



# CN-waterfall: a deep convolutional neural network for multimodal physiological affect detection

Nazanin Fouladgar<sup>1</sup> · Marjan Alirezaie<sup>2</sup> · Kary Främling<sup>1,3</sup>

Received: 12 November 2020 / Accepted: 7 September 2021 / Published online: 24 September 2021  
© The Author(s) 2021

## Abstract

Affective computing solutions, in the literature, mainly rely on machine learning methods designed to accurately detect human affective states. Nevertheless, many of the proposed methods are based on handcrafted features, requiring sufficient expert knowledge in the realm of signal processing. With the advent of deep learning methods, attention has turned toward reduced feature engineering and more end-to-end machine learning. However, most of the proposed models rely on late fusion in a multimodal context. Meanwhile, addressing interrelations between modalities for intermediate-level data representation has been largely neglected. In this paper, we propose a novel deep convolutional neural network, called CN-Waterfall, consisting of two modules: *Base* and *General*. While the *Base* module focuses on the low-level representation of data from each single modality, the *General* module provides further information, indicating relations between modalities in the intermediate- and high-level data representations. The latter module has been designed based on theoretically grounded concepts in the *Explainable AI (XAI)* domain, consisting of four different fusions. These fusions are mainly tailored to *correlation-* and *non-correlation-*based modalities. To validate our model, we conduct an exhaustive experiment on WESAD and MAHNOB-HCI, two publicly and academically available datasets in the context of multimodal affective computing. We demonstrate that our proposed model significantly improves the performance of physiological-based multimodal affect detection.

**Keywords** Multimodal affect detection · Deep convolutional neural network · Physiological-based sensors · Data fusion

## 1 Introduction

Affective computing is an interdisciplinary field of research grounded mainly in neuroscience, psychology and computer science. One of the main purposes of affective

computing is to enable machines to better understand humans emotional state and accordingly assist them in different situations [1]. Given that, the direction of research in affective computing has been conducted by borrowing theories from neuroscience and psychology and constructing a computational model of the human state of mind. Common laboratory procedures in most affect detection systems rely on stimulating individuals emotions and recording their data using specific sensors. Research has shown that when a stimulus is received, a cascade of physiological processes (e.g., increased heart rate) occurs, mobilizing the nervous system and body [2]. This is one of the main rationales behind utilizing wearable sensors to collect individuals' responses (both physiological and physical) and later investigate the relationship between the measured signals and affective states.

Nowadays, physiological sensors are of great importance in different research communities [2]. The sensors on

---

✉ Nazanin Fouladgar  
nazanin@cs.umu.se

Marjan Alirezaie  
marjan.alirezaie@oru.se

Kary Främling  
kary.framling@umu.se

<sup>1</sup> Department of Computing Science, Umeå University, Umeå, Sweden

<sup>2</sup> Centre for Applied Autonomous Sensor Systems (AASS), Örebro University, Örebro, Sweden

<sup>3</sup> School of Science and Technology, Aalto University, Espoo, Finland

which our study focuses consist, primarily, of electrocardiogram (ECG), skin temperature (TEMP), electrodermal activity (EDA) (also called galvanic skin response as GSR), electromyogram (EMG) and respiration (RESP) sensors. The reason for choosing ECG is the high correlation between cardiac activity and affective states [3, 4]. Likewise, it has been argued that signals representing a reaction of the autonomic nervous system have the potential to detect human mental states [5]. Since body temperature is regulated by the human central nervous system, TEMP sensor is also a convenient candidate for affect detection. In addition to ECG and TEMP, EDA sensor measures the flow of electricity through the skin, which is strongly linked to emotional changes, inducing sweat reactions. When sweat glands become more active, they secrete moisture toward the skin surface, consequently affecting its electrical current [6]. In addition to skin surface, the electrical activity of skeletal muscles is also recorded as EMG signals. It has been proven that the EMG value of trapezius muscle activity increases under high mental workload [2, 7]. The respiratory rate as RESP values can also be influenced by the level of mental stress [2], triggering upward or downward trends under stressful or relaxing conditions, respectively. Apart from the aforementioned physiological sensors, the wearable sensors are often equipped with a three-axis accelerometer (ACC) whose values contribute to the measurement of body movements in response to affective stimuli such as stress [8].

To explore the relationship between recorded signals and affective states, a great number of machine learning (ML) solutions have been proposed. These solutions range from classical methods, including support vector machine (SVM) [9–11], AdaBoost [12], simple neural network [12–14], logistic regression (LR) [15], random forest (RF) and decision tree [9], to more sophisticated deep learning methods [16]. For instance, Kim, et al. [17] proposed the extraction of the mean energy of subband spectra as a frequency-domain feature of RESP signal based on linear discriminant analysis (LDA). By contrast, the work in [9] suggested the utilization of SVM to focus on time-domain attributes, such as the standard derivation of GSR signal. The most prominent drawback of classical approaches is their reliance on handcrafted feature engineering, requiring the considerable efforts of experts and domain knowledge professionals to extract the features of modalities (signals). Manual feature extraction approaches also entail the risk of feature redundancy and low discriminative power.

End-to-end deep learning models, however, are capable of removing the manual feature extraction from the loop by automating the process. Recently, models such as deep convolutional neural networks (CNNs) and recurrent neural

networks (RNNs) have been shown to outperform the state-of-the-art ML methods in the context of unimodal as well as multimodal affect detection [16, 18–21]. In the multimodal context, most of the models rely on *late fusion*, where multiple modalities are trained individually and merged in a final-level decision-making to detect human affective states. Such learning processes impede the extraction of information about the interrelation of modalities before decision-level fusion, specifically where a large number of modalities are available. We address this gap by providing a learning process for intermediate-level data representation based on well-established concepts in the domain of *Explainable AI* (XAI) [22]. As a result, the hidden informative features between different modalities are unveiled, allowing highly accurate decision-making.

More specifically, in this paper, we make the following contributions:

- we propose a generic deep convolutional neural network, called CN-Waterfall, applicable mainly to affect detection problems based on multimodal physiological data. Our model consists of two main modules: *Base* and *General*. While the *Base* module focuses on low-level data representation, the *General* module is concerned with intermediate- and high-level data representations.
- we introduce four fusions in our *General* module, based on our previous results [23, 24] in the practice of *Contextual Importance (CI)* and *Contextual Utility (CU)* [22] concepts in the realm of XAI. These fusions emphasize the hidden features of *correlated* and *non-correlated* modalities.
- we compare our model with both the classical and deep learning approaches. We also propose a derivative of our model, called CN-Waterfall-D. All experiments are performed on two publicly and academically available datasets in the multimodal affect detection domain, namely WESAD [25] and MAHNOB-HCI [26], respectively. Extending our approach, we also conduct a comprehensive set of experiments using different configurations of the *Base* and *General* modules.
- we acknowledge that deeper networks do not necessarily provide better performance than shallower ones in either the *Base* or *General* module. In addition, the proposed model is found to be insensitive to number of convolution filters in WESAD, while sensitive to the same setting in MAHNOB-HCI. We also conclude that CN-Waterfall and its derivative model, CN-Waterfall-D, are superior to other comparative approaches.

The rest of the paper is organized as follows. Section 2 discusses related works that have investigated affect detection primarily by means of physiological sensors. Section 3 provides details on the proposed model. We

conduct our experiments in Sect. 4. Lastly, Sect. 5 presents our conclusions and future works.

## 2 Related works

In the literature, affective computing has sometimes been formulated as emotion recognition, as systems are empowered to detect human emotions. Moreover, in the previous works, one could find *stress* as an affective state, since *stress* has been mapped [2, 27] in the prominent circumplex model of Affect [27]. In the following, we provide a brief overview of research works in both emotion and stress detection, relying on wearable (bio) sensors.

### 2.1 Emotion detection with biosensors

In 2004, Haag et al. [13] proposed a data-driven emotion-detection approach relying on different physiological modalities such as EMG, EDA, TEMP, RESP, skin conductivity (SC) and blood volume pulse (BVP). The aforementioned set of data was gathered from a single subject to verify emotion valence and arousal [27, 28] using a one hidden-layer neural network. The results were nevertheless unreliable due to the limited number of participants. Later, in 2007, Regan et al. [29] increased the number of participants to six subjects who were asked to play a computer game. In this study, five emotional states relevant to computer game play were detected, including boredom, challenge, excitement, frustration and fun. By passing the physiological data through a fuzzy logic model, the data were transformed into arousal and valence values and subsequently, by means of another fuzzy logic model, these values were converted into the target affective states. Likewise, with a focus on visual stimulation, Khalili et al. [30] employed 30 trials evoking emotional-annotated images. Different features of modalities such as EEG were extracted for classification by Quadratic Discriminant Analysis (QDA). However, due to the high number of extracted features and thereby the curse of dimensionality, the feature selection procedure was performed using a method based on a genetic algorithm (GA). GA-based methods as well as random forest recursive feature elimination were also examined in [9] to elicit informative variables from physiological modalities. Unlike our approach, the authors in [9] hypothesized that possible dependencies between variables decrease the performance of the classifier. They applied SVM to classify data into three emotions of amusement, sadness and neutral. In addition, some other recent studies have employed SVM [10, 11] for the task of affect detection. The main focus in [11] was selecting a set of low correlated features from the GSR modality, whereas the work in [10] relied on

selecting low-cost and noninvasive biosensors. In two other works [17, 31], physiological data were collected from participants while they listened to a piece of music [17] and engaged in social interaction with two experimenters [31]. The modalities investigated in the former approach [17] were EMG, ECG, SC and RESP, while the latter study [31] captured three modalities: ECG, GSR and brain activity. In a recent work [32], the authors proposed a method emphasizing the temporal sequence of emotions. More specifically, the emotion recognition task was formulated as a spectral–temporal sequence classification problem, for which a deep learning-based model with a temporal loss function was proposed as a solution for computing affective scores.

In contrast to our work, all the aforementioned solutions except [29] suffer from a delicate feature engineering process requiring sufficient domain knowledge, such as signal processing. In addition, the literature focuses on the detection of affective states other than *stress*, while, in our work, we address the detection of affective states including *stress*.

### 2.2 Stress detection with biosensors

Focusing solely on detecting the stress state, [33] presented a statistical analysis of different EMG features over five rest conditions and three interleaved stress statuses. The main aim was to investigate whether the EMG modality alone represented a promising device for the ambulatory monitoring of stress level. The results of the analysis showed that both EMG amplitude and frequency features are potential candidates for discriminating stress from non-stress (rest) conditions. Later, the authors transformed their approach into a multimodal stress detection system, by adding three other biosensors: ECG, RESP and skin conductance [34]. However, a limitation of these works was that they ignored the role of other factors than stress in the variation of modalities. To overcome this challenge, [35] decoupled respiration- and stress-induced heart rate variability (HRV), proposing a new spectral feature, while [36] isolated pressure-related variations of ECG and SC from the features set.

Shifting the focus from the features potentiality to different applications, [37] recorded physiological data from 21 participants in an algorithmic programming contest and characterized the data by several classical machine learning algorithms. In [12], an in-vehicle driving setting accompanied by an unsupervised learning model was proposed. The model consisted of an autoencoder block with a layer-wise training procedure, followed by an AdaBoost block for the purpose of stress classification. A comprehensive review of this application can be found in [38]. In another work [14], a stress recognition system was developed with

metabolic syndrome (MES) patients. The system consisted of 52 biomarkers identified from the participants which were later reduced to 15 features by a principal component analysis (PCA) module. Furthermore, a combination of a neural network and rule-based decision-making was tailored to detect four stress levels. In contrast to previous works, the authors in [39] employed a novel physiological signal, phonocardiography (PCG), to collect the stress-related data of students before an examination. To detect stress, a least-square SVM with tenfold-cross-validation was utilized. In another study, the effects of English and Urdu music tracks on the stress levels of 27 participants were examined [15]. The data were mainly recorded using electroencephalograph (EEG) signals of participants, while they were listening to different tracks. In this regard, four classifiers including logistic regression, stochastic gradient descent, sequential minimal optimization and multilayer perceptron facilitated the stress detection process. The models presented above nevertheless fell short on continuous stress detection in the natural environment. Consequently, the authors in [40] proposed a continuous classifier to predict the perception of stress incorporating prolonged physiological effects of stressors.

