**Evaluating machine learning and deep learning models on WESAD dataset**

**Introduction**

In modern life, stress management has become a very important role in human’s day to day lives. The cause of stress can be due to personal or professional life. Stress can be termed as the body’s response to a particular situation, usually in anxious and hysterical conditions. Whenever a person is in stress condition, the reaction inside the body changes by heavy breath, increase in body temperature, blood flow and heart rate. Individual must understand the responses and relations among the physiological symptoms to detect and recognize stress. These physiological symptoms are cardinal to identify so that one can avoid long term chronic effects. Therefore, in this paper we have comprehended these significant physiological symptoms for prognosticating stress conditions.

The main objective of the proposed work is to detect if an individual is undergoing stress using physiological data. This physiological data is present in the multimodal and time series dataset [1] WESAD. Almost all the features present in the dataset are independent of each other and therefore, are important for drilling and training the model. The dataset was used for multi-classification (transient, stress, baseline, amusement, meditation) and binary classification (stress or no stress). Initially, data visualization and data cleaning are performed to understand the corelations of features and choose the important features. Various sampling and cross validation methods are applied to the dataset for generalization purpose and making the highly imbalanced dataset to balanced dataset. Ensemble machine learning algorithms like AdaBoost, ExtraTrees and XGBoost classifiers and deep learning algorithm like MLP classifiers are compared as predictive models and the best one is perceived.

**Observations**

|  |  |  |  |
| --- | --- | --- | --- |
| **Paper** | **Algorithms used** | **Features used** | **Findings** |
| [2] Stress Detection with ML and DL using Multimodal Physiological Data | KNN, LDA, Random Forest, DT, Adaboost, Kernel SVM, ANN | ACC (RespiBAN), ACC (Empatica E4),ECG, BVP, EDA (RespiBAN), EDA (Empatica E4), EMG, RESP, TEMP | Detection of stress of an individual using WESAD in which LOSO cross validation, PCA, Quantile Transformer and Standard Scaler are used for preprocessing. Kernal SVM and ANN give best performance. |
| [3]User Independent Human Stress Detection | Random Forest (RF), Extra-Trees (ET), Decision Tree (DT) and Gradient Boosting Decision Tree (GBDT) | ACC (RespiBAN), ACC (Empatica E4),ECG, BVP, EDA (RespiBAN), EDA (Empatica E4), EMG, RESP, TEMP | Designed a User Independent (UI) system to predict affective state of individual using WESAD which performs better than User Dependent model. Data prepossessing comprises of Feature Engineering, Temporal segmentation, Feature selection and Oversampling. |
| [4]Dimension-raising Processing Framework for One-dimensional Time Series and its Application in Affect Detection |  |  |  |
| [5]Stress Detection With Single PPG Sensor by Orchestrating Multiple Denoising and Peak-Detecting Methods |  |  |  |
| [6]Evaluating KNN Performance on WESAD dataset |  |  |  |

**Dataset**

The dataset used is [1] WESAD, also known as Wearable Stress and Affect Detection dataset. It is a multimodal dataset, which contains data in the form of numeric values. The dataset is made by conducting experiments on 15 subjects or participants. The dataset is also a multivariate and time-series dataset, where data is collected every second per participant during the experiment time zone.

The [1]data is collected with the help of two devices, RespiBAN and Empatica E4 sensors. RespiBAN sensor is a chest worn device which detects ECG (mV), EDA (μS), EMG (mV), TEMP (°C), XYZ (g), RESP (%). These modalities were sampled at 700 Hz. Empatica E4 is a wrist-worn device which detects ACC (g), BVP, TEMP (°C), and EDA (μS). Of all these features, important modalities are chosen and combined into a pickle document. The important modalities present are Blood Volume Pulse (BVP), Electrocardiogram (ECG), Electrodermal Activity (EDA), Electromyogram (EMG), Respiration (RESP), Body Temperature and Three Axis Acceleration.

**Class Labels**

The [1] dataset contains 8 output labels, however the last 3 are told to be ignored. Hence, the dataset contains 5 significant output labels, namely, Transient, Baseline, Stress, Amusement and Meditation. Baseline condition indicates the neutral affective state of the participant, where they are told to either stand, sit or read a magazine. Stress condition indicated how the participant handled the Trier Social Stress Test (TSST), where they had to talk for 5 minutes in front of a group of people and later solve an arithmetic problem. Amusement condition shows how the participant can react when he/she is shown funny videos. Meditation condition was to deactivate the excitement state to neural state. Transient condition is the time moving from one condition to another.

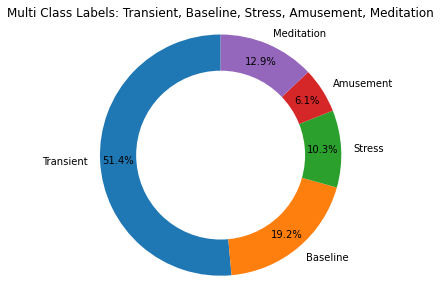


Fig 1. Multiclass Label Distribution

In Fig. 1, each color specifies each class label with a percentage ratio for each one of them. Table 1 gives a number of samples per class label. From the pie chart, we can depict that the number of samples are highest in Transient condition with 2142701 samples and lowest number of samples are present in Amusement condition253400 with samples. Moreover, we see that each class has a different percentage of class labels, indicating that the dataset is an unbalanced dataset.

Table 1 :Distribution of multi class labels

|  |  |  |
| --- | --- | --- |
| **Class Label** | **Number of samples** | **Percentage (%)** |
| Transient | 2142701 | 51.40 |
| Baseline | 800800 | 19.20 |
| Stress | 430500 | 10.30 |
| Amusement | 253400 | 6.10 |
| Medication | 537599 | 12.90 |

In this paper, we have created binary class labels from the multiclass label dataset. The two classes are Stress and No Stress, as shown in Fig. 2. No Stress label is created by merging Transient, Baseline, Amusement and Medication conditions.

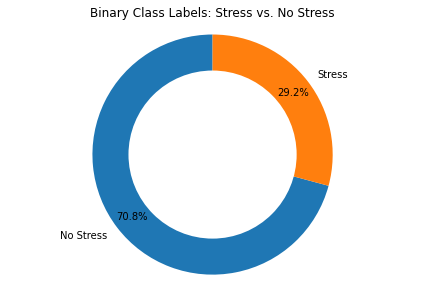


Fig 2. Binary Class Label Distribution

Table. 2 shows the total number of samples of stress and no stress class labels, where the number of stress samples are less as compared to no stress samples.

Table 2. Distribution of binary class labels

|  |  |  |
| --- | --- | --- |
| **Class Label** | **Number of samples** | **Percentage (%)** |
| No Stress | 1044399 | 70.80 |
| Stress | 430500 | 29.20 |

**Features**

The dataset contains 14 features and 1 class label. The 14 features are displayed in Fig 2.

