where y is the output activation vector, f is the activation function, W is the weight matrix, x is the input vector and b is the bias. If every neuron is connected to every neuron in the next layer, this layer is called a fully connected (FC) layer.

The learning of DNNs is based on the forward propagation, calculating the average loss with loss function, and then applying backpropagation. During forward propagation each training input is delivered to the DNN, which eventually produces output. A selected loss function is used to calculate the loss between the target and predicted output values. Training process aims to minimize this loss, and to achieve this, a selected optimization algorithm is used to update the weights and biases inside the DNN through backpropagation. Many optimization algorithms are based on gradient descent, which updates the weights and biases according to the following equation:

$$w_n = w - \alpha \frac{\partial L}{\partial w}, \tag{3.2}$$

where w_n denotes the new weight, w the old weight, α is the learning rate and L stands for the loss function. [15]

Convolutional neural networks (CNNs) are DNNs, which have at least one layer that utilizes the convolution operation, known as the convolutional layer. This convolutional layer consists of neurons that take as input only an enclosed area of the output of the previous layer. The weights and the biases applied on this confined input are called filters or kernels, which aim to extract certain features of the input. [15] In the case of one-dimensional convolutional neural networks (1D-CNNs), the input data is one-dimensional, so the convolution operation kernels take only a certain length of the input data from the output of the previous layer.

3.1.2 Layers of CNNs

The main layers in CNNs are obviously the convolutional layers, which have parameters for the number of kernels, kernel length, padding and stride. The number of kernels is the same as the number of learnable features and the kernel length determines the length of the kernel entries. When input is delivered into the convolutional layer, kernel starts from the beginning of the input data, calculates the dot product between the kernel entries and the input with the applied kernel length. The kernel moves forward in input data and repeats the aforementioned process. The padding establishes if zeroes are added to the beginning and the end of the input that is forwarded to CNN layer, and the stride regulates the amount of steps the kernel moves in input data. After the kernels have slided through all the input data, the resulting feature maps from dot products are forwarded to activation function. This result forms the output of the convolutional layer. The intention is for the network to learn during the training process, which filters activate while detecting some specific type of feature at some position in the input data [15].

Pooling layers downsample the resulting feature maps by applying some operation, mainly max or averaging, to the output of the previous layer. This layer has a pair of similar parameters as the convolutional layer, kernel size and stride. The kernel size determines the dimensions of the window for which the pooling operation is applied, and the stride regulates the amount of steps the kernel moves in the pooling input. For instance, if the pooling operation is max, the kernel size is 5 and the stride is 3, the kernel first takes 5 values from the input and calculates the maximum value. It then moves 3 steps in the input and repeats the process. The purpose of pooling layers is to aggregate features from narrower regions, and because of the downsampling effect, the size of data delivered forward decreases [15]. This consequently increases computational efficiency.

After the convolutional layers and the pooling layers, CNNs typically have also FC layers at the end of the network. These layers aim to produce output for classification purposes through affine transformations. The affine transformations apply multiplication between the output values of the previous layer and the weight values of the neuron connections in the FC layer, proceeded by a vector addition of bias terms [15].

3.1.3 Hyperparameters of CNNs

Hyperparameters can be defined as variables whose value or configuration is selected before training the network begins and whose values or configuration remains unchanged when training completes. These parameters determine the neural network structure and regulate the learning process. Hyperparameter tuning is the process of selecting the best hyperparameters for a neural network model. [15] Properly configuring and setting the hyperparameters of the CNN model according to the task enables the model to learn more efficiently. In the subsequent chapters these parameters are examined more closely.

One hyperparameter is the number of hidden layers and units in the CNN. Determining these numbers affects whether the model achieves desired accuracy. If the numbers are too small, the model may not learn the desired features and thus underfit. Correspondingly when the numbers are too large, the model may overfit. [15]

Drop-out layer is used to change input values entering the layer to zero with a selected drop-out probability. This probability introduces yet another hyperparameter. The intention of drop-out is to diminish the overfitting of the model and simultaneously improve the test and validation accuracy [15].

