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

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY-NC-SA 4.0

SUBMISSION DATE / POSTED DATE

07-02-2021 / 01-11-2021

CITATION

Jiang, Shiyi; Firouzi, Farshad; Chakrabarty, Krishnendu; Elbogen, Eric (2021): A Resilient and Hierarchical IoT-based Solution for Stress Monitoring in Everyday Settings. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.13728199.v2>

DOI

[10.36227/techrxiv.13728199.v2](https://doi.org/10.36227/techrxiv.13728199.v2)

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

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

Abstract—The conventional mental healthcare regime follows a reactive, symptom-focused, and episodic approach in a non-continuous manner, wherein the individual discretely records their biomarker levels or vital signs for a short period prior to a subsequent doctor’s visit. Recognizing that each individual is unique and requires continuous stress monitoring and personally tailored treatment, we propose a holistic hybrid edge-cloud Wearable Internet of Things (WIoT)-based online stress monitoring solution to address the above needs. To eliminate the latency associated with cloud access, appropriate edge models—Spiking Neural Network (SNN), Conditionally Parameterized Convolutions (CondConv), and Support Vector Machine (SVM)—are trained, enabling low-energy real-time stress assessment near the subjects on the spot. This work leverages design-space exploration for the purpose of optimizing the performance and energy efficiency of machine learning inference at the edge. The cloud exploits a novel multimodal matching network model that outperforms six state-of-the-art stress recognition algorithms by 2-7% in terms of accuracy. An offloading decision process is formulated to strike the right balance between accuracy, latency, and energy. By addressing the interplay of edge-cloud, the proposed hierarchical solution leads to a reduction of 77.89% in response time and 78.56% in energy consumption with only a 7.6% drop in accuracy compared to the IoT-Cloud scheme, and it achieves a 5.8% increase in accuracy on average compared to the IoT-Edge scheme.

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 an 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) enable new ways to potentially track and address barriers to stress monitoring and treatment [5], [6], [7], [8], [9]. 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 releases hormones, which change 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) (or GSR—galvanic skin response), respiratory signal, and photoplethysmogram (PPG) can be measured as stress indicators [10], [11], [12], [13], [14]. In addition to physiological signals, daily activities, body movements or gestures, and environmental factors are other potential elements for stress evaluation [15]. Recent studies in the area of stress monitoring have led to machine learning (ML) models for stress evaluation and classification. Aqajari et al. [16] and Oskoei et al. [17] 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 [18], Busso et al. [19] and Jaiswal et al. [20] proposed to use images and audio information captured from recorded videos with pre-designed scenes for stress classification. Additionally, utilizing text messages posted on social media for stress identification was another area explored by Guntuku et al. [21] and others [22], [23]. In addition to using ML methods such as random forest (RF) [16], recent work has also focused on multi-layer perceptron (MLP) and CNNs [24], [25], [26], [27], [28], [29].

Despite this large body of recent work, today’s solutions suffer from several shortcomings and limitations. Current stress monitoring solutions are episodic; the status of a subject may not be precise due to the lack of information before and after the examination. Real-time IoT-based monitoring is therefore essential. In addition, the current reactive, symptom-based approach can be converted into a proactive solution, with IoT-assisted measurement and AI-powered assessments that provide just-in-time interventions. Additionally, an IoT-based solution provides personalization that captures subject-to-subject variations. However, existing works 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. Also, high latency and power-hungry data transmission to remote servers require a careful trade-off between computing at the edge and in the cloud; this trade-off impacts energy consumption,

¹Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA. ²Department of Psychiatry and Behavioral Sciences, Duke University School of Medicine, Durham, NC 27710 USA

Copyright (c) 20xx IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

battery life-time, end-to-end response time, as well as the accuracy/reliability of the ML model [30]. 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 key contributions of this paper are as follows:

- We propose a holistic edge-assisted IoT-based hierarchical solution for processing multimodal data that achieves a balance between accuracy, latency, and energy and significantly outperforms six state-of-the-art classification methods for stress analysis. The proposed offloading technique also ensures low-overhead data offloading and is especially effective for real-time stress monitoring.
- We exploit design-space exploration to optimize model performance and energy efficiency at the edge during data inference. We utilize recently proposed algorithms that have been shown to be effective in the literature. We employ a Spiking Neural Network (SNN), a neural network with Conditionally parameterized Convolutions (CondConv), and a Support Vector Machine (SVM) for comparison. The SVM model achieves the highest accuracy with the lowest computational complexity.
- We present a novel approach that uses a few-shot learning model for stress classification to alleviate the shortcoming of a limited sample size. The proposed model achieves a significant increase in the accuracy and the F1 score compared to state-of-the-art models for stress evaluation.
- We perform real-time stress monitoring simulation to compare our approach with solutions that utilize either only edge-based or cloud-based models. The proposed approach achieves an accuracy of 91.25% with a significant amount of reduction in response time and energy consumption compared to the IoT-Cloud solution. These results further highlight the merits of the proposed method compared to other non-hierarchical models.
- We present a resilient technique to impute the missing data that is likely to occur during data acquisition in connected IoT systems for seamless stress-level monitoring.

The rest of the paper is organized as follows. In Section II, we describe related prior work on stress/emotion classification and the Internet of Medical Things (IoMT). 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 details associated with feature extraction, feature selection, design-space exploration, 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

A. Stress/emotion Recognition

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 [31]. Motivated by the lack of natural interactions between participants and insufficient contextualization of previously developed emotional databases, Busso et al. [19] 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 [18] 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. [20] 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. [32] 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. [22] 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 [23] approached this problem by applying classifiers such as gated Recurrent Neural Network (RNN), CNN, and logistic regression using a dataset from Reddit. They achieved an F1 score of 0.79 using logistic regression. Guntuku et al. [21] 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. [14] 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 [14]. Lin et al. [26] 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. [27] 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% [27].

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. [16] proposed a CNN-based feature extraction on EDA signals and showed significant binary classification accuracy enhancement using K-Nearest Neighbors (KNN), SVM, RF and Naïve Bayes (NB) compared to the use of statistically extracted features. KNN was found to be the most effective model with over 90% accuracy [16]. Oskooei et al. [17] suggested an unsupervised method for binary stress classification using heart-rate variability (HRV) extracted through CAE. Sarkar and Etemad [28] 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 the emotion prediction task. They used four public datasets and achieved 95% accuracy on the WESAD dataset [28]. Stewart et al. [29] 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 [29]. They evaluated the performance based on the area under the curve (AUC), and showed improvement compared to other baseline classifiers: Lasso, SVM and KNN [29].

Kumar et al. [33] used 1-D CNN sub-sub networks for feature extraction and performed model-level fusion from the outputs of two 1-D CNN sub-networks using the WESAD dataset. A CNN was utilized after model-level fusion for 3-class classification. An accuracy of 87.7% and an F1 score of 0.83 were obtained by the model. Khowaja et al. [34] developed novel local maxima and minima features and designed a voting and similarity-based fusion method for binary stress classification using the Drivedb dataset. The proposed method resulted in 92.05% accuracy and an F1 score of 0.9041. Nath and Thapliyal [35] proposed an IoT-based framework for stress evaluation using cortisol level as the ground truth. Physiological data was collected from human subjects by wearable edge devices and was sent to the cloud for analysis using RF. By applying data fusion, an accuracy of 94% and an F1 score of 0.92 were achieved.

Researchers also utilize data with various modalities that can provide auxiliary information for stress prediction. Paredes et al. [15] 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. [24] used ECG in combination with location, weather, and daily activity data for 3-class stress detection (stressed, normal, relaxed) with an NB solution implemented in the cloud and reached over 78% accuracy.

B. Internet of Medical Things (IoMT)

The Internet of Things (IoT) is a network that involves a variety of elements such as sensors, computing devices, software, humans and physical objects, wherein the elements in the network are able to communicate and exchange data with each other [6], [7]. The Internet of Medical Things (IoMT), or Healthcare Internet of Things (HIoT), is a specific type of IoT focusing on medical applications such as seamless health monitoring and proactive disease prevention [36], [37], [38], [39]. By leveraging advances in Wearable IoT, machine learning, and edge/fog computing, IoMT creates a new paradigm with an emphasis on research problems such as task performance optimization, resource allocation, cost minimization, etc. [40], [41], [42].