### 2.3 Affect-stress detection with biosensors

Although the detection of affective states other than stress was discussed in Sect. 2.1, and the detection of stress was scrutinized separately in Sect. 2.2, similar to our work, some research has focused on the detection of multiple affective states, including stress, based on physiological sensors. Such research is discussed in this section.

Schmidt et al. [25] released WESAD (see Sect. 3.1), a publicly available dataset with multimodal data and several affective states including stress. Using this dataset as a benchmark, [21] detected three states—neutral, stress and amusement—by means of four separate modules. Each module was subjected to different modalities, including data collected using different sampling rates. To extract the relevant features, an exclusive deep learning procedure was assigned per module. Similarly, in [19] a feature extraction procedure was performed with separate deep learning models. However, in that study, the models were designed to be homogeneous such that each model extracted the features of a single modality. Moreover, different affective states from those listed in [21], including neutral, stress, amusement and meditation, were detected by fusing the learned features in the final step. In other attempts, [16, 18] focused on a multitask learning in an end-to-end fashion. While in [16] four output branches were trained to simultaneously classify four affective states, including stress, in [18] multiple signal transformations were used to

create a self-supervised network intervening in the detection of neutral, stress, amusement and meditation states.

Adopting a different perspective from previous works, [41] studied a real-time system equipped with complex event processing (CEP), capable of preprocessing modalities from different sources. To detect three affective states—neutral, stress and amusement—an artificial neural network with two hidden layers was created on the server side. Such frameworks could improve the scalability of affective systems in more realistic scenarios, when large numbers of participant data streams are monitored in real time. However, these frameworks' lack of privacy and high latency may be of concern. The work in [20] proposed a scalable deep learning model to tackle these concerns by manipulating the detection structure itself. Two concepts of multiple instance learning and early stopping were employed, emphasizing the fact that only a few sub-instances contain the affective state signature. Moreover, if the classifier predicted the states with reasonable probability, the model could stop iterating over the remaining instances. Here, the focus was the elicitation of three affective states: neutral, stress and amusement.

Generally, the literature either focused on the detection of different affective states from those targeted in our research or tailored the models to application-specific systems. Furthermore, the models fail to address the extraction of information on the interrelation of modalities before any high-level decision-making fusion.

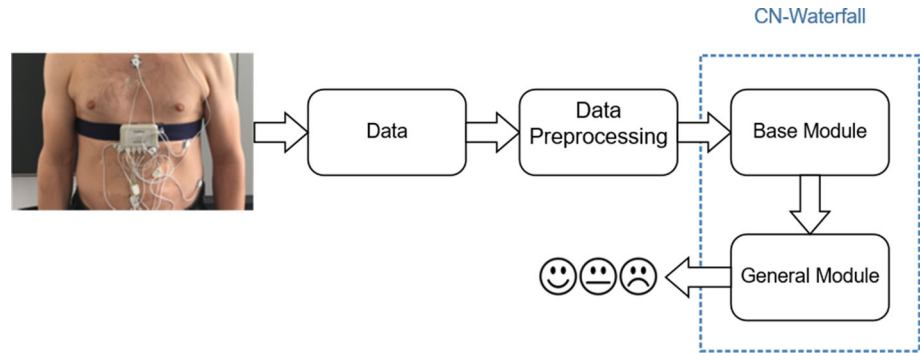
## 3 Materials and methods

Figure 1 provides an overview of our proposed framework for classifying affective states primarily from multiple physiological-based sensor data. As we explain in Sect. 3.1, the multimodal data examined in the *Data* component belong to the WESAD and MAHNOB-HCI datasets. The data are first passed through a preprocessing component (see Sect. 3.2) and then learnt by means of two modules: the *Base* (see Sect. 3.3.1) and *General* modules (see Sect. 3.3.2). These modules are the main components of our proposed model, CN-Waterfall, introduced in Sect. 3.3. In the following, we explain the details of each component as well as the theory-based justification behind the structure of our learning model.

### 3.1 Data

In this section, we describe the two datasets examined in this study and their specifications in terms of modalities as well as the data collection process.

**Fig. 1** An overview of our proposed framework on primarily physiological-based multimodal affect detection (the image on the left refers to [25])



The first dataset, WESAD, was introduced by Schmidt et al. as a publicly available dataset in the multimodal affective computing domain [25]. WESAD was created to address the laboratory-study gap in the collection of both emotional and stress states by means of wearable sensors. Chest- and wrist-worn sensors were two types of wearable sensors employed in this dataset. In our work, we focus solely on chest-worn sensors from a RespiBAN Professional device. The device was equipped with eight modalities: three-axis ACCs (ACC0, ACC1, ACC2), RESP, ECG, EDA, EMG and TEMP. Moreover, all the chest-worn modalities were recorded at a 700 HZ sampling rate.

WESAD was composed of different affective states—*neutral*, *amusement*, *stress* and *meditation*—captured from 15 participants. The *neutral* state was evoked in situations where the participants were equipped with sensors and were asked to either sit or stand at a table or read some materials for 20 minutes. The state of *amusement* was also stimulated by showing 11 funny videos to the participants for 392 seconds. Furthermore, in order to provoke *stress*, the participants were asked to deliver a 5-minute public speech in front of a three-person panel as well as to perform a 5-minute mental arithmetic task. Unlike the previous conditions, the *meditation* state was devised to de-excite the subjects and bring them back to the neutral mode [25]. For this, the participants were instructed to sit in a comfortable position, close their eyes and perform a breathing exercise for 7 minutes. To validate the study protocols, the tests were supplemented with five self-reports in the form of a questionnaire for each subject. In total, the recorded data comprised around 60 million samples.

The second multimodal dataset, MAHNOB-HCI, was introduced by Soleymani et al. as an academically available dataset, in 2012 [26]. One of the goals of data collection in MAHNOB-HCI was emotion recognition. Focusing exclusively on emotion recognition, different physiological sensors were employed to record data gathered from 27 participants. These sensors included 32 electroencephalogram (EEG) channels placed on the

participants' scalp using a head cap, two ECG electrodes attached to the upper right (ECG1) and left (ECG2) corners of the chest below the clavicle bones as well as one ECG electrode placed on the abdomen below the last rib (ECG3), two GSR sensors, namely EDA, placed on the distal phalanges of the middle (GSR1) and index fingers (GSR2), a RESP belt around the abdomen, and a skin temperature (TEMP) sensor placed on the little finger. All the modalities except EEG were sampled at 256 HZ. In the case of EEG, four bands—theta, alpha, beta, and gamma—were utilized, implying different sampling rates.

In MAHNOB-HCI, nine emotional states were stimulated in participants by playing 20 video clips. The emotional states consisted of *sadness*, *joy*, *happiness*, *disgust*, *neutral*, *amusement*, *anger*, *fear*, *surprise* and *anxiety*. After watching each clip, each participant completed a self-assessment form containing five questions on a nine point scale related to the emotional label, arousal, valence, dominance, and predictability. Similar to WESAD, this dataset also contained a considerable number of samples and thus constituted big data.

## 3.2 Data processing

To prepare the data of both datasets for the learning process, we pass the data through several preprocessing steps including *data selection*, *unification*, *downsampling*, *normalization* and *segmentation (windowing)*.

### 3.2.1 Data selection

From the WESAD dataset, as mentioned before, we focus on the modalities from all the chest-worn sensors -ACC0, ACC1, ACC2, ECG, EMG, EDA, RESP and TEMP- and, from the MAHNOB-HCI dataset, we choose seven physiological modalities -ECG1, ECG2, ECG3, GSR1, GSR2, RESP and TEMP- from the emotion recognition experiment.

Regarding the subjects and states, in WESAD, we consider the data from all the participants as well as the four affective states: *neutral*, *amusement*, *stress*, and

*meditation*. However, in MAHNOB-HCI, we choose 7 out of 30 participants and only take account of three emotional states: *amusement*, *happiness* and *surprise*. As it is obvious, WESAD includes *stress* as an affective state which is not studied in the MAHNOB-HCI dataset.

### 3.2.2 Unification and downsampling

To reduce time complexity in the learning process, we downsample the selected data. For both WESAD and MAHNOB-HCI, we perform downsampling at a rate of 10 HZ. More specifically, we randomly sample data from blocks of 700 instances in WESAD and 256 instances in MAHNOB-HCI.

The data recorded for subjects are of different lengths. We follow different unification processes for the two datasets. In order to unify the length of data in WESAD, we consider the minimum amount of data among participants as the basis of data reduction. However, in MAHNOB-HCI we focus on the amount of each emotional state in the subject with minimum data samples as the basis of data reduction. In MAHNOB-HCI, this unification process results in balanced data records for three emotional states, which is not the case for WESAD. It should be noted that, in WESAD, we first perform the downsampling and then follow the unification step, while, in MAHNOB-HCI, the order of two latter steps are reversed. Table 1 shows the number of samples for each emotional state after performing downsampling and unification in both WESAD and MAHNOB-HCI.

### 3.2.3 Normalization

To deal with varying data scales, we pass our downsampled data through a normalization process. We apply the max-min normalization as shown in Eq. 1, where  $x_{\text{norm}}$  stands for normalized data ranges between 0 and 1. Moreover,  $x_{\text{min}}$  and  $x_{\text{max}}$  are the minimum and maximum values of data in each modality:

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \quad (1)$$

**Table 1** Number of affective states in both WESAD and MAHNOB-HCI

Dataset	Affective states				
WESAD	Neutral	Stress	Amusement	Meditation	Total
	176110	99661	55749	101830	433350
MAHNOB-HCI	Amusement	Happiness	Surprised		Total
	4480	4480	4480		13440

### 3.2.4 Segmentation

Segmentation (windowing) is performed after data normalization for both datasets. During the segmentation process, the data are divided into several windows, considering an overlap between two consecutive windows. In our standard setting, we use the window sizes of 3 seconds with 1 second overlaps. Given the sampling rate of 10 HZ, the 3-second window and 1-second stride result in 30 data samples with 10 instances of overlapping. Using data segmentation, we keep temporal relationships among the data within a window.

### 3.3 CN-waterfall

In this section, we present the technical details of our proposed learning model for affective computing: CN-Waterfall. This model is inspired by recently published works based on convolutional neural networks [16, 18, 19, 21] and is equipped with modules for automatic feature extraction from physiological (e.g., ECG) as well as motion (e.g., accelerometer) modalities.

The CN-Waterfall model consists of two main components: the *Base* and *General* modules. Similar to [19], the *Base* module is used to separately extract features from each modality. However, unlike [19], the features are extracted automatically. In fact, the *Base* module provides initial and specific information of each modality in the hidden space. Since such information cannot be used to identify and extract the interrelation of modalities, we extend our learning process to the second module. The design of the second module is based on our recent research [23, 24] on the outcome explanation of fully connected neural network (FCN) and linear discriminant analysis (LDA) models in WESAD and MAHNOB-HCI. In these studies, we examined the importance and utility of modalities in the decision-making of FCN and LDA, using two concepts of *Contextual Importance (CI)* and *Contextual Utility (CU)* [22]. The concepts were coined by Främling in 1996. Theoretically, *CI* and *CU* employ a similar approach to explanation as that used by humans when explaining or justifying a decision to other humans [42]. Applying these concepts to our datasets, we found that, the number of correlated modalities with high importance values (*CI*) are fairly greater than their non-

correlated counterparts. Nevertheless, this result could not negate the impact of non-correlated modalities in the final decision. Based on these findings, in our second module, we design different fusion components to extract and learn the joint information of both correlated and non-correlated modalities. The components direct greater attention to the former modalities primarily by learning their attributes in a gradual process; by contrast, less attention is directed to non-correlated modalities by learning their attributes in an instantaneous process. Finally, we provide a high-level representation component to aggregate decisions corresponding to both correlated and non-correlated based components. Using such procedure, we explicitly present intermediate-level representations that connect the low-level features of the *Base* module to the high-level decision-making. In the following, we present the *Base* and *General* modules in detail.

