# Stress Monitoring in Free-Living Environments

Ramesh Kumar Sah, *Student Member, IEEE*, Michael J Cleveland,
and Hassan Ghasemzadeh, *Senior Member, IEEE*

***Abstract*— Stress monitoring is an important area of research with significant implications for individuals' physical and mental health. We present a data-driven approach for stress detection based on convolutional neural networks while addressing the problems of the best sensor channel and the lack of knowledge about stress episodes. Our work is the first to present an analysis of stress-related sensor data collected in real-world conditions from individuals diagnosed with Alcohol Use Disorder (AUD) and undergoing treatment to abstain from alcohol. We developed polynomial-time sensor channel selection algorithms to determine the best sensor modality for a machine learning task. We model the time variation in stress labels expressed by the participants as the subjective effects of stress. We addressed the subjective nature of stress by determining the optimal input length around stress events with an iterative search algorithm. We found the skin conductance modality to be most indicative of stress, and the segment length of 60 seconds around user-reported stress labels resulted in top stress detection performance. We used both majority undersampling and minority oversampling to balance our dataset. With majority undersampling, the binary stress classification model achieved an average accuracy of 99% and an f1-score of 0.99 on the training and test sets after 5-fold cross-validation. With minority oversampling, the performance on the test set dropped to an average accuracy of 76.25% and an f1-score of 0.68, highlighting the challenges of working with real-world datasets.**

***Index Terms*— stress detection, alcohol addiction, wearables, machine learning**

## I. Introduction

This work was supported in part by National Science Foundation under grant IIS-1954372. Funding for the original study was provided by the Alcohol and Drug Research Program (ADARP) of Washington State University. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

Ramesh Kumar Sah (ramesh.sah@wsu.edu) and Michael J Cleveland (michael.cleveland@wsu.edu) are with Washington State University, Pullman, WA, USA. Hassan Ghasemzadeh (hassan.ghasemzadeh@asu.edu) is with the College of Health Solutions at Arizona State University, Phoenix, AZ, USA.

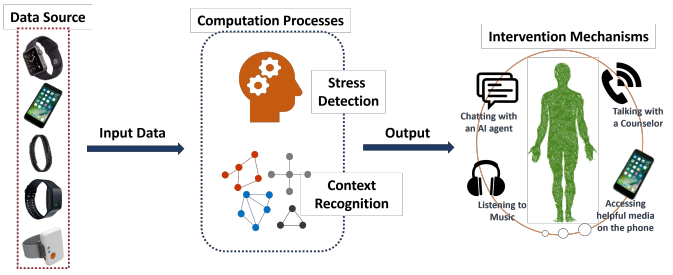


Fig. 1. An mHealth system for automated health assessment and intervention for sustained behavior changes.

STRESS and challenges associated with stress management are prevalent problems of modern life. Many physical and mental health problems are driven by or escalate with the degree of stress. Stress has especially harmful effects on those who suffer from psychological and physical health problems. One such population group is individuals suffering from Alcohol Use Disorder (AUD). Alcohol addiction has increasingly become a serious public health concern, and recent epidemiological data indicate increasing rates of alcohol use and alcohol-related disorders among U.S. adults [1]. National data also suggest that treatment admissions are highest for patients suffering from alcohol use disorders relative to other substances [2]. Due to the intense physiological response induced in the human body because of stressful events, physiological signals such as heart rate variability (HRV), electrodermal activity (EDA), skin temperature, electromyography (EMG), electrocardiography (ECG), and respiration rate lend themselves as best bio-markers for stress monitoring [3].

Stress detection and early intervention are considered critical elements in a treatment strategy toward preventing individuals with alcohol dependence from relapsing. As shown in Fig. 1, mobile health (mHealth) technologies in which patients receive personalized interventions on mobile devices (e.g., through a smartphone app) represent a potential strategy toward helping an individual remain abstinent from alcohol [4]. However, prior research in wearable-based stress monitoring suffers from several shortcomings. First, previous research on

stress monitoring focuses on in-lab experiments where participants take stress tests such as counting backward and preparing for a speech to induce acute stress [5]–[9]. These studies conducted in a controlled environment lack important properties of daily living experiences and fail to generalize in real-world settings [10]. Second, in-lab studies are often conducted with healthy participants, and there is a lack of research involving vulnerable populations such as individuals suffering from AUD. Third, another important characteristic of the state-of-the-art research is the reliance on multiple sensor modalities for stress classification [5], [11]–[13]. Reliance on multiple sensor modalities makes the system design expensive regarding computation and energy requirements. Finally, because prior research focuses on inducing stress in a controlled environment, it assumes that the duration of the stress event is known in advance. However, stress and the response to a stressor are subjective, and the same stress stimuli can elicit different effects in different individuals. The variation in stress response expressed by individuals ultimately gets transformed into the time variation in stress labels indicated by the participants. The shortcomings mentioned above contribute to a lack of knowledge on developing and using mobile health sensor systems in an intervention mechanism in real-world settings.

Previously, we showed that physiological signals captured using a wearable wristband, including heart rate variability and electrodermal activity, are associated with self-reported outcomes, including stress and alcohol cravings in individuals undergoing therapy to remain abstinent from alcohol in the real-world setting [14]. This work aims to investigate the development of efficient machine learning models for stress detection using the same sensor data. We present a data-driven stress detection approach based on Convolutional Neural Networks (CNN) while addressing important research questions. The main contributions of our work can be summarized as follows: (1) we develop a polynomial-time sensor channel selection algorithm to determine the best sensor channel for a machine learning task; (2) we address the lack of knowledge about stress episodes inherent to collecting data in real-world settings by determining the optimal segment length for stress events; (3) we evaluate our algorithms using data collected in a real-world user study to examine the associations between stress and alcohol-related outcomes. We use the collected sensor data to evaluate the performance of our stress detection model; (4) we make our analysis code and the dataset public to encourage further research in this area.

## II. Related Work

Stress detection and classification is an important area of research with significant implications on the physical and mental health of an individual [15]–[17]. Many research articles have reported the usage of multiple modalities for stress detection and classification [5], [11]–[13]. In [5], multiple sensor data collected in lab settings for three affective states - baseline, stress, and amusement were used to train machine learning models to achieve classification accuracies up to $80\%$. Authors in [6] used heart rate variability data with a super short time window for stress detection. Data collected from $20$ participants in a lab setting was used to train a CNN model to detect acute cognitive stress. In [18], the Empatica E4 was used to collect physiological data from participants diagnosed with substance use disorder. The data were collected across four days, and a total of $104$ stress events were reported. Machine learning models trained on multiple modalities data for the binary task of stress classification achieved maximum accuracy of $76.8\%$. In [7], EDA data obtained from $65$ volunteers in controlled conditions was used for stress classification with top accuracy of $94.62\%$. Authors in [8] used blood oxygen saturation to classify human stress with a classification rate up to $95.56\%$. Facial blood oxygen saturation data collected from $42$ participants subjected to two stress conditions was used to train machine learning algorithms.

