History of Medicine, 2024, 10(2): 117–128

DOI: 10.17720/2409-5834.v10.2.2024.15

Automated Stress Detection and Management Using Wearable Sensor Data: A Machine Learning Approach

Dr. B. Sreenaiah[1], Revanth Kannekanti[2], Manisha Poppadi[2], Akhila Chinthanuri[2]

[1]Professor, [2]UG Scholar, [1,2]Department of Computer Science and Engineering
[1,2]Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Hyderabad, Telangana.

ABSTRACT

Stress has become a widespread issue that affects persons of all backgrounds in today's fast-paced society. Recognizing and effectively handling stress is essential for preserving both physical and mental health. The demand for automated stress detection originates from the increasing recognition of the influence of stress on both health and productivity. Conventional methods for evaluating stress typically depend on questionnaires completed by individuals or assessments conducted by clinicians, both of which can be influenced by personal opinions and take up a significant amount of time. Thus, through the utilization of machine learning models, it becomes feasible to offer prompt and precise input to folks, empowering them to proactively regulate their stress levels. The main objective of this project is to create machine learning models that can proficiently assess physiological data collected from wearable sensors (such as the ones available in the WESAD dataset) in order to precisely identify and categorize states of stress impact. This entails generating resilient characteristics from the unprocessed sensor data and developing models with the ability to extrapolate across various individuals and situations. The utilization of machine learning techniques in combination with physiological measures indicates a substantial improvement compared to traditional statistical methods. The utilization of machine learning for stress impact identification represents a paradigm shift in our approach to stress management. This suggested system utilizes wearable sensor data and powerful algorithms to offer users timely insights into their stress levels. This empowers them to proactively take steps towards improving their mental and physical well-being.

Keywords: WESAD dataset, Stress detection, predictive analytics, exploratory data analysis, machine learning.

1. INTRODUCTION

In the current scenario, mental stress has become a social problem. It affects the daily routine work and economy ofan individual and a nation as well. Stressfulin the working style of human age from twenty-five to forty has the effectiveness of stress in their life. Stress has become the common part of daily life which most people struggle in different stages of life. Mental stress has grown to be a social difficulty and could turn out to be a cause of practical incapacity during hobbies work. Mental stress or pressure has been recognized as one of the major contributing variables that prompt different illnesses. The term pressure is characterized into two sections, for example, positive or negative. Positive pressure consistently searches for a chance, development like improvement in execution, while negative pressure developes more serious conditions. Mental stress is one of the contributing factors to health problems. It is defined as the human body's response to mental, physical, and emotional stimuli, as controlled by the sympathetic nervous system (SNS) and the hypothalamus–pituitary–adrenocortical axis (HPA axis) [1]. This expression can be applied to both internal (personality structure) and external (problem solving) issues, resulting in a variety of physiological and negative emotional alterations [2]. Acute stress, episodic stress, and chronic stress are the three types of stress identified in the literature [3]. Acute stress is caused by a brief period of exposure and is not damaging. In our contemporary society, the pervasive issue of stress has become a matter of profound concern, impacting individuals across diverse demographics. Recognizing and effectively managing stress is imperative for the maintenance of both physical and mental well-being. The urgency for automated stress affect detection has emerged due to an increasing awareness of stress's detrimental effects on health and productivity. Conventional methods of stress assessment often rely on subjective self-reported questionnaires or time-consuming clinical evaluations. Consequently, the adoption of machine learning models becomes crucial, offering the potential to deliver timely and precise feedback to individuals, thereby enabling them to take proactive measures in stress management.

