A Resilient and Hierarchical IoT-based Solution for Stress Monitoring in Everyday Settings

Shiyi Jiang¹, Farshad Firouzi¹, Krishnendu Chakrabarty¹, and Eric B. Elbogen²

Abstract—Long-term stress is a global health concern because it impacts our physical and mental health. The emergence of Internet of Things (IoT) and Artificial Intelligence (AI) makes stress monitoring and treatment more accessible compared to today’s physician-centered healthcare system. However, existing solutions either fail to incorporate IoT technology or are not cost-effective. We propose a resilient, hierarchical IoT-based solution for stress monitoring to tackle the above problems. Multimodal data was collected from wearable sensors and underwent preprocessing, feature extraction, and multiple imputation. We applied three feature-selection methods prior to lightweight SVM classification at the edge layer, and utilized a CNN and a matching network model in the cloud layer. We obtained an accuracy of 86.7347% and an F1 score of 0.8725 at the edge using only 10 features selected based on the Fisher score. An accuracy of 98.9247% and an F1 score of 0.9876 was achieved by a matching network model based on electrocardiogram (ECG) data. The trade-off between the communication cost from the edge to the cloud and the overall accuracy was evaluated. Our hierarchical-IoT solution for stress-level evaluation provides insights into the potentiality of IoT and AI technology-based eHealth solutions.

Index Terms—Artificial Intelligence (AI), eHealth, Internet of Things (IoT)

I. INTRODUCTION

Stress is a major societal health concern today. Long-term stress can lead to a wide variety of ailments, e.g., depression, eating disorders, personality disorders, substance abuse, and schizophrenia [1], [2]. It can also alter the physiology of the brain, leading to increased risk of obesity, diabetes, and heart disease [3], [4]. Early treatment of stress is therefore critical; however, today’s physician-centered healthcare model is inadequate for addressing stress-related concerns because it tends to be reactive. In particular, the physician-centered model requires a patient to travel to a clinic multiple times a week, resulting in high cost for the patient and high dropout rates. Another major challenge associated with this model is that of limited observations, which leads to inefficient diagnosis and treatment. Recent advances in eHealth based on the convergence of mobile technology, Wearable Internet of Things (WIoT), and Artificial Intelligence (AI) enables new ways to potentially track and address barriers to stress monitoring and treatment. These technologies improve patient outcomes by offering a holistic, collaborative, and measurement-based solution that utilizes patient-generated insights, real-time remote stress tracking and monitoring, just-in-time intervention, and personalized treatment.

Stress is induced when an individual encounters harmful physical or mental environments, resulting in a fight-or-flight response that release hormones, changing the behavior of the central nervous system in human bodies [2]. Stress is usually accompanied by increased heart rate, quicker breathing, and tighter muscles [2]. Physiological signals such as electrocardiogram (ECG), electromyogram (EMG), electrical dermal activities (EDA), interchangeably, Galvanic skin response (GSR), respiratory signal, and photoplethysmogram (PPG) can be measured as stress indicators [5], [6], [7], [8], [9]. In addition to physiological signals, daily activities, body movements or gestures, and environmental factors are other potential elements for stress evaluation [10]. Recent studies in the area of stress monitoring have led to machine learning (ML) models for stress evaluation and classification. Aqajari et al. [11] and Oskooei et al. [12] proposed effective ways of physiological signal feature extraction through neural networks such as Convolutional Neural Networks (CNN) and Convolutional Autoencoders (CAE). Others such as Tripathi and Beigi [13], Busso et al. [14] and Jaiswal et al. [15] proposed to use images and audio information captured from recorded videos with pre-designed scenes for stress classification. In addition, utilizing text messages posted on social media for stress identification was another area explored by Guntuku et al. [16] and others [17], [18]. In addition to using ML methods such as random forest (RF) [11], recent work has also focused on multi-layer perceptron (MLP) and CNNs [19], [20], [21], [22], [23], [24]..

Despite this large body of recent work, today’s solutions suffer from several shortcomings and limitations. First, they are not suitable for implementation on endpoint IoT devices. The complexity and the number of features associated with these methods require high computational effort, which makes them unsuitable for endpoint IoT devices where the computation resources, memory, and the power source are severely constrained. On the other hand, high latency and power-hungry data transmission to remote servers requires a careful trade-off between computing at the edge and in the cloud; this trade-off impacts energy consumption, battery life-time, end-to-end response time, as well as the accuracy/reliability of the ML model. Another challenge that has not been adequately tackled is related to missing data, which is a common problem for data gathered from sensors that measure physiological signals. Missing data can potentially result in ineffective stress-level classification and long-term stress monitoring. Finally, the limited availability of labeled training data leads to reduced accuracy with today’s solutions.

In this paper, we overcome the above shortcomings by describing a robust and hierarchical IoT-based solution that enables real-time monitoring and early detection of stress situations outside of conventional clinical settings. The main contributions of this work include:

- We present a novel IoT-based solution for processing
multimodal data that quantifies the trade-offs between edge processing and cloud computing, and outperforms state-of-the-art classification methods for stress analysis.

- We describe a holistic hierarchical model that decomposes the problem into two classifiers in two levels of hierarchy, resulting in higher energy savings.
- We propose a few-shot learning model that alleviates the shortcoming of limited sample size and outperforms state-of-the-art stress classification techniques.
- We present a resilient technique to continuously identify stress situations despite the likelihood of missing data in connected IoT systems, e.g., during data acquisition.

The rest of the paper is organized as follows. In Section II, we describe related work on stress/emotion classification. In Section III, we present signal preprocessing, missing value imputation, and a general overview of proposed methods for each level of the hierarchy. We demonstrate detailed methods such as feature extraction, feature selection, and model training for both levels of the hierarchy and their interplay in Section IV. We present experimental results and evaluation methods in Section V, and conclude the paper in Section VI.

II. RELATED PRIOR WORK

Stress/emotion recognition has been a popular research area for decades. Researchers initially used images and audio information for emotion recognition and later shifted their focus to stressed emotion identification using machine learning methods. Research on basic emotion (happiness, sadness and neutral) recognition using speech and facial expressions began two decades ago with statistical models such as the hidden Markov model [25]. Motivated by the lack of natural interactions between participants and insufficient contextualization of previously developed emotional databases, Busso et al. [14] created an interactive emotional dyadic motion-capture database (IEMOCAP), which recorded improvised scenes by two actors with specific emotions: happiness, anger, sadness, frustration and neutral state. They further developed an MSP-IMPROV database similar to IEMOCAP with four designed emotions: anger, sadness, happiness and neutral. They applied Support Vector Machine (SVM) with sequential minimal optimization on speech and facial expression separately, and they reached an optimal accuracy of 70% using facial expression. Accuracy, which ranges from 0% to 100%, and the F1 score, which ranges from 0 to 1, are common metrics used in machine learning for model performance evaluation. The higher these values are, the better performance the model has. Formal definitions are presented in Section V. B.