Rhamani et al. provide a general overview of fog computing and demonstrate the efficiency of an intermediate level of architecture between the edge and the cloud via a medical case study [43]. A variety of healthcare applications have been developed under the IoMT framework, e.g., such as Human Activity Recognition (HAR) [44], [40], [45], seizure detection [46], ECG arrhythmia classification [47], and COVID-19-related symptom detection [48].

Due to the limitations associated with battery life, energy consumption, and computational capability of endpoint IoT devices, performing all the computation locally is not viable. A subset of tasks, therefore, needs to be offloaded, either to a nearby data center, or to the cloud that has abundant computational resources. Furthermore, when multiple endpoint devices are involved, algorithms that allocate resources in an efficient manner are essential. [30] presents a review of recently proposed offloading schemes. Offloading frameworks using reinforcement learning [49], [50], game theory [42], and other algorithms [41], [51], [52], [53] have been proposed and shown to be effective in reducing latency and energy cost.

Additional research topics in the context of IoMT include streaming data preprocessing [54], location prediction of endpoint IoT devices [55], and data privacy [56]. In addition, recent advances in neuromorphic computing highlight the potential of integrating edge devices within the IoMT framework [57].

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

Fig. 1: Proposed general framework for healthcare applications.

were not implemented in the context of IoT. Due to constrained computational resources, battery life, and memory of the endpoint IoT devices, previous models are not suitable for practical deployment. [32] and [24] 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 six selected state-of-the-art methods and alleviates the time and energy costs associated with indiscriminate data transmission.

III. DESCRIPTION OF PROPOSED SOLUTION

A number of IoT systems have been proposed for healthcare and medicine [58], [43], [40]. Today's IoMT frameworks are heterogeneous in nature, and currently there is no consensus with respect to any reference model/architecture [6], [8], [7]. We adapt the architecture introduced in our earlier work [39], [6], [59], [60] shown in Fig. 1, which follows an edge-fog-cloud pattern in which the fog layer extends from the edge to the cloud. The edge layer is where wearable devices, portable sensors, and smartphones are leveraged to monitor a subject's health status by collecting real-time multimodal data (e.g., physical activities, EDA, and ECG). Endpoint IoT devices can be either directly connected to the fog/cloud layer or connected via gateways (e.g., smartphones) as bridges in between. The fog layer is designed based on a multi-layer hierarchical cluster-based pattern, allowing each fog node to communicate vertically and horizontally. Fog nodes locate between the endpoint IoT devices and the cloud, which provide scalable distributed computing, storage, and network connectivity [61]. Finally, data storage, big data processing, and analytics take place in the cloud. The design of this general-purpose architecture ensures that any healthcare application can be mapped to it while satisfying the application requirements.

In this paper, we present a two-tier scheme for stress monitoring by treating the fog layer as part of the cloud. Given the small-sized dataset in our study, the fog alone has sufficient computational resources to perform advanced data analysis. To highlight the importance of the edge-fog/cloud interplay, an offloading mechanism is also presented. The proposed solution

is general and can be expanded by incorporating advanced offloading techniques (e.g., Markov Decision Process and deep reinforcement learning-based offloading schemes) [49], [50]. Although our proposed solution is appropriate for our small dataset, a full edge-fog-cloud architecture that considers task offloading between the fog and the cloud layer will be integrated in future work with larger datasets.

Fig. 2 presents our stress-level evaluation framework using a hierarchical IoT architecture with an edge layer and a cloud 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. In order to optimize the classification performance of the edge model, we construct a recurrent SNN and a CondConv neural network and compare them with SVM for design-space exploration to select the best lightweight edge model. These models are compared in terms of classification accuracy and model computational complexity. Details of the models are presented in Section IV.A. 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 the 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

TABLE I: SUMMARY OF RELATED RESEARCH ON STRESS/EMOTION CLASSIFICATION

Reference	Dataset	Methods	Features	Multimodality	Hierarchical IoT	Missing-value resiliency
[15]	Private	RF	Questionnaires, Demographics, GPS, Calendar, network usage, ACC, Social	Y	N	N
[19]	MSP-IMPROV	SVM with sequential minimal optimization	Audio, Face/Head/Hand movements	Y	N	N
[24]	Private	NB	ECG, Time, Day, Location, Weather, Activity	Y	N	N
[22]	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
[18]	IEMOCAP	MLP, CNN, LSTM	Speech, Face/Head/Hand movements, Text	Y	N	N
[25]	SWELL, WESAD	RF, Extra tree	HRV, EDA	Y	N	N
[26]	WESAD	Multiple CNN with RF	ECG, EDA, ACC, PPG, EMG, Temperature, Respiratory signal	Y	N	N
[27]	Mental Arousal Level Recognition Competition dataset	NSGA-II, Fine tree, Bagged tree, Cubic SVM, Ensemble discriminant subspace	EDA, PPG, Respiratory signal	Y	N	N
[23]	Dreaddit	SVM, Logistic regression, NB, Perceptron, DT, RNN, CNN	Text	N	N	N
[21]	FB, Twitter	Linear regression with regularization, SVM	Text	N	N	N
[16]	WESAD	CNN, KNN, SVM, RF, NB	EDA			
[17]	Private	CAE, LAE, KNN	HRV	N	N	N
[28]	AMIGOS, DREAMER, WESAD, SWELL	Self-supervised multi-task CNN	ECG	N	N	N
[20]	MuSE	Multiple unimodal DNNs	Text, Speech, Facial expression, Temperature, EDA, Heart rate	Y	N	N
[32]	eNTERFACE05 database, Berlin emotional speech database, Kanade-Cohn emotional face database	SVM	Speech, Face images	Y	N	N
[29]	Drivedb, WESAD	KNN, SVM, Logistic regression, Neural process	ECG, EDA	Y	N	N
[33]	WESAD	CNN	ECG, EDA, ACC, PPG, EMG, Temperature, Respiratory signal	Y	N	N
[35]	Private	RF	EDA, BVP, Interbeat interval, Temperature	Y	N	N
[34]	Drivedb	AdaBoost, SVM	GSR, HRV	Y	N	N
Proposed work	WESAD	SVM, CNN, Matching network	ECG, EDA, ACC, PPG, EMG, Temperature, Respiratory signal	Y	Y	Y

Fig. 2: Proposed framework for stress evaluation.

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 [16]. 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 [62].

C. Missing Value Recovery

Many methods exist for resolving missing values in a dataset [63]. 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. According to [63], 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 [64].

We therefore utilize multiple imputation, which is useful for reducing bias and increasing analysis precision, and is

resilient to outliers [65]. 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 [66]. 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. 3. We used a baseline obtained by filling missing data with values chosen uniformly from the dataset for comparison.

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, 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 [67] and [68]. 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 the peak amplitude of EMG were also evaluated. The

Fig. 3: Steps associated with the recovery of missing values.

time intervals between R-peaks (N-N intervals) were identified using the peak detection algorithm [69]. They were then utilized to compute a series of features such as relative and normalized powers and Low Frequency to High Frequency ratio (LF/HF) using four frequency bands: ultra-low frequency (ULF) (0-0.003 Hz), very low frequency (VLF) (0.003-0.04 Hz), LF (0.04-0.15 Hz), and HF (0.15-0.4 Hz) based on [70] and [71]. Fig. 4 shows some examples of signals before and after filtering. All the computed features are listed in Table II for each signal.