Stress and the response to stressors have shown to have both objective and subjective dimensions [19]–[21]. The stressor effects are assumed to occur only when both (a) the situation is appraised as threatening or demanding and (b) insufficient resources are available to cope with the situation [19]. Hence, stressors are only stressors in so far as the individual perceives or appraises them to be so. The cause-and-effect relationship between objective stress events and subjective responses is dependent upon the subjectivity of the individual experiencing the stressor. These subjective variations play out in the biological, physiological, and psychological axes and are encoded in the output of sensors quantifying the effects of stress on an individual. Consequently, mobile health systems designed to detect stress using sensors must consider the subjective response to stress events. Capturing these subjective relationships between stress and stress responses becomes especially challenging in real-world conditions.

## III. System Overview

Our stress detection approach relies on supervised machine learning. To train a machine learning model $f$

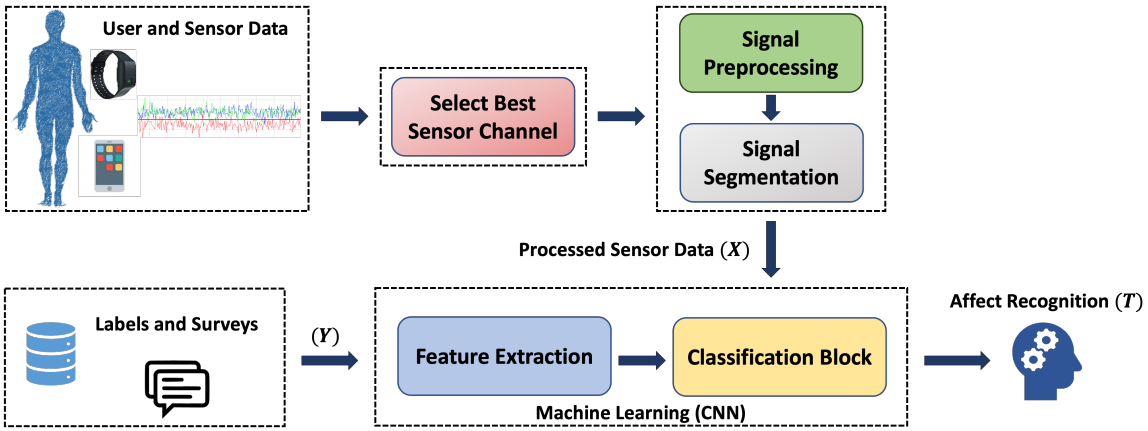


Fig. 2. Proposed system for stress detection. Among many sensor channels available from the sensing device, first the best channel is selected based on the optimal sensor channel selection analysis results. Next, channel specific preprocessing is done and then processed sensor data is divided into segments of length determined with the optimal stress segment length analysis. Segmented raw sensor data from the best sensor modality with optimal stress segment length is used to train convolutional neural networks for stress detection.

for a task $T$ in a supervised manner, we need inputs $X$ and outputs $Y$ to learn the mapping $f : X \rightarrow Y$ using the dataset $D(X, Y)$. Therefore the main component of the machine learning model training pipeline is to create the dataset $D$, which represents the problem space. To develop stress detection algorithms for real-life settings, we use the sensor data collected in our *ADARP* study [22]. As shown in Fig. 2, we use a convolutional neural network architecture for machine learning in our system design. A CNN architecture typically has a feature extraction block followed by a classification block. The feature extraction block generates representations (features) of the input sensor data to be used in the classification block. The classification block learns the mapping between the input representations and output classes, achieving the machine learning goal. Furthermore, a CNN architecture lets us bypass the cumbersome and expensive feature computation and selection process, requiring domain expertise [6], [23].

Designing and training a generalizable stress detection system that is accurate, fast, reliable, and optimized for computational cost and power requirements is challenging. In particular, the transient characteristics of daily life make training machine learning models challenging on real-world data. First, although a wearable device may provide multiple sensor modalities for model training, continuous stress detection in the real-world warrants a system that is efficient in resource consumption and does not require unnecessary and redundant sensor modalities. Therefore, the first question we address deals with the problem of finding sensor channel(s) that result in the highest performance for a machine learning task. Given $n$ number of sensor channels, we want to determine the combination of sensor channels that leads to the best-performing model (Section IV).

Second, in our dataset stress ground truth is obtained through Ecological Momentary Assessment (EMA) and by pressing the event marker button of the Empatica E4 wristband. Since our dataset was collected in real-life settings, the time duration of the stress events is not known a priori. Therefore, the second question that we address (Section V) is the problem of optimal segment length identification for stress events. Stress and response to stress stimuli are subjective, and the occurrence and duration of acute stress events vary from person to person. The subjective nature of stress is reflected in the sensor data through changes in the data and also through the time variation in the instant of acute stress expressed by the participants. Individuals might report the instant of heightened stress with some delay with respect to the actual peak in the sensor data as shown in Fig. 4. Therefore before training a machine learning model, we need to determine the optimal length for all stress events reported by the participants. Furthermore, we also need to ensure that the extracted sensor segment for the stress class adequately accommodates all stress events present in the dataset. The length of stress segments needs to be large enough to capture stress events and, at the same time, should not facilitate the inclusion of sensor data from the baseline or not-stress class.

## IV. Optimal Sensor Channel Selection

Machine learning models trained on data from multiple modalities often outperform models trained on single sensor data [5], [11]. However, the better performance of multi-modal systems comes at the cost of complex

and expensive system design and higher computational requirements. Integrating multiple sensors requires more power during operation and complex processing routines resulting in poor battery life. Using an arbitrarily large number of sensors also increases the system's cost, ultimately discouraging the general population's adoption of the technology. Therefore, one design objective is to optimize the processing pipeline to have a small set of sensors while adhering to application requirements in terms of feasibility (e.g., the number of sensors to include in the final design) and performance (e.g., achieving a minimum accuracy for stress classification). Optimizing the processing pipeline is particularly important in wearable sensor systems where computation requirements are strict and battery life is at a premium.