Selecting the activation functions is another hyperparameter. These functions used in this study are the rectified linear unit (ReLU) and the sigmoid function [15] . The equation for ReLU is represented as:

$$y_{relu} = \max(0, x_{relu}), \tag{3.3}$$

and the sigmoid function as:

$$y_{sig} = \frac{1}{1 + e^{-x_{sig}}}. \tag{3.4}$$

The model of this study uses ReLU after each convolutional layer, and the sigmoid function as the final step when the binary value is calculated.

Train-test split ratio is a hyperparameter that determines how much of the data is used for training and testing the model. There are many commonly used ratios, 80:20, 70:30 and 60:40. One study suggests to use the ratio $\sqrt{p}$:1, where p is the number of parameters in a linear regression model that explains the data adequately [16]. The data used for training may be splitted even further, for training and validation. Validation data is used during the training process to measure how well the model learns. When calculating the accuracy of the model on the validation data, there may be a point when the model starts to overfit. Overfitting means that the model learns the training data too excessively, learning disturbances instead of only desired features. This can be seen during the training process when the accuracy on training data increases, but the previously increasing validation accuracy starts to decrease.

Early stopping is the method of stopping the training process of the model when it starts to overfit. This can be achieved by monitoring the validation accuracy, saving the best accuracy during epochs. Hyperparameters for early stopping are delta and patience. The delta defines how much the change to the best value must be so that it is considered as new improved value. The patience determines how many epochs with no new improved value are run before the training is terminated. Thus, applying early stopping affects another hyperparameter, number of epochs the model uses for training.

Another hyperparameter is the choice of optimization algorithm that updates the weights and biases of the model during backpropagation. One such algorithm is called Adam, the "adaptive moment estimation". As the name suggests, this method estimates the first and second order moments of the gradients and calculates individual adaptive learning rates from these estimates. This algorithm is suitable for large datasets and is memory efficient [17]. For Adam, another hyperparameter must be initialized; the learning rate. This parameter functions as in the function 3.2, determining the extent of the update of weights and biases during each epoch. The larger value may cause the model to learn faster, although it may also prevent the model to converge [15].

The last examined hyperparameter is the batch size. The training data is divided into batches of this size. A batch with this number of samples is forwarded through the network after which the model parameters are updated. Thus, during one epoch all these batches pass through the network. [15]

3.2 Model architecture

In figure 3.1 the model architecture of this study is depicted. It consists of 5 convolutional layers and 2 FC layers. Each convolutional layer has the ReLU as the activation function, and each convolution operation applies 1 padding of zero value to both ends of the input. The convolution operation parameters from the first convolutional layer to the last are: 32 kernels with length of 3, 64 kernels with length of 5, 128 kernels with length of 7, 256 kernels with length of 11 and 64 kernels with length of 3. After each convolutional layer, the output serves as input for the max pooling layer with kernel size of 2 and stride size of 2.

After the last convolutional layer, the output tensor is flattened into a 1D vector, and this serves as input for a linear layer that outputs 1024 features. Then batch normalization is applied to this output, and the result is fed into the ReLU activation function, after which