### 3.3.1 CN-waterfall–Base module

The *Base* module is designed to extract and learn modality-specific features. More concisely, after feeding the pre-processed data of each modality as input, the module consists of two main blocks. The blocks provide the low-level features of each modality as signal representation (SR). Later, SRs contribute in the intermediate-level data representation in our *General* module (see Sect. 3.3.2). Figure 2 depicts the architecture of the *Base* module, explained further in the following.

As shown in Fig. 2, the first block is composed of a one-dimensional convolutional neural network (Conv1D) followed by a *Relu* activation function and a layer of one dimensional max-pooling (MaxPool1D). This block, as expressed in Eq. 2, accepts the preprocessed data of each modality, ( $signal^m \in \mathbb{R}^{ws \times 1}$ ) as input and produces the extreme features of data as output ( $MP_{f_i}^{m,1}$ ). Here, *ws* stands for window size. In the following, we present the computational details of the first block.

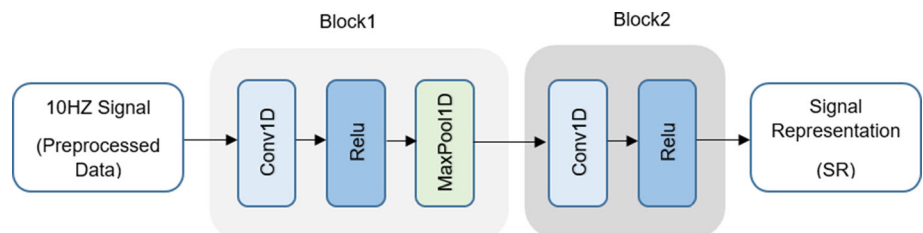
$$\begin{aligned}
 conv_{f_i}^{m,1} &= \left( \sum_{k^1} W_{f_i,k^1}^1 \times signal^m \right) + b_{f_i}^1 \\
 a_{f_i}^{m,1} &= \sigma(conv_{f_i}^{m,1}) \\
 MP_{f_i}^{m,1} &= \max_a \bigcup a_{i,f_i}^{m,1}
 \end{aligned}
 \tag{2}$$

As Eq. 2 indicates, Conv1D provides the sum of the weighted elements of each piece of modality data considering biases. We show the results of this layer as  $conv_{f_i}^{m,1}$ . In fact, applying one-dimensional convolutions enables the network to learn the time dependencies between the examined elements within the segmented data of each modality. In this convolution layer,  $b^1 \in \mathbb{R}^{F^1}$  indicates the first block biases and  $W^1 \in \mathbb{R}^{F^1 \times K^1}$  stands for the vector weights of convolution in the first block with a  $k^1 \in \{0, \dots, K^1 - 1\}$  shape as kernel size. Moreover,  $f^1 \in \{1, \dots, F^1\}$  refers to the number of times we impose different weights vectors as filters. The maximum number of filters in the first block is represented by  $F^1$ . We also set  $F^1$  and  $K^1$  to 32 and 4, respectively. Experiments with the number of filters are presented in Sect. 4.2.2.

Considering nonlinearity among extracted features and providing an appropriate input for the MaxPool1D layer, the output of Conv1D ( $conv_{f_i}^{m,1}$ ) is passed through a *Relu* activation function ( $\sigma$ ). We indicate the output of this function as  $a_{f_i}^{m,1}$ . Later, the process is followed by a downsampling step in the MaxPool1D layer, extracting the maximum values of elements in  $a_{f_i}^{m,1}$ . Finally, the output of the MaxPool1D layer is indicated as  $MP_{f_i}^{m,1} \in \mathbb{R}^{f\_no \times F^1}$ . Due to the downsampling property of this layer, we represent the size of the extracted features as  $f\_no$ .

Moving to the next block of the *Base* module, we learn the extracted features of all filters from the previous block together. The second block, shown in Fig. 2, is also composed of a one-dimensional convolutional neural network (Conv1D) fed by the output of the max-pooling layer in the first block (see Eq. 3). The Conv1D layer in the second block, however, is equipped with a higher number of filters than the first block and a reduced kernel size. This setting allows us to explore more complex features at a more granular level. The output of the Conv1D layer presents the

**Fig. 2** CN-Waterfall: the *Base* module architecture consisting of two blocks



sum of the weighted elements of  $MP_{f^1}^{m,1}$  considering biases. We refer to this output as  $conv_{f^2}^{m,2}$  for each modality as follows:

$$conv_{f^2}^{m,2} = \left( \sum_{k^2} W_{f^2,k^2}^2 \times MP_{f^1}^{m,1} \right) + b_{f^2}^2 \tag{3}$$

$$a_{f^2}^{m,2} = \sigma(conv_{f^2}^{m,2})$$

where  $W^2 \in \mathbb{R}^{F^2 \times K^2}$  refers to the vector weights of convolution in the second block with  $k^2 \in \{0, \dots, K^2 - 1\}$ , as the kernel size. Additionally,  $f^2 \in \{1, \dots, F^2\}$  shows the number of filters in the convolutional layer of the second block and  $F^2$  stands for the maximum number of filters in the same layer. We consider  $F^2$  and  $K^2$  equal to 128 and 1, respectively. Further discussion on the different number of filters can be found in Sect. 4.2.2. We also identify  $b^2 \in \mathbb{R}^{F^2}$  as the second block biases.

Finally, the model is able to learn the nonlinear relations of extracted features by passing the output of the convolutional layer ( $conv_{f^2}^{m,2}$ ) through the *ReLU* activation function ( $\sigma$ ). We refer to  $a^{m,2}$  as the result of this function, applicable for further processing in the *General* module in Sect. 3.3.2. Table 2 provides an overview of the *Base* module specifications.

### 3.3.2 CN-waterfall–General module

The *General* module, depicted in Fig. 3, utilizes the features extracted from the *Base* module and requires the informative attributes between modalities, i.e., correlation- and non-correlation-based information on these modalities. The need for such information is highlighted by our previous work in the practice of the XAI-based concepts, as described earlier. In order to quantify correlations between each pair of modalities and also preserve generalization in our model, we assume the distribution of modalities is non-Gaussian and thus employ the Spearman rank correlation coefficient [43]. In practice, this coefficient assesses the

relationship between two variables based on the rank values rather than real values. To this end, we employ the Pearson correlation [44] method for these rank values. Mathematically, the Spearman rank correlation coefficient is calculated according to Eq. 4:

$$\rho_{rm_i,rm_j} = \frac{cov(rm_i, rm_j)}{\sigma_{rm_i} \sigma_{rm_j}} \tag{4}$$

where  $\rho_{rm_i,rm_j}$  denotes the Spearman correlation coefficient of the ranked modalities  $i$  and  $j$ . We represent the rank modalities as  $rm_i$  and  $rm_j$  where  $i \neq j$  and  $i, j \in \{1, \dots, M\}$ . Here, the total number of modalities is shown as  $M$ . Moreover,  $cov$  and  $\sigma$  stand for covariance and the standard deviations of the rank variables, respectively.

The next step involves calculating the average value of the obtained correlation coefficients over the number of participants examined,  $corr\_avg_{rm_i,rm_j}$  (see Eq. 5). This calculation facilitates exploration of which modality subsets should be integrated for further feature extraction in the intermediate-level data representations. By this level of representation, we refer to different fusion components, which will be introduced later. We formulate this step as follows:

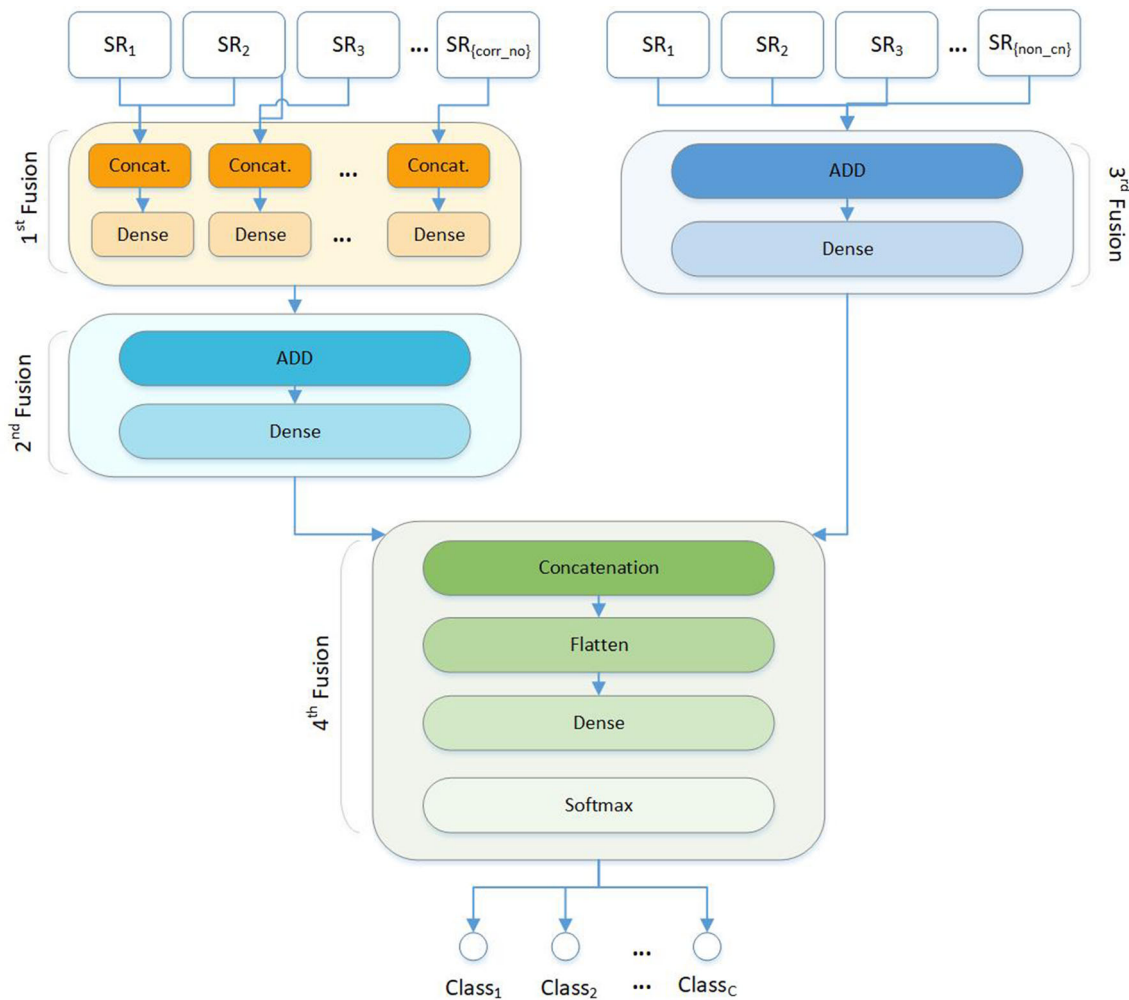
$$corr\_avg_{rm_i,rm_j} = \frac{\sum \rho_{rm_i,rm_j}}{N} \tag{5}$$

where  $N$  indicates the number of participants. If the  $corr\_avg$  value is greater than 0.4, we assume that modalities  $i$  and  $j$  are correlated and therefore incorporate their *Base* module features into the correlation-based subset,  $S_{corr}$ . By contrast, if the value is less than 0.4, we consider the modalities as non-correlated and add their *Base* module attributes to the non-correlated subset  $S_{non-corr}$ . A summary of this procedure is provided in Algorithm 1.