### A. Problem Definition

Given $n$ sensor channels $C_n = \{c_1, c_2, c_3, \ldots c_n\}$, the task of selecting a subset of the channels can take two different forms. In the first case, the aim is to find top-$k$ channels without any constraint on the model performance. We refer to this problem as *top-$k$ channel selection* (KCS). KCS concerns with finding a combination of $k$ channels $C_k$ where $k < n$, such that a machine learning algorithm $f$ trained for task $T$ on input $I_k$ with dimension $k$ achieves the maximum performance. Therefore, the *$k$-channel selection* problem can be formulated as follows.

$$C_k^* = \arg\max_{C_k \in C_n} \mathcal{P}(C_k) \tag{1}$$

where $C_k^*$ denotes the optimal solution and $\mathcal{P}$ represents the performance of algorithm $f$ with respect to the learning task $T$.

In the second case, the goal is to find the subset of channels that guarantee a given performance level $P$. We call this *performance-guaranteed channel selection* (PCS). In PCS, we find a combination of channels $C_p \subseteq C_n$, such that the performance of the machine learning model $f$ with the selected combination input $I_p$ is greater than or equal to $P$ on task $T$.

$$minimize \quad |C_p| \tag{2}$$

Subject to:

$$\mathcal{P}(f(I_p)) \geq P \tag{3}$$

Here, the learning task $T$ is either a classification problem or a regression problem. The model's $(f)$ actual performance is measured using metrics such as accuracy, f1-score, or mean absolute error, depending on the objective of the learning process.

### B. Problem Solution

Selecting a subset of $n$ channels is a combinatorial search problem with exponential time complexity. Combinatorial search is an NP-hard problem because of the explosion of the possible states of the solution space. However, most combinatorial search problems can be solved by efficiently exploring the large solution space of the possible combinations. We present a polynomial-time algorithm based on the decision tree heuristic and greedy approach for our channel selection problems. Greedy algorithms work by selecting a locally optimal choice in each iteration stage until the desired property is satisfied. Our algorithm successively builds the best channel combination in a decision tree fashion and selects a new channel for addition to the current selection using the greedy method.

Before we present our algorithms, let us understand them with an example of KCS problem. Let the total number of channels be $n = 4$, and we want to select the best $k = 3$ channels. Let $C_k$ be the combination with top-3 channels that we want to find. Our algorithm finds the channels in a decision tree fashion. In the tree's first level, there are $4$ channels to choose from, and let us say channel $2$ is the best one, denoted by $*$ in the figure. We add channel $2$ to the current selection such that $C_k = \{2\}$. Following this step, we have $3$ channels to choose from in the second level, and in the third or final level, we have $2$ channels for selection. Here, at each level of the tree, we choose one channel that maximizes the performance of the model $f$ on task $T$ upon addition to the current subset of selected channels $C_k$. For the considered example, when the algorithm stops, the selected channels will be $C_k = \{2, 1, 3\}$.

Algorithm 1 is for the KCS problem and selects $k$ channels out of $n$ channels without any constraint on the final performance. Algorithm 2 describes the PCS problem, which selects a combination of channels that can achieve the performance threshold of at least $P$. In both cases, a learning algorithm $(f)$ is given and at the start, the selected channel set is empty i.e., $C_k = \{\}$ and $C_p = \{\}$. Algorithm 1 is also given the number of channels $(k)$ to select and if the value of $k$ equals the size of the total channel set $C_n$, the algorithm returns $C_n$ and stops. Otherwise, in each run of the while loop, the algorithm creates combinations of channels in sets $C_k$ and $C_n$. Next, we train machine learning models for each created channel combination and select the channel combination with the best score on the task $T$. Each channel in the selected combination is removed from the total channel set $C_n$ and added to the selected channel set, $C_s$, if not already in the set. If the size of $C_k == k$,

then the algorithm stops and returns $C_k$. Otherwise, the process is repeated until the size of $C_k$ becomes equal to $k$.

**Algorithm 1:** $k$-Channel Selection (KCS) Algorithm

```
Data: C_n, k > 0, f
Result: k channels
C_k ← {};
if k = |C_n| then
    return C_n
else
    while |C_k| ≠ k do
        Create combination of channels in C_n and
          C_k, say C_s;
        Train model f_i for each channel combination
          in C_s;
        Select best combination C_b ∈ C_s;
        for Each channel c_i in C_b do
            if c_i ∉ C_k then
                C_k ← c_i
            end
            C_n ← C_n − c_i
        end
    end
    return C_k;
end
```

Algorithm 2 takes as input the performance threshold value $P$ and a small threshold metric $\epsilon$ used to determine the improvement in the model's performance upon the addition of a new channel in the selected set $C_p$. Upon selecting the best channel at the intermediate stage using the greedy method, we check whether the current performance $P_b$ has reached the threshold performance $P$ or the performance improvement compared to the previous stage is less than $\epsilon$ or not. If either of these conditions is true, the algorithm stops and returns the selected channels set $C_p$.

Both algorithms are required to create combinations of channels from two sets. During execution, the set $C_k$ or $C_p$ contains a combination of selected channels, and the set $C_n$ holds all channels that are not yet evaluated. Combinations of channels are created by combining the channel combination in $C_k$ or $C_p$ with each channel in $C_n$ as shown in algorithm 3. For example, if $C_k = \{c1c2\}$ and $C_n = \{c3, c4\}$ then the combinations will be $\{c1c2c3, c1c2c4\}$.

### C. Complexity Analysis of Channel Selection

For the KCS algorithm, a brute force search algorithm must review all possible combinations of $k$ channels out of $n$ channels. Due to the greedy selection of channels at the intermediate stage, the channel's position

**Algorithm 2:** Performance-guaranteed Channel Selection Algorithm (PCS)

```
Data: C_n, P, f, ϵ
Result: k channels
C_p ← {},   q ← 0,   C_q ← {};
while true do
    Create combination of channels in C_n and C_p,
      say C_s;
    Train model f_i for each channel combination in
      C_s;
    Select best combination C_b ∈ C_s with
      performance P_b;
    for Each channel c_i in C_b do
        if c_i ∉ C_p then
            C_p ← c_i;
        end
        C_n ← C_n − c_i;
    end
    if (P_b > P) ∨ (|q − P_b| < ϵ) then
        return C_p;
    end
    q ← P_b;
end
return C_p;
```

**Algorithm 3:** Channel Combinations Algorithm

```
Data: C_n, C_s
Result: Channel Combinations
C ← {};
for Each channel c_i in C_n do
    C ← C_i + C_s;
end
return C;
```

in the selection process matters; consequently, we have a permutation search space instead of a combinatorial search space. Our algorithm selects the best local channel and significantly decreases the solution space. For $n$-total channels, we have $n$ channel choices, and accordingly, $n$ different machine learning models are trained in the first iteration of the algorithm. After selecting the first best channel, $n-1$ machine learning models are trained in the second iteration. This continues, and for choosing $k$ channels out of $n$ channels at the end, $n-(k-1)$ models are trained. If the number of models that need to be trained to select $k$ channels is be $M_k$.