The primary challenge in this research lies in the development of machine learning models capable of proficiently analyzing physiological data acquired from wearable sensors, exemplified by the WESAD dataset, to accurately detect and classify stress affect states. This intricate task involves the creation of robust features extracted from the raw sensor data and the training of models with the capacity to generalize across diverse individuals and scenarios. Noteworthy is the fact that while physiological measures have been utilized previously, the incorporation of machine learning techniques represents a substantial advancement beyond conventional statistical approaches. The application of machine learning for stress affect detection marks a transformative paradigm shift in our approach to stress management. By leveraging the wealth of information provided by wearable sensor data and deploying sophisticated algorithms, the proposed system aims to furnish individuals with timely insights into their stress levels. This empowerment facilitates proactive steps towards enhancing both mental and physical well-being, underscoring the significance of integrating cutting-edge technology in addressing contemporary health challenges.

2. LITERATURE SURVEY

Involution, also known as “Nei Juan” in Chinese pinyin, refers to situations in work and study where individuals put in extra effort, but that effort does not yield a proportional outcome [1]. An influential article [2] written by a Chinese mother narrated her struggles to provide her child with top educational opportunities. Despite earning a monthly salary of RMB 30,000, which is considered relatively high in China, it proved insufficient due to the pressure to enroll her daughter in numerous extracurricular classes in order to excel in all exams, even if those classes covered material beyond the scope of the regular school curriculum. The combined pressures from schools, society, and even families have given rise to a highly stressed generation. Research has indicated that high levels of stress are linked to lower levels of well-being and reduced quality of life. Prolonged exposure to stress can result in severe mental health issues such as anxiety and depression [3].

According to a report by [4], approximately 1100 students out of 100,000 commit suicide each year. Thus, monitoring stress levels can be extremely beneficial for universities and families in supporting students’ academic performance and enhancing their overall well-being. Wagh et al. [5] employed Support Vector Machine (SVM), k-Nearest Neighbor (kNN), and Decision Tree (DT) algorithms to classify positive, neutral, and negative emotions using time and time-frequency domain features extracted from various channels of electroencephalogram (EEG) data. Vijayakumar et al. [6] developed a 1D convolutional neural network (CNN) to classify arousal, valence, and liking based on peripheral physiological signals, including blood volume pressure (BVP), horizontal electrooculogram (hEOG), vertical electrooculogram (vEOG), trapezius electromyogram (tEMG), zygomaticus electromyogram (zEMG), respiration rate (RSP), and skin temperature (SKT) data.

Miao et al. [8] proposed a parallel spatial-temporal 3D deep residual learning framework called MFBPST-3D-DRLF for emotion recognition using EEG signals. This framework utilized multiple frequency bands (delta, theta, alpha, beta, gamma) of EEG signals to generate a 3D representation of features, which were then trained using a 3D deep residual CNN model. It achieved a classification accuracy of 96.67% on the SEED dataset (positive, neutral, and negative emotions) and 88.21% on the SEED-IV dataset (happy, fear, sad, and neutral).

Montero Quispe et al. [9] employed a novel self-supervised learning approach for emotion recognition, consisting of two stages: self-supervised pre-training and emotion recognition model training. In the pre-training stage, the model learned to recognize six signal variants generated by applying noise, scaling, negation, flipping, permuting, and time warping to the original data.

Tang et al. [11] conducted experiments on emotion recognition using EEG data with a proposed model called Spatial-Temporal Information Learning Network (STILN), which achieved an accuracy of 68.31% for arousal and 67.52% for valence. Choi et al. [12] proposed an attention-LRCN model that reduced motion artifacts in collected photoplethysmography (PPG) data. The registration form was created using Wenjuanxing [13], a platform that facilitates the design of questionnaires, exams, voting systems, and rating forms. PPG data were collected using the Polar Verity Sense (PVS) [14], ECG data were collected using the BMD101 device [15], and EEG data were collected using NeuroSky's MindWave Mobile 2 (MV2) [16]. The sampling rates for data collection were set at 55 Hz for PVS, 512 Hz for BMD101, and 512 Hz for MV2. It is important to note that data collection did not occur during the participant's self-reporting periods.