**Table 2** The *Base* module specifications in each layer of the two blocks

Block	Layer	Specification
1st Block	Conv1D	Filter_size = 32, kernel = 4, stride = 1, padding = ‘same’
	Activation	$ReLU(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases}$
	MaxPool1D	Pool_size = 4, kernel = 1, stride <sub>p</sub> = 2
2nd Block	Conv1D	Filter_size = 128, kernel = 1, stride = 1, no padding
	Activation	$ReLU(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases}$





**Fig. 3** CN-Waterfall: *General* module architecture, consisting of four fusion components and one classification layer

**Algorithm 1** Separate correlation- and non-correlation based modalities

**Inputs:**  $m_i, m_j$  : modality data  $i$  and  $j$ ,  $N$ : number of participants

**Output:**  $S_{corr}$  and  $S_{non-corr}$

**Initialization:**  $S_{corr} \leftarrow \emptyset, S_{non-corr} \leftarrow \emptyset$

- 1:  $\rho_{rm_i, rm_j} \leftarrow Spearmanr(m_i, m_j)$
- 2:  $corr\_avg_{rm_i, rm_j} \leftarrow Avg-Spearmanr(\rho_{rm_i, rm_j}, N)$
- 3: **if**  $corr\_avg_{rm_i, rm_j} \geq 0.4$  **then**
- 4:      $S_{corr} \leftarrow S_{corr} \cup \{m_i, m_j\}$
- 5: **else**
- 6:      $S_{non-corr} \leftarrow S_{non-corr} \cup \{m_i, m_j\}$
- 7: **end if**

As mentioned earlier, the generated subsets are added to different fusion modules to extract the interrelation information between modalities. In this regard, we incorporate three fusion modules, two of which gradually learn the hidden attributes of  $S_{corr}$  while one learns the features of the  $S_{non-corr}$  subset instantaneously. Integrating the information from these three modules in terms of intermediate-level representations, we employ a fourth fusion module

providing the final decision of our classification model. Therefore, the latter fusion represents our high-level data representation. In the following, we discuss the details of each fusion model.

**1<sup>st</sup> Fusion Module:** In the first module, each pair of modalities in the  $S_{corr}$  subset is concatenated, enabling our model to learn their joint information in a fully connected neural network. To concatenate the pairs, we employ the

extracted features of the second block from our *Base* module ( $a_{f_2}^{m,2}$ ). We define  $concat^{F_1} \in \mathbb{R}^{f\_no \times (2 * F_1^2)}$  as the concatenation output of each pair in the first fusion module as follows:

$$concat^{F_1} = [a_{f_2}^{cn,2}, a_{f_2}^{cn+1,2}] \tag{6}$$

where  $cn \in \{1, \dots, corr\_no\}$  refers to a modality in  $S_{corr}$  and  $corr\_no$  indicates the total number of modalities in this subset.

To learn the pair-wise concatenation features, we employ a one-hidden-layer neural network with 64 neurons and to detect nonlinearity, we apply *ReLU* as an activation function. We formulate the output of this layer as  $dense^{F_1} \in \mathbb{R}^{f\_no \times 64}$  in Eq. 7:

$$dense^{F_1} = \sigma \left( \left( \sum W^{F_1} \times concat^{F_1} \right) + b^{F_1} \right) \tag{7}$$

where  $\sigma$  represents the *ReLU* function. Here  $W^{F_1} \in \mathbb{R}^{64 \times (2 * F_1^2)}$  and  $b^{F_1} \in \mathbb{R}^{64 \times 1}$  refer to the weights of hidden-layer neurons and bias in the first fusion module ( $F_1$ ), respectively.

**2<sup>nd</sup> Fusion Module:** In the second fusion module, we further exploit all the pair-wised extracted features in an additive fashion accompanied by a fully connected neural network. In other words, we collect the outputs of the first fusion module and add them together. This additive layer reduces the dimensionality of the features and accordingly the learning time. Later, the corresponding information is learned by 64 neurons of one hidden-layer neural network followed by a *ReLU* activation function. We formulate the outputs of the additive layer,  $add^{F_2} \in \mathbb{R}^{f\_no \times 64}$ , and fully connected layer,  $dense^{F_2} \in \mathbb{R}^{f\_no \times 64}$ , as following:

$$add^{F_2} = [dense_1^{F_1} + dense_2^{F_1} + \dots + dense_{(corr\_no-1)}^{F_1}]$$

$$dense^{F_2} = \sigma \left( \left( \sum W^{F_2} \times add^{F_2} \right) + b^{F_2} \right) \tag{8}$$

where  $dense_i^{F_1}, i \in \{1, \dots, corr\_no - 1\}$  indicates the outputs of  $i$ th dense block in the first fusion module. Moreover,  $\sigma, W^{F_2} \in \mathbb{R}^{64 \times 64}$ , and  $b^{F_2} \in \mathbb{R}^{64 \times 1}$  stand for the *ReLU* activation function, weights of hidden neurons, and bias, respectively. The elicited features of this module are further used as the inputs of the 4<sup>th</sup> fusion module in high-level decision-making.

**3<sup>rd</sup> Fusion Module:** Parallel to the first and second fusion modules, the third fusion module is applied to the non-correlated subset,  $S_{non-corr}$ . In line with the theoretically grounded concepts discussed earlier, we employ an instantaneous feature extraction and learning process for the non-correlated-based modalities. To this end, an

additive layer and a fully connected layer are incorporated as the building blocks of this fusion module. Here,  $add^{F_3}$  and  $dense^{F_3}$  refer to the outputs of the former and latter layers, respectively, and are formulated as below:

$$add^{F_3} = [a_{f_2}^{1,2} + a_{f_2}^{2,2} + \dots + a_{f_2}^{non-cn,2}]$$

$$dense^{F_3} = \sigma \left( \left( \sum W^{F_3} \times add^{F_3} \right) + b^{F_3} \right) \tag{9}$$

where  $a_{f_2}^{i,2}, i \in \{1, \dots, non\_cn\}$  indicates the output of the *Base* module, regarding the non-correlated subset modalities. We represent the total number of modalities in this subset as  $non\_cn$ . Additionally,  $\sigma, W^{F_3} \in \mathbb{R}^{F_3 \times 64}$ , and  $b^{F_3} \in \mathbb{R}^{64 \times 1}$  stand for the *ReLU* activation function, the weights of hidden neurons and bias in the third fusion module ( $F_3$ ), respectively.

**4<sup>th</sup> Fusion Module:** The fourth fusion module, representing the high-level features, integrates the information extracted from the correlated and non-correlated subsets. To this end, the outputs of the second and third fusions are concatenated, enabling the model to classify the affective states. For this, a one hidden-layer neural network with the same number of neurons as the number of affective states is employed. However, we first flatten the inputs of this network to provide consistency with the number of affective states in the decision space. In the final step, the network is followed by a *Softmax* activation function, generating the probability of each affective state as the final decision. To define this module, we formulate each layer in the following way:

$$concat^{F_4} = [dense^{F_2}, dense^{F_3}]$$

$$flat^{F_4} = Flatten(concat^{F_4})$$

$$dense^{F_4} = \sum (W^{F_4} \times flat^{F_4}) + b^{F_4} \tag{10}$$

$$out = Softmax(dense^{F_4})$$

where  $concat^{F_4} \in \mathbb{R}^{f\_no \times (2 * 64)}$  indicates the output of the concatenation layer. In addition,  $flat^{F_4} \in \mathbb{R}^{f\_no \times 2 \times 64}$  and  $dense^{F_4} \in \mathbb{R}^{class\_no}$  represent the inputs and outputs of hidden neurons in the neural network. Here,  $class\_no$  refers to the number of affective states in the problem space. For simplicity, we define  $FN = f\_no \times 2 \times 64$  and, respectively, show the weights and bias of the neural network in the fourth fusion model ( $F_4$ ) as  $W^{F_4} \in \mathbb{R}^{FN \times class\_no}$  and  $b^{F_4} \in \mathbb{R}^{class\_no}$ . Finally, *out* stands for the final outputs of CN-Waterfall as human affective states.

## 4 Experiments and results

In the following sections we compare the performance of our proposed method, CN-Waterfall, with that of other models including the classical and deep learning methods. We also present technical details on the evaluation of the CN-Waterfall model.

As mentioned earlier, we conduct our experiments on two multimodal affective computing datasets: WESAD and MAHNOB-HCI. Table 3 provides a summary of the standard parameter settings used to evaluate the CN-Waterfall model.

As demonstrated in Table 3, the experiments are performed 10 times with an epoch size of 20. We also randomly split both datasets into training and test sets with an 80:20 ratio. In each epoch, we consider 50 windows of 3-second data fed into our model.

Finally, we calculate the average performance of the model(s) in terms of accuracy, precision, recall and F1-score. The optimizer used is Adam and the loss function is based on the categorical cross-entropy (CE) as follows:

$$CE = \sum_x P(x) \log Q(x) \quad (11)$$

where  $x$ ,  $P(x)$  and  $Q(x)$  refer to the class, probability of class  $x$  in the target and probability of class  $x$  in prediction, respectively.

### 4.1 Comparison of models

This section compares the performance of CN-Waterfall with classical and deep learning models. For the classical models, we choose linear discriminative analysis (LDA) and AdaBoost and for the deep models we propose Multichannel-CNN and CN-Waterfall-D.

**Table 3** Standard parameter settings in WESAD & MAHNOB-HCI datasets

Parameters	WESAD	MAHNOB-HCI
Window size	3 s (3*10 samples)	3 s (3*10 samples)
Window shift	1 s (1*10 samples)	1 s (1*10 samples)
Sampling rate	10 HZ	10 HZ
No. subjects	15	7
No. states (classes)	4	3
Data distribution	Imbalanced	Balanced
Batch size	50	50
Training epochs	20	20
Iterations	10	10
Optimizer	Adam	Adam

The architecture of the Multichannel-CNN model is inspired by the work in [19] and is depicted in Fig. 4a. For each modality, we assign a channel. The first two blocks also resemble the *Base* module in CN-Waterfall, following a fully connected neural network (dense) with 64 hidden neurons. In the final step, Multichannel-CNN employs the fourth fusion structure of the *General* module as well as the last activation function in CN-Waterfall. Therefore, the captured data representations are concatenated, flattened and categorized using a dense layer, equipped with a softmax activation function.

In addition, CN-Waterfall-D is a derivative of our proposed model that preserves the main CN-Waterfall architecture. As illustrated in Fig. 4b, CN-Waterfall-D follows each signal representation extracted from the *Base* module as well as the four fusion components of the *General* module in CN-Waterfall. However, two additional links, concerned with the second and third fusions, are also incorporated into the CN-Waterfall-D architecture, which constitutes the main difference with CN-Waterfall. The red arrows in Fig. 4b indicate these links.

As mentioned above, the Multichannel-CNN architecture does not include the correlation and non-correlation based fusions. Therefore, comparing CN-Waterfall with Multichannel-CNN provides insights into how the fusions contribute to the performance of CN-Waterfall. Moreover, comparison between CN-Waterfall with CN-Waterfall-D indicates how optimal the decision-level fusion inputs are in the CN-Waterfall structure.