Tripathi and Beigi [13] employed IEMOCAP and applied multiple CNN and Long Short-term Memory (LSTM) on speech, text, image separately and merged them using a fully connected layer for final 4-class emotion detection. The final accuracy was reported to be over 70%. Jaiswal et al. [15] created a multimodal dataset of stressed emotion (MuSE) with audio, visual, lexical, thermal and physiological information, and performed multiple binary stress classification using unimodal deep neural networks. They reported an accuracy of 67% using audio information. Hossain et al. [20] integrated emotion recognition with IoT by feeding audio-visual data to models trained in the cloud.

Researchers have also used plaintext scraped from social media for stress detection. Lin et al. [17] applied CNN in combination with a factor-graph model to text streamed from Weibo, resulting in a 6-9% increase in the F1 score compared to the use of logistic regression, SVM, RF and decision tree. Turcan and McKeown [18] approached this problem by applying classifiers such as Gated Recurrent Neural Network (GRNN), CNN, and logistic regression using a dataset from Reddit. They achieved an F1 score of 0.79 using logistic regression. Gunakku et al. [16] manipulated text from Facebook and Twitter with regularized linear regression and SVM, demonstrating that the addition of social media language outperformed other sociodemographic features.

Additionally, researchers have utilized physiological signals for stress classification. Schmidt et al. [9] developed a Wearable Stress and Affect Detection (WESAD) dataset with multiple physiological signals such as EDA, ECG and blood volume pressure (BVP), and built five models for 3-class (stressed, neutral, amusement) and binary classification from extracted features. AdaBoost provided the highest performance with around 80% accuracy for 3-class classification, and Linear Discriminant Analysis (LDA) generated the highest accuracy (over 92%) for binary classification [9]. Lin et al. [21] processed the WESAD dataset for 3-class stress classification by building multiple CNN models using single or small groups of physiological signals and the result was then fed to RF for a final decision; an accuracy of 85%. Mozafari et al. [22] utilized EDA, PPG, abdominal and thoracic respiration signals from the mental arousal level recognition competition dataset and performed 5-level stress classification using the Nondominated Sorting Genetic Algorithm II (NSGA-II) with a set of machine learning methods: fine tree, bagged tree, cubic SVM and ensemble discriminant subspace. The best performance using ensemble discriminant subspace was over 68% [22].

In recent years, attempts have also been made to increase classification accuracy based on a single type of signal and data fusion method. Aqajari et al. [11] proposed a CNN-based feature extraction on EDA signals and showed significant binary classification accuracy enhancement using K-Nearest Neighbors (KNN), SVM, RF and Naive Bayes (NB) compared to the use of statistically extracted features. KNN was found to be the most effective model with over 90% accuracy [11]. Oskooei et al. [12] suggested an unsupervised method for binary stress classification using heart-rate variability (HRV) extracted through CAE. Sarkar and Etemad [23] proposed a self-supervised ECG learning for classifying emotion arousal, valence and stress by first training a multi-tasking CNN using transformed signals and then transferring the network for emotion prediction task. They used four public datasets and achieved 95% accuracy on the WESAD dataset [23]. Stewart et al. [24] explored the effectiveness of meta-learning methods. They chose a neural process, composed of a classification network and an encoder network, for binary stress classification using statistical features extracted from the ECG and EDA signals from the WESAD and Drivedb datasets [24]. They evaluated the performance
based on AUC, and showed improvement compared to other baseline classifiers: Lasso, SVM and KNN [24].

Researchers also utilize data with various modalities that can provide auxiliary information for stress prediction. Paredes et al. [10] collected multimodal data such as demographics, weather, daily activity, questionnaires, geolocations and body movements, and used RF to predict the stress level and provide recommendations. Alharthi et al. [19] used ECG in combination with location, weather and daily activity data for 3-class stress detection (stressed, normal, relaxed) with a NB solution implemented in the cloud and reached over 78% accuracy.

To present a clear comparison between prior work and our proposed solution, we summarize stress/emotion recognition work in terms of features, methods, data modality, IoT architecture and missing value resiliency. The comparison is shown in Table I based on the publicly available datasets listed in Table A.1 in the appendix. From Table I, we note that previous studies suffer from several limitations. Missing values during data collection or feature extraction can impair the performance of stress-level evaluation. To the best of our knowledge, prior work on stress recognition has not considered model resiliency. In addition, most prior solutions were not implemented in the context of IoT. [26] and [19] considered IoT by training models and making predictions in the cloud. However, these solutions suffer from high latency and energy consumption during data transmission from endpoint IoT devices to the cloud. To tackle these problems, we propose a resilient, hierarchical-IoT solution for stress-level evaluation that outperforms the state-of-the-art methods and alleviates the time and energy costs associated with indiscriminate data transmission.

III. DESCRIPTION OF PROPOSED SOLUTION

Fig. 1 presents our stress-level evaluation framework using a hierarchical IoT architecture with an edge layer and a cloud layer. The edge is composed of a multitude of devices such as sensors, smartwatches and smartphones that connect a local area network to a more extensive network. The cloud refers to servers that can be accessed over the internet to provide services with abundant computational resources.

We next introduce the workflow at the edge layer. Multimodal data such as ECG and respiratory signals collected from wearable sensors is preprocessed at the edge to ensure clean data for the next step. This process includes feature extraction. Imputation recovers missing values that are identified during feature extraction. Feature selection determines a subset of features that yields the best classification performance at the edge. Due to the constrained computational power at the edge, we utilize an SVM for lightweight classification using the selected features.

We can obtain predictions at the edge with rapid response time, but the edge prediction may suffer from unsatisfactory classification accuracy. Training models and performing inference in the cloud ensure high performance, but they lead to high time and energy costs for data transmission from the edge to the cloud. To tackle this problem, we propose a hierarchical-IoT solution that partitions the data between the edge and the cloud to obtain enhanced overall performance, while avoiding the time and energy costs associated with unnecessary communication and cloud computation. The criteria used for data partitioning are elaborated in Section IV. Samples extracted from the data segments that satisfy the criteria remain at the edge for lightweight classification and prediction. Samples that do not satisfy the criteria are sent to the cloud, and are then converted to images for advanced analysis. In the cloud, we utilize a CNN and a matching network model, and assess their performance. The model that has the better performance is selected for prediction.

A. Biosignals

Here we explain the biosignals that we utilized in the study of stress-level assessment. ECG measures electrical activity of the heart in millivolts (mV) using multiple electrodes placed on the body. EDA reflects the activity of the human sympathetic network via sweat glands, which can be measured as the conductance in microsiemens (µS). Respiratory-signal measures displacement variations based on volume changes of the thorax or abdomen during inhalation and exhalation. EMG measures electrical signals associated with the muscle fibers in millivolts (mV) sent from motor sensory neurons in the brain. Accelerometer data tracks body movements in X, Y, Z coordinates. BVP is obtained from PPG, which measures blood volume change optically by recording changes of light absorption in the skin. The skin temperature is measured in Celsius.