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 [72], Joint Mutual Information (JMI) [73], and Double Input Symmetrical Relevance (DISR) [74] 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 , μ_i^j and σ_i^j are the mean and standard deviation of feature i

(a) Raw and filtered ECG signal

(b) Raw and filtered EDA signal

(c) Raw and filtered respiratory signal

Fig. 4: Comparison between raw and filtered ECG, EDA, and respiratory signal.

in samples from class j , and μ_i is the mean of feature i among all samples.

$$\mathcal{F}(i) = \frac{\sum n_j (\mu_i^j - \mu_i)^2}{\sum n_j (\sigma_i^j)^2}. \quad (1)$$

The JMI measure is derived based on Mutual Information (MI) [75], which computes the amount of shared information between two variables X and Y as defined in Equation (2) where $\mathcal{H}$ is the entropy [75].

$$\begin{aligned}\mathcal{I}(X; Y) &= \mathcal{H}(X) - \mathcal{H}(X|Y) \\ &= \sum_{x \in X} \sum_{y \in Y} p(xy) \log \frac{p(xy)}{p(x)p(y)}.\end{aligned}\quad (2)$$

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.

$$\mathcal{I}_{JMI}(X_i) = \sum_{X_j \in S} \mathcal{I}(X_i, X_j; Y) \quad (3)$$

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 [74], DISR is suitable for dealing with datasets that have high feature-to-sample ratio [73].

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

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, the distance from data point x_n to the hyperplane is defined as $y(x_n)$, as indicated in Equation (5), where w represents the weight and b represents the bias. The objective of SVM is to maximize the margin, as described by Equation (6), where $t_n \in \{-1, +1\}$ represents the label of the test point. The ideal constraint on margin maximization for the hyperplane to perfectly separates all the data points is $t_n y(x_n) \geq 1$. However, most real-world datasets are non-separable. In order to solve this problem, SVM loosens the constraint by introducing a penalty term ϵ_n , where $\epsilon_n \geq 0$ as shown in Equation (7), adding tolerance on data misclassification.

$$y(x_n) = w^T x_n + b \quad (5)$$

$$\operatorname{argmax}_{w, b} \frac{1}{\|w\|} \min_n t_n y(x_n) \quad (6)$$

$$t_n y(x_n) \geq 1 - \epsilon_n, \forall x_n \quad (7)$$

4) *Spiking Neural Networks (SNNs)*: Neuromorphic computing mimics the learning patterns of neurons and synapses in the human brain [76]. It possesses an architecture with collocation of processor and memory, which enables faster processing and lower power consumption compared to the classical von Neumann architecture [76]. Spiking Neural Networks (SNNs) are novel models that utilize neuromorphic computing building blocks.

In contrast to neurons in artificial neural networks, neurons in SNNs output time-dependent binary values called spikes. The behavior of a spiking neuron mimics a biological neuron, where the output of neuron j at time t (Out_j^t) is threshold-based, as shown in Equation (8). When the potential accumulated in the neuron (V_j^t) is above the threshold V_{thres} , the neuron is activated to generate a spike. It restores the resting membrane potential V_{rest} after activation. However, when the potential is less than V_{thres} , the neuron remains inactivated and outputs a zero. There are many types of spiking neurons. One of the most common types of spiking neurons is Leaky Integrate-and-Fire (LIF) neurons. The dynamics of the potential accumulated in an LIF neuron is characterized by the differential equation presented in Equation (9), where V_j^t is the potential of neuron j at time t , V_{rest} is the resting membrane potential, τ and α are constants, w_{ij} is the synaptic weight between neuron i and neuron j , and x_i is the output from neuron i . A single spiking neuron has been shown to have higher computational power than a canonical neuron with sigmoid activation function [77].

$$Out_j^t = \begin{cases} 1, & \text{if } V_j^t > V_{thres} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$$\tau \cdot \frac{d}{dt} V_j^t = -(V_j^t - V_{rest}) + \alpha \sum_{i=1}^n w_{ij} \cdot x_i \quad (9)$$

Similar to canonical artificial neural network architectures such as fully connected neural networks, CNNs, and RNNs, architectures composed of spiking neurons have been developed and have been shown to achieve high performance in handling physiological signals such as ECG and EMG [78], [79], [80]. To handle the problem of non-differentiability inherent in Equation (9), algorithms for approximating the gradient descent mechanism were developed [81], [82].

Inspired by the architectures proposed in previous work [80], [79], we develop an SNN architecture with one recurrent cell unit and one fully connected layer utilizing LIF neurons. We encode input data based on Equation (8) and Equation (9), where x_i is the input.

5) *Conditionally Parameterized Convolutions (CondConv)*: CNNs are widely used for various computer vision tasks [83], [84]. The performance of a model can be optimized by increasing the capacity of the network, such as adding more convolutional layers. However, this approach increases the computational cost. The convolutional layer applies convolution using the same set of filters to all input data. [85] proposed conditionally parameterized convolutions (CondConv) using customized convolutional filters for each input sample to enhance model performance with low computational cost. The output from a CondConv layer is calculated based on Equation

(10), where x is the input data, $W_1, \dots, W_n$ are convolutional filters, $\alpha_1, \dots, \alpha_n$ are weights learned via gradient descent, and σ is a non-linear activation function [85].

$$\text{Output}(x) = \sigma((\alpha_1 \cdot W_1 + \dots + \alpha_n \cdot W_n) * x) \quad (10)$$

We utilize CondConv layers to construct an advanced neural network at the edge. We perform design-space exploration by varying the number of CondConv layers followed by one fully connected layer as the architecture. The details of the evaluated architectures are explained in Section V.A. The description of the conventional CNNs is provided in Section IV.B.

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 (CNNs)*: CNNs are 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 [86], 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	3×3	1	128 × 128 × 3	126 × 126 × 32
2	MaxPool	2×2	2	126 × 126 × 32	63 × 63 × 32
3	Conv2d	3×3	1	63 × 63 × 32	61 × 61 × 64
4	MaxPool	2×2	2	61 × 61 × 64	30 × 30 × 64
5	Conv2d	3×3	1	30 × 30 × 64	28 × 28 × 128
6	Conv2d	3×3	1	28 × 28 × 128	26 × 26 × 256
7	Conv2d	3×3	1	26 × 26 × 256	24 × 24 × 256
8	Linear	-	-	24 × 24 × 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.

The aim of supervised learning algorithms is to learn a hypothesis that predicts the optimal approximation of the

ground truth. The objective is shown in Equation (11), where $R(\cdot)$ is the expected risk, i.e., how well an algorithm performs, $\tilde{h}$ is the optimal hypothesis that minimizes the risk, h^* is the best approximation of $\tilde{h}$ in the hypothesis space $\mathcal{H}$, and $h_I \in \mathcal{H}$ is the best hypothesis can be obtained with I samples that minimizes the risk [87]. As shown in Equation (11), the error is composed of two parts, with the first portion being the approximation error, and the latter the estimation error. When I is small, $R(h_I)$ grows, increasing the estimation error. Metric-based learning models such as matching networks reduce the estimation error by constraining $\mathcal{H}$ using prior knowledge learned by their embedding functions [87].

$$E[R(h_I) - R(\tilde{h})] = E[R(h^*) - R(\tilde{h})] + E[R(h_I) - R(h^*)] \quad (11)$$

As shown in [88], metric-based learning models can strengthen relevant information, exclude redundant information, and prevent model overfitting. Koch et al. used twin neural networks that share the same weights to learn the best embedding function that reflects the similarity between within-class and between-class sample pairs in the embedded space [89]. Their model suggests the robustness of metric-based learning methods on few-shot image recognition tasks. Matching networks utilize attention mechanism and LSTM to learn the embedding function. LSTM is a specialized RNN cell that takes the form presented in Equation (12). Given an input sequence x , at each time step k , it generates a hidden state h_k as the output of the cell, and a cell state c_k , which contains the information memorized from previous time steps.

$$h_k, c_k = \text{LSTM}(x, h_{k-1}, c_{k-1}) \quad (12)$$

We use the matching networks approach described in [90] for one-shot learning. It first embeds all samples (labeled support set and unlabeled samples) via a convolutional encoder. [90] modified the original embedding method via full context embeddings $f(\hat{x}, S)$ as presented in Equation (13), where $\hat{x}$ is the test sample, S is the support set, 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. att-LSTM is an attention-based LSTM model, which is further explained in Equations (14)-(17). The attention mechanism att is shown in Equation (14), where $x_i \in S$, h_{k-1} is the hidden state at step $k-1 \in K$. It is applied to compute the attention score r_{k-1} as shown in Equation (15). The quantity r_{k-1} is then fed into a conventional LSTM model presented in Equation (16). The final output h_k at step k is computed as shown in Equation (17). [90] shows that this re-embedding is especially useful when the two samples from different classes are similar.

$$f(\hat{x}, S) = \text{att-LSTM}(f'(\hat{x}), g(S), K) \quad (13)$$

$$\text{att}(h_{k-1}, g(x_i)) = \text{Softmax}(h_{k-1}^T \cdot g(x_i)) \quad (14)$$

$$r_{k-1} = \sum_{i=1}^{|S|} \text{att}(h_{k-1}, g(x_i)) \cdot g(x_i) \quad (15)$$

$$\hat{h}_k, c_k = \text{LSTM}(f'(\hat{x}), [h_{k-1}, r_{k-1}], c_{k-1}) \quad (16)$$

$$h_k = \hat{h}_k + f'(\hat{x}) \quad (17)$$

Fig. 5: Description of the matching network architecture.

