Stress Detection Using Machine Learning on Multimodal Dataset for Wearable Stress and Affect Detection (WESAD)

1st Kamal Andani
Faculty of Engineering And Applied Science
Queen's University
Kingston, Ontario, Canada
19kaa3@queensu.ca
Student ID: 20188032

2nd Anubhav Bhatti
Faculty of Engineering And Applied Science
Queen's University
Kingston, Ontario, Canada
anubhav.bhatti@queensu.ca
Student ID: 20169150

***Abstract*—Affective computing's goal is to make the computer understand the emotional state of humans and then provide improvised responses based on the emotional states of the humans, and hence, improving the human-machine interaction. In recent times, detection of stressful states of humans by using machine learning algorithms has attracted a lot of attention from the research community. Using different techniques to detect stress has become important since having a stress for a longer period of time can have implications on the well-being of the persons. For this project, our motive is to apply various machine learning and deep learning algorithms on dataset WESAD [1], a multimodal dataset for wearable stress and affect detection, to detect stress and non-stress conditions in humans with better accuracy i.e., a binary classification problem. The reason for choosing the WESAD dataset is that it has physiological and motion data, recorded from chest- and wrist-worn device. For this dataset, 17 subjects (2 subjects' sensor malfunctioned) wore sensors RespiBAN professional, and Empatica E4 on chest and wrist, respectively. The sensor modalities included are blood volume pulse (BVP), electrocardiogram (ECG), electrodermal activity (EDA), electromyogram (EMG), respiration (RESP), body temperature (TEMP), and three-axis acceleration (ACC) [2]. For predicting stress condition i.e., stress, and non-stress, we have used various machine learning and deep learning algorithms such as Neural Networks, Random Forest, SVMs, Logistic Regressions. Out of these algorithms, Neural Networks and Logistic Regression Classifiers seemed to perform better then other algorithms on predicting the stress and non-stress class with an average accuracy of 90% and 91%, respectively. We also tried to compare the affect of adding/dropping different features, while training the algorithms, on the overall accuracy of the classifier.**

***Index Terms*—Affective Computing, Machine Learning, Deep Learning, Emotion Recognition, Stress Detection**

I. Introduction

Affective computing is an evolving field with a focus of improving human to machine interaction by designing and developing machines that can detect the emotional states of humans and based on their emotions can change their own behaviour to improve the overall human-machine interaction. Among the various affective states of human beings, stress is the most important state.

Stress has become an unavoidable part of our lives and it can be triggered by various reasons or situations. As per the information got from American Institute of Stress, approx. $300 billion is spent each year on the treatment of stress related disorders [3]. Stress is also basic cause of 60% of all human illness and disease [3]. Long-term stress can have various physical and health related issue in human beings. Long-term stress can lead to sleeplessness, sadness, anger, or irritability [4]. Due to this ever-increasing problem of Stress in human being, it has become very important to consistently detect and monitor stress situations in human beings. The detection of stress at early stage can help to prevent some serious health related problems with the help of proper stress prevention techniques. Stress consists of two parts: Stressor and Reaction. Stressor is the activity or event that is responsible for triggering the stress. As a result of stress, many physiological parameters of human body deviates from their normal levels, and that is called reaction. Therefore, one of the most reliable methods to detect stress in human body is by using physiological parameter values.

Various studies have shown that electrodermal activity (EDA) in combination of heart rate can be used to detect stress levels in human beings [5]. Electrodermal Activity (EDA) signals measure the changes in the electrical characteristics of the human skin by measuring the changes in resistance of the skin to a small electrical current. These changes in the resistance of the skin can be produced by various emotional and/or physical stimuli. The EDA is controlled by sympathetic nervous system (SNS) that is why it is the most ideal signal for measurement of stress. This analysis also shows that the average accuracy of the classifier greatly improves if we include all of the features derived from the EDA signal. For the current analysis, only phasic, tonic, and smna components were consider for classification.

II. Problem Statement and Literature Review:

Since, changes in the physiological parameters are directly related to stress and emotional stimuli, these signals have been used in recent times to train machine learning models to predict the emotional state of the human beings. Many recent works done in the field of affective computing study the use of deep learning to automatically determine the emotional state of the subjects from speech and visual data. For speech based emotional recognition system, Seyedmahdad et al. [6] proposed a strategy to pool features using local attention to focus on the specific regions of speech signal that are emotionally salient.

Also, Panagiotis et al. [7] proposed another emotion recognition system that uses speech and visual modalities to detect emotions in the subject. For extracting features from the speech, they used Convolutional Neural Networks, while Deep Residual network with 50 layers were used to extract features from visual modalities. Lu et al. [8] proposed a StressSense system that determines stress in unconstrained acoustic environments. Their system uses the user's smartphone to recognize stress in human voice in real-time.

These methods have practical limitations as they are computationally expensive, hence cannot be implemented on embedded devices such as wearable devices. Also, continuous recording and processing of audio/visual data is required could be quite intrusive in terms of privacy as nobody would like any system to record their personal conversations. Due to these limitations, there is a need for less computationally expensive method that can be implemented on wearable devices such as activity tracker, smart watch and using modalities that do not intrude the privacy of the use in any manner.