B. Signal Preprocessing

We apply downsampling, filtering, min-max normalization, and segmentation for signal preprocessing.

a) Downsampling: We perform downsampling to reduce processing time and memory usage at the edge. ECG, EDA, EMG, respiratory signal, temperature and accelerometer data were downsampled from 700 Hz to 100 Hz. BVP was maintained at the original 64 Hz sampling rate.

b) Filtering, Normalization, and Segmentation: We calculated the moving average with a 1-second window and min-max normalization to all signals to remove noise caused by environmental differences in participants as suggested by [11]. All signals were sliced with a 90-second window for feature extraction. For cloud analysis, EDA and respiratory signal were converted to images of size 128 × 128 based on 90-second window segments. ECG was first split into 90-second windows, and for each 90-second window, it was further divided into images of size 128 × 128 that contain one R-peak per image using the R-peak detection algorithm [27].

C. Missing Value Recovery

Many methods exist for resolving missing values in a dataset [28]. The simplest one is deletion by directly removing sample points. Imputation-based methods such as KNN imputation estimate missing values by weighing previous k nearest samples. Model-based methods construct either statistical models such as maximum likelihood estimation or machine learning models such as SVM and neural networks based on previous data.
| Reference | Dataset | Methods | Features | Multimodality | Hierarchical IoT | Missing-value resiliency |
|-----------|---------|---------|----------|---------------|------------------|------------------------|
| [10]      | Private | RF      | Questionnaires, Demographics, GPS, Calendar, Social network usage | Y              | N                | N                      |
| [11]      | WESAD   | SVM with sequential minimal optimization | Audio, Face/Head/Hand movements | Y              | N                | N                      |
| [12]      | Private | NB      | ECG, Time, Day, Location, Weather, Activity | Y              | N                | N                      |
| [13]      | Weibo   | CNN, CAE, Factor graph model, SVM, RF, DT, Logistic regression | Text | N              | N                | N                      |
| [14]      | WESAD   | DT, AdaBoost, KNN, RF, LDA | ECG, EDA, ACC, PPG, EMG, Temperature, Respiratory signal | Y              | N                | N                      |
| [15]      | IEMOCAP | MLP, CNN, LSTM | Speech, Face/Head/Hand movements | Y              | N                | N                      |
| [16]      | SWELL, WESAD | RF, Extra tree | HRV, EDA | Y              | N                | N                      |
| [17]      | WESAD   | Multiple CNN with RF | ECG, EDA, ACC, PPG, EMG, Temperature, Respiratory signal | Y              | N                | N                      |
| [18]      | Mental Arousal Level Recognition Competition dataset | NSGA-II, Fine tree, Bagged tree, Cubic SVM, Ensemble discriminant subspace | EDA, PPG, AR, TR | Y              | N                | N                      |
| [19]      | Dreaddit | SVM, Logistic regression, NB, Perceptron, DT, GRNN, CNN | Text | N              | N                | N                      |
| [20]      | FB, Twitter | Linear regression with regularization, SVM | Text | N              | N                | N                      |
| [21]      | WESAD   | CNN, KNN, SVM, RF, NB | EDA | N              | N                | N                      |
| [22]      | Private | CAE, LAE, KNN | HRV | N              | N                | N                      |
| [23]      | AMIGOS, DREAMER, WESAD, SWELL | Self-supervised multi-task CNN | ECG | N              | N                | N                      |
| [24]      | MuSE    | Multiple unimodal DNNs | Text, Speech, Facial expression, Temperature, EDA, HR | Y              | N                | N                      |
| [25]      | Drivedb, WESAD | SVM | Speech, Face images | Y              | N                | N                      |
| [26]      | Proposed work | KNN, SVM, Logistic regression, Neural process | ECG, EDA | Y              | N                | N                      |

According to [28], when the missing window is small (below one hour), KNN imputation achieves the lowest root-mean-square error compared to other types of methods. The above methods are categorized as single imputation, where the missing value is replaced by one value. However, single imputation may lead to high bias since it fails to reveal sampling variability [29].

We therefore utilize multiple imputation, which is useful
for reducing bias and increasing analysis precision, and is resilient to outliers [30]. Multiple imputation generates \( n \) distinct datasets with missing values in the original dataset filled by \( n \) potential values. To impute the dataset, we utilize sequential regression imputation by estimating one feature with missing values at a time based on regression models established using the other features. Creating separate regression models is beneficial to adjusting for covariates in targeted analysis and for bias reduction [31]. We repeat the imputation \( n \) times to obtain \( n \) distinct datasets. We train a lightweight classification model at the edge for each dataset and obtain the predictions. We average the predictions from the \( n \) models as the final result of multiple imputation. The overall workflow is shown in Fig. 2. We used a baseline obtained by filling missing data with values chosen uniformly from the dataset for comparison.

In order to obtain customized imputations, auxiliary data (signals collected from other sensors, information acquired from surveys, etc.) can be utilized in combination with the outputs of multiple imputation. However, due to the limited sample size of the selected dataset, we were not able to perform customization for missing value imputation.

IV. STRESS RECOGNITION USING HIERARCHICAL-IoT

A. Stress Recognition at the Edge

1) Feature Extraction: We computed maximum, minimum, sum, standard deviation, mean, skewness and kurtosis as statistical features for each segment of all seven signals. Furthermore, additional features based on the specific properties of signals were computed. For accelerometer data, the area under the curve (AUC) and the mean peak amplitude along each axis were calculated. The slope and dynamic range were calculated for temperature. The respiratory signal was filtered to distinguish between cycles of inhalation and exhalation, and we obtained the mean, standard deviation of inhalation and exhalation, ratio of inhalation to exhalation, range of respiration, and mean respiratory rate as features. The skin conductance response (SCR) was extracted from EDA based on [32] and [33]. The mean, standard deviation, sum amplitude, number of SCR peaks, AUC, slope and dynamic range were used as features. The dynamic range, AUC, median, 10th percentile, 90th percentile, mean, standard deviation, number of peaks and sum of peak amplitude of EMG were also evaluated. The time intervals between R-peaks (N-N intervals) were identified using the peak detection algorithm [34]. They were then utilized to compute a series of features such as relative and normalized powers and LF/HF ratio using four frequency bands: ULF (0-0.003 Hz), VLF (0.003-0.04 Hz), LF (0.04-0.15 Hz), and HF (0.15-0.4 Hz) based on [35] and [36]. Fig. 3 shows some examples of signals before and after filtering. All the computed features are listed in Table II for each signal.

![Comparison between raw and filtered ECG signal](a)
![Comparison between raw and filtered EDA signal](b)
![Comparison between raw and filtered respiratory signal](c)

Fig. 3: Comparison between raw and filtered ECG, EDA, and respiratory signal.
Fig. 2: Steps associated with the recovery of missing values.