3. PROPOSED METHDOLOGY

This work outlines a machine learning project focused on stress detection using the WESAD dataset. The project involves the following steps:

- Dataset Loading: The initial step is to load the WESAD dataset, specifically the S6.pkl file, using the pickle library. The data is stored in the 's6_data' variable, which contains various physiological signals recorded from both chest and wrist sensors.
- Signal Extraction: The script extracts specific signals such as acceleration (ACC), electrocardiogram (ECG), electromyogram (EMG), electrodermal activity (EDA), temperature (Temp), and respiration (Resp) from both chest and wrist sensors. These signals are stored in individual variables for further analysis.
- Data Combination and DataFrame Creation: The extracted signals are combined to form a NumPy array, which is then converted into a Pandas DataFrame named 'df.' This DataFrame consists of features such as chest and wrist accelerometer values, ECG, EMG, EDA, temperature, respiration, and the corresponding stress labels.
- Data Analysis and Cleaning: Basic statistical information about the DataFrame is obtained using the 'describe' and 'info' methods. Outliers are identified and removed using the Interquartile Range (IQR) method, resulting in the 'df_out' DataFrame. Mean normalization is applied to the data, ensuring zero mean for each feature.
- Feature and Target Variable Separation: The features (X) and the target variable (y) are separated from the DataFrame. The target variable represents stress labels, while the features include various physiological signals.
- Train-Test Split: The dataset is split into training and testing sets using the 'train_test_split' method from Scikit-Learn. This step is crucial for evaluating the models on unseen data.
- Logistic Regression Model: A Logistic Regression model is trained on the training set, and predictions are made on the test set. The accuracy and classification report, including precision, recall, and F1-score, are displayed. Additionally, a confusion matrix is visualized.

- Decision Tree Classifier: A Decision Tree Classifier is implemented and trained on the training set. Similar to the Logistic Regression model, accuracy and classification report metrics are presented, along with a confusion matrix.
- Conclusion: In summary, the project involves loading, preprocessing, and analyzing physiological signals from the WESAD dataset. The main focus is on stress detection using two different machine learning models: Logistic Regression and Decision Tree Classifier. The performance of these models is evaluated using various metrics, providing insights into their effectiveness in predicting stress levels. Visualizations, such as confusion matrices, enhance the interpretation of model performance.

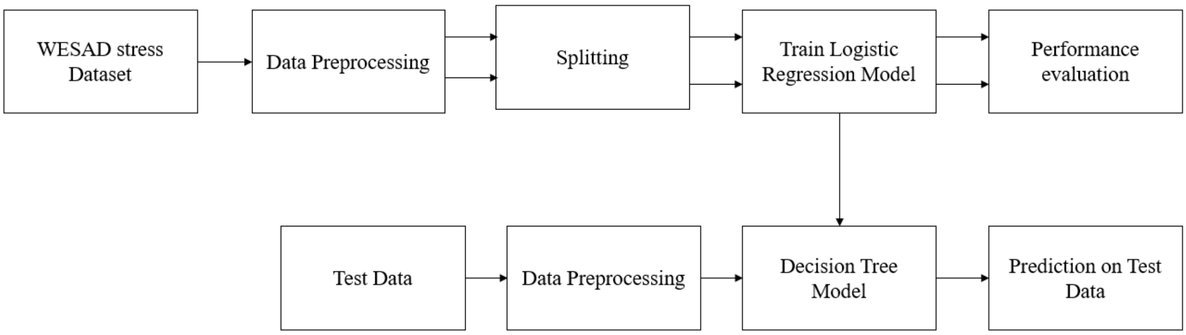


Figure 1: Block diagram of proposed model.

3.1 Data Preprocessing

Data Cleaning: Remove any missing or erroneous data points that -may compromise the integrity of the dataset.

Address outliers or anomalies in the physiological sensor readings that could distort the analysis.

Normalization: Normalize the sensor data to ensure that all features have a consistent scale. This is essential for the proper functioning of DNNs, as it facilitates convergence during the training process.