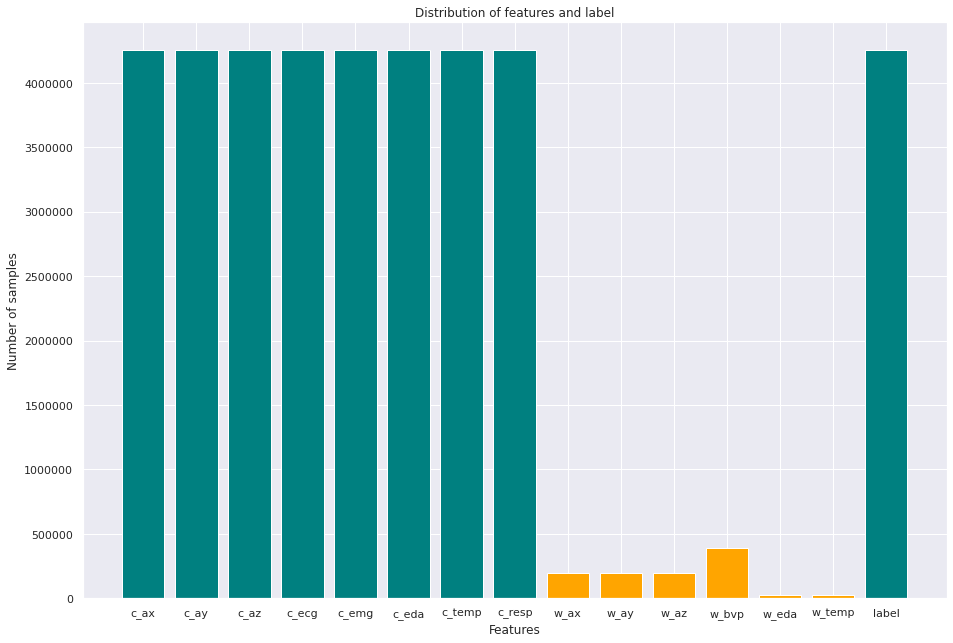


Fig 3. Bar chart for distribution of features and label

In Fig. 3, we can identify that chest worn device data is stable and more as compared to the wrist worn device data. Hence, again we can estimate that the dataset is unbalanced. As the number of samples in wrist data is low, we have decided to ignore them and continue our experiment on the chest worn device data and the class label.

As mentioned before, the dataset is the pickle format, meaning, the data is present in the character stream. Hence, we convert the pickle format data into pandas data frame with the significant features and class label. Code snippet for the same is shown in Fig. 4.

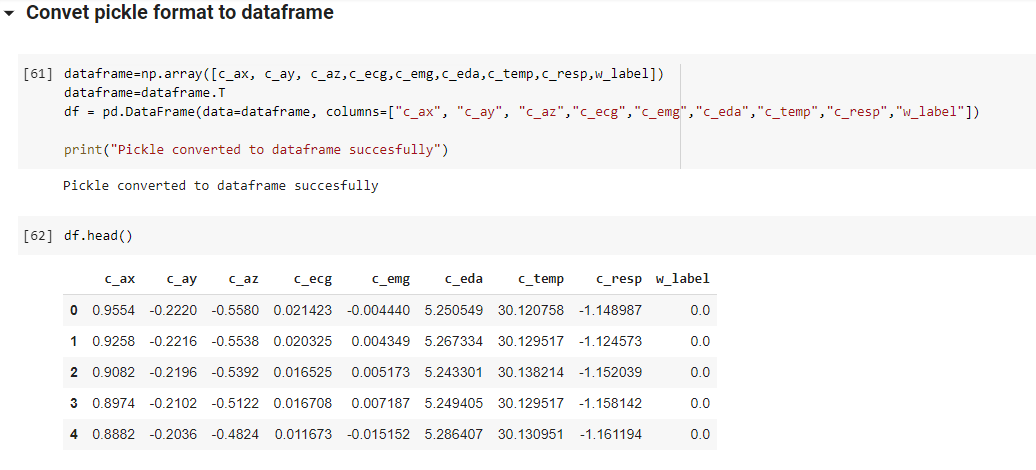
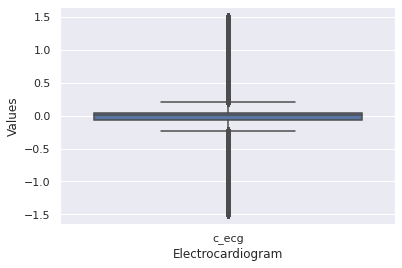
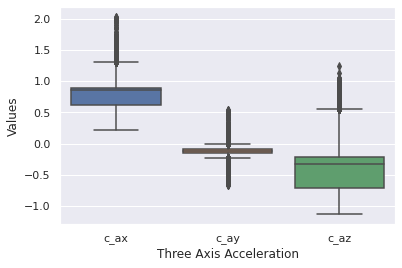


Fig 4. Pickle dictionary to pandas data frame conversion

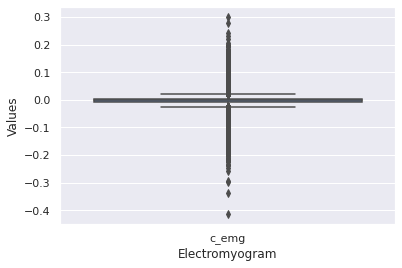
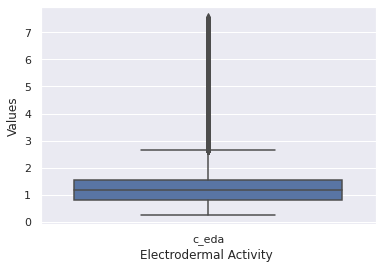
The data frame has 4255300 instances and 9 features. It has no null values and the data type of each feature is float64. The memory usage is 292.2 MB and is a numerical dataset.

**Data Cleaning**

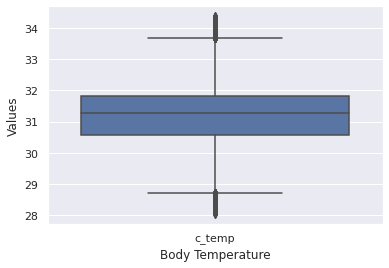
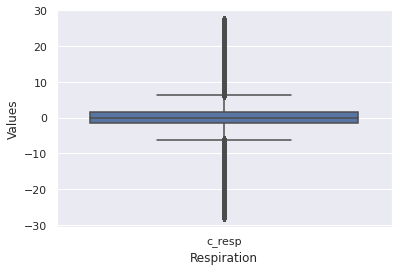
To clean the data, we have tried to find out if outliers are present in each of the features in the dataset. Outlier is a point which is present in an isolated or secluded area from the significant points**.** These outliers can cause significant impact on the performance. Hence these minor points can be removed.Box plots have been used for the visualization of outliers present in each feature.