### TABLE II: ADDITIONAL FEATURES EXTRACTED FROM SIGNALS

| Signals          | Additional features                                      |
|------------------|---------------------------------------------------------|
| Accelerometer data | AUC and mean peak amplitude for each axis               |
| Temperature      | Slope, dynamic range                                    |
| Respiratory signal | Mean, std of inhalation and exhalation, inhalation/exhalation ratio, stretch, mean respiratory rate |
| EDA              | Slope, dynamic range, mean, std of SCR peaks, # of SCR peaks, sum of SCR amplitude, AUC |
| EMG              | Dynamic range, AUC, median, 10th percentile, 90th percentile, mean, std of peaks, # of peaks, sum of peak amplitude |
| ECG              | Sum of peaks, LF/HF ratio, total power from ULF, VLF, LF and HF, relative and normalized powers, RMSSD |

2) **Feature Selection**: We applied the Fisher score [37], Joint Mutual Information (JMI) [38], and Double Input Symmetrical Relevance (DISR) [39] to the extracted features described above, and compared their effectiveness in selecting crucial features for stress assessment. We selected these methods based on their computational efficiency and ability to minimize overfitting. The Fisher score computes a score for every feature independently based on Equation (1) below, where $\mathcal{F}(i)$ is the Fisher score for feature $i$, $n_j$ is the number of samples in class $j$, $\mu_j$ and $\sigma_j^2$ are the mean and standard deviation of feature $i$ in samples from class $j$, and $\mu_i$ is the mean of feature $i$ among all samples.

$$\mathcal{F}(i) = \frac{\sum n_j(\mu_j^i - \mu_i)^2}{\sum n_j(\sigma_j^i)^2}.$$  

The JMI measure is derived based on Mutual Information (MI) [40], which computes the amount of shared information between two variables $X$ and $Y$ as defined in Equation (2) where $\mathcal{H}$ is the entropy [40].

$$I(X;Y) = \mathcal{H}(X) - \mathcal{H}(X|Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{2}$$

Note that JMI measures shared information between the target variable $Y$ and the set of variables $X_i$ and $X_j$ as indicated in Equation (3), where $S$ is the set of currently selected features. It evaluates whether the addition of feature $X_i \notin S$ expands the amount of shared information.

$$I_{\text{JMI}}(X_i) = \sum_{X_j \in S} I(X_i;X_j;Y) \tag{3}$$

DISR is a method built upon JMI that measures the symmetrical relevance of all combinations of two variables in a set. It first selects a feature that has the maximum MI with target variable $Y$. It then performs a greedy step of adding features one-by-one to the selected features that yields a maximum score. The score is computed based on Equation (4) below, with $S$ being the set of selected features and $X_i \notin S$. As shown in [39], DISR is suitable for dealing with datasets that have high feature-to-sample ratio [38].

$$I_{\text{DISR}}(X_i) = \sum_{X_j \in S} \frac{I(X_i;X_j;Y)}{\mathcal{H}(X_i,X_j;Y)} \tag{4}$$

Features were then ranked based on the selection metrics. For each method, we selected the set of features that lead to the highest accuracy using the lightweight classifier. We further compared the performance and evaluated the efficacy of feature-selection methods.

3) **Support Vector Machine (SVM)**: SVM is a widely-used supervised learning algorithm that is known for its effectiveness and memory efficiency. We therefore utilize SVM for lightweight stress classification on computational resource-restricted devices. It aims at finding a hyperplane that separates data from different classes. It finds the hyperplane by maximizing the margin, i.e., the distance of the closest data point to the hyperplane. Formally, SVM finds the margin $y(x)$ as indicated in Equation (5), where $w$ represents the weight and $b$ represents the bias. For linearly separable data, it maximizes the margin as defined by Equation (6), where $t_i$
represents the label of the test point. For linearly non-separable data, SVM introduces a penalty term \( \epsilon_n \) for misclassification in the constraints as shown in Equation (7), where \( \epsilon_n = 0 \) for correct classification, \( \epsilon_n > 1 \) for incorrect classification, and \( 0 < \epsilon_n \leq 1 \) for correct classification inside the margin.

\[
y(x) = w^T x + b
\]

\[
\text{argmax}_{w,b} \frac{1}{n} \min_{n} t_n y(x_n)
\]

\[
\begin{align*}
& t_n y(x_n) \geq 1, \forall x_n, \text{ linearly separable} \\
& t_n y(x_n) \geq 1 - \epsilon_n, \forall x_n, \text{ linearly non-separable}
\end{align*}
\]

B. Stress Recognition in the Cloud

To ensure overall high performance, samples selected by our decision criteria are delivered to the cloud, and converted to images for advanced analysis. We utilized a CNN and a matching network that are known for their high performance on image classification.

1) Convolutional Neural Network (CNN): A CNN is a category of neural networks that consists of combinations of convolutional layers, pooling layers, and fully connected layers; it is typically used for image classification. The convolutional layer applies convolution between a set of filters and the image, converting the image to feature maps. The pooling layer subsamples values in non-overlapping neighbors in the maps by either taking the maximum or the average in the neighborhood. The fully connected layer applies linear transforms to the input data. The activation function introduces non-linearity to the output of the fully connected layer.

All parameters are adjusted through back-propagation to minimize a pre-defined loss function based on the output and the ground truth. Inspired by the CNN architecture proposed in [41], we design a CNN with the architecture detailed in Table III.

| TABLE III: PROPOSED CNN ARCHITECTURE |
|--------------------------------------|
| Layer | Layer type | Kernel size | Stride | Input dimensions | Output dimensions |
|-------|------------|-------------|--------|------------------|-------------------|
| 1     | Conv2d    | 3 x 3       | 1      | 128 x 128 x 3    | 126 x 126 x 32    |
| 2     | MaxPool   | 2 x 2       | 2      | 126 x 126 x 32   | 63 x 63 x 32      |
| 3     | Conv2d    | 3 x 3       | 1      | 63 x 63 x 32     | 61 x 61 x 64      |
| 4     | MaxPool   | 2 x 2       | 2      | 61 x 61 x 64     | 30 x 30 x 64      |
| 5     | Conv2d    | 3 x 3       | 1      | 30 x 30 x 64     | 28 x 28 x 128     |
| 6     | Conv2d    | 3 x 3       | 1      | 28 x 28 x 128    | 26 x 26 x 256     |
| 7     | Conv2d    | 3 x 3       | 1      | 26 x 26 x 256    | 24 x 24 x 256     |
| 8     | Linear    | -           | -      | 24 x 24 x 256    | 2048              |
| 9     | Linear    | -           | -      | 2048             | 2                 |

2) Matching Networks: Matching networks are used for few-shot learning, which is aimed at achieving high classification accuracy with very few training samples in each class. These networks are typically characterized in the form of \( k \)-way \( n \)-shot learning, where \( k \) is the number of classes in the dataset and \( n \) is the number of samples in each class. In the scenario of limited sample size in the publicly available datasets for stress assessment, matching networks have the potential to achieve higher performance than typical deep learning models.

We use the matching networks approach described in [42] for one-shot learning. It first embeds all samples (labeled support set and unlabeled samples) via a convolutional encoder. Modifications are made on the original embedding via full context embeddings \( f(\hat{x}, S) \) as presented in Equation (8), where \( S \) is the support set, \( \hat{x} \in S \), \( f \) is a function that denotes the CNN for the original embedding, \( g \) is a bidirectional LSTM, and \( K \) is the number of unrolling steps in LSTM. [42] shows that this re-embedding is especially useful when samples \( x_i \) and \( x_j \) from different classes are similar.

\[
f(\hat{x}, S) = \text{att-LSTM}(f(\hat{x}), g(S), K)
\]

After the embedding step, pairwise distances between support and unlabeled samples are calculated based on the given matching function. The Softmax function is applied to pairwise distances and the results are multiplied with true labels from the support sets to generate predicted outputs for unlabeled samples.