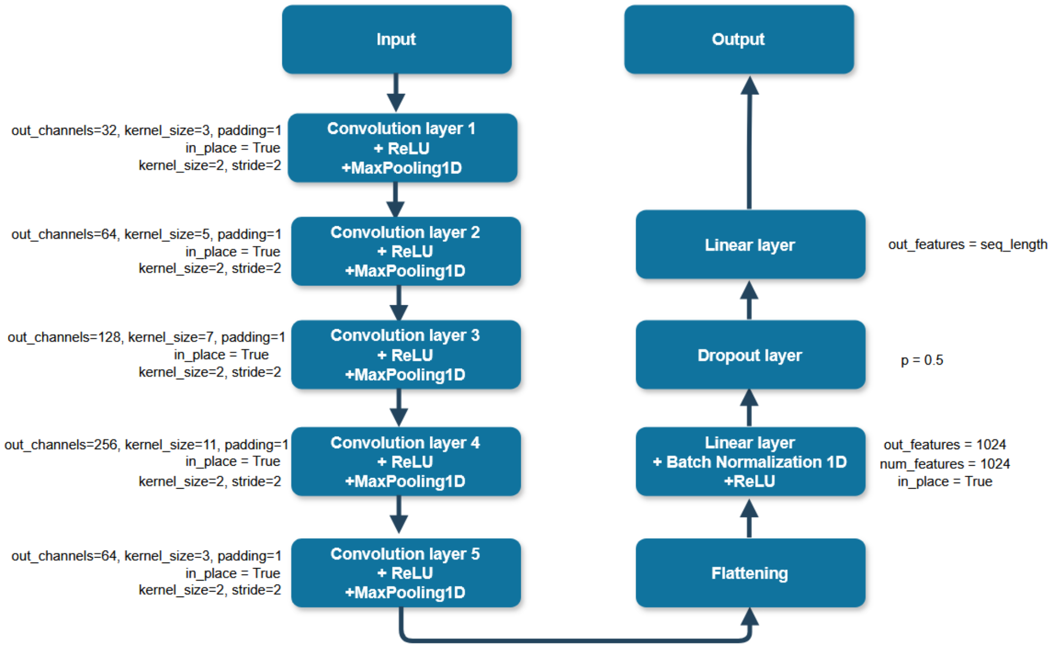

Figure 3.1. *The model architecture for 1D-CNN*

the dropout layer is applied with a probability of 50%. The result is served for the final FC layer, which outputs the result into the same length as the sequence length of one batch.

3.3 Training setup

The training data is divided into training samples of fixed sequence length. The batch size corresponds to the number of these training samples that are presented to the model network after which the model parameters are updated. In this study the batch size is set to 128 and the sequence length to 31815.

The initial number of epochs is set to 1000. Early stopping is implemented by setting the patience to 60 epochs and the delta parameter to 0.01. The binary cross-entropy (BCE) loss is selected as the loss function for the validation step, but with the PyTorch setup, the sigmoid activation function and BCE loss function are combined into one class (BCEWithLogitsLoss). Thus, the output from the last layer of the model is not served to the sigmoid function in the model itself, but in the loss function calculation step. The Adam optimizer is selected as the optimizer algorithm with the learning rate of 0.0001.

For the loss function of the training step, the same BCEWithLogitsLoss is selected, except that the parameter pos_weight is set according to the fraction:

$$p = \frac{N_n}{N_p}, \tag{3.5}$$

where p is the pos_weight-parameter, N_n is the number of negative classes in the training data and N_p is the number of positive classes in the training data.

This parameter helps the loss function to account for the class imbalance in the training dataset.

4. EXPERIMENTS

4.1 WESAD dataset description

This study analyzes WESAD, a publicly available multimodal dataset for wearable stress and affect detection. The subjects were exposed to study protocol, which aimed to elicit 3 different affective states; neutral, stress and amusement. The data was recorded from chest- and wrist-worn devices. Totally, measurements from 15 subjects were eventually accepted into dataset. This paper focuses on analyzing the raw data from modalities of chest-worn device, previously mentioned ECG, EDA, EMG, RESP and TEMP. [2]

The data was collected using measurements from chest-worn device RespiBAN Professional and wrist-worn device Empatica E4. Because of the focus on this study, only the measurements from RespiBAN are examined. The RespiBAN device was placed around the subject's chest, where the device measures RESP with its built-in sensor, a respiration inductive plethysmograph sensor. All other sensors of rest of the four modalities were connected to RespiBAN's four analog ports. The ECG was measured with a standard three point ECG, and the EDA data was recorded on the rectus abdominis. The EMG singal was measured on the upper trapezius muscle on both sides of the spine, and the TEMP signal was measured on the sternum. All these signals were sampled at 700 Hz. [2] In the subsequent paragraphs, the study protocol is described.