Smartphones and other Wearable devices for tracking user activities and fitness are already very popular in consumer market and widely available. So, the current focus should be to use these devices to infer stress-levels and emotional states in human beings. Many studies use the physiological data captured from these devices to train stress detection systems. Plarre et al. [9] proposed two models for continuous prediction of stress from physiological signals captured by wearable sensors strapped on chest. The wearable sensors were used to monitor cardiovascular, respiratory, and thermoregulatory systems that were further used to determine stress and other psychologically and physically demanding conditions.

Hovsepian et al. [10] proposed a stress model that used ECG, and accelerometer to detect stress in a subject. Liu et al. [11] proposed a real time movie induced emotion recognition system, wherein the system can predict the emotion of a subject in real time using EEG signals. In their system, they measured the EEG signals with 14-electrodes placed on the subject's scalp. Akmandor et al. [12] proposed an automatic stress detection and alleviation system (SoDA) that used the electrocardiogram (ECG), galvanic skin response (GSR), respiration rate, blood pressure, and blood oximeter from the wearable medical sensors (WMSs) to monitor and mitigate stress in subjects. Interestingly, Gjoreski et al. [13] proposed a method for continuously detection of stressful events by using physiological data from commercial wrist device. These systems were able to determine the stress and emotional states of the subject by using the peripheral physiological data with an impressive accuracy but since, the setup used by these systems is very difficult to put in place (excluding the Gjoreski et al.); hence, making them un-practical for consumer-based market. To overcome the above challenges, we have tried to train our model using the modalities that can be collected through the wearable devices that can be easily setup.

III. Dataset Description

In this section, we would provide more details in the dataset we used, along with, number of subjects, types of sensors and their placement on the subjects' body.

A. Sensor Placement

For this collection of physiological data, two different devices were worn by the subjects, one on chest and the other one on wrist.

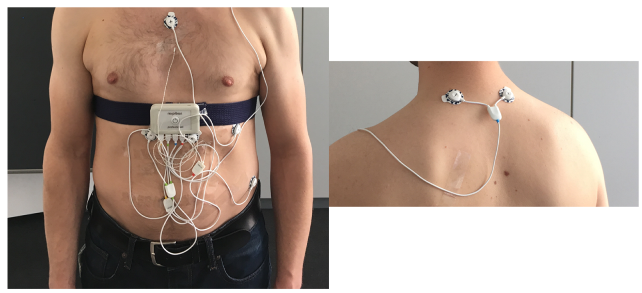

Fig. 1. Shows the placement of the sensors on chest of a Subject [2]

1) RespiBAN Professional:

- The first device worn chest was a RespiBAN Professional[1], and it is equipped with sensor for measuring acceleration via accelerometer (ACC) and respiration (RESP). Placement of the RespiBan and the ECG, EDA, EMG, TEMP sensors are shown in Figure 1.
- Along with these, the device was used to collect ECG, EDA, EMG, and TEMP from the subject's chest.
- All the signals recorded from RespiBAN Professional were recorded at a sampling frequency of 700Hz.
- The respiration (RESP) was measured using a respiration inductive plethysmograph sensor. Inductive plethysmography sensor is used for non-invasive monitoring of breathing patterns in subjects [14]
- ECG was recorded using standard three-point ECG and they were placed on the rectus abdominis since it has more density and is ideal for measuring ECG signal [2].

2) Empatica E4:

- Empatica E4 is a medical-grade wearable device that can detect the physiological data in real-time[2]
- E4 was used to record the following modalities:
    1. Blood Volume Pulse (BVP) at a sampling frequency of 64 Hz,

[1] https://www.biosignalsplux.com/index.php/respiban-professional
[2] https://www.empatica.com/research/e4/

2. Electrodermal Activity (EDA) at a sampling frequency of 4 Hz,

3. Subject's Temperature at sampling frequency of 4 Hz, and;

4. Subject's motion (ACC) at a sampling frequency of 32 Hz.

- All the subjects wore the E4 on their non-dominant hand.

B. Subjects

For the current study, 17 graduate students with the exclusion criteria of pregnancy, heavy smoking, mental disorders, chronic and cardiovascular diseases were selected. However, data of 2 subjects were not provided in the dataset due to sensor malfunction. Out of the remaining 15 subjects, twelve subjects were male other three were females with an average age of 27.5 ± 2.4 years.