We use the matching network architecture described in Fig 4. Built upon the architecture depicted in [42], we constructed an encoder with five convolutional layers with batch normalization and five max-pooling layers for the respiratory signal and ECG. The encoder for EDA was assembled using four convolutional layers with batch normalization and four max-pooling layers.

C. Edge-Cloud Interplay

As introduced in Section III, the performance at the edge layer is constrained by low computational power and limited resources. To address this problem, we utilize the cloud layer and offload a subset of data to the cloud for analysis, and then acquire results from the cloud. Through this interplay between the edge and the cloud, we achieve better overall performance. However, time and energy costs for data offloading from the edge to the cloud have to be considered. We therefore investigate the criteria for data offloading in order to ensure low transmission cost in terms of communication delay and energy consumption.

We next present our criteria for dispatching samples from the edge to the cloud. We handle the output from the SVM at the edge as follows. Instead of outputting the predicted labels, we output the probability of a sample being in each of the classes as shown in Equation (9), where \( p_j(x_i) \) is the probability of the sample \( x_i \) being in class \( j \), and \( n \) is the number of classes. We use the maximum probability \( \max(p_1(x_i), ..., p_n(x_i)) \) as a score for the sample to make data transmission decision. We establish a decision threshold \( c \), so that samples with scores below the threshold are offloaded to the cloud, and the rest of the samples remain at the edge, as shown in Equation (10). The samples left at the edge are assigned to the class where they have the maximum output probability.

\[
p_j(x_i) = p(j|x_i), 1 \leq j \leq n
\]

\[
\begin{align*}
& \max\{p_1(x_i), ..., p_n(x_i)\} < c, \frac{1}{n} \leq c \leq 1, \text{ cloud sample} \\
& \max\{p_1(x_i), ..., p_n(x_i)\} \geq c, \frac{1}{n} \leq c \leq 1, \text{ edge sample}
\end{align*}
\]

Because we perform binary classification, the threshold \( c \geq 0.5 \). Based on empirical evidence from experimentation, we
sampled 15 thresholds spaced uniformly from 0.55 to 1 to ensure that the number of samples dispatched to the cloud at each threshold is reasonable. For each threshold, we estimated the communication latency and energy consumption during data transmission. We examined the trade-off between classification accuracy and data transmission cost at the edge.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

1) Stress Dataset: We used the Wearable Stress and Affect Detection (WESAD) dataset to evaluate our solution and compare it with previous work [11], [23], [21], [24], [9]. The WESAD dataset was collected from 17 graduate student subjects but data from two of the subjects was excluded after evaluation. The data was recorded using RespiBAN, a wearable device placed on the chest, and Empatica E4 wristbands. RespiBAN measures ECG, EDA, EMG, temperature, accelerometer data, and respiratory signal at a 700 Hz sampling rate. Empatica E4 records accelerometer data sampled at 32 Hz, temperature sampled at 4 Hz, BVP sampled at 64 Hz, and EDA sampled at 4 Hz.

The participants were measured under three conditions: baseline, stress and amusement. The baseline setting includes 20-minute sitting/standing/reading magazines. The stress setting was based on Trier Social Stress Test (TSST), which includes public speaking and mental arithmetic tasks [43]. The amusement setting includes watching a set of funny video clips. There was a meditation interval between each condition.

The data was recorded with RespiBAN sampled at 700 Hz were downsampled to 100 Hz with normalization. The processed data was then divided into clips. There was a meditation interval between each condition.

2) Computation Platform: We performed signal pre-processing, feature extraction, missing value imputation, feature selection and lightweight classification on a MacBook Pro with a 2.6 GHz Dual-Core Intel Core i5 processor and 8 GB 1600 MHz DDR3 memory. Image classification in the cloud was performed on the Google Colab Pro platform with a Tesla V100 GPU.

3) Training Setup: As described in Section III, all signals recorded with RespiBAN sampled at 700 Hz were downsampled to 100 Hz with normalization. Signal filtering, segmentation, and feature computation were performed using the Scipy and pyHRV packages [27]. Multiple imputation for recovering missing values was performed using Bayesian ridge regression implemented in the scikit-learn package. According to [44], we imputed n = 5 and n = 9 distinct datasets after introducing missing values at rates of 5% and 10%, respectively. Feature selection was performed based on [45]. Due to a limited number of samples, the synthetic minority oversampling technique was applied and we trained a leave-one-out SVM using radial basis function kernel with tuned hyperparameters chosen via grid search using the scikit-learn package.

The CNN was trained using 70% of the samples with a batch size of 128 for 200 epochs. These hyperparameters were selected using random search. Data augmentation was performed with random affine transformation and random erasing to increase data diversity [46]. As a typical choice for binary classification, binary cross entropy (L_{ce}), as shown in Equation (11), was selected as the loss function, where P is the true distribution of the data and Q is the estimated distribution.

\[ L_{ce} = -P \log_2 Q - (1 - P) \log_2 (1 - Q) \] (11)

Based on [42] and random search, the matching network was trained using 70% of the samples with a batch size of 128 for 400 epochs. Cosine similarity (L_{cs}), as shown in Equation (12), was selected as the matching function to measure the distance between the prediction and the ground truth.

\[ L_{cs} = \frac{\hat{y} \cdot y}{\|y\|_2 \cdot \|\hat{y}\|_2} \] (12)

Adam optimization [47], [48] was applied to the CNN with learning rates of 1×10^{-3} for the respiratory signal and 1×10^{-4} for EDA and ECG. A weight decay of 4×10^{-2} was applied to the ECG model. The same optimization was applied to the matching network with learning rates of 1×10^{-4} for the respiratory signal, 3×10^{-4} for EDA, and 1×10^{-5} for ECG. These hyperparameters were tuned using random search, and all other hyperparameters were maintained at default settings from Pytorch.

B. Evaluation Metrics

By convention, we assess the performance of classifiers using the metrics of classification accuracy, precision, recall and F1 score. Precision measures the percentage of correct predictions in the positive class and recall calculates the percentage of correct prediction in samples that are all predicted to be positive, as described in Equation (13). The F1 score is a combined measurement of precision and recall values as shown.
in Equation (14). The goal is to ensure high performance of the classifier.

\[
\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}
\]

\[
\text{F1 score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]  

(C. Performance Evaluation)

1) Feature Selection: As described in Section III, a total of 100 features were computed. To test the robustness of our model, we introduced missing values at rates of 5% and 10% by convention [49], [50], and performed multiple imputation accordingly. Features were ranked based on the Fisher score, JMI, and DISR, accordingly. Considering the computational resource restriction at the edge layer, we evaluated the impact of feature size on the classification accuracy, in order to obtain the minimum number of features that yields the optimum accuracy. Fig. 6 shows how the classification accuracy varies based on the top \( k \) features chosen using three selection methods, where \( k \) ranges from 3 to 15 to capture the growth in accuracy and ensure that convergence is reached. As the number of chosen features increases, the accuracy of the SVM increases and finally converges. We chose top ten features selected by each of these three methods to ensure that the SVM reaches its optimum accuracy under all three conditions. Selected features using different methods are listed in Table IV in the order of their rankings. The percentage contributions of each signal are presented in Fig. 6. We see that EDA and the respiratory signal are major contributors to the classification accuracy for all three feature selection methods.