$$M_k = \sum_{j=1}^{k}(n-j+1) \tag{4}$$

$$M_k = n + (n-1) + (n-2) + \cdots + (n-k+1) \tag{5}$$

Also, for the worst case of selecting $k = n-1$

channels, the total choices of channel combinations will be

$$= n + (n-1) + (n-2) + (n-3) + \cdots + 2 \quad (6)$$

$$= (n * \frac{n-1}{2}) - 1 \quad (7)$$

$$= \frac{n^2}{2} - \frac{n}{2} - 1 \quad (8)$$

Therefore, we will have a polynomial time complexity of $O(n^2)$ for the KCS algorithm. For the PCS algorithm, the naive approach will give us an exponential runtime since for $n$-channels, there are $2^n$ possible choices. Our greedy approach selects the best channel/channel combination at each stage and consequently runs in $O(n^2)$. This again follows from the fact that at each stage, the number of choices keeps decreasing, and the total number of options will be that shown in equation (6).

### D. Time Complexity of Model Training

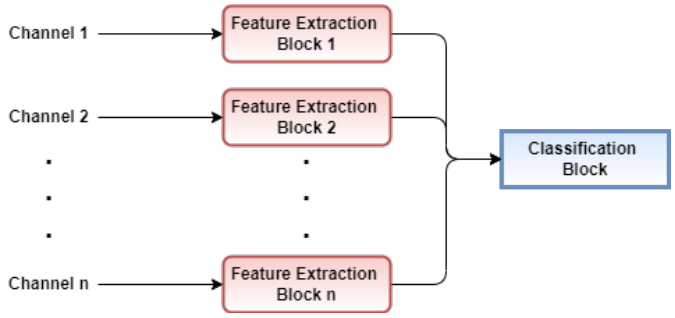


Fig. 3. Modular design of the convolutional neural network used in the KCS and PCS algorithms. We have a feature extraction block for each channel, and outputs from all feature extractors are fed to the classification block.

For the KCS and PCS channel selection algorithms, we also need to consider the time complexity of training machine learning models. During the successive iteration of our channel selection algorithms, the number of input channels to the machine learning model grows linearly. Consequently, the model architecture is modified to account for this change. We achieve this by having separate feature extractors and classification blocks in the model architecture as shown in Fig 3. The classification block remains fixed, and the number of feature extractor blocks equals the number of input channels. Since the size or number of parameters of the feature extractor block is fixed, increasing the number of input channels will linearly increase the number of trainable parameters of the complete model. Furthermore, the training complexity of a machine learning model depends linearly on the number of parameters of the model or has a polynomial relationship with the size of the dataset [24]–[26]. Therefore, the overall time complexity of our channel selection algorithms will still be polynomial-time $O(n^2)$.

## V. Optimal Stress Segment Length

Sensor systems continuously sense some physical phenomena and generate data spread in time. These are time-series data used for training and evaluating machine learning algorithms. The training data needs to capture the properties of the task so that the trained model learns to differentiate between the classes in a generalized way. In our study, participants were asked to mark stress events in time by pressing the push button on E4 when they felt stressed. However, the event markers may not necessarily be at the moment of heightened stress. The event marker could be before, after, or during the stress episodes, as shown in Fig. 4. These variations stem from the subjective nature of stress such that the same external stimuli could have different effects on different individuals. Therefore, the sensor segment extracted around event markers needs to have a length that accommodates the subjective nature of stress expressed through temporal variations in the event markers. Earlier works have used a 40 minute segment length around stress events, i.e., 20 minutes before and 20 minutes after the event marker [18], but fail to provide any reason for their choice. In our understanding, extracting a 40 minutes segment for the stress class around each event marker covers the varying effects of stress on different individuals. However, the question of optimal segment length remains unanswered. In our context, optimality means the sensor segment length around event markers that yields the best discriminative data between the stress and not-stress class and facilitates a model with the highest performance or lowest generalization error on stress detection.


Fig. 4. Variation of the event marker and the actual stress event peak in the sensor data. The event marker can be before, during, and after the signal peak associated with the stress event. An stress event starts with the rise and terminate during the decline of the signal from its peak.

Let $l$ be the length of the input time-series segment $x$. One way to ensure that the input data contain the maximum amount of information - consider all variations in the participant's stress response - is to use an infinitely

long input segment, i.e., $l = \infty$. However, this makes the system unusable in daily life for activities such as stress detection, where the system needs to be fast enough to respond to events and changes happening in real-time. Therefore, we need a segment length $l = L$ that is sufficiently small and captures the information about the classes so that the machine learning model trained on the dataset created for the input length of $L$ has the lowest generalization error. This is an optimization problem where we find the length of the sensor segment used for training the machine learning model $f$ such that the generalization error $\epsilon_g(f)$ is the lowest or below some threshold $\delta$.

$$|x| = L \tag{9}$$

$$s.t. \quad \epsilon_g(f(x)) \leq \delta \tag{10}$$

We determine the optimal segment length around stress events by training machine learning models for stress detection at different values of segment length. Consequently, the best-performing model will be the result of optimal segment length. For segment lengths $L = \{l_1, l_2, \ldots, l_n\}$ around event markers, we train machine learning models $F = \{f_1, f_2, \ldots, f_n\}$ on the respective datasets $D = \{d_1, d_2, \ldots, d_n\}$. The model with the best performance $f_o = \max\{F\}$ will give us the optimal segment length $l_o$. In our analysis, the segment length for stress events is varied from 60 seconds to 3600 seconds with an increment of 60 seconds. The sensor segments for the not-stress class have a start and end point of 60 minutes before, and after the event markers, and for all values of stress segment lengths, there is no data overlap between the stress and not-stress classes as shown in Fig. 6.

## VI. Evaluation Approach

### A. User Study and Dataset

We conducted a user study to collect sensor and survey data for use in this project. After approval from the Institutional Review Board at Washington State University (IRB #17018), we recruited 11 participants receiving treatment for mental health and alcohol use disorder at a treatment agency in the state of Washington. Each participant was asked to wear an Empatica E4 (shown in Fig 5) wristband to capture in real-time continuous physiological markers of stress, press the event marker button on E4 whenever they felt stressed, and complete the surveys sent to their phone 4 times daily for 14 days. Prior research studies have studied the sensor data obtained from E4 and have found a good correlation with standard clinical ground truth [27], [28]. Empatica E4 measures skin conductance or electrodermal activity (EDA), skin temperature (TEMP), 3-axial body acceleration (ACC-X, ACC-Y, ACC-Z), blood volume pulse (BVP), and heart rate (HR). A total of 1698 hours of physiological data were collected and 409 moments of stress were identified using the button available on the E4. For more information about the study and dataset please see our paper [22].