C. Dataset Structure

The dataset was organised in a WESAD folder with sub-directories for each subject. In the WESAD directory, each subject has a folder named 'SX', where the 'X' stands for the subject ID e.g., subdirectory 'S2' contains the physiological data collect from a subject with subject ID '2'. Each subdirectory contains multiple files including information regarding the sensor data and subject. Following files are included in the sub-directories:

- SX_readme.txt: It includes the information about the subject, data collection and data quality.
- SX_quest.csv: It includes the information about obtaining the ground truth (not considered in our analysis)
- SX_respiban.txt: It contains the data collect from the RespiBAN Professional device.
- SX_E4_Data.zip: It contains the data collect from the Empatica E4 device.
- SX.pkl: The python pickle file contains the synchronized raw data from both the devices. For synchronization, the double tap signal pattern was used to manually synchronize the raw data. The python pickle file is in the form of a dictionary and has the following keys with definition:

  1. 'subject': This key contains the subject ID of the subject e.g., 'S2'

  2. 'signal': This key contains the data from the two devices stored in the form of another dictionary with two keys i.e., 'chest' and 'wrist'.

  3. 'label': this key contains the ID of respective study protocol condition (sampled at 700 Hz). The key has the following IDs: 0 = not defined/transient, 1 = baseline, 2 = stress, 3 = amusement, 4 = meditation, 5/6/7 are ignored.

For the current analysis, only baseline, stress, and amusement categories were considered. Since we are trying to classify between stress and non-stress conditions, we have merged the baseline and amusement categories into a single category.

D. Labels

The dataset was prepared with goal to elicit three different affective states in the subjects i.e., neutral/baseline, stress, and amusement. Baseline condition was recorded for 20 minutes after each subject wore the sensors and sensors were synchronized using the double tap gesture. The goal for baseline condition was to induce a neural affective state. After being in the baseline for approximately 20 minutes, subjects were shown a set of funny video clips to elicit amusement affective state. For eliciting stress condition, the subjects were told to speak in public and perform a mental arithmetic task. A Meditation and Recovery condition was followed by stress condition to de-excite the subjects and remove the sensor equipment after synchronization using double tap gesture.

IV. Pre-Processing and Feature Extraction

In this section, we have mentioned all of the pre-processing steps that we have performed on the individual signals to remove the noise and extract features from the raw signals. The below mentioned steps for pre-processing were performed for each subject individually.

After loading the python pickle file in memory, we observed that the total length of data for each subject was approximately 101 minutes i.e., 1.68 hours. For this analysis, we have only considered the following modalities from the sensor placed in wrist i.e., EDA, BVP, and TEMP, while we only considered RESP from the sensor placed on chest.

Also, due to different sampling frequencies of signals, the number of data points were different. So, before joining the data from different sensors in a single data frame we had to pre-process the raw sensor data, it is mentioned in the subsection IV-A. Different raw signal data from the sensors are shown in the Figure 2.

A. Pre-Processing

- Noise Reduction

  EDA (Electrodermal Activity) refers to electrical potential on skin surface. EDA measured using wearable devices contains high frequency noise caused by electrical interference. Therefore, to remove high frequency noise in EDA, 1Hz Butterworth low pass filter of 6th order was used here. Moreover, Respiration data also contains high-frequency noise, so same low pass filter was used to remove that noise (see Figure 3 & 4).
- Extraction of Different components from EDA signal

  EDA signal contains phasic and tonic component, which are very important feature in determining emotional state of a person. So here, phasic and tonic components from EDA signal were obtained using cvxEDA – Convex Optimization approach to Electrodermal Activity Processing [15]. It describes EDA as sum of three terms: Phasic component, Tonic component and Additive white Gaussian noise. The algorithm implemented by [16] was used to extract these components
- Dataframe creation and Filling Missing Labels