According to Table IV, we see some overlapping key features, namely mean respiratory rate, standard deviation of inhalation, the mean and standard deviation of SCR, and EDA AUC, selected by these three methods. The selected features belong to the respiratory signal and EDA, which is consistent with the results from Fig. 6.

The Fisher score is efficient for computation, but one major disadvantage associated with it is that it computes each feature independently without considering feature redundancy. We can also see from Table IV that the Fisher score has the worse performance compared to the other two methods, when noise was introduced by multiple imputation at higher missing rates. By examining the distributions of the features, we found that the feature with similar intra-class and inter-class mean and variance can lead to a relatively high Fisher score, even though the feature introduces significant noise to the whole dataset, degrading the performance of the classifier.

JMI solves the problem of feature redundancy by measuring the amount of mutual information obtained by adding a new feature. DISR is considered to be a normalized form of JMI that measures the relative reduction of uncertainty when a new feature is added. Among these two information-based feature selection methods, we see from Table IV that DISR is less impaired by missing values, and it generates a better result compared to that from JMI. We therefore conclude that DISR is a better method for physiological feature selection.

2) SVM Classification at the Edge: In Table V we present the performance of the SVM classifier at the edge layer before and after missing-value imputation using the top ten features selected by the Fisher score, JMI, and DISR. Signals are segmented in 90-second windows; some features cannot be calculated due to the lack of available patterns in the windows, resulting in a missing rate of 0.22%. According to Table V at the original missing rate, the features selected by the Fisher score yield the best accuracy of 86.7347% and an F1 score of 0.8725. At higher missing rates, the accuracy and the F1 score after imputation lead to at least a 5% increase compared to the scores obtained before imputation using all three methods. By comparing the performance at all the missing rates, we conclude that DISR outperforms the other two methods.

| Methods | Selected features |
|---------|-------------------|
| Fisher  | mean respiratory rate, inhalation std, SCR std, EDA AUC, SCR mean, BVP sum, ECG kurtosis, BVP mean, BVP skewness, SCR amplitude |
| JMI     | mean respiratory rate, inhalation std, SCR std, SCR mean, EDA AUC, ECG mean, BVP mean, ECG skewness, Respiratory signal sketchness |
| DISR    | mean respiratory rate, relative power, inhalation std, BVP sum, SCR std, ECG kurtosis, SCR mean, EDA AUC, normalized power, respiratory signal sketchness |

2) SVM Classification at the Edge: In Table IV, we present the evaluation results for the CNN and the matching network model. We see that the performance of the matching network using all three signals, we pooled ECG predictions based on the second window. To maintain consistency of the outputs from the CNN individually. Since ECG was segmented by R-peaks, a considerable number of images were generated in one 90-second window. To maintain consistency of the outputs from all three signals, we pooled ECG predictions based on the 90-second windows using a majority vote. Table V presents the evaluation results for the CNN and the matching network model. We see that the performance of the matching network model is superior to that for the CNN model. The CNN achieves comparable performance using ECG with an

| Methods | Before imputation | After imputation |
|---------|-------------------|------------------|
| Missing rate (%) | Method | Accuracy (%) | F1 score | Accuracy (%) | F1 score |
| 0.22 | Fisher | 85.2041 | 0.8549 | 86.7347 | 0.8725 |
| DISR | 80.6122 | 0.8041 | 85.4592 | 0.8613 |
| 5 | Fisher | 71.9387 | 0.6646 | 76.2755 | 0.7223 |
| DISR | 70.2983 | 0.6498 | 85.4592 | 0.8613 |
| 10 | Fisher | 58.9286 | 0.566 | 83.1632 | 0.8366 |
| DISR | 80.8673 | 0.812 | 85.4592 | 0.8585 |

3) Stress Classification in the Cloud: As indicated in Section IV, we established two neural network models for image classification using the respiratory signal, EDA and ECG individually. Since ECG was segmented by R-peaks, the results from Table V are presented in order to obtain the minimum number of features that yields the optimum accuracy. Fig. 6 shows how the classification accuracy varies based on the top \( k \) features chosen using three selection methods, where \( k \) ranges from 3 to 15 to capture the growth in accuracy and ensure that convergence is reached. As the number of chosen features increases, the accuracy of the SVM increases and finally converges. We chose top ten features selected by each of these three methods to ensure that the SVM reaches its optimum accuracy under all three conditions. Selected features using different methods are listed in Table IV in the order of their rankings. The percentage contributions of each signal are presented in Fig. 6. We see that EDA and the respiratory signal are major contributors to the classification accuracy for all three feature selection methods.

According to Table IV, we see some overlapping key features, namely mean respiratory rate, standard deviation of inhalation, the mean and standard deviation of SCR, and EDA AUC, selected by these three methods. The selected features belong to the respiratory signal and EDA, which is consistent with the results from Fig. 6.

The Fisher score is efficient for computation, but one major disadvantage associated with it is that it computes each feature independently without considering feature redundancy. We can also see from Table IV that the Fisher score has the worse performance compared to the other two methods, when noise was introduced by multiple imputation at higher missing rates. By examining the distributions of the features, we found that the feature with similar intra-class and inter-class mean and variance can lead to a relatively high Fisher score, even though the feature introduces significant noise to the whole dataset, degrading the performance of the classifier.

JMI solves the problem of feature redundancy by measuring the amount of mutual information obtained by adding a new feature. DISR is considered to be a normalized form of JMI that measures the relative reduction of uncertainty when a new feature is added. Among these two information-based feature selection methods, we see from Table IV that DISR is less impaired by missing values, and it generates a better result compared to that from JMI. We therefore conclude that DISR is a better method for physiological feature selection.
accuracy of 98.9247% and an F1 score of 0.9873. Additionally, the overall performance in the cloud significantly exceeds the results obtained at the edge.

We next compare the performance of our models in the cloud with prior work that utilizes selected signals and labels from the WESAD dataset; the results are shown in Table VII. Considering the performance obtained from prior work that used the same evaluation metrics, our models in the cloud outperform prior methods with at least a 2% increase in the accuracy and the F1 score.

Based on the performance of the models using different signals, we conclude that ECG serves as the best indicator for stress level using the matching network in the cloud. Based on Table VII, the matching network outperforms the CNN model for all three signals, indicating the robustness of the few-shot learning models under the scenario of a limited number of samples in each class.

4) Trade-off between overall accuracy and cost: The results in Table V and Table VI show that inferring using samples at the edge results in inadequate accuracy due to limited computational power; therefore, a subset of samples were sent to the cloud to obtain higher overall accuracy in our hierarchical architecture. In order to investigate the trade-off between the overall accuracy and the energy and communication costs during data transmission from the edge to the cloud, we sampled a series of decision thresholds to obtain the corresponding partitioned samples and the performance at the edge and the
cloud, respectively. Here we use the CNN model with ECG input in the cloud as an example.