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. 5. Built upon the architecture depicted in [90], 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 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 (18), 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 $\max\{p_1(x_i), \dots, p_n(x_i)\}$ 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 (19). The samples left at the edge are assigned to the class where they have the maximum output probability.

$$p_j(x_i) = p(j|x_i), 1 \leq j \leq n \quad (18)$$

$$\begin{cases} \max\{p_1(x_i), \dots, p_n(x_i)\} < c, \frac{1}{n} \leq c \leq 1, & \text{cloud sample} \\ \max\{p_1(x_i), \dots, p_n(x_i)\} \geq c, \frac{1}{n} \leq c \leq 1, & \text{edge sample} \end{cases} \quad (19)$$

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.

Our work provides a data offloading scheme between a single endpoint edge device and the cloud, with an emphasis on maintaining high accuracy, for stress level evaluation with low latency and energy cost during data transmission. Offloading schemes that consider operations among multiple endpoint edge devices, edge device battery consumption, queuing models, and communication networks while not in the scope of this work, will be explored in future research.

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 [16], [28], [26], [29], [14]. 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 [91]. The amusement setting includes watching a set of funny video clips. A baseline was first measured and followed by a stress setting and an amusement setting with a meditation interval in between. The stress and amusement settings were interchanged for each participant. Another meditation period was measured at the end of the experiment. The devices recorded a total of 100 minutes for each participant.

In accordance with a considerable amount of work using the WESAD dataset, we selected the baseline and stress conditions for binary classification [16], [29], [14].

2) *Computation Platform*: We performed signal preprocessing, 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 [62]. Multiple imputation for recovering missing values was performed using Bayesian ridge regression implemented in the scikit-learn package. According to [92], we imputed $n = 5$ and $n = 9$ distinct datasets after introducing missing values at rates of 5% and 10%, respectively. Due to a limited number of samples, the synthetic minority oversampling technique was applied. Feature selection was performed based on [93].

In order to determine the best edge model, design-space exploration was performed on a recurrent SNN, a CondConv neural network, and an SVM. All the extracted features were used and shaped into a size of 10×10 as the input for the recurrent SNN and the CondConv neural network. The selected hyperparameters and values are listed in Table IV, where C is the regularization strength and γ is the parameter for RBF kernel. The range of the hyperparameters was determined based on [94] and [95]. The recurrent SNN and the CondConv neural network were trained using 70% of the samples with a batch size of 32 for 30 epochs. The number of training epochs was increased to 100 for the CondConv neural network with a larger architecture. The negative log-likelihood loss (L_{nll}) shown in Equation (20) was applied to both models, where $\hat{y}_n$ is the predicted value, y_n is the ground truth, and N is the batch size. Adam optimization [96], [97] with a learning rate of 10^{-3} was applied to both models. Hyperparameters for training these models were chosen based on random search. The recurrent SNN was implemented using the Norse library [98]. We trained a leave-one-out SVM using the Radial Basis Function (RBF) kernel with tuned hyperparameters chosen via grid search using the scikit-learn package.

$$L_{nll} = -\frac{1}{N} \sum_{n=1}^N \log(\hat{y}_n \cdot y_n) \quad (20)$$

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 [99]. As a typical choice for binary classification, binary cross entropy (L_{ce}), as shown in Equation (21), was selected as the loss function, where P is the true distribution of the data and Q is the estimated distribution.

$$L_{ce} = -P \log_2 Q - (1 - P) \log_2 (1 - Q) \quad (21)$$

Based on [90] and random search, the matching network was trained using 70% of the samples with a batch size of 128

TABLE IV: Design-space exploration on models with selected hyperparameter setting

Model	Hyperparameter	Test points
SVM	C	10 points linearly spaced in $[1, 10^4]$
	γ	20 points linearly spaced in $[10^{-9}, 10^{-8}]$
recurrent SNN	No. of hidden neurons	10, 28, 46, 64, 82, 100
	No. of time steps	10, 15, 20, 25, 30, 35
CondConv neural network	No. of experts	1, 2, 3
	No. of CondConv layers	1, 2, 3
	No. of channels	3, 8, 16, 32, 64
	Kernel size	3, 5

for 400 epochs. Cosine similarity (L_{cs}), as shown in Equation (22), was selected as the matching function to measure the distance between the prediction $\hat{y}$ and the ground truth y .

$$L_{cs} = \frac{\hat{y} \cdot y}{\|\hat{y}\|_2 \cdot \|y\|_2} \quad (22)$$

Adam optimization 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.

4) *Baseline Techniques/Architectures*: We select six state-of-the-art methods for stress-level classification using the WESAD dataset as the baselines for comparison. The baseline models include conventional CNNs [28], [33], a neural process model [29], a deep fusion network [26], a KNN [16], and an LDA [14]. The neural process model is a neural network that learns a distribution over functions and adapts to the given context. The deep fusion network is a neural network composed of multiple sub-networks, including CNNs and LSTMs.

For real-time simulation, we compared our work with models executed on only edge-based and cloud-based architectures. As stated in Section III, we aim at showing that our proposed hierarchical model achieves a better trade-off between the overall accuracy and the cost than non-hierarchical models utilizing only edge or the cloud. Comparisons with other existing IoMT architectures are beyond the scope of this work.

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

Fig. 6: Accuracy of SVM using features selected by different methods.

Fig. 7: Feature category contributions to the SVM performance.

positive, as described in Equation (23). The F1 score is a combined measurement of precision and recall values as shown in Equation (24). The goal is to ensure high performance of the classifier.

$$\text{Precision} = \frac{TP}{TP + FP}, \text{Recall} = \frac{TP}{TP + FN} \quad (23)$$

$$\text{F1 score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (24)$$

In addition, for the purpose of edge model comparison, we used the number of floating-point operations per inference sample (FLOPs) as the metric for model computational complexity. Based on [100], FLOPs for the SVM is $O(M \cdot N_s)$, where M is the dimensionality of input data and N_s is the number of support vectors utilized by the model. FLOPs for the recurrent SNN at layer l ($FLOPs_{SNN}^l$) is computed in Equation (25) [101], where $FLOPs_{ANN}^l$ is FLOPs computed using the same layer architecture with non-spiking neurons, and S_A^l is the average fraction of activated neurons at layer l in one time step. The FLOPs computation for the CondConv layer is based on Equation (10).

$$FLOPs_{SNN}^l = FLOPs_{ANN}^l \cdot S_A^l \quad (25)$$

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 [102], [103] 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 V in the order of their rankings.

Fig. 6 shows that SVM based on only ten features can achieve an accuracy of around 85%. Although the accuracy is slightly worse compared to previous work [14], [26], we must also consider the limited computational resource available at the edge layer. Our model requires fewer features and thus is more feasible for implementation at the edge. The percentage contributions of each signal are presented in Fig. 7. We see that EDA and the respiratory signal are major contributors to the classification accuracy for all three feature selection methods. This observation is consistent with [14] and [26]. According to Table V, we see some 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. This highlights the consistency of the importance of the top features between the different methods for stress-level classification.