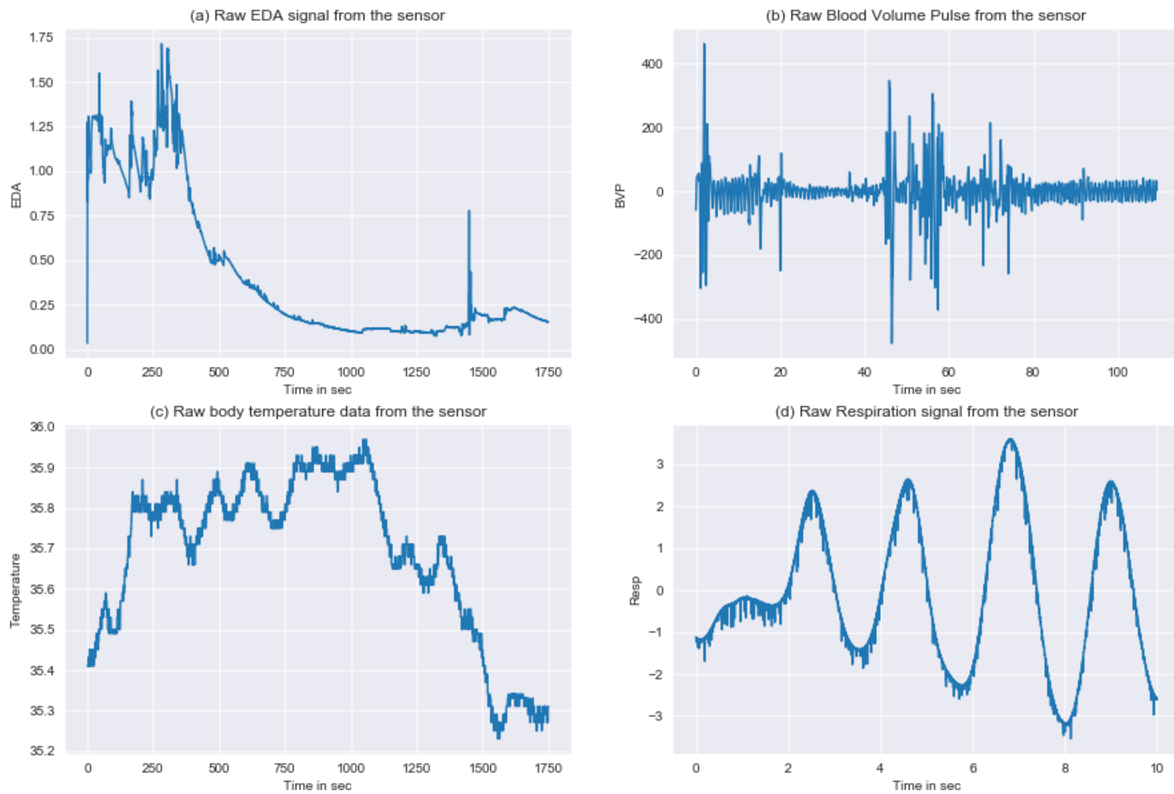

Fig. 2. Shows Raw Signals from the Sensors for Subject 'S2'

As the given data did not contain time information, it was needed to convert it to time series. This was done by converting indices of signals to time with the help of sampling rate and length of signal. EDA, BVP and Temperature data from wrist-worn Empatica E4 device and Respiration data from RespiBAN device were combined with labels to create a dataframe.As a result of outer join, around 4200000 data points were created out of which approx. 350000 data points did not have labels. So, to fill the missing values of labels, forward filling method was used. Forward filling uses previous known data value to fill missing value in time-series data.

- Group the combined data by labels

  The original data contained 7 labels from which 3 labels were useful in this task of stress detection. Those labels were Stress, baseline and amusement. So, baseline, stress and amusement samples were obtained by grouping the data by labels.

- Feature Extraction

  After that, the signals were cut into 10 second window with no overlap and statistical features like mean, standard deviation, min and max values of signals were computed on each window. Moreover peak frequency of BVP signal was obtained with the help of Welch's method for each window. Also, number of peaks of BVP signal and Resp signal on each window was counted. Stressful condition also impacts body temperature. So temperature slope was calculated from Temperature data for each window.

After all the pre-processing and feature extraction, the features for individual subjects were stored in a .csv file as well as the features were combined in a main .csv file for model training.

V. Classification Algorithms

After pre-processing the raw signal data and combining them in a single feature matrix, we preformed multiple classification techniques to classify stress and non-stress conditions. For this analysis, we trained five classifiers and performed hyperparameter tuning for them to get the best parameters (more information about the architecture and hyperparameter tuning has been provided in the below sections).

A. *Neural Networks*

For designing the architecture for the neural networks, we trained multiple neural networks with different number of hidden layers and units at each layer. However, the best performance from the neural networks was achieved by the neural network architecture that had 5 hidden layers. In this architecture, the first layer is the input layer that have inputs equal to the shape of feature matrix and outputs of 256 nodes. In this layer, we used 'ReLU' i.e., Rectified Linear Unit as the activation function with a 'l2' regularizer (l = 0.001). After that layer, we used the dropouts of 0.5. After this, in the second

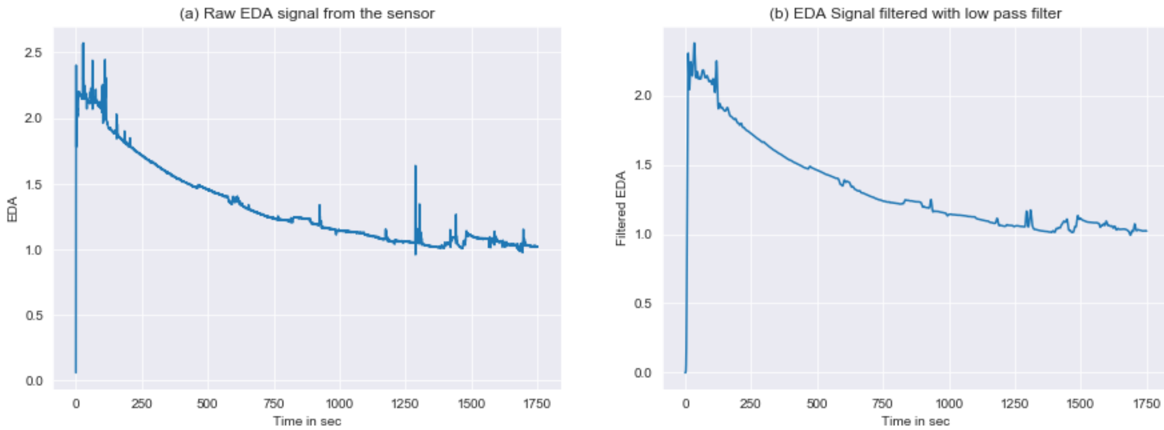

Fig. 3. Comparison between Raw EDA Signal And Filtered EDA signal from subject S2