Comparing the results in WESAD, Table 4 shows that CN-Waterfall outperforms all other models except CN-Waterfall-D. The difference between the results of CN-Waterfall and the classical models is around 28% on average, whereas the difference between CN-Waterfall and Multichannel-CNN is an average of roughly 4%. In the case of CN-Waterfall-D, similar results to CN-Waterfall are achieved, implying that feeding compound attributes to the late fusion fail to provide the model with additional information. This argument is also valid for the MAHNOB-HCI dataset (Table 5). According to this table, CN-Waterfall again performs better than the classical approaches and Multichannel-CNN in MAHNOB-HCI. Here, the performance difference between CN-Waterfall and LDA is around 40%, while CN-Waterfall outperforms AdaBoost and Multichannel-CNN by around 9% and 6% respectively.

Comparing the results of the two datasets, we can observe that AdaBoost performs much better in the MAHNOB-HCI dataset than in WESAD, with around 15% higher accuracy, precision, recall and F1-score. However, the performance of LDA and Multichannel-CNN decreases by an average of around 10% and 2%, respectively. Regarding CN-Waterfall and CN-Waterfall-D, both models

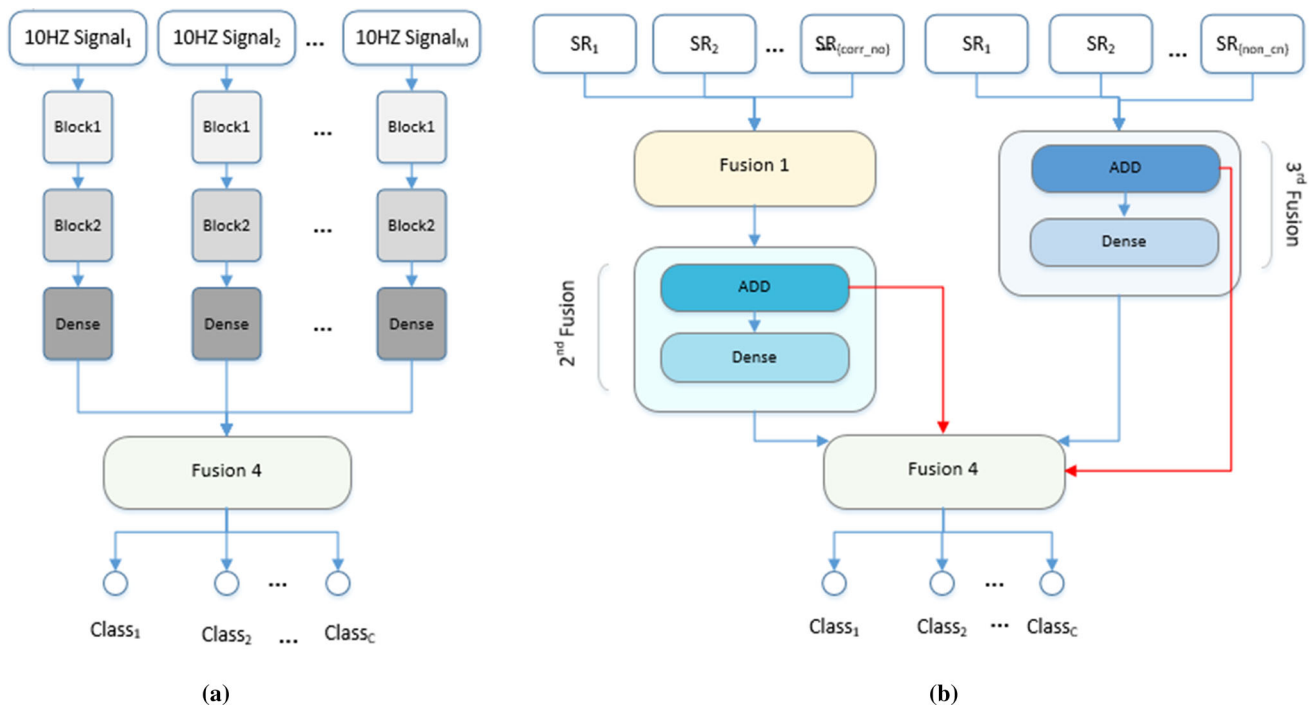


Fig. 4 a Multichannel-CNN and b CN-Waterfall-D architectures

Table 4 Comparison of deep vs. classical models in WESAD

Models	Accuracy	Precision	Recall	F1-score
Linear discriminative analysis (LDA)	0.709	0.687	0.709	0.688
AdaBoost	0.740	0.723	0.740	0.721
Multichannel-CNN	0.952	0.944	0.939	0.952
CN-Waterfall	0.990	0.988	0.989	0.990
CN-Waterfall-D	0.990	0.988	0.988	0.990

Table 5 Comparison of deep vs. classical models in MAHNOB-HCI

Models	Accuracy	Precision	Recall	F1-score
Linear discriminative analysis (LDA)	0.597	0.597	0.597	0.595
AdaBoost	0.894	0.894	0.894	0.894
Multichannel-CNN	0.924	0.927	0.925	0.924
CN-Waterfall	0.987	0.988	0.987	0.987
CN-Waterfall-D	0.987	0.988	0.987	0.987

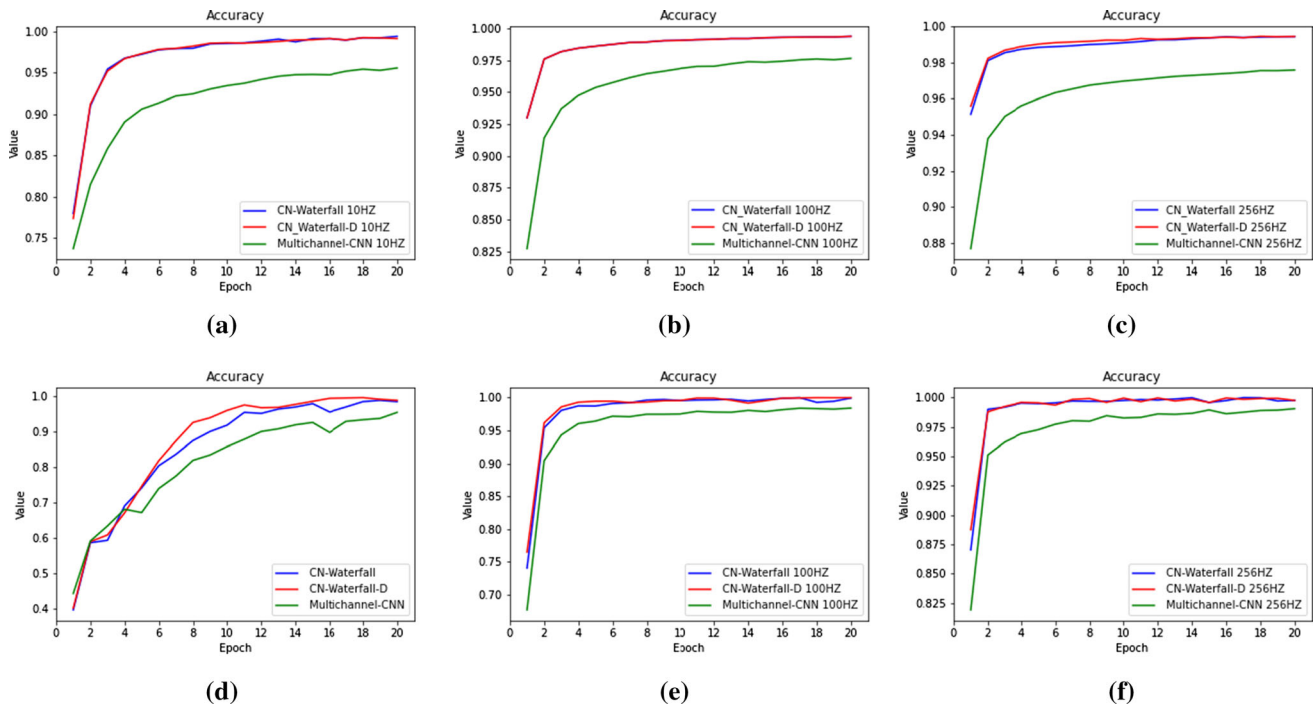
perform quite similarly in both datasets in terms of the aforementioned metrics.

#### 4.1.1 Effects of different sampling rates in deep learning models

In this section, we present our experiments on different data sampling rates. The results indicate the sensitivity of the models to the granularity of the data in terms of accuracy. Due to the considerable superiority of the deep learning models (see Tables 4 and 5), in this section we

exclude the classical models and only investigate the performance of deep learning models at data sampling rates of 10 HZ, 100 HZ and 256 HZ.

Figure 5 illustrates the result of our experiments on both datasets. As can be inferred from Fig. 5a–c, in WESAD, CN-Waterfall and CN-Waterfall-D perform similarly for the data of all sampling rates. However, the performance of the CN-Waterfall family is clearly superior to Multichannel-CNN. This superiority is an average of 4% in 10 HZ data and 2% in both 100 HZ and 256 HZ data. At a more granular level, the higher the sampling rate is, the higher



**Fig. 5** Comparison of CN-Waterfall, CN-Waterfall-D and Multichannel-CNN performances with **a** 10 HZ, **b** 100 HZ and **c** 256 HZ sampling rates in WESAD and **d** 10 HZ, **e** 100 HZ and **f** 256 HZ in MAHNOB-HCI

the accuracy will be in the primitive epochs of all models. CN-Waterfall and CN-Waterfall-D achieved an accuracy of 80% within the 10 HZ data, rising to 92% and 95% for the 100 HZ and 256 HZ data respectively, at the beginning of learning. In case of Multichannel-CNN, the accuracy is found to be around 75% for the 10 HZ data, and increasing to 82% and 88% at 100 HZ and 256 HZ sampling rates, respectively. Likewise, the results reveal that CN-Waterfall and CN-Waterfall-D achieve an accuracy of around 95% at the flattening point within the 10 HZ data, increasing to 97% and 98% at 100 HZ and 256 HZ sampling rates, respectively. The same was found for Multichannel-CNN, yet with lower accuracy than the two other models: around 90%, 95% and 96% are reported at the beginning of the flattening point for 10 HZ, 100 HZ, and 256 HZ data, respectively.

Regarding the results with MAHNOB-HCI datasets, Fig. 5d–f shows the models performance at 10 HZ, 100 HZ and 256 HZ sampling rates, respectively. Unlike in WESAD, CN-Waterfall-D demonstrates clear superiority over CN-Waterfall with 10 HZ data after six epochs. However, this is no longer the case at 100 HZ and 256 HZ sampling rates. In the latter experiments, a high overlap is observed in the performance of CN-Waterfall-D and CN-Waterfall. Although for 10 HZ data there are some fluctuations in all three models performance, it is clear that CN-Waterfall-D and CN-Waterfall perform better than the Multichannel-CNN model overall. Moreover, the former

models preserve their superiority at 100 HZ and 256 HZ sampling rates as well. The results show that while CN-Waterfall-D and CN-Waterfall generate average accuracies of around 98%, 99% and 99% for 10 HZ, 100 HZ, and 256 HZ data respectively, Multichannel-CNN underperforms, with an average accuracy of around 92%, 97%, and 98% for 10 HZ, 100 HZ, and 256 HZ data. We also see a slight arc in the performance curves of the three models in the 10 HZ experiments. The curves, however, arc more deeply at the 100 HZ and 256 HZ sampling rates, subsequent to the consistent accuracy of the models in the early stages.

#### 4.1.2 Effects of different windowing on deep learning models

As mentioned before, we report our experiments on the deep learning models due to their superiority over the classical models. In this section, we investigate how different window sizes and overlapping data blocks influence the performance of the three deep learning models. The results indicate the duration of the physiological and motion responses required to recognize the affective states. As mentioned in Sect. 3.2 and shown in Table 3, in the standard setting, we use a window size (ws) and shift size (ss) of 30 and 10 samples, respectively. To demonstrate the impact of windowing on the performance of the models, we further examine the models with two more window sizes of 60 and 120 samples as well as shift sizes of 20, 40 and 80.