Fig. 8: Computational complexity and classification accuracy comparison between edge models.

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 VI 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 VI 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.

TABLE V: SELECTED FEATURES USING DIFFERENT SELECTION METHODS

Methods	Selected features
Fisher score	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, Acc_x mean, BVP mean, ECG skewness, Acc_z std, respiratory signal skewness
DISR	mean respiratory rate, relative power, inhalation std, BVP sum, SCR std, ECG kurtosis, SCR mean, EDA AUC, normalized power, respiratory signal skewness

2) *Design-space exploration on the edge models*: To optimize the performance of the edge model, we constructed a recurrent SNN and a CondConv neural network and compared them with the SVM model. We evaluate a model in terms of classification accuracy and model computational complexity as shown in Fig. 8. Points with high FLOPs and low accuracy, which indicate severe model overfitting were truncated. In Fig. 8.a, we see that SVM can achieve a maximum accuracy of 88% with approximately 10,000 FLOPs. Utilizing fewer or

more support vectors degrades classification performance. A reduced number of support vectors leads to a narrower margin, resulting in model overfitting. An increased number of support vectors results in a wider margin, increasing the number of misclassifications.

We present Fig. 8.b for the recurrent SNN model. To identify the upper bound on the parameter S_A^l in Equation (25) for the FLOPs associated with the recurrent SNN, we trained the model with each combination of the hyperparameters listed in Table IV for a total of 10 times. Based on the experiments, the upper bound on S_A^l is determined to be 0.6 for recurrent SNN architectures explored in our study, which is consistent with [101]. We see that for the recurrent SNN, increased computational complexity leads to increased accuracy, but with large fluctuations on accuracy occurred at low model complexity. We observe that performance of the recurrent SNN with a small number of hidden neurons is affected by high variance of the input spikes induced during data encoding. The effect of input data variance on model performance is reduced when the model complexity increases.

We also observe fluctuations in the accuracy for the CondConv neural network in Fig. 8.c. This observation can be attributed to the fact that both the width and depth of the network are increasing during design-space exploration. An increase in the width or depth of a network increases model complexity, but the resulting predictions from the network show large variations for each individual sample [104]. Furthermore, as the number of model parameters increases, given the limited amount of training data, the model is not able to learn most of the parameters, resulting in low accuracy. When we compare the three models on the basis of computational complexity and accuracy, we find that SVM achieves the highest accuracy with the lowest computational complexity. Given a limited amount of training data, SVM obtains better performance compared to the other two data-hungry neural networks. Note that design-space exploration has to be conducted to re-evaluate the quality of a model when larger datasets are available.

3) *Stress Classification at the Edge*: Based on the results of design-space exploration shown in Fig. 8 and the discussion in Section V.C.2, we chose SVM as the edge model. In Table VI, 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.

Fig. 9: Accuracy and sample partitioning obtained using ECG in the overall system, and at the edge and in the cloud.

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 VI, 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.

TABLE VI: PERFORMANCE OF SVM AT EDGE BEFORE AND AFTER IMPUTATION

Missing rate (%)	Method	Before imputation		After imputation	
		Accuracy (%)	F1 score	Accuracy (%)	F1 score
0.22	Fisher	85.2041	0.8549	86.7347	0.8725
	JMI	82.3979	0.8253	81.6326	0.8172
	DISR	85.7143	0.8613	86.7347	0.8725
5	Fisher	71.9387	0.6646	76.2755	0.7223
	JMI	52.0408	0.1296	84.4388	0.8508
	DISR	80.6122	0.8041	85.4592	0.8613
10	Fisher	71.6836	0.6498	76.7857	0.7331
	JMI	58.9286	0.566	83.1632	0.8366
	DISR	80.8673	0.812	85.4592	0.8585

4) *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, 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 VII presents the evaluation results for the CNN and the matching network model. We see that the performance of the matching network using all three signals is superior to that of the CNN model. The CNN achieves comparable performance using ECG with an 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.

TABLE VII: PERFORMANCE OF CNN AND MATCHING NETWORK USING RESPIRATORY SIGNAL, EDA AND ECG

Models		Accuracy (%)	Precision	Recall	F1 score
Respiratory signal	CNN	94.6236	1	0.875	0.9333
	Matching network	96.4286	1	0.9091	0.9524
EDA	CNN	90.3226	0.9697	0.8	0.8767
	Matching network	96.4286	1	0.9091	0.9524
ECG	CNN	98.9247	1	0.975	0.9873
	Matching network	98.9247	0.9756	1	0.9876

We next compare the performance of our models in the cloud with six prior methods that utilize selected signals and labels from the WESAD dataset; the results are shown in Table VIII. 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. Note that [28] utilized ECG signals from four datasets to train the model, and therefore resulting in a higher accuracy compared to other prior work. Without comparing with [28], our models can achieve at least a 5% increase in the accuracy and a 7% increase in 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. Table VII shows that 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. As mentioned earlier in this section, the number of ECG image samples is much greater than the number of EDA or respiratory signal image samples. A conventional CNN that requires large amounts of data for training can obtain comparable performance to the matching network model using the ECG signal, but the performance is worse when only the EDA and respiratory signals are used. This experimental finding

is also supported by our theoretical explanation in Section IV.B.

TABLE VIII: PERFORMANCE COMPARISON WITH PRIOR WORK USING WESAD

Reference	Signals	No. of classes	Accuracy (%)	F1 score	AUC	Precision
[28]*	ECG	4	96.9	0.963	-	-
[26]	All	3	85	0.86	-	-
[33]	All	3	87.7	0.83	-	-
[29]	EDA, ECG	2	-	-	0.984	0.957
[16]	EDA	2	91.6	-	-	-
[14]	All Chest	2	93.12	0.9147	-	-
Proposed work	Respiratory Signal	2	96.4286	0.9524	-	-
	EDA	2	96.4286	0.9524	-	-
	ECG	2	98.9247	0.9876	-	-

In addition, we compare the proposed work with prior work using different stress datasets. As introduced in Section II, [34] and [35] achieved an accuracy of 92.05% and 94%, respectively, on binary stress classification. Our work outperforms the previous models with at least a 4% increase in accuracy, and a 6% increase in the F1 score.

5) *Trade-off between overall accuracy and cost:* The results in Table VI and Table VII show that inferencing 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. 9. 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, Long-Term Evolution (LTE), and Wi-Fi using TestMy.net [105] on a mobile phone; see Table IX. 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 [106]. Equation (26) describes the upload power model, where P_{upload} is the power level for uploading, α_u (mW/Mbps) and β (mW) are parameters with the values provided in [106], 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.

$$P_{Upload} = \alpha_u t_u + \beta \quad (26)$$

TABLE IX: UPLOAD BANDWIDTH AND DATA RATE OF DIFFERENT NETWORKS

	Bandwidth	Data rate
Wi-Fi	6.1 Mbps	759 KB/s
LTE	830 Kbps	104 KB/s
3G	55 Kbps	7 KB/s

Fig. 10: Data transfer energy at different thresholds for different networks.

Fig. 11: Communication latency at different thresholds for different networks.

Fig. 10 and Fig. 11 present the energy consumption and the communication latency during data transmission. As the

*This work utilized ECG signals from four datasets (including WESAD) to train the model.

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.

Fig. 12: Communication latency and energy costs at different thresholds with overall accuracy (%) labeled.

We illustrate the trade-off between the overall accuracy, the communication latency, and the energy consumption during data transmission from the edge to the cloud layer. As shown in Fig. 12, 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. 9(b). The overall accuracy shares a similar trend with the cost as indicated in Fig. 9(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.