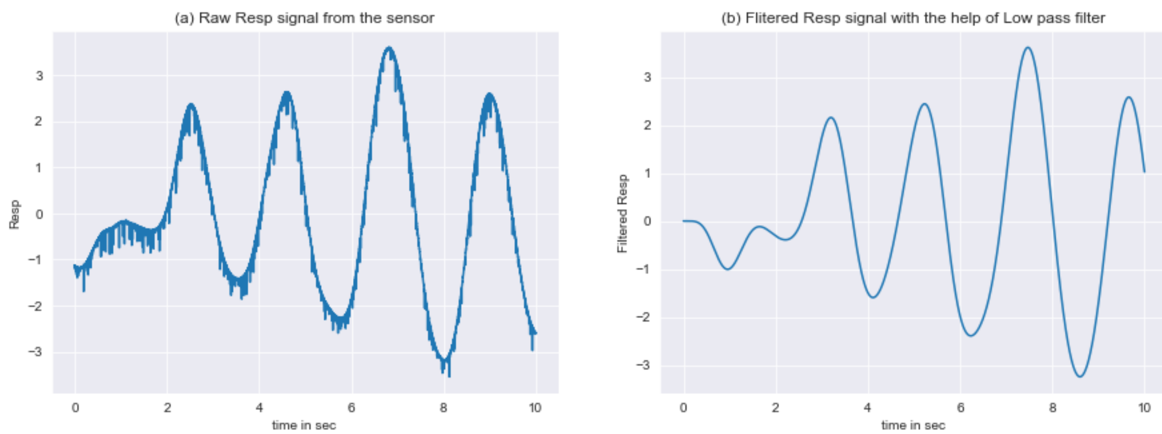

Fig. 4. Comparison between Raw RESP Signal And Filtered RESP signal from subject S2

layer, we used 512 nodes as output with the same activation function, regularizer, and dropouts. For the third layer we used the same layer as the second layer with the same activation function, regularizer and dropouts. For the fourth layer, we increase the number of output layers to 1024 and kept other parameters same as the above layers. The fifth, final layer, was a fully connected dense layer with 2 units as output and 'Softmax' as the activation function. For this model, we used 'Categorical Cross Entropy' as the loss and 'Adam' as the optimizer with a learning rate of '0.001'. For training, we used a batch size of '95' for 50 epochs with a validation split of '0.2' i.e., 80% of the training data would be used for training and 20% would be used for validation. More information on the evaluation method is given in the next section. Due to the limited computational power, we could not perform a thorough hyperparameter tuning for neural networks and identified the hyperparameter with 'hit and trial' method.

B. Logistic Regression

Logistic Regression is an algorithm for classification that transforms its output using logistic sigmoid function to return a probability value. Then the target label corresponding to highest probability is selected as the target label. For training the Logistic Regression (LR), we performed a grid search for tuning the parameter 'C' of the model. The 'C' parameter in LR is known as Inverse regularization parameter i.e., lower value of 'C' would represent high value of Lambda regulator. After the grid search, we trained the Logistic Regression Classifier on the training dataset with the best parameters identified in the grid search.

C. Support Vector Machine (SVM)

For training the Support Vector Machine, we used Support Vector Classification (SVC) from Scikit-learn package. The implementation of SVC is based on 'libsvm' library. For SVC, same as Logistic Regression, we performed a grid search for the parameter 'C' with Radial Basis Function as the kernel and kept the default value of 'gamma' i.e., gamma = 1. After performing the grid search, we used the best value of 'C' parameter to train on the training dataset and evaluated its performance on test dataset.

D. Random Forest Classifier

Random forest classifier is a very powerful, yet very simple, machine learning algorithm that use ensemble technique.

TABLE I
PRE-PROCESSING STEPS AND EXTRACTED FEATURES

Signal	Pre-processing on the Signal	Extracted Features *(with a window of 10 seconds)*
EDA	Noise reduction using low pass filter, Obtaining Phasic & Tonic Components	Statistical Features like Mean, Std., Min., Max. for: 1. EDA Signal 2. Phasic Component 3. SMNA driver of Phasic component 4. Tonic component
Respiration	Noise reduction using low pass filter	Statistical Features like Mean, Std., Min., Max. Number of Peaks
BVP	N/A	Statistical Features like Mean, Std., Min., Max. peak frequency Number of Peaks
Temperature	N/A	Statistical Features like Mean, Std., Min., Max. Slope

Random Forest is an estimator that fits number of Decision Tree Classifiers on sub-samples of the dataset. For training the classifier, we used a grid search to identify the optimized number of trees for the forest. After performing the grid search, we used the best value of 'n_estimators' parameter to train on the training dataset and evaluated its performance on test dataset.

E. AdaBoost

Adaptive Boosting is a boosting algorithm that combines multiple "weak classifiers" into a "better more stronger classifier". For training the adaboost classifier, we used a grid search to identify the best parameter for number of trees and then used that value to train on the train set.