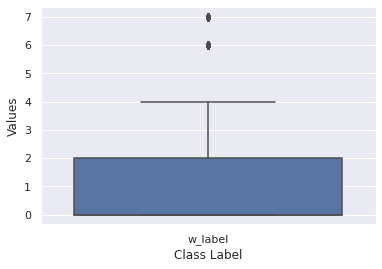
(a) (b)



(c) (d)



(e) (f)



(g)

Fig 5. Outlier detection using bar plot of each feature

In Fig. 5, the black thick line represents the outlier for each of the features. Of all the features, we notice that EDA (Fig. 5(c)), EMG (Fig. 5(d)), Resp (Fig. 5(e)) and Temp (Fig. 5(f)) sensors have the maximum number of outliers. The below table shows the exact number of outliers for each feature. To calculate the outliers, we have calculated the IQR, also called The Interquartile Range.

Table 3. IQR values of all features

|  |  |
| --- | --- |
| **Feature** | **IQR value** |
| c\_ax | 0.271200 |
| c\_ay | 0.054000 |
| c\_az | 0.507400 |
| c\_ecg | 0.112335 |
| c\_emg | 0.012314 |
| c\_eda | 0.746918 |
| c\_temp | 1.240173 |
| c\_resp | 3.100586 |
| w\_label | 2.000000 |

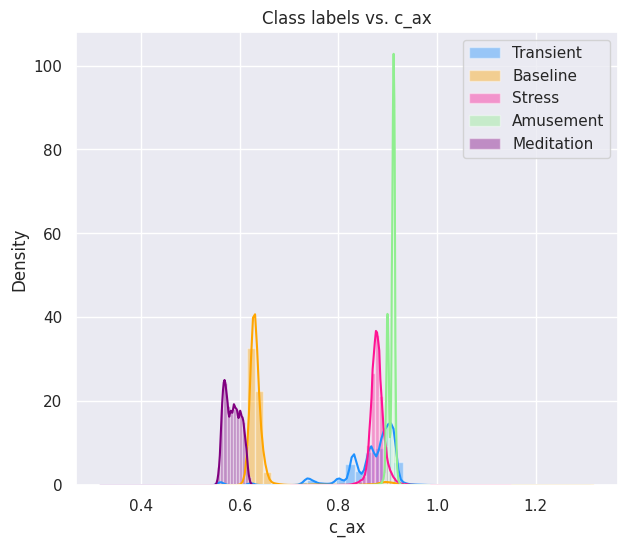
After removing the outliers, the shape of the data frame is reduced, as shown in Table. 4.

Table 4. Shape of dataset before and after outlier removal

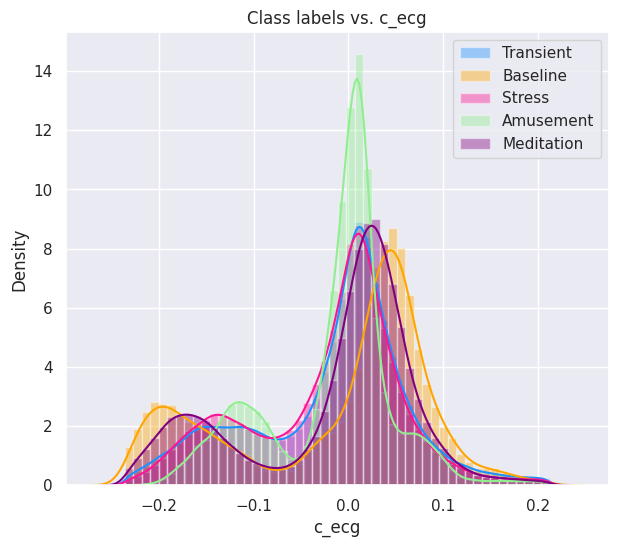
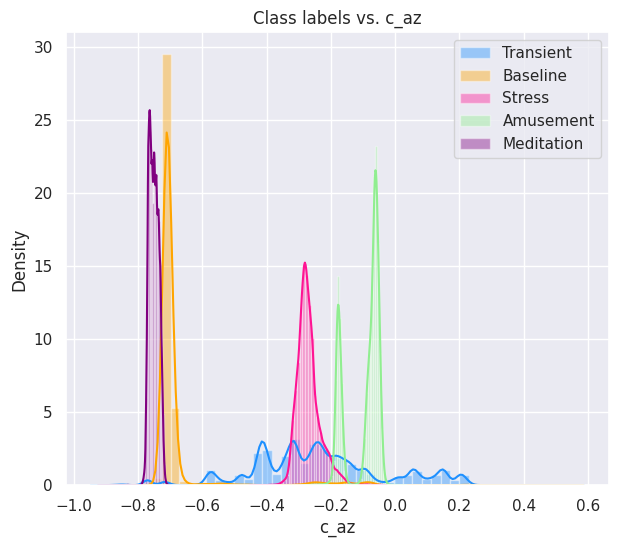
|  |  |
| --- | --- |
| **Before Outlier Removal** | **After Outlier Removal** |
| (4255300, 9) | (2100456, 9) |

**Histograms**

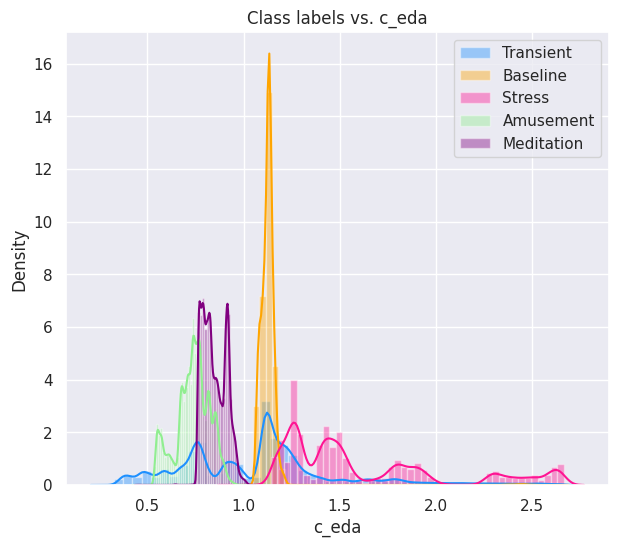
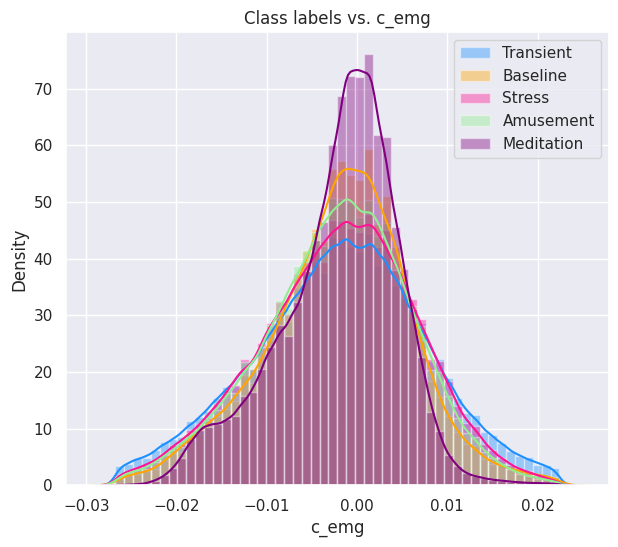
To visualize the WESAD dataset, the data visualization technique used is Histogram. It helps to summarize the distribution of a univariate dataset. Histograms are created for each feature present in the WESAD dataset.