Standardize numerical values to have a mean of 0 and a standard deviation of 1, promoting uniformity across different features.

Temporal Alignment: Since wearable sensors may record data at varying intervals, align the temporal aspect of the dataset to a standardized time scale. This ensures uniformity and coherence in the time series data.

Feature Extraction: Extract relevant features from the raw sensor data that are indicative of stress affect states. These could include heart rate variability, skin conductance, and accelerometry features.

Utilize signal processing techniques to derive time or frequency domain features that capture essential patterns in the physiological signals.

Dimensionality Reduction: Employ dimensionality reduction techniques, such as Principal Component Analysis (PCA), to reduce the complexity of the dataset while retaining the most critical information. This is especially important when dealing with a large number of features.

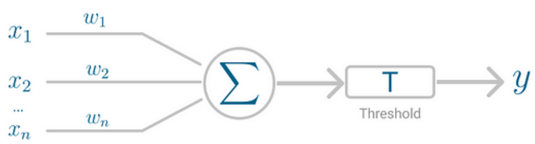
Handling Imbalances: Check for class imbalances in the stress affect labels and employ strategies such as oversampling or undersampling to address this issue. Balancing the dataset ensures that the model is not biased towards the majority class.

Data Splitting: Divide the dataset into training, validation, and test sets. This is crucial for evaluating the model's performance on unseen data and preventing overfitting.

Data Augmentation (Optional): If the dataset is limited, consider data augmentation techniques to artificially increase the diversity of the training set. This can enhance the model's ability to generalize to different scenarios.

3.2 DNN

Although today the Perceptron is widely recognized as an algorithm, it was initially intended as an image recognition machine. It gets its name from performing the human-like function of perception, seeing, and recognizing images. Interest has been centered on the idea of a machine which would be capable of conceptualizing inputs impinging directly from the physical environment of light, sound, temperature, etc. — the "phenomenal world" with which we are all familiar — rather than requiring the intervention of a human agent to digest and code the necessary information. Rosenblatt's perceptron machine relied on a basic unit of computation, the neuron. Just like in previous models, each neuron has a cell that receives a series of pairs of inputs and weights. The major difference in Rosenblatt's model is that inputs are combined in a weighted sum and, if the weighted sum exceeds a predefined threshold, the neuron fires and produces an output.



$$y = \begin{cases} 1, \text{if } \overbrace{\sum_i w_i x_i}^{\text{weighted sum}} - \underbrace{T}_{\text{threshold}} > 0 \\ 0, \text{otherwise} \end{cases}$$

Fig. 2: Perceptron neuron model (left) and threshold logic (right).

Threshold T represents the activation function. If the weighted sum of the inputs is greater than zero the neuron outputs the value 1, otherwise the output value is zero.

Perceptron for Binary Classification

With this discrete output, controlled by the activation function, the perceptron can be used as a binary classification model, defining a linear decision boundary.

It finds the separating hyperplane that minimizes the distance between misclassified points and the decision boundary. The perceptron loss function is defined as below:

$$\underbrace{D(w,c)}_{\text{distance}} = -\sum_{\underbrace{i \in \mathrm{M}}_{\text{misclassified observations}}} \overbrace{y_i}^{\text{output}} (x_i w_i + c)$$

To minimize this distance, perceptron uses stochastic gradient descent (SGD) as the optimization function. If the data is linearly separable, it is guaranteed that SGD will converge in a finite number of steps. The last piece that Perceptron needs is the activation function, the function that determines if the neuron will fire or not. Initial Perceptron models used sigmoid function, and just by looking at its shape, it makes a lot of sense! The sigmoid function maps any real input to a value that is either 0 or 1 and encodes a non-linear function. The neuron can receive negative numbers as input, and it will still be able to produce an output that is either 0 or 1.