VI. EVALUATION MATRIX

For evaluating our classifiers, we used the accuracy and F-score as evaluation metrics. Accuracy evaluation gives us the number of correctly predicted data points out of all the data points by the classifier, while F1-score gives us the measure of test's accuracy. F1-score is defined as the weighted harmonic mean of the test's precision and recall. Since, in this analysis, the data was unbalanced as the number of datapoints in non-stress class was more than the number of datapoints in stress class, so it is recommended to use the F1-score as evaluation metrics.

Also, all models were evaluated using Leave-one-subject-out (LOSO) procedure. In this, we divided our dataset into two parts i.e., train, and test, where all the subjects except one subject were used for training and the model was evaluated on the subject that was not included in the train dataset. The motive of using LOSO was to indicate how well the classifiers did on the data from previously unseen subject. Finally, an average accuracy and F1-score was also calculated.

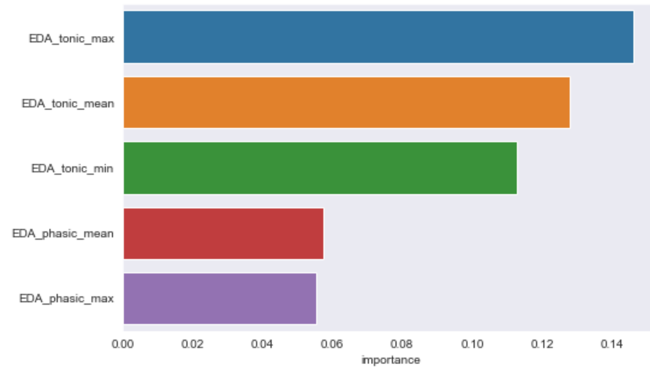

Fig. 5. Bar plot showing Importance of Features

VII. EFFECT OF PARAMETERS

In order to access the importance of features after features extraction on the classification accuracy, we used random forest to predict the target variable with best parameters i.e., no. of trees equal to 700. For feature importance, we used Random Forest Classifier from Scikit-learn package. This algorithm has an attribute called 'feature importance'. The higher the value for feature importance for a feature the more important the feature for classification. Based on the feature importance, we observed that the accuracy of the classifier degrades abruptly if we remove the features related to EDA (see Figure 5).

TABLE II
IMPORTANCE OF EDA FEATURES FOR CLASSIFICATION

Classifier	F1-score for Stress Class		
	With Only EDA	Without EDA	EDA + Other
Logistic Regression	0.82	0.61	0.85
AdaBoost	0.80	0.64	0.83
Random Forest	0.79	0.62	0.82
SVC	0.78	0.50	0.82
Neural Net	0.78	0.49	0.85

Also, to check the importance of EDA feature, we trained neural networks, random forest, logistic regression, SVC, and adaboost on EDA features only, without EDA features and with EDA and all other features and evaluated them on the test dataset. By this analysis, we observed that EDA features greatly impacts the overall accuracy since, without EDA features the f1-score of all classifiers decrease abruptly (see Table II). We also tried to implement dimensionality reduction by using PCA and LDA; however, the classifiers did not perform well on the transformed features, so we had to drop that approach.

Subject	Affective Condition	Logistic Regression		Random Forest		Neural Network		SVC		AdaBoost	
		f1-score	Accuracy	f1-score	Accuracy	f1-score	Accuracy	f1-score	Accuracy	f1-score	Accuracy
S2	Non-Stress	0.94	0.91	0.96	0.93	0.93	0.91	0.93	0.90	0.94	0.91
	Stress	0.85		0.87		0.85		0.82		0.82	
S3	Non-Stress	0.76	0.72	0.83	0.76	0.79	0.74	0.77	0.72	0.82	0.77
	Stress	0.64		0.63		0.66		0.65		0.70	
S4	Non-Stress	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00
	Stress	1.00		1.00		1.00		0.99		1.00	
S5	Non-Stress	1.00	1.00	1.00	1.00	1.00	1.00	0.95	0.93	0.99	0.99
	Stress	0.99		0.99		0.99		0.88		0.99	
S6	Non-Stress	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.98
	Stress	1.00		1.00		1.00		0.99		0.96	
S7	Non-Stress	0.99	0.99	0.97	0.96	0.97	0.96	0.97	0.96	0.96	0.95
	Stress	0.99		0.93		0.94		0.94		0.91	
S8	Non-Stress	0.99	0.98	1.00	1.00	0.98	0.97	0.98	0.97	0.99	0.99
	Stress	0.97		1.00		0.96		0.94		0.99	
S9	Non-Stress	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00
	Stress	0.99		1.00		0.99		0.98		1.00	
S10	Non-Stress	0.99	0.98	0.93	0.89	1.00	1.00	0.99	0.98	0.99	0.99
	Stress	0.97		0.79		0.99		0.97		0.98	
S11	Non-Stress	0.85	0.77	0.82	0.48	0.82	0.73	0.86	0.78	0.90	0.85
	Stress	0.47		0.48		0.49		0.50		0.70	
S13	Non-Stress	0.98	0.98	0.99	0.99	0.98	0.97	0.98	0.97	0.96	0.94
	Stress	0.97		0.98		0.95		0.95		0.88	
S14	Non-Stress	0.88	0.83	0.11	0.34	0.82	0.76	0.88	0.84	0.03	0.31
	Stress	0.71		0.48		0.67		0.73		0.46	
S15	Non-Stress	0.98	0.98	1.00	1.00	0.96	0.95	0.91	0.89	0.95	0.93
	Stress	0.96		0.99		0.92		0.84		0.88	
S16	Non-Stress	0.99	0.99	1.00	1.00	0.96	0.95	1.00	1.00	0.96	0.95
	Stress	0.99		1.00		0.92		0.99		0.91	
S17	Non-Stress	0.66	0.54	0.63	0.48	0.72	0.63	0.67	0.53	0.61	0.50
	Stress	0.27		0.17		0.43		0.67		0.31	