Fig. 5. Empatica E4 wristband with embedded sensors.

### B. Filtering and Noise Removal

For EDA signals, a low-pass filter is suitable for removing high-frequency noise. In our analysis before the normalization and segmentation stages, we use a second-order low-pass Butterworth filter with a cutoff frequency of 1.25 Hz to filter out high-frequency noise from EDA signals [29], [30]. In our earlier work [14], we estimated the quality of EDA data collected in our study across two dimensions using standard tools like EDA Explorer and LedaLab. We estimated the proportion of clean signals after noise and artifact removal and the distribution of skin conductance response (SCR) using Trough-to-Peak analyis (TTP) and Continuous Decomposition Analysis (CDA). We found 87.86% of the EDA signals to be clean and hence we have not used any other processing routines except the low-pass filter to process the EDA signals.

### C. Dataset Construction

Collected sensor data is partitioned into the stress and not-stress class based on an event marker (stress ground truth) as shown in Fig. 6. Around each event marker, we have a buffer zone and the data for the stress class lies within the buffer region. All the data outside the buffer zone is considered for the not-stress class. We acknowledge that collecting data in free-living environment impedes a through verification of stress event reported by the participants. One way to verify stress events reported using the E4 in our study was to find report of subjective stress/not-stress in Ecological Momentary Assessment (EMA) surveys completed by the participants after the event timestamp for that particular day. We designed our study to be able to verify labels using cross checking

between stress events reported by the participants using the sensor device E4 and EMA surveys. However, only 26% of E4 stress events could be corroborated using survey responses. Since the adherence of stress events was low in EMA surveys, we decided to use only stress labels obtained from E4 in our analysis. However, we do note that subjects may not always realize that they have a stressful situation in a free-living environment or forget to push the button on E4. Verification of labels is one of the limitations of real-life studies, and the standard approach is to validate labels using EMA surveys which was also the aim of our study. Moreover we also believe event markers generated using the push button on E4 are more likely to correspond to the actual event of stress than events reported in surveys later due to forgetting.


Fig. 6. Partition of the sensor data into stress and not-stress class. Shown is a EDA signal segment with event marker or stress ground truth represented by the red line. Data for stress class is extracted around the event marker and other parts of the segment is considered into not-stress class.

## D. Class Imbalance

Due to the low number of stress ground truth, we observe a large class imbalance between stress and not-stress classes in the dataset. The number of samples for the stress class depend on the length of sensor segment extracted around each event marker. Smaller values of sensor segment length around event markers result in a lower number of samples for the stress class and a higher degree of class imbalance between the stress and not-stress classes.

We posit the class imbalance to be one of the consequences of collecting data in real-world settings. Our analysis uses minority class oversampling and majority class undersampling to balance the classes before training machine learning algorithms. Artificial training data is generated using the Sythetic Minority Over-Sampling (SMOTE) [31] method in minority class oversampling and training samples from the majority class is randomly dropped in majority class undersampling. We believe majority undersampling will create a better dataset for stress detection since both classes will be expressed equally with preserved variance in the training and testing sets. In contrast, minority oversampling will generate artificial samples very similar to the original stress data. The resulting dataset will have less variance in the stress class. Consequently, the trained machine learning model will overfit the noise in the data and will have poor generalization performance [32].

## E. Performance Metrics

We report precision, recall, f1-score, and accuracy on the training and test sets. Accuracy measure how many samples the model got right among all samples. Precision measures the ability of the model to not make a mistake on the negative class (not-stress class), and recall is the ability of the model to find all positive samples (stress class). Precision and recall are defined as

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

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

where $TP$ is the true positive, $FP$ is the false positive, and $FN$ is the false negative. F1-score is the weighted average of precision and recall and gives the holistic measure of performance.

$$\text{F1-score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

## F. Network Architecture and Hyperparameters

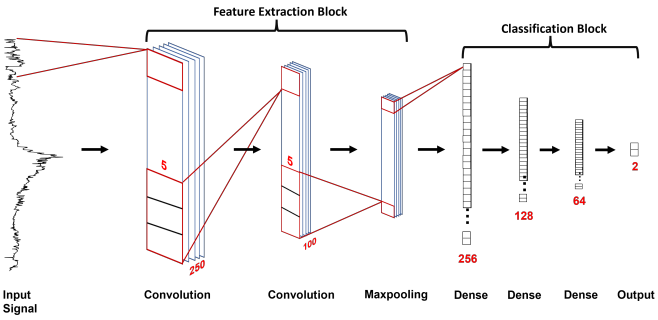


Fig. 7. Architecture of 1D Convolutional Neural Network (CNN) model.

In our experiments, we have reused the CNN architecture we proposed in our earlier work [22] (shown in Fig 7). The CNN model has two $1D$ convolutional layers and three fully-connected layers. Before the first fully connected layer, there is a global max-pooling layer to aggregate embedding obtained from the convolutional block and between the first and second fully connected layers there is a drop-out layer. In the CNN architecture convolutional layers belong to the feature extraction block and fully connected layers constitutes the classification block. The learning rate was set to 0.001 in all our experiments, and categorical cross-entropy loss

with Adam [33] optimizer was used for training. The values of all other hyperparameters were chosen after a randomized search over a set of hyperparameter values.

## VII. Results and Discussion

We used one-dimensional CNNs in our analysis because CNN allows us to create the modular architecture required in our channel selection algorithm. Also, 1D CNNs have shown to have superior performance on classification problems with time-series data in prior research [6], [23], [34]. CNNs can directly learn features and associations between the classes from the raw sensor data during the training process.