The two different versions of study protocols are depicted in figure 4.1. The different phases discussed in this section are baseline condition, stress condition and amusement condition. The more detailed depiction of the study protocol can be found in WESAD paper [2].

First, the subjects are equipped with the sensors. Then, the baseline condition was achieved by letting the subjects sit/stand at a table and providing neutral reading material. This condition intended to induce neutral affective state. During this 20 minute period, a baseline was recorded. [2] As can be observed in figure 4.1, in both versions of the study protocol the baseline condition is recorded first.

The stress condition was constructed using the previously mentioned TSST. For the public speaking task in the WESAD version, the subjects first had to deliver a 5 minute speech on their personal traits in front of a three-person panel, concentrating on their strengths

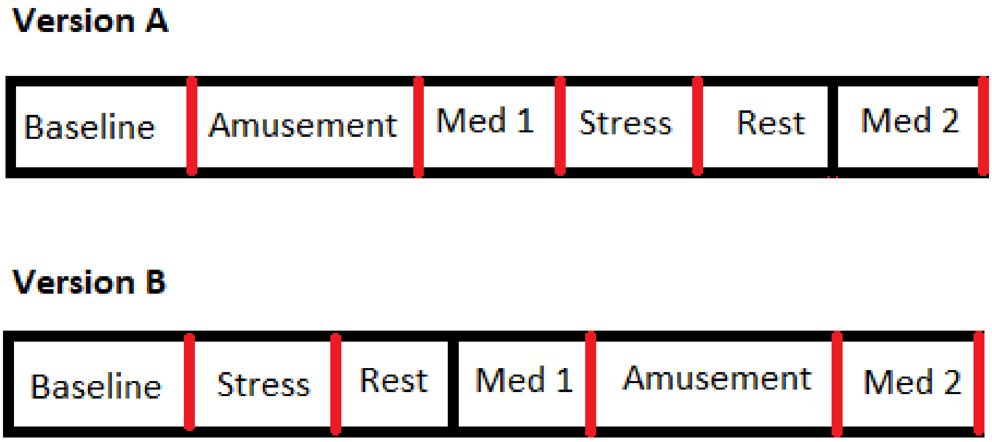

Figure 4.1. *The two different versions of study protocol. The red/dark lines refer to filling in self-reports.*

and weaknesses. For the subjects, the three panel members were portrayed as human resources specialists. The subjects were told to try to leave the best possible impression in order to improve their career options. The subjects were given 3 minutes to prepare their speech but they were not permitted to use their notes during the speech. The mental arithmetic task was constructed by asking the subjects to count from 2023 to zero, doing steps of 17. This was asked by panel, after the speech. If the subjects made a mistake, they had to start over. This task also lasted 5 minutes, so the duration of the TSST was about 10 minutes. Also, a 10 minute rest period was granted for subjects after the TSST. [2] This rest period can be seen in figure 4.1 in both versions of study protocol, following immediately after the stress condition.

In order to achieve the affective state of amusement, the subjects were shown a set of 11 funny video clips. After each clip, a short neutral sequence of 5 seconds followed. This amusement condition had a length of 392 seconds. [2]

After both the stress and amusement conditions, a guided meditation was organized for the subjects. The meditation lasted 7 minutes, during which the subjects listened to an audio track instructing a controlled breathing exercise, while sitting in comfortable position with their eyes closed. The intention of this meditation period was to bring the subjects close to neutral affective state. [2]

The study protocol was validated using five self-reports for each subject. The red lines in figure 4.1 refer to writing these self-reports. The questionnaires in these reports were formulated using several varied methods, such as Positive and Negative Affect Schedule (PANAS), Likert scale, State-Trait Anxiety Inventory (STAI), Self-Assessment Manikins (SAM) and Short Stress State Questionnaire (SSSQ). [2] However, as was previously mentioned, this study uses only the study protocol as ground truth.