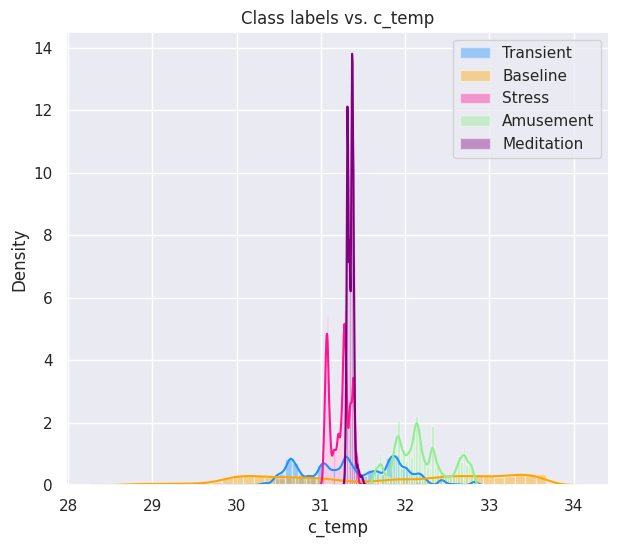
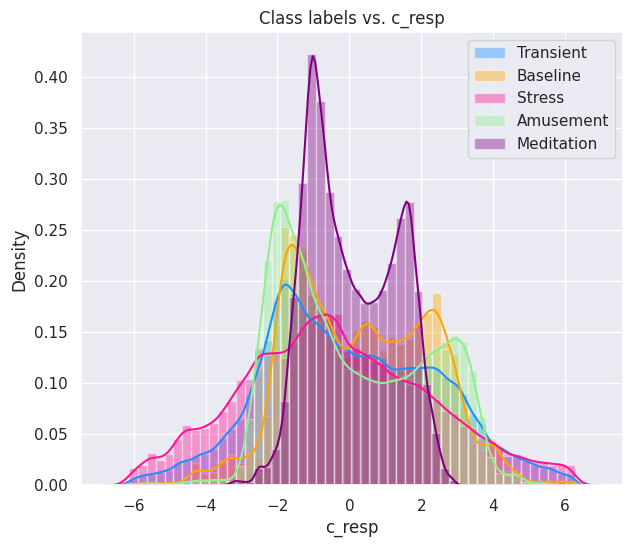
(a) (b)



(c) (d)



(e) (f)



(g) (h)

Fig 6. Histogram representation of each feature in WESAD

Fig. 6 shows that all features have Gaussian Distribution, meaning the graphs are symmetric about the mean. They have a bell shape graph. Basically, the data which is present in the centre is more frequent than the data which is present far from the mean.

Each feature is compared with each of the class labels with different colours. The meditation condition is highly dependent on c\_ecg, c\_emg and c\_resp (see Fig. 6(d,e,g)). The heart beat count and the breathing count are the most significant for a person to perform meditation. While, the amusement condition is highly dependent on most of the features like c\_az, c\_ecg, c\_c\_emg, c\_eda and c\_resp. Also, the other conditions are significantly dependent on all the other features. Each of the features are important for prediction, as each feature is connected to each class.

**Correlation Matrix**

Using correlation matrix, we get to know the degree of correlations between each feature. The light colored points indicate that the features are highly related, while dark points indicate they are weakly related to each other.

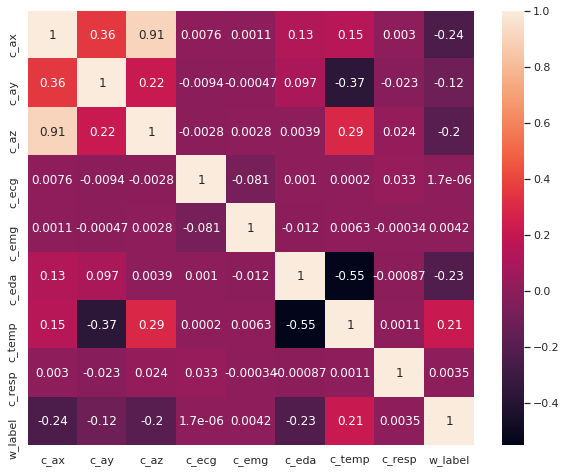


Fig 7. Heatmap of all features

In Fig. 7, we can estimate that c\_az and c\_za are highly and strongly related to each other with correlation values of 0.91. However, other features are not strongly related to each other. Hence, we can say that all the features are independent of each other. Hence, for evaluation all the features will be considered. Fig 8. shows the correlation between c\_az and c\_az.

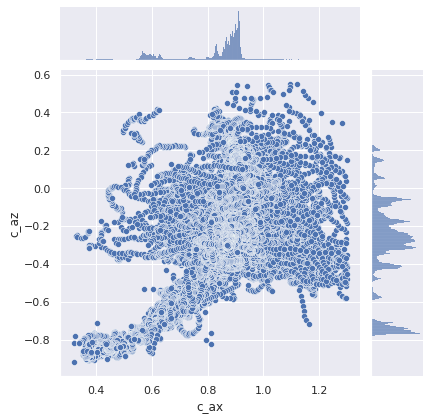


Fig 8. Correlation between c\_az and c\_ax

**Data Sampling and Cross Validation**

The WESAD dataset has an unbalanced number of samples for each of the output labels (see Fig. 1). Due to unbalance, we can have misclassification, where it will become most biased to the Transient label of all labels due to the higher number of samples. Hence, to avoid this limitation we perform K Fold Cross Validation and Data Sampling.