### A. Optimal Sensor Channel

The data collected in our study contains 7 sensor channels with 3 channels for body acceleration (ACC-X, ACC-Y, ACC-Z) and 1 channel each for electrodermal activity (EDA), blood volume pulse (BVP), skin temperature (TEMP), and heart rate (HR). We include both HR and BVP in our analysis because HR only quantifies the heart rate whereas BVP also contains information about Heart Rate Variability (HRV). To determine the best sensor channel, we use the *KCS Algorithm* defined in 1 with $k = 1$. The set $C_n$ is the 7 channels of bio-markers, and the learning algorithm $f$ is the CNN with architecture and hyperparameters values defined in section VI-F. We extract 40 minutes of sensor segment around event markers for the stress class data, and the remaining data belongs to the not-stress class. We decided to use a 40 minutes length around event markers because we think 40 minute is a big-enough window to account for stress variation and its effect on an individual. Our aim is to include all information about stress response even if redundant information are also included with a large stress event segment size. Extracted segments for the stress and not-stress class is further divided into windows of size 60 seconds with 50% overlap between consecutive windows. After data preprocessing, the dataset is split into training and testing sets with a 70 : 30 split. The models are trained for 100 epochs with a batch size of 100. Each model is trained 3 times with random initialization, and the average values of loss, accuracy, and f1-score of the trained model on the training and testing set are shown in Fig. 8.

The $y-$channel of the body acceleration achieved the best performance on the training set. The EDA channel has a similar performance on the training set but performed much better on the testing set. Since the generalizability of a machine learning model is assessed by measuring the model performance on a held-out test set, we conclude EDA sensor channel is the best modality for stress classification. Also, ACC signal in general has more noise than EDA making the ACC model prone to overfitting and ACC signal is less qualified to capture stress responses physiology. These we believe to be the reasons for lower performance of ACC model on the testing set. Next, we use the EDA sensor channel to determine the optimal segment length around stress events.

### B. Optimal Stress Segment Length

To determine optimal stress segment length, we experimented with stress segment length from 60 seconds to 3600 seconds with 60 seconds increment. Using the EDA channel data, we trained individual stress detection model for each stress segment length. We first extracted stress segments around the event markers for each segment length as shown in Fig. 9. We then used overlapping windowing to get the stress samples for training machine learning algorithms. The same test set was used to evaluate trained models at different values of the stress segment length for a fair comparison. Stress class samples for the segment length of 60 seconds belong to stress samples for all other segment lengths; since the segment length of 60 seconds is the lowest value. Therefore, we extracted stress samples for the segment length of 60 seconds, randomly selected 30% of the samples (39 samples), and kept them as the stress samples for the test set. The remaining 70% of the 60 seconds stress data (89 samples) is used in the training set and mixed with the stress samples obtained for all other segment lengths greater than 60 seconds. We also selected 39 samples from the not-stress data and created the balanced test set to evaluate machine learning models trained for different stress segment lengths. The remaining not-stress data (163845 samples) are kept for the training set.

Figure Fig. 10 shows the loss, accuracy, and f1-score of machine learning models trained for different stress segment lengths on the training and testing sets. The x-axis shows the stress segment length with an increment of 180 seconds. Before training, we balanced the training set using the majority class undersampling. We found the best stress classification performance for the segment length of 60 seconds. The accuracy and f1-score on the training and test set decreased with the increase in the stress segment length.

### C. Stress Classification

We determined EDA as the best sensor channel and an optimal segment length of 60 seconds around stress

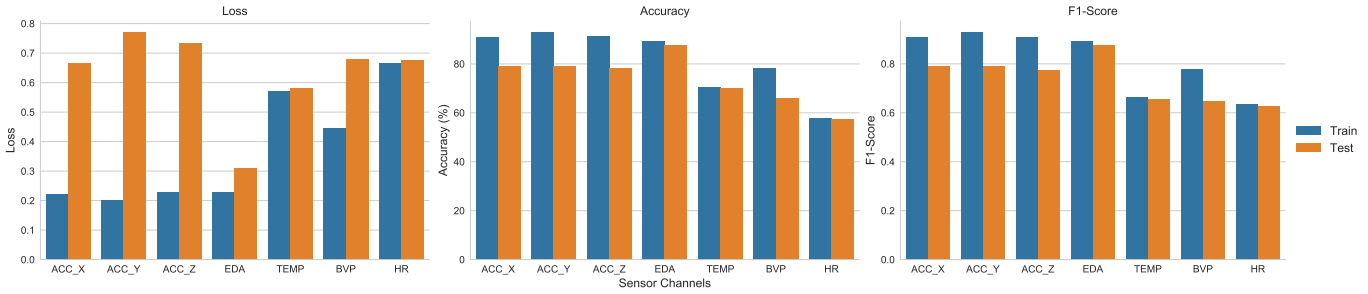


Fig. 8. Training and test set loss, accuracy, and f1-score of models trained on different sensor channel data with random undersampling to balance the classes. The EDA sensor channel has the best performance on both the training and testing set.

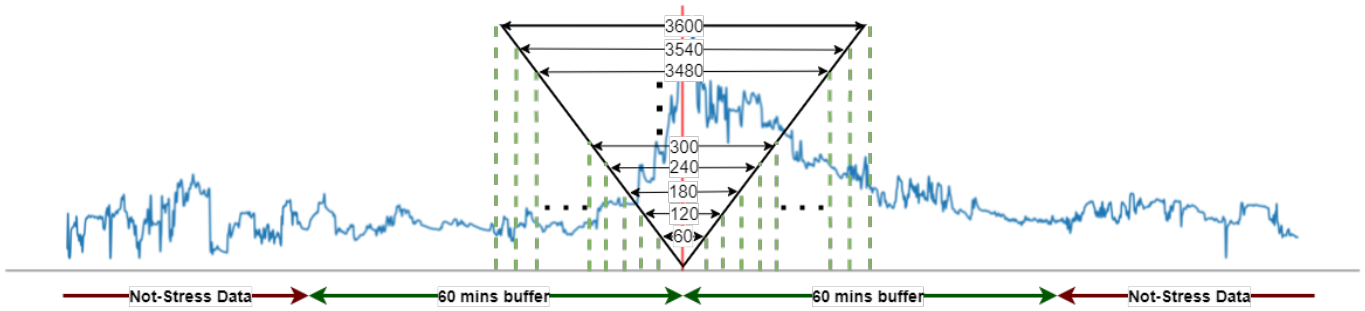


Fig. 9. Stress sensor segment extracted for different segment lengths. The not-stress class data starts before and after one hour buffer around the event marker.