But, if you look at Deep Learning papers and algorithms from the last decade, you'll see the most of them use the Rectified Linear Unit (ReLU) as the neuron's activation function. The reason why ReLU became more adopted is that it allows better optimization using SGD, more efficient computation and is scale-invariant, meaning, its characteristics are not affected by the scale of the input. The neuron receives inputs and picks an initial set of weights random. These are combined in weighted sum and then ReLU, the activation function, determines the value of the output.

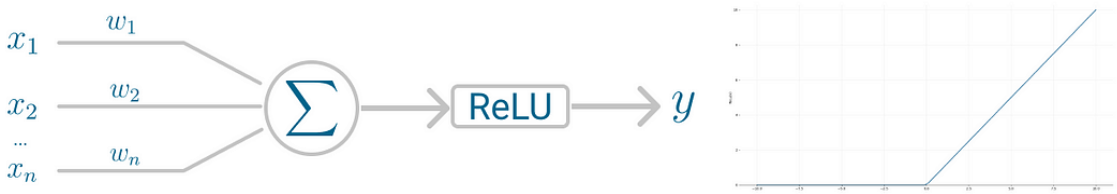


Fig. 3: Perceptron neuron model (left) and activation function (right).

Perceptron uses SGD to find, or you might say learn, the set of weight that minimizes the distance between the misclassified points and the decision boundary. Once SGD converges, the dataset is separated into two regions by a linear hyperplane. Although it was said the Perceptron could represent any circuit and logic, the biggest criticism was that it couldn't represent the XOR gate, exclusive OR, where the gate only returns 1 if the inputs are different. This was proved almost a decade later and highlights the fact that Perceptron, with only one neuron, can't be applied to non-linear data.

4.3.2 DNN

The DNN was developed to tackle this limitation. It is a neural network where the mapping between inputs and output is non-linear. A DNN has input and output layers, and one or more hidden layers with many neurons stacked together. And while in the Perceptron the neuron must have an activation function that imposes a threshold, like ReLU or sigmoid, neurons in a DNN can use any arbitrary activation function. DNN falls under the category of feedforward algorithms, because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron. But the difference is that each linear combination is propagated to the next layer. Each layer is feeding the next one with the result of their computation, their internal representation of the data. This goes all the way through the hidden layers to the output layer.

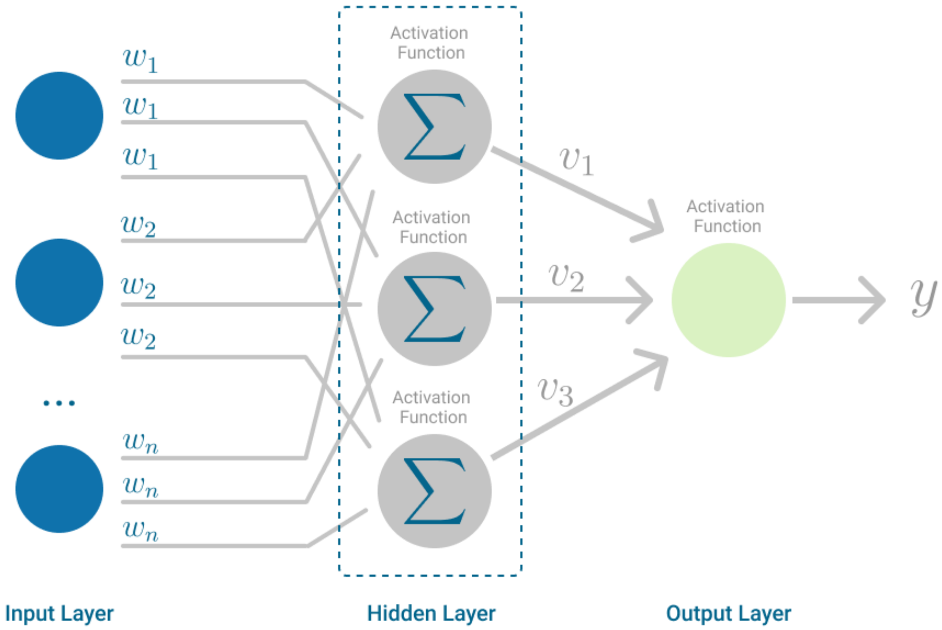


Fig. 4: Architecture of DNN.