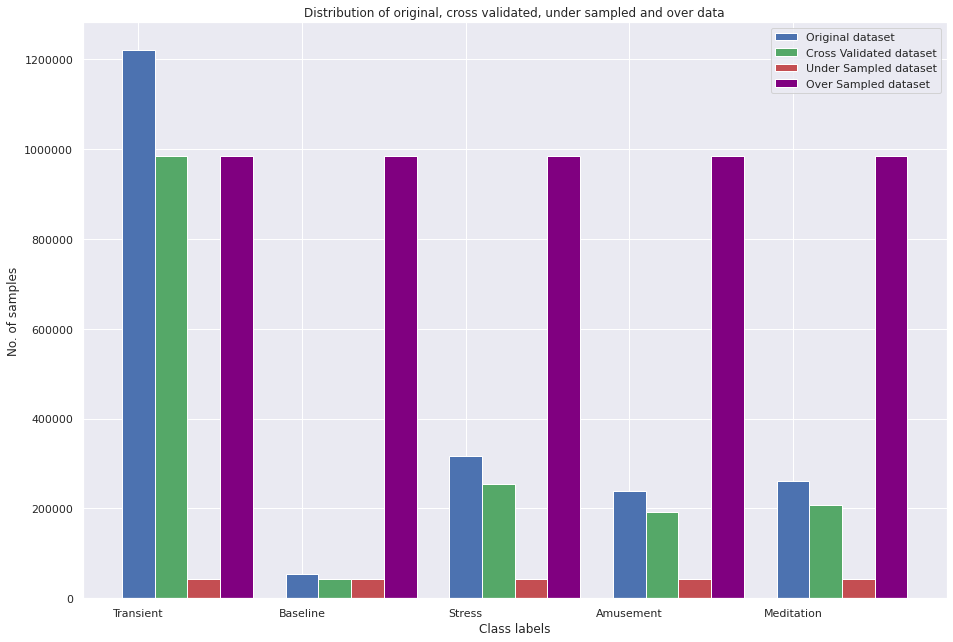


Fig 9. Distribution of original, cross validated, under sampled and over sampled data for multiclass classification

K-Fold Cross Validation helps to create k folds of a dataset, which is used as a training and testing set. K-1 folds are used for the training set, while the kth fold is used as the testing set. Fig 9. Shows the working of Stratified K -Fold Cross Validation, where the number of samples of each class label are equally divided in each k-1 fold.

Sampling is performed on each k-1 fold, where all the labels are tried to be kept equal by either under sampling or over sampling. The testing set (kth fold) is not sampled, as the data must always be new, random and unbiased for testing the model performance. Fig 9. Shows under sampled and over sampled data. The red bars represent under sampling of data, where transient, stress, amusement and meditation data is under sampled to the value of baseline data. While, the purple bars represent over sampling of data, where baseline, stress, amusement and meditation data is over sampled (by randomly choosing data points of their class to duplicate data) to the value of transient data.

The same has been shown in Fig 10. For binary classification.

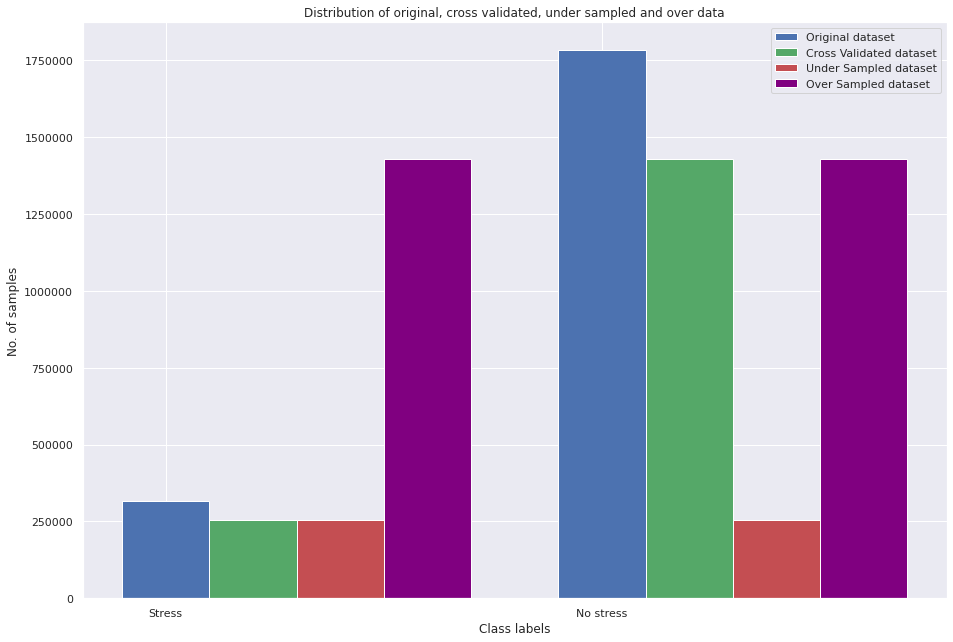


Fig 10. Distribution of original, cross validated, under sampled and over sampled data for binary classification

**Performance Metrics used**

Various performance metrics are used to evaluate the models for binary and multiclass classification. The performance metrics used are Accuracy, Precision, Recall, F1 score, Cohen’s Kappa and Precision-Recall Curve. All the metrics mainly depend on the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values.

As seen in Fig. 9 and Fig. 10, the highly imbalanced dataset is converted to balanced dataset using sampling methods, hence we have used accuracy to measure the performance of the models. It is the ratio of correctly predicted values over all the values. The formula for accuracy is given below.

Accuracy = (TP + TN)/(TP + TN + FP + FN)

Precision is the metric which determines the proportion of subjects that model predicted as stress, actually were stressed. The formula for precision is given below.

Precision = TP/(TP + FP)

Recall is the metric which determines what proportion of subjects who were actually in stress was predicted as stressed by the model.

Recall = TP/(TP + FN)

Another metrics used for evaluation is done by taking arithmetic mean of precision and recall, which is also called as F1 score. Therefore, the formula of F1 score is given below.

F1 score = (2 \* Precision \* Recall) / Precision + Recall

Cohen’s kappa is the metric which is mainly used for multiclass classification problems. The precision recall curve is used to show the relationship between precision and recall.

**Results**

Machine learning and deep learning algorithms are used to predict if a person is under stress or not. In machine learning, ensemble methods are used such as AdaBoost, ExtraTrees and XGBoost classifiers, while MLP classifier is used in deep learning. All these algorithms are compared with various parameters and have chosen the best performing algorithm of all in both binary classification and multi-classification.

Ensemble algorithms are implemented as shown in Table 5. The sampling method and cross validation method is kept constant. Stratified K-Fold Cross Validation is used, with a value of k as 5, while the sampling method used is the Random Under Sampling method.