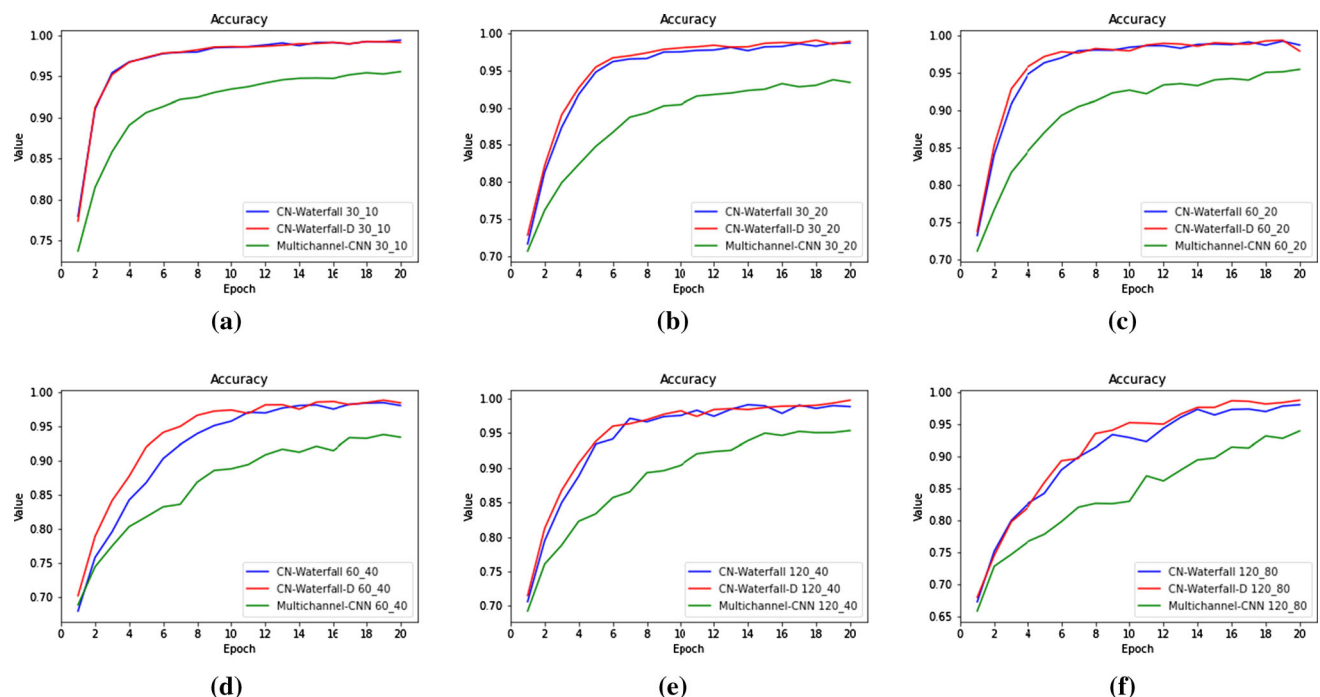
Figure 6 shows the results of our experiments in WESAD. As discussed before, the performance of CN-Waterfall and CN-Waterfall-D overlaps in the standard setting (see Fig. 6a). The same is true when  $ws = 30$  and  $ss = 20$  as well as  $ws = 60$  and  $ss = 20$  (see Fig. 6b and c). Although there are slight fluctuations in the performances of CN-Waterfall and CN-Waterfall-D for  $ws = 120$  and  $ss = 40$ , their accuracy, on average, overlaps (see Fig. 6e). In all the experiments, these two models considerably outperform Multichannel-CNN. The largest difference (about 8%) is when  $ws = 60$  and  $ss = 40$ , as the average accuracy of both CN-Waterfall and CN-Waterfall-D models is around 98% while the comparable accuracy of Multichannel-CNN is around 90% (see Fig. 6d). In addition, all the models perform best with the standard setting, with 99% accuracy for CN-Waterfall and CN-Waterfall-D and around 95% accuracy for the Multichannel-CNN model. Moreover, the lowest performance is when  $ws = 120$  and  $ss = 80$  (see Fig. 6f), as the curve performs a shallow arc in comparison with the other experiments. In addition, the fluctuations in the models performance increase when the window size is enlarged. The highest variations among all the models are evident in the last experiment (Fig. 6f).

Regarding the MAHNOB-HCI dataset, Fig. 7 indicates the effects of windowing on the performance of the aforementioned models. As can be seen, a high fluctuation over 20 epochs is evident in all the experiments except the

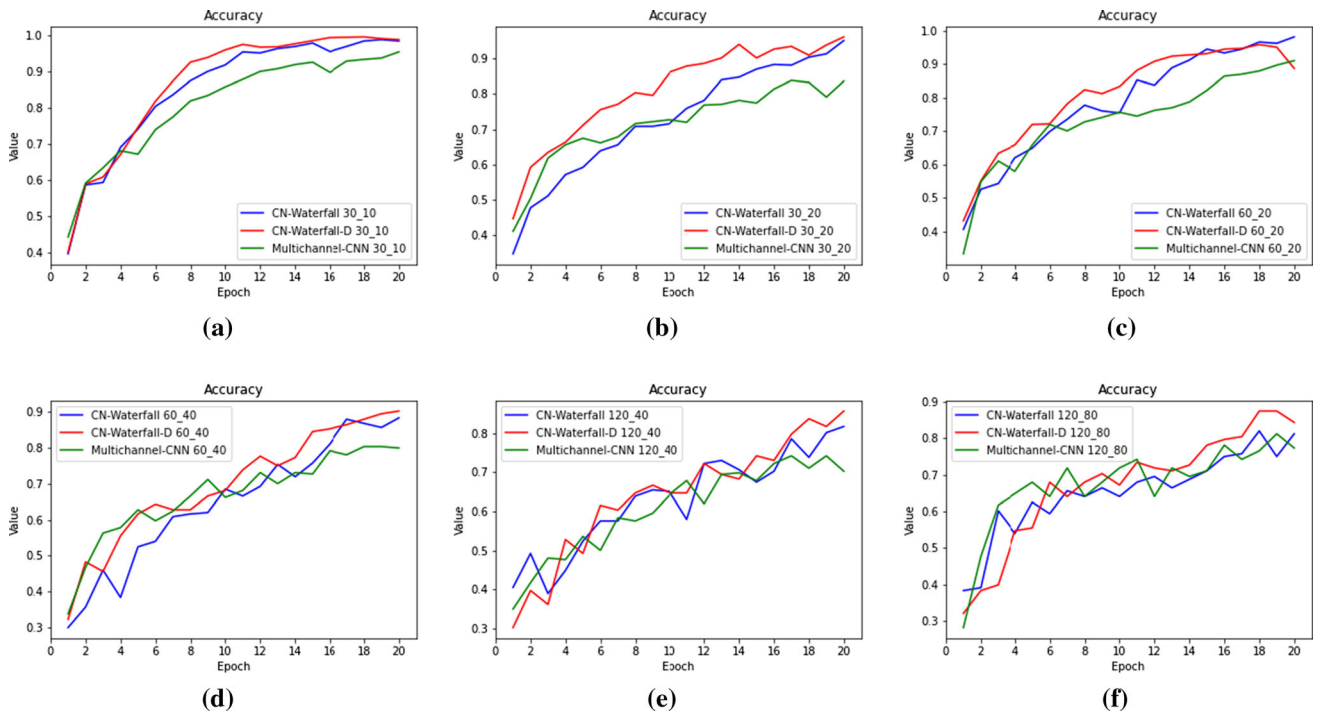
standard setting (see Fig. 7a). These fluctuations impede arc-shaped curves and thereby convergent accuracies. We also find the three models achieve their best performance with  $ws = 30$  and  $ss = 10$  (standard setting) compared to the other settings. Similar to the WESAD result, the models perform worst when  $ws = 120$  and  $ss = 80$  (see Fig. 7f). In this experiment, the average accuracy of Multichannel-CNN, CN-Waterfall and CN-Waterfall-D are around 66%, 71% and 77%, respectively. Focusing on the three models, CN-Waterfall and CN-Waterfall-D demonstrate superior accuracy over Multichannel-CNN in all the experiments. Moreover, the former models perform rather similarly in the standard setting despite small fluctuations in both models performance. It should also be noted that given the same window size, as the difference between the window and shift sizes decreases, the accuracy of all three models decreases. For instance, the accuracy achieved by the models for the  $ws = 60$  and  $ss = 40$  settings is considerably lower than for  $ws = 60$  and  $ss = 20$ .

## 4.2 Base module experiments

Here, we investigate the *Base* module with a different number of blocks and convolutions filters.



**Fig. 6** Comparison of the performance of CN-Waterfall, CN-Waterfall-D and Multichannel-CNN with window and shift sizes of **a** 30 and 10, **b** 30 and 20, **c** 60 and 20, **d** 60 and 40, **e** 120 and 40, **f** 120 and 80, respectively, in WESAD



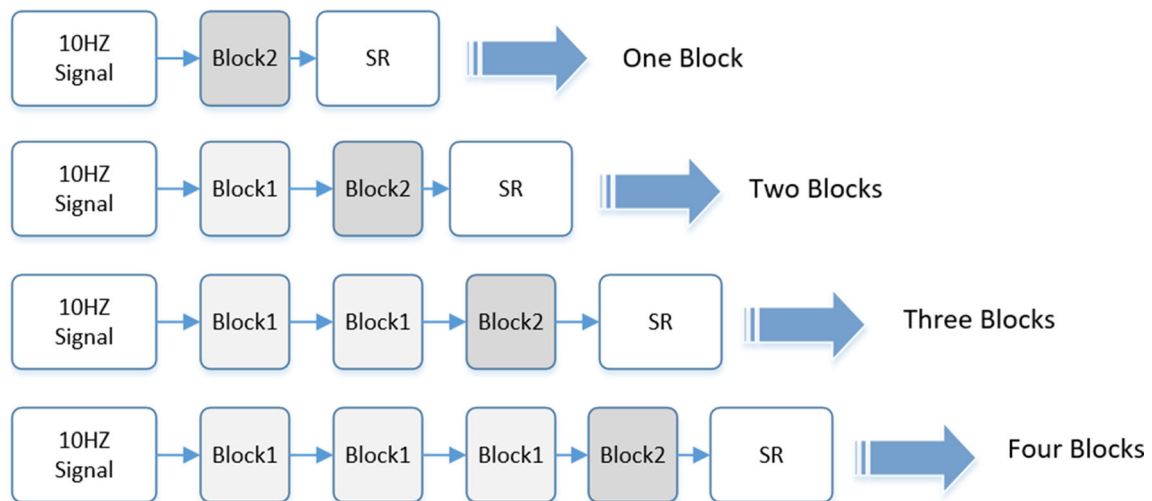
**Fig. 7** Comparison of CN-Waterfall, CN-Waterfall-D and Multichannel-CNN performances with different window and shift sizes of **a** 30 and 10, **b** 30 and 20, **c** 60 and 20, **d** 60 and 40, **e** 120 and 40, **f** 120 and 80, respectively, in MAHNOB-HCI

**4.2.1 Effects of deep layers**

To examine CN-Waterfall with respect to the number of layers in the *Base* module, we use three different settings: one shallow layer and two deeper layers than the *Base* module standard setting. The shallow network encompasses only the second block of the *Base* module, whereas the deeper ones contain the first block of the *Base* module repeated two and three times, with three and four blocks in

total, respectively. Figure 8 illustrates the aforementioned settings.