The WESAD dataset includes a readme file with the following information. The data for each subject is in a PKL file, which is a dictionary. The "subject" key gives the subject ID. The "signal" key includes all the raw data in two fields, "chest" and "wrist". The "label" key gives the ID of the respective study protocol condition. The IDs are: 0 = not defined/transient, 1 = baseline, 2 = stress, 3 = amusement, 4 = meditation, 5/6/7 = ignored in this dataset. In this paper, only the chest signals are analyzed, and only the previously mentioned modalities.

4.2 Data preprocessing

Initially for each subject, the modalities ECG, EDA, EMG, RESP and TEMP from chest signals are selected. Only the data points corresponding to labels baseline, stress and amusement are included. Because this study examines the BC problem of stress and non-stress data points, data points corresponding to labels baseline and amusement are classified as non-stress points. The points corresponding to the stress are labeled positive as 1, and 0 is used as the negative label for the labels corresponding to amusement and baseline.

Then among the 15 subjects, the shortest timeseries data length was selected. All other subject's data was reshaped into this length by removing the appropriate amount of data points from the beginning of the timeseries data corresponding to baseline label. The figure 4.1 displays that all the versions of study protocol have baseline data points in the beginning of the timeseries data. Now the length of the timeseries data of all the modalities between all subjects is the same. After the modification, min-max normalization was applied for each subject and its modality differently, rescaling the values between $[-1, 1]$.

For this study, the dataset was divided by two different methods. In the first method the data of each subject is divided by 54-16-30 split, 54% for training, 16% for validation and 30% for testing. Then the method concatenates these divided data parts from all subjects into complete training, validation and testing sets.

The second method randomly selected 12 subjects for training and 3 subjects for testing. Among the training subjects, the data is divided by 80-20 split, 80 % for training and 20% for validation. Then the training and validation sets are concatenated from all the training subjects. Testing set for each modality is concatenated from the modality data of all test subjects. This second method corresponds to dividing the whole concatenated timeseries data of all subjects by 64-16-20 split, which grants more data for training and less to testing compared to the first method.

4.3 Software and Hardware

The whole experiment in this study was conducted using the open-source deep learning library PyTorch. The hardware used was limited to a NVIDIA GeForce GTX 1050 Ti GPU, with CUDA enabled for the training and testing the model. CUDA is "a parallel computing platform and programming model developed by NVIDIA for general computing on graphical processing units" [18]. It enables the use of GPU instead of CPU for the complex calculation tasks during the training and testing the model.

4.4 Training procedure and evaluation strategy

For each data modality, the model is trained and tested individually with both the previously described methods of dividing the dataset; 54-16-30 split, and randomly selecting 12 subjects for training and 3 for testing. The selected hyperparameters defined in the training setup section are the same for both models, except of course the value of pos_weight, which depends on the training data. The 1D-CNNs are trained with early stopping using the validation dataset and finally evaluated with the test dataset.

In the evaluation step for each trained 1D-CNN different statistics are calculated. These metrics are derived from true positives (TPs), true negatives (TNs), false positives (FPs) and false negatives (FNs). True positives are the number of correctly positively predicted instances, and correspondingly true negatives are the number of correctly negatively predicted instances. Respectively, false positives/negatives are the number of incorrectly positively/negatively predicted instances. [19]

The accuracy metric is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP}, \tag{4.1}$$

where TP = true positives, FN = false negatives, TN = true negatives and FP = false positives. The accuracy denotes the proportion of correctly labeled data points. [19]

The precision metric depicts the proportion of correctly predicted positives out of all positively predicted data points. It is defined as [19]:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{4.2}$$

The recall is determined by equation [19]:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{4.3}$$

This metric describes the proportion of correctly predicted positives out of all real positives.