Table 5. Performance of ensemble algorithms in multiclass and binary classification

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | | **Recall** | | **F1 Score** | | **Cohen's kappa** | **Precision Recall Curve** | **Time per k fold** | |
| **Macro** | **Wtd** | **Macro** | **Wtd** | **Macro** | **Wtd** | **Training (s)** | **Testing (s)** |
| **Multi-classification** | | | | | | | | | | | |
| AdaBoost | 41.23 | 67.25 | 77.80 | 78.63 | 41.23 | 56.38 | 26.98 | 32.21 | 78.08 | 13.66 | 5.39 |
| ExtraTrees | 98.31 | 97.49 | 98.38 | 99.41 | 98.31 | 98.32 | 99.41 | 97.25 | 100 | 11.88 | 6.08 |
| XGBoost | 97.19 | 95.89 | 97.36 | 98.94 | 97.19 | 97.34 | 97.20 | 95.44 | 99.98 | 45.69 | 5.89 |
| **Binary Classification** | | | | | | | | | | | |
| AdaBoost | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 20.93 | 31.28 | 0.61 |
| ExtraTrees | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 18.30 | 16.45 | 1.25 |
| XGBoost | 98.78 | 96.38 | 98.86 | 99.11 | 98.78 | 95.38 | 98.79 | 95.38 | 66.38 | 29.75 | 3.04 |

From Table. 5, we can observe that the Extra Tree classifier performs the best in both binary and multi-classification. AdaBoost classifier performs extremely well with binary classification, but performs poorly with multi-classification. XGBoost classifier works well with both binary and multiclass data, however not best as Extra Tree classifier.

Table 6. Performance of ExtraTreesClassifier with random over and under sampler

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sampling** | **Accuracy** | **Precision** | | **Recall** | | **F1 Score** | | **Cohen's kappa** | **Precision Recall Curve** | **Execution time (min)** |
| **Macro** | **Wtd** | **Macro** | **Wtd** | **Macro** | **Wtd** |
| **Multi-classification** | | | | | | | | | | |
| Random Under Sampler | 98.31 | 97.49 | 98.38 | 99.41 | 98.31 | 98.32 | 99.41 | 97.25 | 100 | 8 |
| Random Over Sampler | 99.49 | 99.37 | 99.47 | 99.61 | 99.47 | 99.49 | 99.47 | 99.49 | 100 | 55 |
| **Binary Classification** | | | | | | | | | | |
| Random Under Sampler | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 98.94 | 60.47 | 2 |
| Random Over Sampler | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 98.94 | 60.47 | 4 |

As Extra Tree classifier performs finer with an accuracy of 97.19%, we try to increase its performance by changing parameters. So, this classifier is run on both random under sampler and random over sampler, as shown in Table. 6. Stratified K-Fold cross validation with k=5 is kept constant. For multi-classification, over-sampler gives better performance as compared to under-sampler and has achieved an highest accuracy of 99.49%. However, the only limitation of over sampling is that it takes a long execution time compared to under sampling. For binary classification, there is no significant change between both the samplers. Both provide high accuracy of 99.99%.

Table 7. Performance of ExtraTreesClassifier with different values of k

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Value of k** | **Accuracy** | **Precision** | | **Recall** | | **F1 Score** | | **Cohen's kappa** | **Precision Recall Curve** | **Time per k fold** | |
| **Macro** | **Wtd** | **Macro** | **Wtd** | **Macro** | **Wtd** | **Training (s)** | **Testing (s)** |
| **Multi-classification** | | | | | | | | | | | |
| 5 | 98.31 | 97.49 | 98.38 | 99.41 | 98.31 | 98.32 | 99.41 | 97.25 | 100 | 11.88 | 6.08 |
| 10 | 98.35 | 97.54 | 98.42 | 99.42 | 98.35 | 98.45 | 98.36 | 97.31 | 100 | 14.25 | 3.25 |
| 20 | 98.38 | 97.58 | 98.44 | 99.43 | 98.38 | 98.48 | 98.38 | 97.35 | 100 | 14.75 | 1.73 |
| **Binary Classification** | | | | | | | | | | | |
| 5 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 98.94 | 60.47 | 16.45 | 1.25 |
| 10 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 98.94 | 60.47 | 18.44 | 0.63 |
| 20 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 98.94 | 60.47 | 18.44 | 0.63 |

In Table. 7, different values of k are implemented on Extra Trees classifier with random under sampler. For multi-classification, as the number of k increases, we can observe minimal increase in the performance. However, it does not defeat the performance of Extra Trees classifier with random over sampler (see Table. 6). While, in binary classification, the values remain constant, even when the value of k increases.

Table 8. Performance of ExtraTreesClassifier with different cross validation techniques

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Cross Validation** | **Accuracy** | **Precision** | | **Recall** | | **F1 Score** | | **Cohen's kappa** | **Precision Recall Curve** | **Time per k fold** | |
| **Macro** | **Wtd** | **Macro** | **Wtd** | **Macro** | **Wtd** | **Training (s)** | **Testing (s)** |
| **Multi-classification** | | | | | | | | | | | |
| Stratified Cross Validation | 98.35 | 97.54 | 98.42 | 99.42 | 98.35 | 98.45 | 98.36 | 97.31 | 100 | 14.25 | 3.25 |
| K Fold Cross Validation | 98.31 | 97.48 | 98.38 | 99.41 | 98.31 | 98.31 | 99.41 | 97.21 | 100 | 12.43 | 6.32 |
| Monte Carlo | 98.31 | 97.48 | 98.38 | 99.41 | 98.31 | 98.31 | 99.41 | 97.21 | 100 | 13.21 | 9.85 |
| **Binary Classification** | | | | | | | | | | | |
| Stratified Cross Validation | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 98.94 | 60.47 | 16.45 | 1.25 |
| K Fold Cross Validation | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 98.94 | 60.47 | 15.33 | 1.29 |
| Monte Carlo | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 99.99 | 98.94 | 60.47 | 16.55 | 1.48 |

Table. 8 shows the performance of Extra Trees classifier with various cross validators. All cross validations give the same performance.

Table 9. Performance of ExtraTrees on feature selection technique

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Feature Selection** | **Features** | **Accuracy** | **Precision** | | **Recall** | | **F1 Score** | | **Cohen's kappa** | **Precision Recall Curve** |
| **Macro** | **Wtd** | **Macro** | **Wtd** | **Macro** | **Wtd** |
| **Multi-classification** | | | | | | | | | | |
| Information Gain | c\_ax, c\_ay, c\_az, c\_eda, c\_temp | 98.45 | 97.70 | 98.50 | 99.42 | 98.45 | 98.53 | 98.45 | 97.46 | 100 |
| Mean Absolute Distance | c\_ax,c\_ay,c\_az, c\_ecg, c\_emg | 86.56 | 83.62 | 83.62 | 93.33 | 86.56 | 87.41 | 87.02 | 79.40 | 96.26 |
| **Binary Classification** | | | | | | | | | | |
| Information Gain | c\_ax, c\_ay, c\_az, c\_eda, c\_temp | 99.57 | 98.66 | 99.59 | 99.57 | 99.73 | 99.18 | 99.58 | 98.37 | 52.00 |
| Mean Absolute Distance | c\_ax,c\_ay,c\_az, c\_ecg, c\_emg | 89.46 | 79.24 | 93.10 | 91.92 | 89.46 | 83.30 | 90.39 | 67.06 | 66.12 |

Two feature extraction methods, information gain and mean absolute distance are used to evaluate the performance of ExtraTrees classifier, as shown in Table. 9. However, we can observe that the performance of the model gets reduced by using feature selection techniques. Hence, these observations represent that all the features are important for prediction, as with all the features we get best results.