If the algorithm only computed the weighted sums in each neuron, propagated results to the output layer, and stopped there, it wouldn't be able to learn the weights that minimize the cost function. If the algorithm only computed one iteration, there would be no actual learning. This is where Backpropagation comes into play.

Backpropagation: Backpropagation is the learning mechanism that allows the DNN to iteratively adjust the weights in the network, with the goal of minimizing the cost function. There is one hard requirement for backpropagation to work properly. The function that combines inputs and weights in a neuron, for instance the weighted sum, and the threshold function, for instance ReLU, must be differentiable. These functions must have a bounded derivative because Gradient Descent is typically the optimization function used in DNN. In each iteration, after the weighted sums are forwarded through all layers, the gradient of the Mean Squared Error is computed across all input and output pairs. Then, to propagate it back, the weights of the first hidden layer are updated with the value of the gradient. That's how the weights are propagated back to the starting point of the neural network. One iteration of Gradient Descent is defined as follows:

$$\underbrace{\Delta_w(t)}_{\text{Gradient Current Iteration}} = -\overbrace{\varepsilon}^{\text{Bias}} \frac{\overbrace{dE}^{\text{Error}}}{\underbrace{dw_{(t)}}_{\text{Weight vector}}} + \overbrace{\alpha}^{\text{Learning Rate}} \underbrace{\Delta_{w(t-1)}}_{\text{Gradient Previous Iteration}}$$

This process keeps going until gradient for each input-output pair has converged, meaning the newly computed gradient hasn't changed more than a specified convergence threshold, compared to the previous iteration.

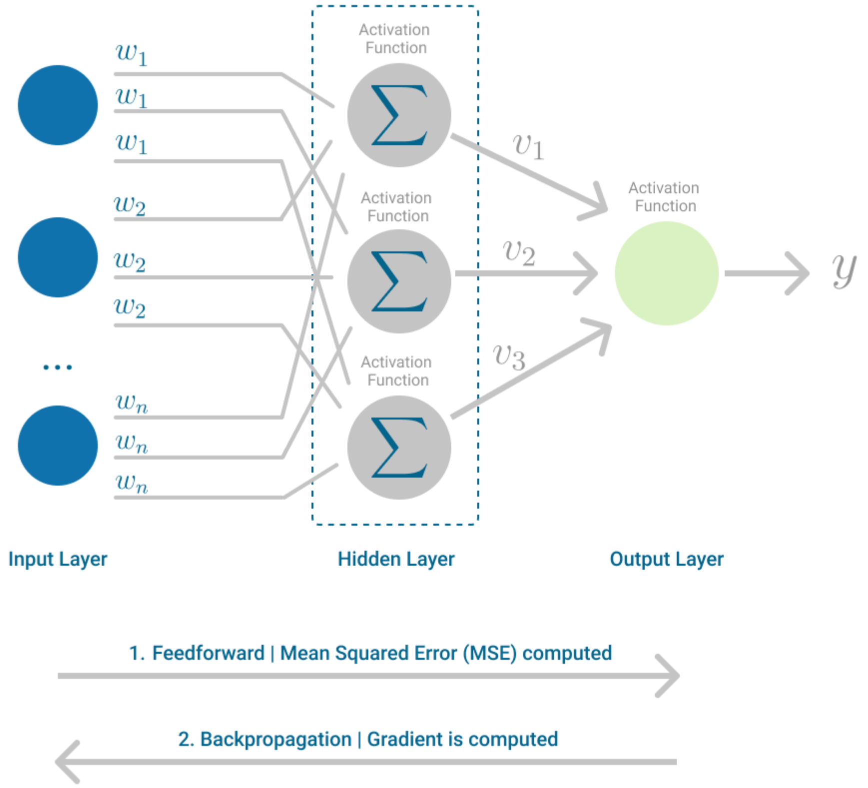


Fig. 5: DNN, highlighting the Feedforward and Backpropagation steps.