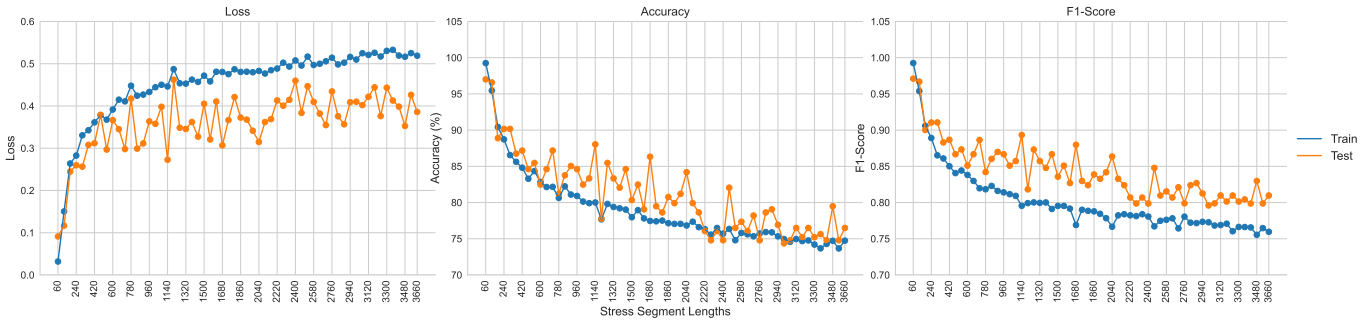


Fig. 10. Loss, accuracy, and f1-score of the trained CNN model on the training and test sets at different values of stress segment length. Models are trained on the EDA sensor channel data and the training set is balanced using random undersampling.

events from our analysis. Next, we trained a CNN model for stress classification using the EDA sensor data with a stress segment length of 60 seconds. After extracting sensor data for the stress and not-stress class and overlapping segmentation, we get 203 samples in the training set and 50 samples in the test set. We balance the training set using the majority class undersampling method. Table I shows the average values for loss, accuracy, precision, recall, and f1-score for the training and testing set after 5-fold cross-validation. Our CNN model achieves an average accuracy of 99% and an f1-score of 0.99 on training and testing sets.

TABLE I
STRESS CLASSIFICATION WITH EDA STRESS SEGMENT LENGTH OF **60** SECONDS AND RANDOM UNDERSAMPLING TO BALANCE THE CLASSES.

| Dataset | Loss | Accuracy (%) | Precision | Recall | f1-Score |
|---|---|---|---|---|---|
| Training | 0.009 | 99.707 | 0.994 | 1.0 | 0.997 |
| Testing | 0.075 | 99.215 | 0.985 | 1.0 | 0.992 |

## D. Results with Oversampling

In this section, we present the results of our analysis for the best sensor channel, optimal stress segment length, and stress classification using the minority oversampling method to balance the classes in the training set. For minority class oversampling, we used

the SMOTE method to generate synthetic samples for the stress class. All other details of the analyses are kept the same, and only the class balancing method is changed. Due to the nature of SMOTE, after balancing the classes, the training set contains a very high number of similar samples for the stress class. The EDA sensor channel again outperformed all other sensor channels and achieved the highest accuracy and f1-score on the test set. The results of the optimal stress segment length analysis were mixed. In terms of accuracy, a segment length of 180 seconds came on top, and for f1-score segment length of 660 seconds achieved the highest value.

TABLE II
STRESS CLASSIFICATION WITH EDA STRESS SEGMENT LENGTH OF **180** SECONDS AND SMOTE OVERSAMPLING TO BALANCE THE CLASSES.

| Dataset | Loss | Accuracy (%) | Precision | Recall | f1-Score |
|---|---|---|---|---|---|
| Training | 0.018 | 99.33 | 0.992 | 0.994 | 0.993 |
| Testing | 3.974 | 76.25 | 0.99 | 0.529 | 0.689 |

For stress classification, the trained model achieved an accuracy of 99% and an f1-score of 0.99 on the training set as shown in table II. The model performance decreased on the test set, with an accuracy of 76.25% and an f1-score of 0.68. The recall value of 0.52 on the test set indicates that the trained model misclassifies not-stress samples as stress samples. We believe the underwhelming performance in the case of minority oversampling is due to the overfitting of the model on the training data. Since the training data contains highly similar stress class samples, the model learns specific details or noise of the stress class during training and consequently has higher generalization errors.

## VIII. CONCLUSIONS

Stress and stress management is an integral part of our daily lives. Early detection and classification of stress are especially beneficial for developing intervention strategies designed to improve the lives of individuals suffering from depression, anxiety, and addiction. Our analysis and results show a viable method for stress detection using sensor data collected in real-world conditions from individuals diagnosed with alcohol use disorder and undergoing treatment to abstain from alcohol. We presented a data-driven approach for stress detection based on convolutional neural networks while addressing the problems of multi-modal wearable sensor systems and the lack of knowledge about stress episodes in real life. Our analysis of the best sensor channel and optimal stress segment length answers two fundamental questions about using sensor systems for stress detection in daily life. We found the electrodermal activity (EDA) or skin conductance to be most indicative of stress, and the segment length of 60 seconds around stress events gave the top stress detection performance. Using majority undersampling to balance the classes, the stress detection model trained on the EDA sensor data with a stress segment length of 60 seconds achieved an average accuracy of 99% and f1-score of 0.99 on training and test sets after 5 fold cross-validation. The stress detection performance dropped with minority oversampling with an average accuracy of 76.25% and an f1-score of 0.68 on the test set. Our work has the following limitations: (1) a more comprehensive analysis of the channel selection algorithms is needed to ascertain the importance of presented algorithms properly, and (2) dataset collected in our study and used in the analysis had a significant class imbalance.

The analysis of optimal stress segment length is subjective to our dataset and constraints of the study. The result might be different for a dataset collected using different sensor devices, different stressors or population groups, and sensing modalities. Also, stress and effects of stress have many facets including individual characteristics and environmental factors. Hence, the optimal stress segment length obtained from our analysis may not be the same for stress events in general. For example, stress caused by taking an exam vs. getting involved in an accident would have different intensities and duration. The size or type of the model should have minimal influence on the results if the model is not underfitting. In regard with channel selection, we want to note that in cases when the sensor system has multiple modalities such as the E4, the main utility of channel selection is to manage the trade off between complex algorithms and computational load and better situation assessment. For example, by using other sensor channels present in the E4, type and intensity of stress can also be assessed. Furthermore, additional modalities can provide contextual information that can be used to remove confounding factors such as exercise and sleep to better assess stress levels. In the future, we plan to study our channel selection algorithms in detail and also make the segment length determination dynamic.

The data[1] and code[2] used in our analysis are made public to facilitate future research.

## REFERENCES

[1] B. F. Grant, S. P. Chou, T. D. Saha, R. P. Pickering, B. T. Kerridge, W. J. Ruan, B. Huang, J. Jung, H. Zhang, A. Fan

[1] *https://zenodo.org/record/6640290*
[2] *https://github.com/rameshKrSah/ADARP_Dataset*

*et al.*, "Prevalence of 12-month alcohol use, high-risk drinking, and dsm-iv alcohol use disorder in the united states, 2001-2002 to 2012-2013: results from the national epidemiologic survey on alcohol and related conditions," *JAMA Psychiatry*, vol. 74, no. 9, pp. 911–923, 2017.