For deep learning technique, the algorithm used is MLP classifier. Initially, MLP classifier is run with ReLu activation function and Adam solver, as shown in Table. 10, the number of nodes per hidden layer are changed to achieve better performance. The learning rate is kept constant at 0.001 and the number of epochs at 10. For multi-classification and binary classification, we can observe that as the hidden layer nodes increase, the performance of the model decreases.

Table 10. Performance of MLP classifier with different no. of hidden layers

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No. of nodes per hidden layer** | **No. of iterations**  **(epochs)** | **Accuracy** | **Precision** | | **Recall** | | **F1 Score** | | **Execution Time**  **(min)** |
| **Macro** | **Wtd** | **Macro** | **Wtd** | **Macro** | **Wtd** |
| **Multi-classification** | | | | | | | | | |
| 1 | 10 | 0.71 | 0.58 | 0.62 | 0.53 | 0.71 | 0.54 | 0.65 | 2 |
| 10 | 10 | 0.93 | 0.91 | 0.94 | 0.96 | 0.93 | 0.93 | 0.93 | 21 |
| 15 | 10 | 0.97 | 0.96 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 50 |
| 20 | 10 | 0.97 | 0.96 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 33 |
| **Binary Classification** | | | | | | | | | |
| 10 | 10 | 0.98 | 0.96 | 0.98 | 0.97 | 0.98 | 0.97 | 0.98 | 11 |
| 20 | 10 | 0.97 | 0.93 | 0.97 | 0.93 | 0.97 | 0.93 | 0.97 | 9 |

The best performance with the given conditions are shown for multi-classification when hidden layers node is 15, with an accuracy of 97%. While, for binary classification at 10 hidden layer nodes, the accuracy achieved is 98%.

Table 11. Performance of MLP classifier with different solvers and hidden layers

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Solver** | **No. of hidden layers** | **Accuracy** | **Precision** | | **Recall** | | **F1 Score** | | **Execution Time**  **(min)** |
| **Macro** | **Wtd** | **Macro** | **Wtd** | **Macro** | **Wtd** |
| **Multi-classification** | | | | | | | | | |
| Adam | 10 | 0.93 | 0.91 | 0.94 | 0.96 | 0.93 | 0.93 | 0.93 | 21 |
| Adam | 15 | 0.97 | 0.96 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 50 |
| SGD | 10 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 20 |
| SGD | 15 | 0.94 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 32 |
| **Binary Classification** | | | | | | | | | |
| Adam | 10 | 0.98 | 0.96 | 0.98 | 0.97 | 0.98 | 0.97 | 0.98 | 11 |
| Adam | 15 | 0.98 | 0.96 | 0.98 | 0.97 | 0.98 | 0.96 | 0.98 | 9 |
| SGD | 10 | 0.95 | 0.91 | 0.95 | 0.89 | 0.95 | 0.90 | 0.95 | 2 |
| SGD | 15 | 0.95 | 0.90 | 0.95 | 0.91 | 0.95 | 0.91 | 9.95 | 3 |

In Table. 11, the performance of the MLP classifier is calculated by changing the solver, and keeping other parameters constant. The number of epochs is taken as 10 and the learning rate is kept at 0.001. For both binary and multi-classification, the best results are found with Adam solver and number of hidden layers as 15, with accuracy of 97% and 98% respectively.

Table 12. Performance of MLP classifier with different learning rate and no. of epochs

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Learning Rate** | **No. of iterations**  **(epochs)** | **Accuracy** | **Precision** | | **Recall** | | **F1 Score** | | **Execution Time**  **(min)** |
| **Macro** | **Wtd** | **Macro** | **Wtd** | **Macro** | **Wtd** |
| **Multi-classification** | | | | | | | | | |
| 0.001 | 10 | 0.97 | 0.96 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 50 |
| 0.005 | 10 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 | 19 |
| 0.001 | 20 | 0.95 | 0.95 | 0.96 | 0.97 | 0.96 | 0.96 | 0.96 | 47 |
| **Binary classification** | | | | | | | | | |
| 0.001 | 10 | 0.98 | 0.96 | 0.98 | 0.97 | 0.98 | 0.97 | 0.98 | 11 |
| 0.005 | 10 | 0.99 | 0.97 | 0.99 | 0.98 | 0.99 | 0.97 | 0.99 | 14 |
| 0.001 | 20 | 0.97 | 0.93 | 0.97 | 0.96 | 0.97 | 0.95 | 0.97 | 15 |

The next parameter changed is the learning rate, as shown in Table. 12. As the learning rate or number of epochs increases in multi-classification, the performance of the model decreases. While, in binary class, as the learning rate is increased, the model gives best accuracy of 99.64%. However, as the number of epochs is increased, the performance reduces.



Fig 11. Confusion Matrix of Extra Trees classifier in Multi-classification

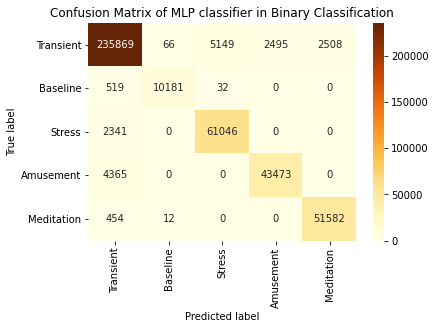


Fig 12. Confusion Matrix of MLP classifier in Multi-classification

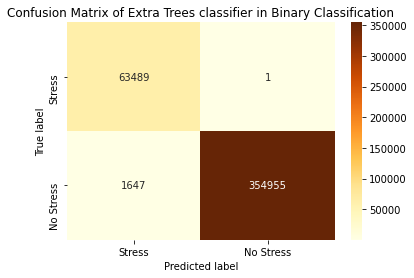


Fig 13. Confusion Matrix of Extra Trees classifier in binary classification

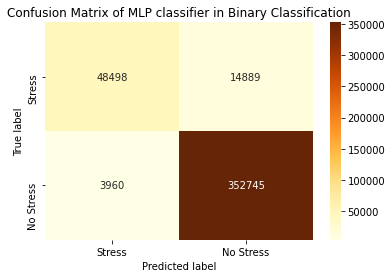


Fig 14. Confusion Matrix of MLP classifier in binary classification

Fig. 11, fig. 12, fig. 13 and fig. 14 shows the confusion matrix for the best models in binary classification and multi-classification. It is observable that the predicted true labels are high in count, hence the performance of Extra Trees and MLP classifiers is really good.