Fig. 6. Table showing Subject-wise evaluation of Classifiers

VIII. RESULTS

After training all the machine learning algorithms, evaluation of these algorithms was performed based on the evaluation criteria mentioned in the section evaluation matrix VI. After comparing the results, we could observe that the Neural Networks and Logistic Regression classifiers performed best on the test subject's dataset. Both could classify the Stress condition in test subject with an average F1-Score of 85%, and with 93% F1-score for classifying Non-Stress conditions. It was very interesting to note that the Logistic Regression could match the performance of Neural Networks while taking less time to train and requiring very less computational power compared to Neural Networks. However, by observing the Loss curve of Neural Networks (see Figure 7), we can obverse that the validation and training loss was still decreasing at the end of 50 epochs. Hence, we can say that Neural Networks could have performed better if we would have trained them for more epochs i.e., running for 100 epochs then 50 epochs. It is also interesting to note that both Neural Networks and Logistic Regression classifiers could not perform better on the test subject 'S11' and 'S13'. While, AdaBoost's performance on the subject 'S13' was way better than the other algorithms with a F1-score of 70% and for subject ‘S17’ SVC classifier performed best with a F1-Score of 67%.

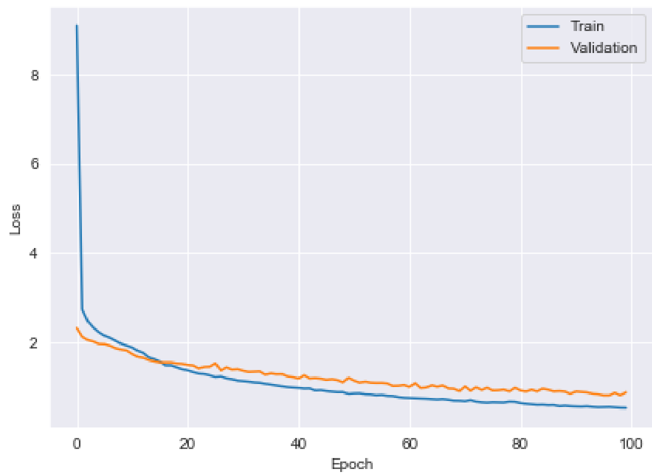

Fig. 7. Figure showing Loss vs Epoch curve for Neural Networks

TABLE III
EVALUATION OF CLASSIFIERS

Classifier	**Accuracy**	**F1-Score**	
		Stress	Non Stress
Logistic Regression	0.91	0.85	0.93
Neural Network	0.90	0.85	0.93
AdaBoost	0.87	0.83	0.87
Random Forest	0.87	0.82	0.88
SVC	0.90	0.82	0.92

IX. CONCLUSIONS

Objective of this project was to use the publicly available dataset WESAD, a multimodal dataset for wearable stress and affect detection, and implement different machine learning and deep learning algorithms to predict the stress and non-stress conditions in human beings. We implemented 5 different machine learning algorithms on the dataset and evaluated their performance with Leave-one-subject-out (LOSO) cross-validation procedure. By evaluating the algorithms, we observed that Logistic Regression and Neural Networks performed best on the dataset. However, Logistic Regression Classifier was more efficient than Neural Networks as it took less training and scoring time with less computational power. However, Neural Networks showed potential of performing better on the dataset set, if they were trained for more epochs, since the validation loss vs epochs curve was gradually decreasing even at the end of 50 epochs. Also, by evaluating the features importance, we could identify that with just EDA features, stress and non-stress conditions could be predicted to a great extent; however, adding other features like BVP and Body Temperature the overall prediction accuracy is increased. This indicates that by only using the wrist sensor data promising results can be achieved; hence making practical for commercial use. For this analysis, we did not consider the data from chest sensor and the raw accelerometer data. Adding the accelerometer data to the above analysis could improve the overall prediction accuracy and F1-Score.