6) *Real-time stress monitoring simulation*: We have shown in Section V.C.4 that the accuracy of the proposed cloud models is higher than that of existing state-of-the-art methods. We also assessed the trade-off between the overall accuracy of the proposed hierarchical model and the cost during data transmission between the edge and the cloud. To further highlight the merit of our proposed solution in terms of classification accuracy, response time of data inference, and energy consumption, we performed a real-time stress level monitoring simulation in a two-hour window. We next compare our proposed hierarchical solution with the use of only the edge-based or the cloud-based models. We randomly selected 80 samples with each sample being a 90-second window containing stress information from the test set described in Section V.A. Inference using the samples was conducted under

IoT-Edge (edge-based model using SVM), IoT-Cloud (cloud-based model using CNN), and our proposed hierarchical scheme with an offloading threshold of 0.68 selected as described in Section V.C.5 based on Fig. 12. We measured the response time of each sample in the two-hour window for each scheme as shown in Fig. 13. The response time for each sample inferred in the cloud includes data transmission latency using Wi-Fi computed from Equation (26). Fluctuations in the response time in IoT-Edge are caused by other task executions on the CPU. To minimize this effect, we measured the response time 10 times and recorded the lowest time for each sample.

In addition, we estimated the energy consumption based on FLOPs computed from the edge model. We excluded energy consumption in the cloud but considered energy consumed during data transmission for the IoT-Cloud scheme. Based on [107], energy consumed per double-precision arithmetic operation on an x86-64 processor is 387 pJ/op. In Table X, we present the average response time, the average accuracy, and the average energy consumption for all three architectures. According to Table X, the proposed hierarchical edge-cloud scheme obtains an accuracy of 91.25% with high efficiency in terms of latency and energy consumption. Our proposed solution leads to a reduction of 77.89% in response time and 78.56% in energy consumption with only a 7.6% drop in accuracy compared to the IoT-Cloud scheme and it achieves a 5.8% increase in accuracy on average compared to the IoT-Edge scheme.

TABLE X: Results obtained using real-time simulation under different IoT architectures.

	Accuracy (%)	Average response time per sample (ms)	Average energy consumption per sample (μ J)
IoT-Edge	86.25	0.21	0.43
IoT-Cloud	98.75	152.46	176.08
Proposed work	91.25	33.65	37.75

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. Design-space exploration was conducted to select the most effective model at the edge. An accuracy of 86.7347% was achieved by the selected model at the edge using only 10 physiological features. An accuracy of 98.9247% was obtained by the cloud models, which is significantly better than the accuracy of the six selected state-of-the-art methods. 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. We developed an offloading scheme to address the interplay between the edge and the cloud. The trade-off between communication latency, data transfer energy, and overall accuracy was evaluated when making offloading decisions. We also conducted real-time

Fig. 13: Response time for different schemes based on two-hour real-time simulation.

stress monitoring simulation to demonstrate the benefits of the proposed solution. The next steps in this research include 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. Yariyebeygi, 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] L. Greco, G. Percannella, P. Ritrovato, F. Tortorella, and M. Vento, "Trends in IoT based solutions for health care: Moving AI to the edge," *Pattern Recognition Letters*, vol. 135, pp. 346–353, 2020.
- [6] B. Farahani, F. Firouzi, V. I. C. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," *Future Gener. Comput. Syst.*, vol. 78, pp. 659–676, 2018.
- [7] P. Pace, G. Aloï, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta, "An Edge-Based Architecture to Support Efficient Applications for Healthcare Industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 481–489, 2019.
- [8] I. Azimi, J. Takalo-Mattila, A. Anzanpour, A. Rahmani, J. Soininen, and P. Liljeberg, "Empowering Healthcare IoT Systems with Hierarchical Edge-Based Deep Learning," in *2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pp. 63–68, 2018.
- [9] M. Mozafari, F. Firouzi, and B. Farahani, "Towards IoT-enabled Multimodal Mental Stress Monitoring," in *2020 International Conference on Omni-layer Intelligent Systems (COINS)*. IEEE, 2020, pp. 1–8.
- [10] K. Plarre, A. Rajj, 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.
- [11] 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.
- [12] M. 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.
- [13] 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.
- [14] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. Van Laerhoven, "Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection," in *2018 20th ACM International Conference on Multimodal Interaction (ICMI)*, 2018, pp. 400–408.
- [15] P. Paredes, R. Gilad-Bachrach, M. Czerwinski, A. Roseway, K. Rowan, and J. Hernandez, "PopTherapy: Coping with Stress through PopCulture," in *2014 8th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, 2014, pp. 109–117.
- [16] S. A. H. Aqajari, E. K. Naeini, M. A. Mehrabadi, S. Labbaf, 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.
- [17] A. Oskoei, S. M. Chau, J. R. M. Weiss, A. Sridhar, M. R. Martínez, and B. Michel, "DeStress: Deep Learning for Unsupervised Identification of Mental Stress in Firefighters from Heart-rate Variability (HRV) Data," 2019, arXiv:1911.13213.
- [18] S. Tripathi and H. Beigi, "Multi-Modal Emotion recognition on IEMOCAP Dataset using Deep Learning," 2018, arXiv:1804.05788.
- [19] C. Busso, S. Parthasarathy, A. Burmanian, M. AbdelWahab, N. Sadoughi, and E. M. Provost, "MSP-IMPROV: An Acted Corpus of Dyadic Interactions to Study Emotion Perception," *IEEE Transactions on Affective Computing*, vol. 8, no. 1, pp. 67–80, 2017.
- [20] 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.
- [21] S. C. Guntuku, A. Buffone, K. Jaidka, J. C. Eichstaedt, and L. H. Ungar, "Understanding and measuring psychological stress using social media," in *2019 The Thirteenth International AAAI Conference on Web and Social Media (ICWSM)*, 2019, pp. 214–225.
- [22] 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.
- [23] E. Turcan and K. McKeown, "Dreaddit: A Reddit Dataset for Stress Analysis in Social Media," 2019, arXiv:1911.00133.
- [24] 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.
- [25] K. Nkurikiyeyezu, A. Yokokubo, and G. Lopez, "The effect of person-specific biometrics in improving generic stress predictive models," 2019, arXiv:1910.01770.
- [26] J. Lin, S. Pan, C. S. Lee, and S. Oviatt, "An explainable deep fusion network for affect recognition using physiological signals," in *28th ACM International Conference on Information and Knowledge Management (CIKM '19)*, 2019, pp. 2069–2072.