With the structure mentioned above as the *Base* module, the performance of CN-Waterfall is evaluated in terms of accuracy, precision, recall and f1-score for the two datasets. According to the results shown in Fig. 9, there is almost no difference in accuracy, f1-score, and precision values for the *Base* module with the shallow and deeper networks applied to WESAD. A trivial 0.7% difference



**Fig. 8** The architecture of the shallow *Base* module with one block, the standard *Base* module with two blocks and deeper *Base* modules with three and four blocks

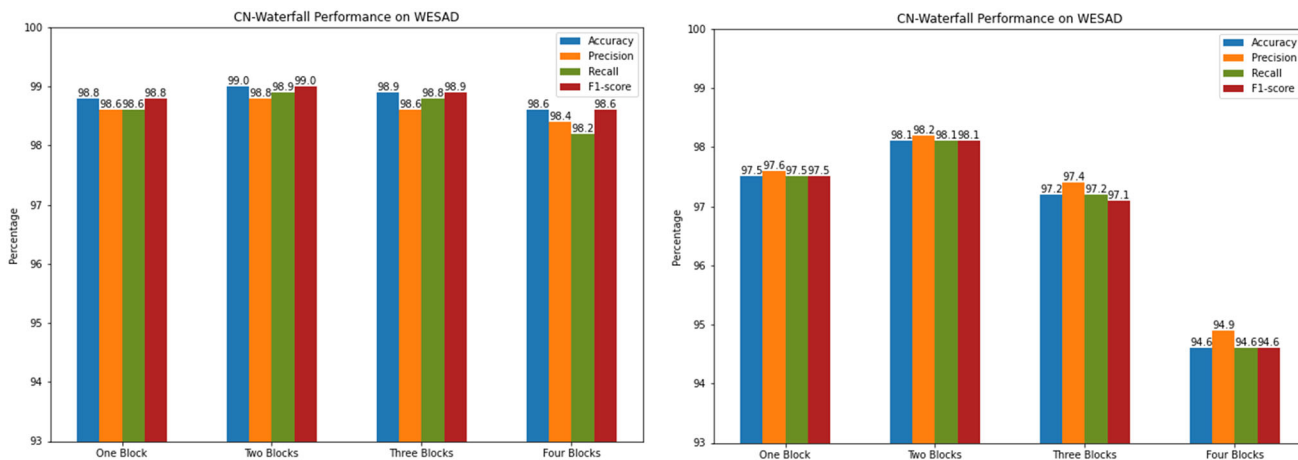


Fig. 9 The performance of CN-Waterfall with different *Base* modules architecture (Fig. 8) on a WESAD and b MAHNOB-HCI

between the recall values of the *Base* module with two and four blocks can be observed. This difference, however, increases considerably to 4% for MAHNOB-HCI. In fact, in the latter dataset, the performance of the deepest *Base* module decreases significantly compared to the standard *Base* module. What we conclude from the results is that deeper networks, in particular on balanced data, do not necessarily perform better than shallower networks.

4.2.2 Effects of the convolutions filter number

Here, we report the results of CN-Waterfall in terms of accuracy with a different number of convolution filters in the *Base* module with two-blocks. Figure 10 shows the results of our model applied to the two datasets, with 128, 64, 32 and 16 convolution filters in the two blocks of the *Base* module.

As demonstrated in Fig. 10, in WESAD, regardless of the number of convolution filters in both blocks, accuracy increases smoothly over epochs. However, in MAHNOB-HCI, we observe slight fluctuations in all the settings. Moreover, in WESAD, accuracy rises to more than 95% after just four epochs, preserving the high performance from the primitive stages of learning, while in MAHNOB-HCI the same accuracy is only achieved in the late stages of learning.

Given the results, we can conclude that the CN-Waterfall model is insensitive to the number of convolution filters in WESAD. However, in MAHNOB-HCI, the model with 128 filters in the second block of the *Base* module shows, on average, better accuracy than with 16 and 32 filters in the same block (Fig. 10d). Therefore, the same conclusion is not valid for the second dataset.

4.3 General module experiments

This section evaluates the *General* module of the CN-Waterfall model in terms of modality correlation and deeper fusions.

4.3.1 Modality correlation

Following Algorithm 1, we calculate the Spearman rank correlation coefficient between 8 and 7 modalities in WESAD and MAHNOB-HCI, respectively. Then, the average of the correlations over the number of participants is calculated. In WESAD, the results reveal that 4 out of 8 modalities meet the requirements of  $S_{corr}$  in the algorithm and the rest are included in the  $S_{non-corr}$  subset. More precisely, the ACC0 and ACC2 modalities show a correlation of 0.84, and EDA and TEMP have an average correlation of 0.44. Therefore, we follow  $S_{corr}$  and  $S_{non-corr}$  in WESAD as below:

$$S_{corr} \leftarrow \{ACC0, ACC2, EDA, TEMP\} \tag{12}$$

$$S_{non-corr} \leftarrow \{ACC1, RESP, EMG, ECG\} \tag{13}$$

Regarding the MAHNOB-HCI dataset, the ECG1 and ECG2 modalities have a 0.52 correlation, and the ECG2 and ECG3 modalities show a correlation of 0.6. We add these modalities to the  $S_{corr}$  subset and the rest of the modalities to the  $S_{non-corr}$  subset in the following:

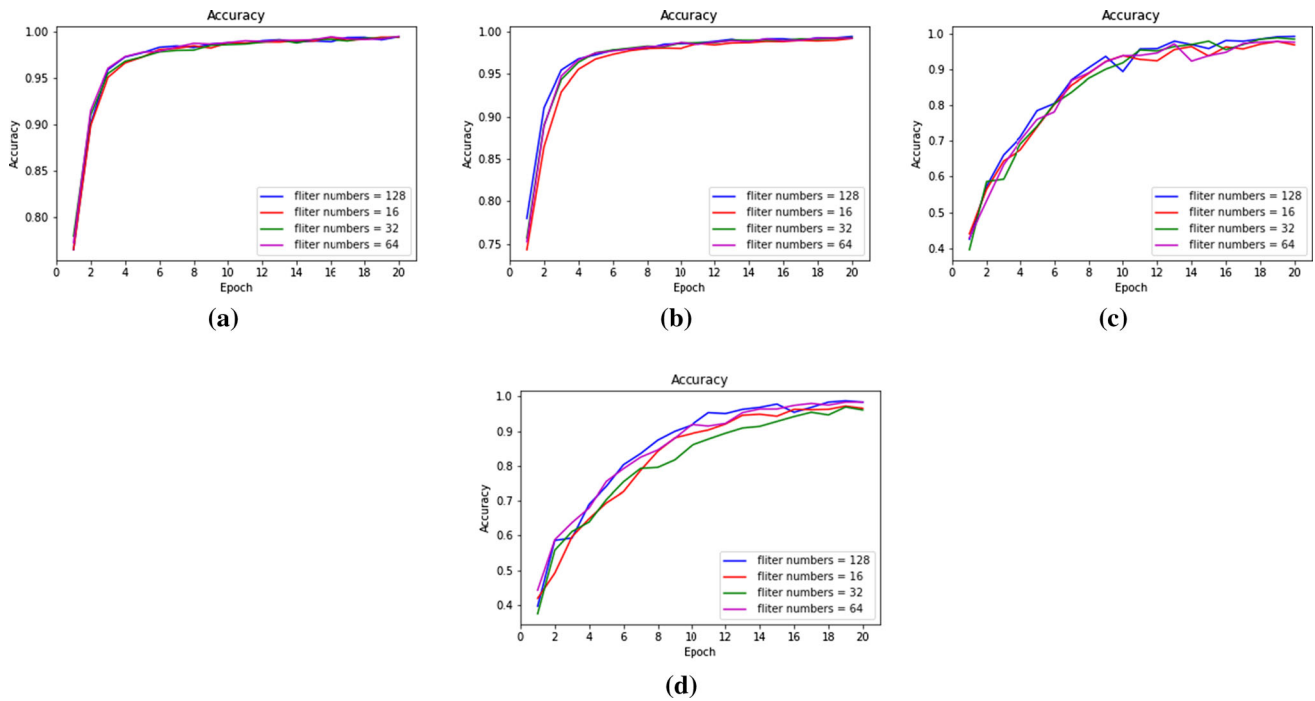
$$S_{corr} \leftarrow \{ECG1, ECG2, ECG3\} \tag{14}$$

$$S_{non-corr} \leftarrow \{TEMP, GSR1, GSR2, RESP\} \tag{15}$$

4.3.2 Effects of deep fusions

In this section, we discuss how fusions with deeper layers in the *General* module (Fig. 3) influence CN-Waterfall





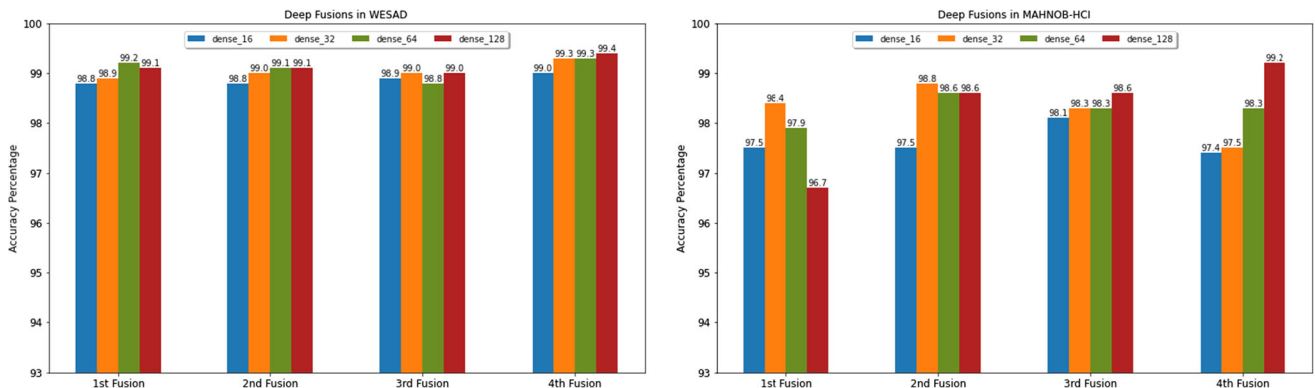
**Fig. 10** The performance of CN-Waterfall with a different number of convolution filters number in the **a** first block and **b** second block of the *Base* module in WESAD and the **c** first block and **d** second block of the *Base* module in MAHNOB-HCI

performance. For this, we connect one hidden-layer fully connected neural network to the fusion components. We, then, examine the impact of 16, 32, 64 and 128 hidden neurons on CN-Waterfall performance.

As shown in Fig. 11, in WESAD, regardless of the number of hidden neurons, the accuracy values of all deeper fusions are closely inline with the accuracy value of the model in the standard setting (Table 4). In other words, CN-Waterfall performance is not influenced by deeper intermediate- and high-level representations in the *General* module.

In MAHNOB-HCI, the deeper networks of the first and second fusions with 32 hidden neurons (*dense\_32*) perform

quite similarly to the standard setting (Table 5) in terms of accuracy. This finding is also valid in the case of the networks with 64 and 128 hidden neurons (*dense\_64* and *dense\_128*, respectively) in the second fusion. However, other deeper networks of the first and second fusions underperform the standard setting, with a rather tangible difference in accuracy. The largest difference, of around 2%, is observed in the network with 128 neurons (*dense\_128*) in the first fusion. This difference also indicates that model performance is a function of the number of hidden neurons in the deeper layers of the first and second fusions.



**Fig. 11** The performance of CN-Waterfall with deeper fusions of *General* module by adding one hidden-layer neural network to each fusion component and examining different settings of 16, 32, 64 and 128 hidden neurons in **a** WESAD and **b** MAHNOB-HCI

Regarding the third fusion, the model seems rather insensitive to the number of hidden neurons in the deeper layers. Moreover, all the networks of this fusion perform in a fairly similar manner to the standard setting in terms of accuracy.