Based on empirical evidence from experimentation, we fed 30% of the data to the edge for inference. For each decision threshold, using the criteria defined in Section IV, samples with a score below the threshold were offloaded to the cloud for advanced classification. We performed lightweight classification using the rest of the samples remained at the edge. Confusion matrices were computed at both the edge and the cloud, and both these matrices were utilized for the overall performance assessment. The number of samples and the accuracy obtained at the edge and in the cloud, and their combined accuracy are presented in Fig. 7. We note that the accuracy drops slightly as the number of samples increases at both the edge and the cloud. As the decision threshold increases, more samples are sent to the cloud, resulting in an increase in the overall accuracy.

To compute the communication latency and the energy consumption during data offloading to the cloud, we first measured the upload bandwidth of 3G, LTE, and Wi-Fi using TestMy.net [51] on a mobile phone; see Table VIII. For each decision threshold, we calculated the total size of the data offloaded to the cloud in KB by multiplying the number of samples with the size of a 90-second-window signal sample (72 KB). We then divided the total cloud sample size by the upload internet speed to obtain the communication latency. We computed the energy consumption based on the power model presented in [52]. Equation (15) describes the upload power model, where $P_{\text{upload}}$ is the power level for uploading, $\alpha_u$ (mW/Mbps) and $\beta$ (mW) are parameters with the values provided in [52], and $t_u$ is the upload bandwidth in Mbps. The power value was then converted to J/s and multiplied with the communication latency at each decision threshold to determine the energy consumption during data transmission.

$$P_{\text{upload}} = \alpha_u t_u + \beta$$ (15)

Fig. 8 and Fig. 9 present the energy consumption and the communication latency during data transmission. As the decision threshold increases, the cost using 3G increases rapidly, and the cost escalates at higher thresholds using LTE. Wi-Fi incurs the minimum energy and time cost at all thresholds compared to LTE and 3G. We therefore assess the trade-off between the overall performance and the data transmission cost using only Wi-Fi.

| Network | Bandwidth (Mbps) | Data rate (KB/s) |
|---------|------------------|-----------------|
| Wi-Fi   | 6.1              | 759             |
| LTE     | 830              | 104             |
| 3G      | 55               | 7               |

Fig. 8: Data transfer energy at different thresholds for different networks.

Fig. 9: Communication latency at different thresholds for different networks.
data transmission from the edge to the cloud layer. As shown in Fig. 10, the time and energy costs grow slowly at lower thresholds, but they grow faster at intermediate thresholds. The escalation of the cost declines at higher thresholds, because only a few new samples are offloaded to the cloud at each of the high thresholds, according to Fig. 7(b). The overall accuracy shares a similar trend with the cost as indicated in Fig. 7(c). The decision threshold is determined by the point with the minimal communication latency and energy consumption given the overall system accuracy. To illustrate this selection, suppose our target is an overall accuracy of 93.55%; in this case, a threshold of either 0.68 or 0.71 should be chosen to achieve the minimum communication latency and energy consumption for data transmission.

VI. CONCLUSION

Stress is becoming a significant concern with the potential of causing mental and physical health impairment in the long term. We have presented a resilient hierarchical-IoT system for stress level evaluation. The optimal features selected from multiple feature selection methods yield lightweight classification performance at the edge. Our results demonstrate the performance superiority of the proposed CNN and the matching network models in the cloud compared to the state-of-the-art methods. The trade-off between communication latency, data transfer energy, and the overall accuracy was evaluated. The next steps in this research include developing real-time stress monitoring, integrating a personalization component to the solution, incorporating feedback from clinical experts for model adjustment, and establishing a recommendation system for the subjects.

REFERENCES

[1] C. Hammen, “Stress and depression,” Annual Review of Clinical Psychology, vol. 1, pp. 293–319, 2005.
[2] H. Yaribeygi, Y. Panahi, H. Sahraei, T. Johnston, and A. Sahebkar, “The impact of stress on body function: A review,” EXCLI Journal, vol. 16, pp. 1057–1072, 2017.
[3] J. Kim and K. Yoon, “Stress: metaplastic effects in the hippocampus,” Trends in neurosciences, vol. 21, no. 12, pp. 505–509, 1998.
[4] S. Lupien, B. McEwen, M. Gunnar, and C. Heim, “Effects of stress throughout the lifespan on the brain, behaviour and cognition,” Nature Reviews Neuroscience, vol. 10, pp. 434–445, 2009.
[5] K. Plarre, A. Raij, and M. Scott, “Continuous inference of psychological stress from sensory measurements collected in the natural environment,” in 2011 10th International Conference on Information Processing in Sensor Networks (IPSN), 2011, pp. 97–108.
[6] K. Hovsepian, M. alAbsi, and S. Kumar, “cstress: Towards a gold standard for continuous stress assessment in the mobile environment,” in 2015 Ubicomp, 2015, pp. 493–504.
[7] A. Gjoreski, H. Gjoreski, and M. Gams, “Continuous stress detection using a wrist device: In laboratory and real life,” in 2016 Ubicomp, 2016, pp. 1185–1193.
[8] J. Kim and E. Andr, “Emotion recognition based on physiological changes in music listening,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 12, pp. 2067–2083, 2008.
[9] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. Van Laerhoven, “Introducing weasad, a multimodal dataset for wearable stress and affect detection,” in 2018 20th ACM International Conference on Multimodal Interaction (ICMI), 2018, pp. 400–408.
[10] P. Paredes, R. Gilad-Bachrach, M. Czerwinski, A. Roseway, K. Rowan, and J. Hernandez, “Poptherapy: Coping with stress through pop-culture,” in 2014 8th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2014, pp. 100–117.
[11] V. R. A. H. Qaqarji, E. K. Naeini, M. A. Mehrabadi, S. Labbah, A. Rahmani, and N. Dutt, “Gsr analysis for stress: Development and validation of an open source tool for noisy naturalistic gsr data,” 2020, arXiv:2005.01834.
[12] A. Oskooei, S. M. Chau, J. R. M. Weiss, A. Sridhar, M. R. Martinez, and B. Michel, “Destress: Deep learning for unsupervised identification of mental stress in firefighters from heart-rate variability (hrv) data,” 2019, arXiv:1911.13213.
[13] S. Tripathi and H. Beigi, “Multi-modal emotion recognition on iemocap dataset using deep learning,” 2018, arXiv:1804.05788.
[14] C. Bussu, S. Parthasarathy, A. Burmania, M. AbdelWahab, N. Sadoughi, and E. M. Provost, “Map-improv: An acted corpus of dyadic interactions to study emotion perception,” IEEE Transactions on Affective Computing, vol. 8, no. 1, pp. 67–80, 2017.
[15] M. Jaiswal, C.-P. Bara, Y. Luo, M. Burzo, R. Mihalcea, and E. M. Provost, “Muse: A multimodal dataset of stressed emotion,” in 2020 12th Conference on Language Resources and Evaluation (LREC), 2020, pp. 1499–1510.
[16] S. C. Guntuku, A. Buffone, K. Jaidka, J. C. Eichstaedt, and L. H. Ungar, “Understanding and measuring psychological stress using social media,” in the Thirteenth International AAAI Conference on Web and Social Media (ICWSM), 2019, pp. 214–225.
[17] H. Lin et al., “Detecting stress based on social interactions in social networks,” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, vol. 29, no. 9, pp. 1820–1833, 2017.
[18] E. Turcan and K. McKeown, “Dreaddit: A reddit dataset for stress analysis in social media,” 2019, arXiv:1911.00133.
[19] R. Alharthi, R. Alharthi, B. Guthier, and A. E. Saddik, “Casp: context-aware stress prediction system,” Multimed Tools Appl, vol. 78, pp. 9011–9031, 2019.
[20] K. Nkurikiyeyeu, A. Yokokubo, and G. Lopez, “The effect of person-specific biometrics in improving generic stress predictive models,” 2019, arXiv:1910.01770.
[21] J. Lin, S. Pan, C. S. Lee, and S. Oviatt, “An explainable deep fusion network for affect recognition using physiological signals,” in 2019 28th ACM International Conference on Information and Knowledge Management (CIKM ’19), 2019, pp. 2069–2072.
[22] M. Mozafar, E. Khazaei, S. Bagheri, N. Hasanazadeh, B. TaghiBeyglou, and F. Ghassemi, “Mental stress level detection using physiological signals and evolutionary algorithm,” in 2019 26th National and 4th International Iranian Conference on Biomedical Engineering (ICBME), 2019, pp. 75–81.
[23] P. Sarkar and A. Etemad, “Self-supervised ecg representation learning for emotion recognition,” in 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 3217–3221.
[24] C. Steward, A. Folarin, and R. Dobson, “Personalized acute stress classification from physiological signals with neural processes,” 2020, arXiv:2002.04176.
[25] Y. Lin and G. Wei, “Speech emotion recognition based on hmm and svm,” in 2005 International Conference on Machine Learning and Cybernetics, vol. 8, 2005, pp. 4898–4901.