Finally, the F1-score is calculated by using equation [19]:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{4.4}$$

The F1-score is the harmonic mean of precision and recall, and it encapsulates both of these into single metric. The closer the value is to 1, the better the model performs in predicting positive labels.

5. RESULTS

The evaluation metrics for trained 1D-CNNs of different data modalities are represented in Tables 5.1 and 5.2 corresponding to previously described ways to divide the physiological data. The indicative baseline, the best results from all different methods in WESAD study [2] for BC problem, are depicted in Table 5.3.

Table 5.1. *Evaluation metrics for 1D-CNNs trained with 54-16-30 split*

	ECG	EDA	EMG	TEMP	RESP
Accuracy	29.065	77.927	27.232	65.421	33.006
F1-score	6.802	85.653	27.065	79.074	16.946
Precision	79.623	81.352	50.172	70.719	87.598
Recall	3.553	90.435	18.531	89.667	9.380

Table 5.2. *Evaluation metrics for 1D-CNNs trained with 12 subjects' whole data*

	ECG	EDA	EMG	TEMP	RESP
Accuracy	63.357	87.027	31.571	78.644	67.444
F1-score	22.641	80.725	47.212	74.652	47.618
Precision	33.682	75.760	31.167	59.555	48.194
Recall	17.052	86.387	97.306	100.000	47.055

Table 5.3. *Some evaluation metrics on the BC task using LDA from WESAD study [2]*

	ECG	EDA	EMG	TEMP	RESP
Accuracy	85.44	81.70	67.10	69.49	88.09
F1-score	81.31	74.51	52.49	41.00	85.61

For the 1D-CNNs trained with the 54-16-30 split, the two modalities with the best accuracy and F1-score were EDA and TEMP. With the EDA data, the model achieved accuracy of 77.93% and F1-score of 85.65%. The model trained with the TEMP data accomplished accuracy of 65.42% and F1-score of 79.07%.

Correspondingly for the 1D-CNNs trained with 12 subject's whole data, the two modalities with the best accuracy and F1-score were again EDA and TEMP. With the EDA data, the model accomplished accuracy of 87.03% and F1-score of 80.73%. The model trained with the TEMP data achieved accuracy of 78.64% and F1-score of 74.65%. As can be perceived, for all models of different modalities all the accuracy and F1-scores have better, larger values with the data division using the 12 subjects' whole data. This is not unexpected, because this method provides more data for the training part of the model.

The best performing data modalities of indicative baseline in Table 5.3 are RESP and ECG. When compared to all the values of the indicative baseline modalities, the results of evaluation metrics for models trained in this study are lower in all modalities, except in the case of the EDA and TEMP data. The models trained for EDA and TEMP data achieved better F1-score with both data division methods, but only better accuracy with the 12 subjects' whole data method.

Because of computational limitations cross-validation was not applied in this study. This method would provide more reliable results and prevent overfitting.

6. CONCLUSIONS

As can be interpreted from the results, EDA and TEMP are the best data modalities for 1D-CNNs, EDA being clearly the best. Although the accuracies did not surpass 90 %, by experimenting more with the hyperparameters the results may be improved. One reason why LDA method achieved better results in indicative baseline study could be attributed to different feature vectors calculated from the data modalities [2]. This could be another way to enhance the accuracies, applying at least some feature calculation from the data before feeding feature vectors for the 1D-CNN. Although, this calculation should not be too robust, because the idea behind using the 1D-CNNs is to reduce the computation needed for classification task.

Utilizing 1D-CNNs in stress detection clearly has significance, and further experimentation is needed for better results. Collecting physiological data that satisfies quality requirements is a challenging task especially for stress detection, hence improving the classification methods in speed and accuracy is crucial for benefiting from current and future stress data.

REFERENCES

[1] Bolpagni, M., Pardini, S., Dianti, M. and Gabrielli, S. Personalized Stress Detection Using Biosignals from Wearables: A Scoping Review. eng. *Sensors (Basel, Switzerland)* 24.10 (2024), pp. 3221–. ISSN: 1424-8220.