In the high-level representation, the fourth fusion, accuracy is improved stepwise with respect to the number of hidden neurons. The difference between the lowest step (dense\_16) and the highest step (dense\_128) is around 2%, demonstrating the model sensitivity to the number of hidden neurons. In addition, the network with 128 neurons outperforms the standard setting by about 0.5%. This implies that the model explores more informative attributes at the intersection of both correlated- and non-correlated-based features. However, since the latter difference is not significant, one could argue that the deeper layer's increased complexity outweighs any gains in accuracy.

In general, none of the aforementioned networks considerably outperforms the standard setting on MAHNOB-HCI. Thus, we can conclude that, as with the WESAD dataset, CN-Waterfall with the standard setting performs rather optimally.

## 5 Conclusions and future works

In this paper, we proposed a novel deep convolutional neural network, CN-Waterfall. The model was designed as a solution to the problem of autonomous multimodal affect recognition, using mostly physiological-based sensors. One of the major characteristics of our proposed model was its provision of intermediate-level data representation, based on our findings for two theoretically grounded concepts, *CI* and *CU*, in the XAI domain. To this end, CN-Waterfall was composed of two major components, the *Base* and the *General* modules where the former dedicated to learning representations of each single modality, and the latter concerned with learning joint information of multiple modalities. We focused on four fusion components in the *General* module to fuse the features extracted from the correlated and non-correlated modalities. To validate our model, we utilized WESAD and MAHNOB-HCI, two publicly and academically available datasets in the domain of multimodal affect detection, respectively. Through rigorous experimental procedures on different layers and parameter settings, we demonstrated the superiority of CN-Waterfall over developed approaches. Due to the promising results achieved, we believe that the application of our model could be wide; i.e., it may be capable of managing fusions in deep neural networks in general, rather than being exclusive to the domain of affect detection.

Future extensions of CN-Waterfall will deal with the limitations of the current work. First, despite our

comprehensive experiments, deep learning models are known as black-boxes due to their complex nature. There is no bright map in this study identifying why a specific affective state has been selected. In this sense, recent breakthroughs in the field of *explainable AI* could provide valuable insights for better understanding the model's outcome. Second, the datasets we examined were not designed for studying the relationships between context-aware specifications and affective states. Therefore, the validity of the proposed model for such datasets requires further investigation. Third, in the present study, we did not investigate the sensitivity of model to the degree of correlation between modalities. Additional experiments are thus required to scrutinize this aspect of the model. Finally, the CN-Waterfall *Base* module represents a homogeneous feature learning process for all modalities. As each physiological modality shows its own characteristics, it would be useful to focus on heterogeneous feature extraction and learning processes instead.

**Authors' contributions** N.F. and M.A. took over the conceptualization and debugging. N.F. devised the main idea and carried out all the implementations. N.F. performed all the experiments and analysis. N.F. wrote the original draft. N.F. and M.A. reviewed and edited the original draft. M.A. and K.F. supervised the project.

**Funding** Open access funding provided by Umea University. This research was funded by Umeå University. Additionally, this work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

**Availability of data and material** Not applicable.

**Code availability** Not applicable.

## Declarations

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

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Picard R (1995) Affective computing. MIT Technical Report
- Giannakakis G, Grigoriadis D, Giannakaki K, Olympia S, Alexandros R, Manolis T (2019) Review on psychological stress detection using biosignals. *IEEE Trans Affect Comput*
- Jing C, Liu G, Hao M (2009) The research on emotion recognition from ecg signal. In: 2009 international conference on information technology and computer science, volume 1, pp 497–500
- Foteini A, Dimitrios H, Anderson Adam K (2012) Ecg pattern analysis for emotion detection. *IEEE Trans Affect Comput* 3(1):102–115
- Christiaan V, Renske P, Juliane H, Joris V, Klaessens John HGM, Berend O, Cor K (2013) The effect of stress on core and peripheral body temperature in humans. *Stress (Amsterdam, Netherlands)* 16:06
- Andrius D, Arturas K, Vytautas B (2020) Human emotion recognition: review of sensors and methods. *Sensors* 20:592
- Lawrence S, Thomas S, Scott K, Jeffrey C, Ronald L, Brad H (2008) Mental stress and trapezius muscle activation under psychomotor challenge: a focus on emg gaps during computer work. *Psychophysiology* 45(356–65):06
- Sun F-T, Kuo C, Cheng H-T, Buthpitiya S, Collins P, Griss M (2012) Activity-aware mental stress detection using physiological sensors. In: International conference on mobile computing, vol 76. applications, and services. Springer, Berlin, Heidelberg, pp 282–301
- Domínguez-Jiménez JA, Campo-Landines KC, Martínez-Santos JC, Delahoz EJ, Contreras-Ortiz SH (2020) A machine learning model for emotion recognition from physiological signals. *Biomed Signal Process Control* 55:101646
- Girardi D, Lanubile F, Novielli N (2017) Emotion detection using noninvasive low cost sensors. In: 2017 seventh international conference on affective computing and intelligent interaction (ACII), pp 125–130
- Liu M, Di F, Zhang X, Gong X (2016) Human emotion recognition based on galvanic skin response signal feature selection and svm. In: 2016 international conference on smart city and systems engineering (ICSCSE), pp 157–160
- Wang K, Guo P (2020) An ensemble classification model with unsupervised representation learning for driving stress recognition using physiological signals. In: *IEEE transactions on intelligent transportation systems*, pp 1–13
- Andreas H, Silke G, Peter S, Jason W (2004) Emotion recognition using bio-sensors: first steps towards an automatic system. *Affective dialogue systems*. Springer, Berlin, pp 36–48
- Patlar AF, Baris I, Aydin A (2020) Wearable sensor-based evaluation of psychosocial stress in patients with metabolic syndrome. *Artif Intell Med* 104:101824
- Anum A, Muhammad M, Muhammad AS (2019) Human stress classification using eeg signals in response to music tracks. *Comput Biol Med* 107:182–196
- Schmidt P, Dürichen R, Reiss A, Van Laerhoven K, Plötz T (2019) Multi-target affect detection in the wild: an exploratory study. In: Proceedings of the 23rd international symposium on wearable computers, ISWC '19, pp 211–219
- Jonghwa K, Elisabeth A (2008) Emotion recognition based on physiological changes in music listening. *IEEE Trans Pattern Anal Mach Intell* 30(12):2067–2083
- Pritam S, Ali E (2020) Self-supervised ecg representation learning for emotion recognition. *IEEE Trans Affect Comput*
- Chakraborty S, Aich S, Joo M., Sain M, Kim H-C (2019) A multichannel convolutional neural network architecture for the detection of the state of mind using physiological signals from wearable devices. *J Healthc Eng*
- Ragav A, Krishna NH, Narayanan N, Thelly K, Vijayaraghavan V (2019) Scalable deep learning for stress and affect detection on resource-constrained devices. In: 18th IEEE international conference on machine learning and applications (ICMLA), pp 1585–1592
- Lin J, Pan S, Lee CS, Oviatt S (2019) An explainable deep fusion network for affect recognition using physiological signals. In: Proceedings of the 28th ACM international conference on information and knowledge management, CIKM '19, pp 2069–2072. Association for Computing Machinery
- Främling K (1996) Explaining results of neural networks by contextual importance and utility. In: Proceedings of the AISB'96 conference, Brighton, UK, 1–2
- Fouladgar N, Alirezaie M, Främling K (2020) Decision explanation: applying contextual importance and contextual utility in affect detection. In: Italian workshop on explainable artificial intelligence (AI\*AI2020)
- Fouladgar N, Alirezaie M, Främling K (2021) Exploring contextual importance and utility in explaining affect detection. In: AIXIA 2020—advances in artificial intelligence: XIXth international conference of the Italian association for artificial intelligence., volume 12414, pp 3–18
- Schmidt P, Reiss A, Duerichen R, Marberger C, Van Laerhoven K (2018) Introducing wesad, a multimodal dataset for wearable stress and affect detection. In: Proceedings of the 20th ACM international conference on multimodal interaction, Association for Computing Machinery, pp 400–408
- Mohammad S, Jeroen L, Thierry P, Maja P (2012) A multimodal database for affect recognition and implicit tagging. *IEEE Trans Affect Comput* 3(1):42–55
- Russell James A (1980) A circumplex model of affect. *J Pers Soc Psychol* 39(6):1161–1178
- Jonathan P, James R, Bradley P (2005) The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology. *Dev Psychopathol* 17:715–34
- Mandryk Regan L, Stella AM (2007) A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. *Int J Hum Comput Stud* 65(4):329–347
- Khalili Z, Moradi MH (2009) Emotion recognition system using brain and peripheral signals: using correlation dimension to improve the results of EEG. In: International joint conference on neural networks, pp 1571–1575
- Laura F, Gianmaria M, Francesco S, Hamido F, Filippo C (2020) Unsupervised emotional state classification through physiological parameters for social robotics applications. *Knowl Based Syst* 190:105217
- Hyung KB, Sungho J (2020) Deep physiological affect network for the recognition of human emotions. *IEEE Trans Affect Comput* 11(2):230–243
- Wijsman J, Grundlehner B, Penders J, Hermens H (2010) Trapezius muscle emg as predictor of mental stress. In: ACM transactions on embedded computing systems (TECS), volume 12, pp 155–163, 01
- Wijsman J, Grundlehner B, Liu H, Penders J, Hermens H (2013) Wearable physiological sensors reflect mental stress state in office-like situations. In: 2013 humane association conference on affective computing and intelligent interaction, pp 600–605
- Choi J, Ahmed B, Gutierrez-Osuna R (2012) Development and evaluation of an ambulatory stress monitor based on wearable sensors. *IEEE Trans Inf Technol Biomed* 16(2):279–86
- Vila G, Godin C, Charbonnier S, Labyt E, Sakri O, Campagne A (2018) Pressure-specific feature selection for acute stress detection from physiological recordings. In: 2018 IEEE international

- conference on systems, man, and cybernetics (SMC), pp 2341–2346
37. Yekta C, Niaz C, Deniz E, Cem E (2019) Continuous stress detection using wearable sensors in real life: algorithmic programming contest case study. *Sensors* 19:04
  38. Mohammad Naim RASTGOO, Bahareh N, Andry R, Vinod C, Dian T (2018) A critical review of proactive detection of driver stress levels based on multimodal measurements. *ACM Comput Surv* 51(5):1–35
  39. Amandeep C, Mandeep S (2019) An application of phonocardiography signals for psychological stress detection using non-linear entropy based features in empirical mode decomposition domain. *Appl Soft Comput* 77:24–33
  40. Plarre K, Raji A, Hossain SM, Ali AA, Nakajima M, al' Absi M, Ertin E, Kamarck T, Kumar S, Scott M, Siewiorek D, Smailagic A, Wittmers LE (2011) Continuous inference of psychological stress from sensory measurements collected in the natural environment. In: Proceedings of the 10th ACM/IEEE international conference on information processing in sensor networks, pp 97–108
  41. Marković D, Vujičić D, Stojić D, Jovanović Ž, Pešović U, Randić Siniša (2019) Monitoring system based on iot sensor data with complex event processing and artificial neural networks for patients stress detection. In: 2019 18th international symposium INFOTEH-JAHORINA (INFOTEH), pp 1–6
  42. Främling K (2020) Explainable ai without interpretable model
  43. Patrick S, Christa B, Lothar S (2018) Correlation coefficients: appropriate use and interpretation. *Anesthesia Analgesia* 126:1
  44. Werner R, Valev D, Danov D (2009) The pearson's correlation -a measure for the linear relationships between time series? In: Fundamental space research

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