- [27] M. Mozafari, E. Khazaei, S. Bagheri, N. Hasanzadeh, 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.
- [28] 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.
- [29] C. Stewart, A. Folarin, and R. Dobson, "Personalized acute stress classification from physiological signals with neural processes," 2020, arXiv:2002.04176.
- [30] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Gener. Comput. Syst.*, vol. 87, pp. 278–289, 2018.
- [31] 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.
- [32] 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.
- [33] A. Kumar, K. Sharma, and A. Sharma, "Hierarchical deep neural network for mental stress state detection using IoT based biomarkers," *Pattern Recognit. Lett.*, vol. 145, pp. 81–87, 2021.
- [34] S. A. Khowaja, A. G. Prabono, F. Setiawan, B. N. Yahya, and S.-L. Lee, "Toward soft real-time stress detection using wrist-worn devices for human workspaces," *Soft Comput.*, vol. 25, pp. 2793–2820, 2021.
- [35] R. K. Nath and H. Thapliyal, "Smart wristband-based stress detection framework for older adults with cortisol as stress biomarker," *IEEE Transactions on Consumer Electronics*, vol. 67, pp. 30–39, 2021.
- [36] M. Usak, M. Kubiak, M. Shabbir, O. Dudnik, K. Jermisittiparsert, and L. Rajabion, "Health care service delivery based on the Internet of things: A systematic and comprehensive study," *Int. J. Commun. Syst.*, vol. 33, 2020.
- [37] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. Vasilakos, and S. Kim, "The Future of Healthcare Internet of Things: A Survey of Emerging Technologies," *IEEE Communications Surveys & Tutorials*, vol. 22, pp. 1121–1167, 2020.
- [38] F. Alshehri and G. Muhammad, "A Comprehensive Survey of the Internet of Things (IoT) and AI-Based Smart Healthcare," *IEEE Access*, vol. 9, pp. 3660–3678, 2021.
- [39] F. Firouzi, B. Farahani, M. Ibrahim, and K. Chakrabarty, "Keynote Paper: From EDA to IoT eHealth: Promises, Challenges, and Solutions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, pp. 2965–2978, 2018.
- [40] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I. D. Munari, "IoT Wearable Sensor and Deep Learning: An Integrated Approach for Personalized Human Activity Recognition in a Smart Home Environment," *IEEE Internet of Things Journal*, vol. 6, pp. 8553–8562, 2019.
- [41] Y. Qiu, H. Zhang, and K. Long, "Computation Offloading and Wireless Resource Management for Healthcare Monitoring in Fog-Computing based Internet of Medical Things," *IEEE Internet of Things Journal*, 2021, early Access.
- [42] Z. Ning, P. Dong, X. Wang, X. Hu, L. Guo, B. Hu, Y. Guo, T. Qiu, and R. Y. K. Kwok, "Mobile Edge Computing Enabled 5G Health Monitoring for Internet of Medical Things: A Decentralized Game Theoretic Approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, pp. 463–478, 2021.
- [43] A. Rahmani *et al.*, "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, 2018.
- [44] X. Zhou, W. Liang, K. Wang, H. Wang, L. Yang, and Q. Jin, "Deep-Learning-Enhanced Human Activity Recognition for Internet of Healthcare Things," *IEEE Internet of Things Journal*, vol. 7, pp. 6429–6438, 2020.
- [45] F. Samie, L. Bauer, and J. Henkel, "Hierarchical Classification for Constrained IoT Devices: A Case Study on Human Activity Recognition," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8287–8295, 2020.
- [46] F. Samie, S. Paul, L. Bauer, and J. Henkel, "Highly Efficient and Accurate Seizure Prediction on Constrained IoT Devices," in *2018 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2018, pp. 955–960.
- [47] B. Farahani, M. Barzegari, F. S. Aliee, and K. A. Shaik, "Towards collaborative intelligent IoT eHealth: From device to fog, and cloud," *Microprocess. Microsystems*, vol. 72, 2020.
- [48] M. A. Rahman and M. S. Hossain, "An Internet of Medical Things-Enabled Edge Computing Framework for Tackling COVID-19," *IEEE Internet of Things Journal*, 2021, early Access.
- [49] M. Min, X. Wan, L. Xiao, Y. Chen, M. Xia, D. Wu, and H. Dai, "Learning-Based Privacy-Aware Offloading for Healthcare IoT With Energy Harvesting," *IEEE Internet of Things Journal*, vol. 6, pp. 4307–4316, 2019.
- [50] X. Liu, J. Yu, J. Wang, and Y. Gao, "Resource Allocation With Edge Computing in IoT Networks via Machine Learning," *IEEE Internet of Things Journal*, vol. 7, pp. 3415–3426, 2020.
- [51] X. Q. Pham and E. N. Huh, "Towards task scheduling in a cloud-fog computing system," *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–4, 2016.
- [52] Z. Zhou, S. Yu, W. Chen, and X. Chen, "CE-IoT: Cost-Effective Cloud-Edge Resource Provisioning for Heterogeneous IoT Applications," *IEEE Internet of Things Journal*, vol. 7, pp. 8600–8614, 2020.
- [53] Z. Zhou, Z. Wang, H. Yu, H. Liao, S. Mumtaz, L. Oliveira, and V. Frascolla, "Learning-Based URLLC-Aware Task Offloading for Internet of Health Things," *IEEE Journal on Selected Areas in Communications*, vol. 39, pp. 396–410, 2021.
- [54] S. Din, A. Paul, N. Guizani, S. H. Ahmed, M. Khan, and M. Rathore, "Features Selection Model for Internet of E-Health Things Using Big Data," *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–7, 2017.
- [55] S. Ghosh, A. Mukherjee, S. K. Ghosh, and R. Buyya, "Mobi-IoST: Mobility-Aware Cloud-Fog-Edge-IoT Collaborative Framework for Time-Critical Applications," *IEEE Transactions on Network Science and Engineering*, vol. 7, pp. 2271–2285, 2020.
- [56] M. S. Hossain and M. Ghulam, "Emotion recognition using secure edge and cloud computing," *Inf. Sci.*, vol. 504, pp. 589–601, 2019.
- [57] O. Krestinskaya, A. P. James, and L. Chua, "Neuromemristive Circuits for Edge Computing: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, pp. 4–23, 2020.
- [58] A. Emam, A. Abdellatif, A. M. Mohamed, and K. A. Harras, "Edge-Health: An Energy-Efficient Edge-based Remote mHealth Monitoring System," *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–7, 2019.
- [59] F. Firouzi, B. Farahani, and A. Marinšek, "The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT)," *Information Systems*, p. 101840, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306437921000776>
- [60] F. Firouzi and B. Farahani, "Architecting IoT Cloud," in *Intelligent Internet of Things*, F. Firouzi, K. Chakrabarty, and S. Nassif, Eds. Springer, 2020, pp. 173–241.
- [61] National Institute of Standards and Technology. Accessed: 2021-08. [Online]. Available: <https://www.nist.gov>
- [62] 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.
- [63] I. Azimi, T. Pahikkala, A. Rahmani, H. Niela-Vilén, A. Axelín, 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.
- [64] 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.
- [65] S. van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate Imputation by Chained Equations in R," *Journal of Statistical Software*, vol. 45, no. 3, pp. 1–67, 2011.
- [66] 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.
- [67] H. Gamboa, "Multi-Modal Behavioral Biometrics based on HCI and Electrophysiology," Ph.D. dissertation, Instituto Superior Técnico, 2008.
- [68] 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.
- [69] 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.
- [70] M. Malik and C. Terrace, "Heart rate variability. standards of measurement, physiological interpretation, and clinical use," *European Heart Journal*, vol. 17, pp. 354–381, 1996.