AUTHORS’ CONTRIBUTION

Due to very subjective nature of the project, most of the tasks performed to achieve the final outcome of the project were performed in collaboration of both authors. Almost each task performed the author was thoroughly discussed and cross-validated with/by the other author. It would be very difficult to draw boundary between task as none of the tasks were executed individually. However, for the sake of reference, we have tried to segregate the tasks and the role of both the authors in those tasks.

A. Python Programming

Initially, Kamal and Anubhav individually explored the dataset to understand the pickle file and how the data in the pickle file is arranged. After understanding the structure of data, pre-processing steps like converting the data into time-series, merging the different raw signals, and designing the window for sampling was done by Kamal, while Anubhav worked on designing the butter-worth filter, extracting features EDA signal, and filling missing values in labels. Tasks like feature extraction e.g., peak detection, number of peaks, statistical features were done in collaboration. For training the models, since, Kamal had better resource, in terms of computational power, Neural Networks were trained on the Kamal’s computer. Anubhav provided the architecture of neural networks e.g., number of layers, hyperparameters, etc. that were then implemented by Kamal on his computer. Logistic Regression and Random Forest with its hyperparameter tuning was implemented by Kamal, while Anubhav implemented SVC and AdaBoost and performed its hyperparameter tuning.

B. Report

For report, the section of literature review was divided equally among both of the authors. While the sections like Abstract, Problem statement, Classification Algorithms, Effect of Parameters, Results were written by Anubhav. Kamal wrote the following sections Introduction, Pre-Processing and Feature Extraction, Evaluation Matrix, and Conclusion.

REFERENCES

[1] UCI Machine Learning Repository: WESAD (Wearable Stress and Affect Detection) Data Set, archive.ics.uci.edu/ml/datasets/WESAD (Wearable Stress and Affect Detection).

[2] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. V. Laerhoven, “Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection,” Proceedings of the 2018 on International Conference on Multimodal Interaction - ICMI 18, pp. 400–408, Oct. 2018.

[3] “Stress is Killing You,” The American Institute of Stress, 23-Aug-2012. [Online]. Available: http://www.stress.org/stress-is-killing-you. [Accessed: 28-Apr-2020].

[4] "5 Things You Should Know About Stress," National Institute of Mental Health. [Online]. Available: https://www.nimh.nih.gov/health/publications/stress/index.shtml. [Accessed: 29-Apr-2020].

[5] R. Sioni and L. Chittaro, "Stress Detection Using Physiological Sensors," in Computer, vol. 48, no. 10, pp. 26-33, Oct. 2015.

[6] S. Mirsamadi, E. Barsoum and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 2227-2231.

[7] P. Tzirakis, G. Trigeorgis, M. A. Nicolaou, B. W. Schuller and S. Zafeiriou, "End-to-End Multimodal Emotion Recognition Using Deep Neural Networks," in IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1301-1309, Dec. 2017.

[8] H. Lu, D. Frauendorfer, M. Rabbi, M. S. Mast, G. T. Chittaranjan, A. T. Campbell, D. Gatica-Perez, and T. Choudhury, "StressSense," Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp 12, pp. 351–360, Sep. 2012.

[9] K. Plarre et al., "Continuous inference of psychological stress from sensory measurements collected in the natural environment," Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, Chicago, IL, 2011, pp. 97-108.

[10] K. Hovsepian, M. Alabsi, E. Ertin, T. Kamarck, M. Nakajima, and S. Kumar, "cStress," Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp 15, pp. 493–504, Sep. 2015.

[11] Y. Liu, M. Yu, G. Zhao, J. Song, Y. Ge and Y. Shi, "Real-Time Movie-Induced Discrete Emotion Recognition from EEG Signals," in IEEE Transactions on Affective Computing, vol. 9, no. 4, pp. 550-562, 1 Oct.-Dec. 2018.

[12] A. O. Akmandor and N. K. Jha, "Keep the Stress Away with SoDA: Stress Detection and Alleviation System," in IEEE Transactions on Multi-Scale Computing Systems, vol. 3, no. 4, pp. 269-282, 1 Oct.-Dec. 2017.

[13] M. Gjoreski, H. Gjoreski, M. Luštrek, and M. Gams, "Continuous stress detection using a wrist device," Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, Sep. 2016.

[14] G. Brüllmann, K. Fritsch, R. Thurnheer, and K. E. Bloch, "Respiratory Monitoring by Inductive Plethysmography in Unrestrained Subjects Using Position Sensor-Adjusted Calibration," Respiration, vol. 79, no. 2, pp. 112–120, 2010.

[15] A. Greco, G. Valenza, A. Lanata, E. P. Scilingo and L. Citi, "cvxEDA: A Convex Optimization Approach to Electrodermal Activity Processing," in IEEE Transactions on Biomedical Engineering, vol. 63, no. 4, pp. 797-804, April 2016.

[16] Lciti, "lciti/cvxEDA," GitHub, 07-Feb-2017. [Online]. Available: https://github.com/lciti/cvxEDA. [Accessed: 29-Apr-2020].

[17] WJMatthew. "WJMatthew/WESAD." GitHub, 21 May 2019, github.com/WJMatthew/WESAD.