Therefore, we deduce that the Extra Tree classifier gives an superior accuracy of 99.99% for binary classification, when stratified k-fold cross validation (k=5) and random under sampler are used for generalization of model, while in multi-classification the Extra Tree classifier gives an high accuracy of 99.64% with when stratified k-fold cross validation (k=5) and random over sampler are used for generalization of mode. Moreover, the deep learning model MLP classifier also gives fine accuracy of 98.64% in binary classification, with ReLu activation function, Adam solver, number of epochs as 10 and learning rate as 0.005. On the other hand, for multi-classification MLP classifier gives accuracy of 97%, with ReLu activation function, Adam solver, number of epochs as 15 and learning rate as 0.001.

**Conclusion**

The proposed research work is done on the multimodal dataset WESAD to predict if the subject is undergoing stress or not. The WESAD dataset was visualized and cleaned to be used for machine learning and deep learning algorithms. The dataset was used for multi-classification (transient, stress, baseline, amusement, meditation) and binary classification (stress or no stress). Stratified K Fold Cross Validation was used for generalization of the model keeping k as 5. Sampling was used to make the highly imbalanced dataset to balanced dataset. Ensemble machine learning methods such as AdaBoost, ExtraaTrees and XGBoost classifiers were used as predictive models. Of the three machine learning models, ExtraTrees classifier gave the best performance with an accuracy of 99.99% for binary classification and 99.49% for multi-classification. While, in deep learning MLP classifiers were used and gave performance of 98.64% for binary classification and 97.00% for multi-classification. Various parameters were changed and tweaked in order to achieve best performance.

**Benchmarking Table**

Table 17. Benchmarking Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Paper** | **Dataset** | **Technique** | **Classification** | **Accuracy** | **F1 Score** | **Year** |
| Our paper | WESAD | ExtraTrees | Binary | 99.99 | 99.99 | 2021 |
| Multiclass | 99.49 | 99.47 |
| XGBoost | Binary | 98.78 | 98.79 |
| Multiclass | 97.19 | 97.20 |
| MLPClassifier | Binary | 98.64 | 99.00 |
| Multiclass | 97.00 | 97.00 |
| [2] Stress Detection with ML and DL using Multimodal Physiological Data | WESAD | Kernel SVM | Binary | 93.20 | 92.31 | 2020 |
| Multiclass | 81.65 | 73.57 |
| ANN | Binary | 95.21 | 94.24 |
| Multiclass | 84.23 | 78.71 |
| [3]User Independent Human Stress Detection | WESAD | Extra Trees | Binary | 93.00 | 93.00 | 2020 |
| Multiclass | 83.00 | 94.00 |
| [4]Dimension-raising Processing Framework for One-dimensional Time Series and its Application in Affect Detection | WESAD | CNN | Multiclass | 97.62 | 97.91 | 2020 |
| [5]Stress Detection With Single PPG Sensor by Orchestrating Multiple Denoising and Peak-Detecting Methods | WESAD | OMPD (integrated model) | Binary | 96.50 | 93.36 | 2021 |
| Multiclass | 74.20 | 61.70 |
| [6]Evaluating KNN Performance on WESAD dataset | WESAD | KNN | Multiclass | - | 90.00 | 2020 |
| [7]Feature Selection Framework for XGBoost Based on Electrodermal Activity in Stress Detection | WESAD | XGBoost | Mutliclass | - | 92.38 | 2019 |

**References**

[1]Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger and Kristof Van Laerhoven. 2018. Introducing WESAD, a multimodal dataset for Wearable Stress and Affect Detection. In 2018 International Conference on Multimodal Interaction (ICMI â€™18), October 16â€“20, 2018, Boulder, CO, USA. ACM, New York, NY, USA, 9 pages.

[2]P. Bobade and M. Vani, "Stress Detection with Machine Learning and Deep Learning using Multimodal Physiological Data," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 51-57, doi: 10.1109/ICIRCA48905.2020.9183244.

[3]R. Murugappan, J. J. Bosco, K. Eswaran, P. Vijay and V. Vijayaraghavan, "User Independent Human Stress Detection," 2020 IEEE 10th International Conference on Intelligent Systems (IS), Varna, Bulgaria, 2020, pp. 490-497, doi: 10.1109/IS48319.2020.9199928.

[4]Z. Ye, F. Deng, J. Zhao and M. Lu, "Dimension-raising Processing Framework for One-dimensional Time Series and its Application in Affect Detection," 2020 IEEE 16th International Conference on Control & Automation (ICCA), Singapore, 2020, pp. 307-311, doi: 10.1109/ICCA51439.2020.9264431.

[5]S. Heo, S. Kwon and J. Lee, "Stress Detection With Single PPG Sensor by Orchestrating Multiple Denoising and Peak-Detecting Methods," in IEEE Access, vol. 9, pp. 47777-47785, 2021, doi: 10.1109/ACCESS.2021.3060441.

[6]D. Bajpai and L. He, "Evaluating KNN Performance on WESAD Dataset," 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), Bhimtal, India, 2020, pp. 60-62, doi: 10.1109/CICN49253.2020.9242568.

[7]C. Hsieh, Y. Chen, W. Beh and A. A. Wu, "Feature Selection Framework for XGBoost Based on Electrodermal Activity in Stress Detection," 2019 IEEE International Workshop on Signal Processing Systems (SiPS), Nanjing, China, 2019, pp. 330-335, doi: 10.1109/SiPS47522.2019.9020321.

[8]M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo and J. Santos, "Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier]," in IEEE Computational Intelligence Magazine, vol. 13, no. 4, pp. 59-76, Nov. 2018, doi: 10.1109/MCI.2018.2866730.

[9]Hossin, Mohammad & M.N, Sulaiman. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. International Journal of Data Mining & Knowledge Management Process. 5. 01-11. 10.5121/ijdkp.2015.5201.

[10]Dietterich T.G. (2000) Ensemble Methods in Machine Learning. In: Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science, vol 1857. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/3-540-45014-9_1>

[11]Zhen Zhang, Weimin Shao and Hong Zhang, "A learning algorithm for multilayer perceptron as classifier," IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339), 1999, pp. 1681-1684 vol.3, doi: 10.1109/IJCNN.1999.832627.

[12]Miao, Jiaju and W. Zhu. “Precision-Recall Curve (PRC) Classification Trees.” *ArXiv* abs/2011.07640 (2020): n. Pag.