[2] S. Abuse, M. H. S. A. US, O. of the Surgeon General (US *et al.*, "Early intervention, treatment, and management of substance use disorders," in *Facing Addiction in America: The Surgeon General's Report on Alcohol, Drugs, and Health [Internet]*. US Department of Health and Human Services, 2016.

[3] J. Choi, B. Ahmed, and R. Gutierrez-Osuna, "Development and evaluation of an ambulatory stress monitor based on wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 2, pp. 279–286, 2011.

[4] S. Carreiro, M. Newcomb, R. Leach, S. Ostrowski, E. D. Boudreaux, and D. Amante, "Current reporting of usability and impact of mhealth interventions for substance use disorder: A systematic review," *Drug and Alcohol Dependence*.

[5] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. Van Laerhoven, "Introducing wesad, a multimodal dataset for wearable stress and affect detection," in *Proceedings of the 20th ACM International Conference on Multimodal Interaction*.

[6] J. He, K. Li, X. Liao, P. Zhang, and N. Jiang, "Real-time detection of acute cognitive stress using a convolutional neural network from electrocardiographic signal," *IEEE Access*, vol. 7, pp. 42 710–42 717, 2019.

[7] A. Greco, G. Valenza, J. Lázaro, J. M. Garzón-Rey, J. Aguiló, C. De-la Camara, R. Bailón, and E. P. Scilingo, "Acute stress state classification based on electrodermal activity modeling," *IEEE Transactions on Affective Computing*, 2021.

[8] X. Liu, Y. Shan, M. Peng, H. Chen, and T. Chen, "Human stress and sto2: database, features, and classification of emotional and physical stress," *Entropy*, vol. 22, no. 9, p. 962, 2020.

[9] F.-T. Sun, C. Kuo, H.-T. Cheng, S. Buthpitiya, P. Collins, and M. Griss, "Activity-aware mental stress detection using physiological sensors," in *International Conference on Mobile Computing, Applications, and Services*. Springer, 2010.

[10] J. Healey and R. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, 2005.

[11] P. Bobade and M. Vani, "Stress detection with machine learning and deep learning using multimodal physiological data," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2020, pp. 51–57.

[12] M. Gjoreski, H. Gjoreski, M. Luštrek, and M. Gams, "Continuous stress detection using a wrist device: in laboratory and real life," in *2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 1185–1193.

[13] C.-P. Hsieh, Y.-T. Chen, W.-K. Beh, and A.-Y. A. Wu, "Feature selection framework for xgboost based on electrodermal activity in stress detection," in *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*. IEEE, 2019, pp. 330–335.

[14] P. Alinia, R. K. Sah, M. McDonell, P. Pendry, S. Parent, H. Ghasemzadeh, M. J. Cleveland *et al.*, "Associations between physiological signals captured using wearable sensors and self-reported outcomes among adults in alcohol use disorder recovery: Development and usability study," *JMIR Formative Research*, vol. 5, no. 7, p. e27891, 2021.

[15] D. R. Dacunhasilva, Z. Wang, and R. Gutierrez-Osuna, "Towards participant-independent stress detection using instrumented peripherals," *IEEE Transactions on Affective Computing*, 2021.

[16] S. Koldijk, M. A. Neerincx, and W. Kraaij, "Detecting work stress in offices by combining unobtrusive sensors," *IEEE Transactions on Affective Computing*, vol. 9, no. 2, 2018.

[17] J. Aigrain, M. Spodenkiewicz, S. Dubuisson, M. Detyniecki, D. Cohen, and M. Chetouani, "Multimodal stress detection from multiple assessments," *IEEE Transactions on Affective Computing*, vol. 9, no. 4, pp. 491–506, 2018.

[18] S. Carreiro, K. K. Chintha, S. Shrestha, B. Chapman, D. Smelson, and P. Indic, "Wearable sensor-based detection of stress and craving in patients during treatment for substance use disorder: A mixed methods pilot study," *Drug and Alcohol Dependence*, vol. 209, p. 107929, 2020.

[19] S. Cohen, T. Kamarck, and R. Mermelstein, "A global measure of perceived stress," *Journal of Health and Social Behavior*, pp. 385–396, 1983.

[20] R. H. Birk, "On stress and subjectivity," *Theory & Psychology*, vol. 31, no. 2, pp. 254–272, 2021.

[21] A. Rapoport, "Culture and the subjective effects of stress," *Urban Ecology*, vol. 3, no. 3, pp. 241–261, 1978.

[22] R. K. Sah, M. McDonell, P. Pendry, S. Parent, H. Ghasemzadeh, and M. J. Cleveland, "Adarp: A multi modal dataset for stress and alcohol relapse quantification in real life setting," in *IEEE-EMBS International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2022, pp. 1–4.

[23] R. K. Sah and H. Ghasemzadeh, "Stress classification and personalization: Getting the most out of the least," *arXiv preprint arXiv:2107.05666*, 2021.

[24] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5353–5360.

[25] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the computational efficiency of training neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'14. MIT Press, 2014, p. 855–863.

[26] D. Bienstock, G. Muñoz, and S. Pokutta, "Principled deep neural network training through linear programming," *Discrete Optimization*, vol. 49, p. 100795, 2023.

[27] M. Matsubara, O. Augereau, C. L. Sanches, and K. Kise, "Emotional arousal estimation while reading comics based on physiological signal analysis," in *Proceedings of the 1st International Workshop on Comics Analysis, Processing and Understanding*, 2016, pp. 1–4.

[28] S. Ollander, C. Godin, A. Campagne, and S. Charbonnier, "A comparison of wearable and stationary sensors for stress detection," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 004 362–004 366.

[29] H. F. Posada-Quintero and K. H. Chon, "Innovations in electrodermal activity data collection and signal processing: A systematic review," *Sensors*, vol. 20, no. 2, p. 479, 2020.

[30] J. Tao, T. Tan, and R. Picard, *Affective computing and intelligent interaction*. Springer, 2006, vol. 3784.

[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[32] G. M. Weiss, K. McCarthy, and B. Zabar, "Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?" *Dmin*, vol. 7, 2007.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR*, 2015.

[34] L. Liakopoulos, N. Stagakis, E. I. Zacharaki, and K. Moustakas, "Cnn-based stress and emotion recognition in ambulatory settings," in *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2021.