Fig. 10: Communication latency and energy costs at different thresholds with overall accuracy (%) labeled.
[26] M. Hossain, M. Ghulam, M. F. Alhamid, B. Song, and K. A. Mutib, “Audio-visual emotion recognition using big data towards 5g,” Mobile Networks and Applications, vol. 21, pp. 753–763, 2016.

[27] P. Gomes, P. Margaritoff, and H. Silva, “pyhrv: Development and evaluation of an open-source python toolbox for heart rate variability (hrv),” in Proc. Int’l Conf. on Electrical, Electronic and Computing Engineering (IcETRAN), 2019, pp. 822–828.

[28] I. Azimi, T. Pahikkala, A. Rahmani, H. Niela-Vilén, A. Axelin, and P. Liljeberg, “Missing data resilient decision-making for healthcare iot through personalization: A case study on maternal health,” Future Generation Computer Systems, vol. 96, pp. 297–308, 2019.

[29] A. Donders, G. van der Heijden, T. Stijnen, and K. Moons, “Review: A gentle introduction to imputation of missing values,” Journal of Clinical Epidemiology, vol. 59, no. 10, pp. 1087–1091, 2006.

[30] S. van Buuren and K. Groothuis-Oudshoorn, “mice: Multivariate imputation by chained equations in r,” Journal of Statistical Software, Articles, vol. 45, no. 3, pp. 1–67, 2011.

[31] Nurzaman, T. Siswantining, S. Soemartojo, and D. Sarwinda, “Application of sequential regression multivariate imputation method on multivariate normal missing data,” in 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), 2019, pp. 1–6.

[32] H. Gamboa, “Multi-modal behavioral biometrics based on hci and electrophysiology,” Ph.D. dissertation, Instituto Superior Técnico, 2008.

[33] A. Greco, G. Valenza, A. Lanata, E. P. Scilingo, and L. Citi, “cvxeda: A convex optimization approach to electrodermal activity processing,” IEEE Transactions on Biomedical Engineering, vol. 63, no. 4, pp. 797–804, 2016.

[34] I. Christov, “Real time electrocardiogram qrs detection using combined adaptive threshold,” Biomed Eng Online, vol. 3, no. 1, 2004, doi: 10.1186/1475-925X-3-28.

[35] H. Gamboa, “Multi-modal behavioral biometrics based on hci and electrophysiology,” Ph.D. dissertation, Instituto Superior Técnico, 2008.

[36] P. Welch, “The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” IEEE Transactions on Audio and Electroacoustics, vol. 15, pp. 70–73, 1967.

[37] X. He, D. Cai, and P. Niyogi, “Laplacian score for feature selection,” in Advances in Neural Information Processing Systems 18, ser. NIPS’05. Cambridge, MA, USA: MIT Press, 2006, pp. 507–514.

[38] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, “Conditional likelihood maximisation: A unifying framework for information theoretic feature selection,” J. Mach. Learn. Res., vol. 13, pp. 27–66, 2012.

[39] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2018, arXiv:1711.05101.

[40] J. L. Schafer, “Multiple imputation: a primer,” Statistical Methods in Medical Research, vol. 8, no. 1, pp. 3–15, 1999.

[41] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, “A close examination of performance and power characteristics of 4g lte networks,” in Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys’12). Association for Computing Machinery, 2012, pp. 225–238.
# Appendix A

## Supplementary Table

### Table A.1: Public Stress/Emotion Datasets Used in Prior Work

| Reference | Dataset               | Average time recorded per subject (min) | No. of subjects | Modality          | Attribute         | No. of classes |
|-----------|-----------------------|----------------------------------------|-----------------|-------------------|-------------------|----------------|
| [14]      | MSR-IMPROV            | -                                      | 12              | Audio, Visual     | Emotion           | 4              |
| [11], [23], [21], [24] | WESAD             | 100                                    | 15              | Physiological     | Stress            | 3              |
| [14], [13] | IEMOCAP             | 72                                     | 10              | Audio, Visual     | Emotion           | 10             |
| [23], [20] | SWELL               | 75                                     | 25              | Lexical, Physiological, Visual | Stress | 3              |
| 22        | Mental Arousal Level Recognition Competition Dataset | 10                                      | 30              | Physiological     | Stress            | 5              |
| 23        | DREAMER              | 18                                     | 23              | Physiological     | Arousal/Valence   | 5              |
| 18        | Dreaddit             | -                                      | -               | Lexical           | Stress            | 2              |
| 23        | AMIGOS               | -                                      | 40              | Audio, Physiological, Visual | Arousal/Valence | 9              |
| 15        | MuSE                 | 45                                     | 28              | Audio, Lexical, Physiological, Visual | Stressed emotion | 3              |
| 26        | eNTERFACE05 database | 1.5                                    | 42              | Audio, Visual     | Emotion           | 6              |
| 26        | Berlin emotional speech database | 2.2                                    | 10              | Audio             | Emotion           | 7              |
| 26        | Kanade-Cohn emotional face database | -                                      | 182             | Visual            | Emotion           | 6              |
| 24        | Drivedb              | 50                                     | 24              | Physiological     | Stress            | 3              |