[2] Schmidt, P., Reiss, A., Duerichen, R., Marberger, C. and Van Laerhoven, K. Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection. eng. *Proceedings of the 20th ACM International Conference on Multimodal Interaction*. New York, NY, USA: ACM, 2018, pp. 400–408. ISBN: 9781450356923.

[3] Fink, G. *Stress: Definition and History*. eng. 2009.

[4] MultiMedia LLC. *Stress*. 2024. URL: `https://www.health.harvard.edu/topics/stress` (visited on 10/10/2025).

[5] Russell, J., Shipston, M. and Ali, N. *Neuroendocrinology of stress*. eng. 1st ed. Masterclass in Neuroendocrinology Series. Chichester, England: Wiley Blackwell, 2015 - 2015. ISBN: 9781118921715.

[6] Burani, K., Brush, C. J., Eckel, L. A. and Hajcak, G. Acute stress-induced reductions in neural response to reward are related to acute stress-related increases in cortisol. eng. *Psychophysiology* 61.12 (2024), e14683–n/a. ISSN: 0048-5772.

[7] Cohen, S., Janicki-Deverts, D. and Miller, G. E. Psychological Stress and Disease. eng. *JAMA : the journal of the American Medical Association* 298.14 (2007), pp. 1685–1687. ISSN: 0098-7484.

[8] Provider, O. U. P. C. and Martin, E. A. *Concise medical dictionary*. eng. Eighth edition. Oxford paperback reference Concise medical dictionary. Oxford: Oxford University Press, 2010. ISBN: 0-19-280697-1.

[9] Gamboa, P., Varandas, R., Mrotzeck, K., Silva, H. P. d. and Quaresma, C. Electrodermal Activity Analysis at Different Body Locations. eng. *Sensors (Basel, Switzerland)* 25.6 (2025), pp. 1762–. ISSN: 1424-8220.

[10] O'Toole, M. T., Miller, B. F. and Keane, C. B. *Encyclopedia and dictionary of medicine, nursing, and allied health*. eng. 7. ed. Philadelphia (Pa.): Saunders, 2003. ISBN: 0-7216-9791-7.

[11] Vors, O., Marqueste, T. and Mascret, N. The Trier Social Stress Test and the Trier Social Stress Test for groups: Qualitative investigations. eng. *PloS one* 13.4 (2018), e0195722–. ISSN: 1932-6203.

[12] Li, J. and Washington, P. A Comparison of Personalized and Generalized Approaches to Emotion Recognition Using Consumer Wearable Devices: Machine Learning Study. eng. *JMIR AI* 3 (2024), e52171–. ISSN: 2817-1705.

[13] Islam, T. and Washington, P. Personalization of Stress Mobile Sensing using Self-Supervised Learning. eng. *arXiv.org* (2023). ISSN: 2331-8422.

[14] Sah, R. K., Cleveland, M. J., Habibi, A. and Ghasemzadeh, H. Stressalyzer: Convolutional Neural Network Framework for Personalized Stress Classification. eng. *2022 44th Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*. Vol. 2022. IEEE, 2022, pp. 4658–4663. ISBN: 1728127823.

[15] Munir, A., Kong, J. and Qureshi, M. A. *Accelerators for convolutional neural networks*. eng. Hoboken, New Jersey: Wiley, 2024. ISBN: 978-1-394-17189-7.

[16] Joseph, V. R. Optimal ratio for data splitting. eng. *Statistical analysis and data mining* 15.4 (2022), pp. 531–538. ISSN: 1932-1864.

[17] Kingma, D. P. and Ba, J. L. Adam: A method for stochastic optimization. eng. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. 2015.

[18] NVIDIA Corporation. *CUDA Toolkit*. 2025. URL: `https://developer.nvidia.com/cuda-toolkit` (visited on 11/15/2025).

[19] Powers, D. M. W. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. eng. *arXiv.org* (2020). ISSN: 2331-8422.