- [71] 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.
- [72] 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.
- [73] G. Brown, A. Pock, 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.
- [74] P. E. Meyer and G. Bontempi, "On the use of variable complementarity for feature selection in cancer classification," in *Applications of Evolutionary Computing*, ser. EuroGP'06, vol. 3907. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 91–102.
- [75] L. Paninski, "Estimation of entropy and mutual information," *Neural Computation*, vol. 15, no. 6, pp. 1191–1253, 2003.
- [76] C. D. Schuman *et al.*, "A survey of neuromorphic computing and neural networks in hardware," 2017, arXiv:1705.06963.
- [77] W. Maass, "Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons," in *Advances in Neural Information Processing Systems*, vol. 9, 1997, pp. 211–217.
- [78] A. Amirshahi and M. Hashemi, "ECG Classification Algorithm Based on STDP and R-STDP Neural Networks for Real-Time Monitoring on Ultra Low-Power Personal Wearable Devices," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1483–1493, 2019.
- [79] Y. Ma, B. Chen, P. Ren, N. Zheng, G. Indiveri, and E. Donati, "EMG-Based Gestures Classification Using a Mixed-Signal Neuromorphic Processing System," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, pp. 578–587, 2020.
- [80] F. Corradi *et al.*, "ECG-based Heartbeat Classification in Neuromorphic Hardware," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.
- [81] E. Neftci, H. Mostafa, and F. Zenke, "Surrogate Gradient Learning in Spiking Neural Networks," 2019, arXiv:1901.09948.
- [82] Y. Jin, W. Zhang, and P. Li, "Hybrid Macro/Micro Level Backpropagation for Training Deep Spiking Neural Networks," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*. Curran Associates Inc., 2018, pp. 7005–7015.
- [83] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [84] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017, arXiv:1704.04861.
- [85] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "CondConv: Conditionally Parameterized Convolutions for Efficient Inference," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.
- [86] T. J. Jun, H. M. Nguyen, D. Kang, D. Kim, D. Kim, and Y. Kim, "ECG arrhythmia classification using a 2-D convolutional neural network," 2018, arXiv:1804.06812.
- [87] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a Few Examples: A Survey on Few-Shot Learning," *ACM Comput. Surv.*, vol. 53, no. 3, 2020, article no.63, 34 pages.
- [88] M. Fink, "Object classification from a single example utilizing class relevance metrics," in *Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS'04)*. MIT Press, 2004, pp. 449–456.
- [89] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-Shot Image Recognition," in *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, vol. 37, 2015.
- [90] O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," 2016, arXiv:1606.04080.
- [91] K. P. C. Kirschbaum and D. Hellhammer, "The 'Trier Social Stress Test'—a tool for investigating psychobiological stress responses in a laboratory setting," *Neuropsychobiology*, vol. 28, no. 1–2, pp. 76–81, 1993.
- [92] J. Graham, A. Olchowski, and T. Gilreath, "How Many Imputations are Really Needed? Some Practical Clarifications of Multiple Imputation Theory," *Prevention Science*, vol. 8, pp. 206–213, 2007.
- [93] J. Li *et al.*, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 94:1–94:45, 2018.
- [94] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification," National Taiwan University, Tech. Rep., 2003.
- [95] T. Elsken, J. H. Metzen, and F. Hutter, "Neural Architecture Search: A Survey," 2019, arXiv:1808.05377.
- [96] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, arXiv:1412.6980.
- [97] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019, arXiv:1711.05101.
- [98] C. Pehle and J. E. Pedersen. (2021, January) Norse-A deep learning library for spiking neural networks. Zenodo. Documentation: <https://norse.ai/docs/>. [Online]. Available: <https://doi.org/10.5281/zenodo.4422025>
- [99] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," 2017, arXiv:1708.04896.
- [100] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.
- [101] P. Panda, S. A. Aketi, and K. Roy, "Toward Scalable, Efficient, and Accurate Deep Spiking Neural Networks With Backward Residual Connections, Stochastic Softmax, and Hybridization," *Frontiers in Neuroscience*, vol. 14, 2020.
- [102] J. L. Schafer, "Multiple imputation: a primer," *Statistical Methods in Medical Research*, vol. 8, no. 1, pp. 3–15, 1999.
- [103] Y. Dong and C. Peng, "Principled missing data methods for researchers," *Springer Plus*, vol. 2, 2013, doi: 10.1186/2193-1801-2-222.
- [104] T. Nguyen, M. Raghu, and S. Kornblith, "Do Wide and Deep Networks Learn the Same Things? Uncovering How Neural Network Representations Vary with Width and Depth," 2021, arXiv:2010.15327.
- [105] TestMy.net Internet Speed Test. Accessed: 2020-12. [Online]. Available: <https://testmy.net>
- [106] 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.
- [107] D. Molka, D. Hackenberg, R. Schne, and M. S. Mller, "Characterizing the energy consumption of data transfers and arithmetic operations on x8664 processors," in *International Conference on Green Computing*, 2010, pp. 123–133.

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
[19]	MSR-IMPROV	-	12	Audio, Visual	Emotion	4
[16], [28], [26], [29], [14], [25]	WESAD	100	15	Physiological	Stress	3
[19], [18]	IEMOCAP	72	10	Audio, Visual	Emotion	10
[28], [25]	SWELL	75	25	Lexical, Physio- logical, Visual	Stress	3
[27]	Mental Arousal Level Recognition Competition Dataset	10	30	Physiological	Stress	5
[28]	DREAMER	18	23	Physiological	Arousal/Valence	5
[23]	Dreaddit	-	-	Lexical	Stress	2
[28]	AMIGOS	-	40	Audio, Physiolog- ical, Visual	Arousal/Valence	9
[20]	MuSE	45	28	Audio, Lexical, Physiological, Visual	Stressed emotion	3
[32]	eNTERFACE05 database	1.5	42	Audio, Visual	Emotion	6
[32]	Berlin emotional speech database	2.2	10	Audio	Emotion	7
[32]	Kanade-Cohn emotional face database	-	182	Visual	Emotion	6
[29]	Drivedb	50	24	Physiological	Stress	3

Shiya Jiang received a B.A. degree (magna cum laude) from Boston University, Boston, MA, in 2018. She received the M.Sc. degree in Electrical and Computer Engineering with Data Analytics & Machine Learning concentration from Duke University, Durham, NC, in 2020, where she is currently pursuing the Ph.D. degree. Her current research focuses on techniques to preserve privacy, enhance and personalize healthcare data analysis in the IoT system.

Eric Elbogen received a bachelors degree from Cornell University and a masters degree from Harvard University before earning a Ph.D. in Clinical Psychology at the University of Nebraska-Lincoln. He completed a Clinical Psychology Internship at Harvard Medical School/Massachusetts Mental Health Center and a Health Services Research Post-doctoral Fellowship at the Duke University School of Medicine. His research interests include use of mobile technology to reduce stress and chronic pain in patients with posttraumatic stress disorder and

traumatic brain injury.

Dr. Farshad Firouzi is an Adj. Assistant Professor at the Electrical and Computer Engineering Department of Duke University. Dr. Firouzi is a top-producing expert and technical leader with 10+ years' experience offering strong performance in all aspects of AI/ML, Smart Data, Computer Architecture, VLSI, and IoT including R&D, consulting services, strategic planning, and technology solutions, across vertical industries, e.g., Semiconductor, Automotive, Finance, Manufacturing, Logistics, and eHealth. Dr. Firouzi authored 60+ Conference/Journal papers and served

as Guest/Associate Editor of several well-known Journals (e.g., IEEE IoT-J, IEEE TVLSI, IEEE TCAD, Elsevier FGCS, Elsevier MICPRO, Elsevier JNCA, and Elsevier Information Systems) as well as the chair of 10+ international conferences on AI/IoT/eHealth.

Krishnendu Chakrabarty received the B. Tech. degree from the Indian Institute of Technology, Kharagpur, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively. He is now the John Cocke Distinguished Professor and Department Chair of Electrical and Computer Engineering, and Professor of Computer Science, at Duke University.

Prof. Chakrabarty is a recipient of the National Science Foundation CAREER award, the Office of Naval Research Young Investigator award, the

Humboldt Research Award from the Alexander von Humboldt Foundation, Germany, the IEEE Transactions on CAD Donald O. Pederson Best Paper Award (2015), the IEEE Transactions on VLSI Systems Prize Paper Award (2021), the ACM Transactions on Design Automation of Electronic Systems Best Paper Award (2017), multiple IBM Faculty Awards and HP Labs Open Innovation Research Awards, and over a dozen best paper awards at major conferences. He is also a recipient of the IEEE Computer Society Technical Achievement Award (2015), the IEEE Circuits and Systems Society Charles A. Desoer Technical Achievement Award (2017), the IEEE Circuits and Systems Society Vitold Belevitch Award (2021), the IEEE-HKN Asad M. Madni Outstanding Technical Achievement and Excellence Award (2021), the Semiconductor Research Corporation Technical Excellence Award (2018), and the IEEE Test Technology Technical Council Bob Madge Innovation Award (2018). He is a Research Ambassador of the University of Bremen (Germany) and he was a Hans Fischer Senior Fellow at the Institute for Advanced Study, Technical University of Munich, Germany during 2016-2019. He is a 2018 recipient of the Japan Society for the Promotion of Science (JSPS) Invitational Fellowship in the "Short Term S: Nobel Prize Level" category.

Prof. Chakrabarty's current research projects include: design-for-testability of integrated circuits and systems (especially 3D integration and system-on-chip); testing of AI accelerators; microfluidic biochips; hardware security; AI for failure prediction and healthcare; neuromorphic computing systems. He is a Fellow of ACM, IEEE, and AAAS, and a Golden Core Member of the IEEE Computer Society. He was a Distinguished Visitor of the IEEE Computer Society (2005-2007, 2010-2012), a Distinguished Lecturer of the IEEE Circuits and Systems Society (2006-2007, 2012-2013), and an ACM Distinguished Speaker (2008-2016). Prof. Chakrabarty served as the Editor-in-Chief of *IEEE Design & Test of Computers* during 2010-2012, *ACM Journal on Emerging Technologies in Computing Systems* during 2010-2015, and *IEEE Transactions on VLSI Systems* during 2015-2018.