4. RESULTS AND DISCUSSION

Dataset Description

The dataset is WESAD (Wearable Stress and Affect Detection), which is a multimodal dataset designed for stress and affect recognition. The dataset includes physiological signals from both chest-worn and wrist-worn sensors.

- Here's a breakdown of the key components of your dataset:
- Physiological Signals (Chest):
- ACC (Accelerometer): Three-dimensional accelerometer data with X, Y, and Z components.
- ECG (Electrocardiogram): Electrocardiogram data.
- EMG (Electromyogram): Electromyogram data.
- EDA (Electrodermal Activity): Electrodermal activity data.
- Temp (Temperature): Temperature data.
- Resp (Respiration): Respiration data.
- Physiological Signals (Wrist):
- ACC (Accelerometer): Three-dimensional accelerometer data with X, Y, and Z components.
- BVP (Blood Volume Pulse): Blood volume pulse data.
- EDA (Electrodermal Activity): Electrodermal activity data.
- TEMP (Temperature): Temperature data.
- Other Information:
- label: An array of labels indicating the presence of stress or affect.
- subject: The subject or participant identifier (e.g., 'S6').

Results and Description

Figure 2 provides insights into the distribution of labels within the specified column. It allows users to visually assess the balance or imbalance of the dataset concerning the target variable.

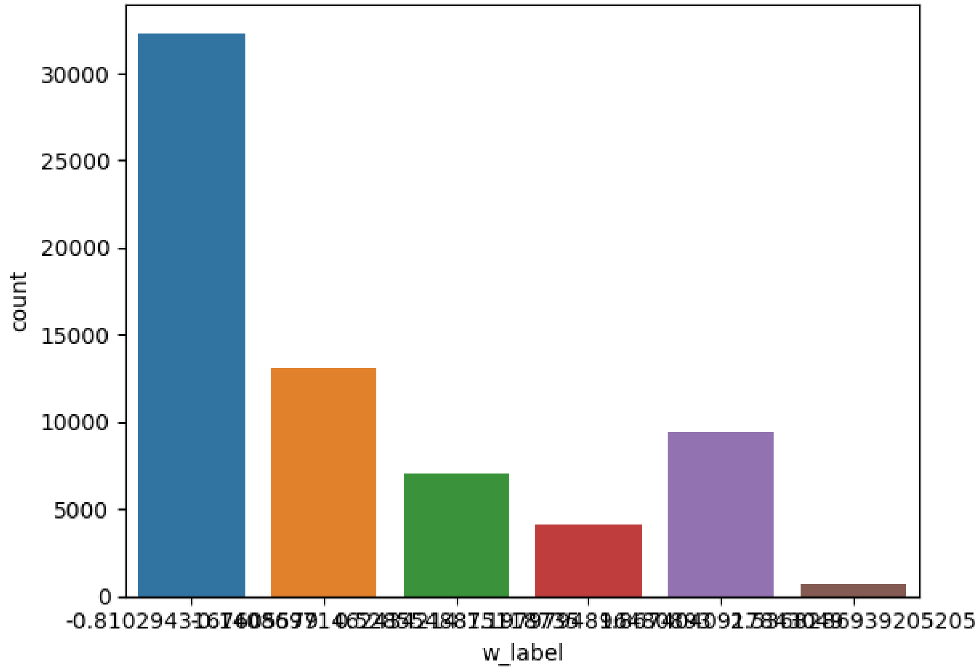


Figure 2: Displays the count plot for w_label column.

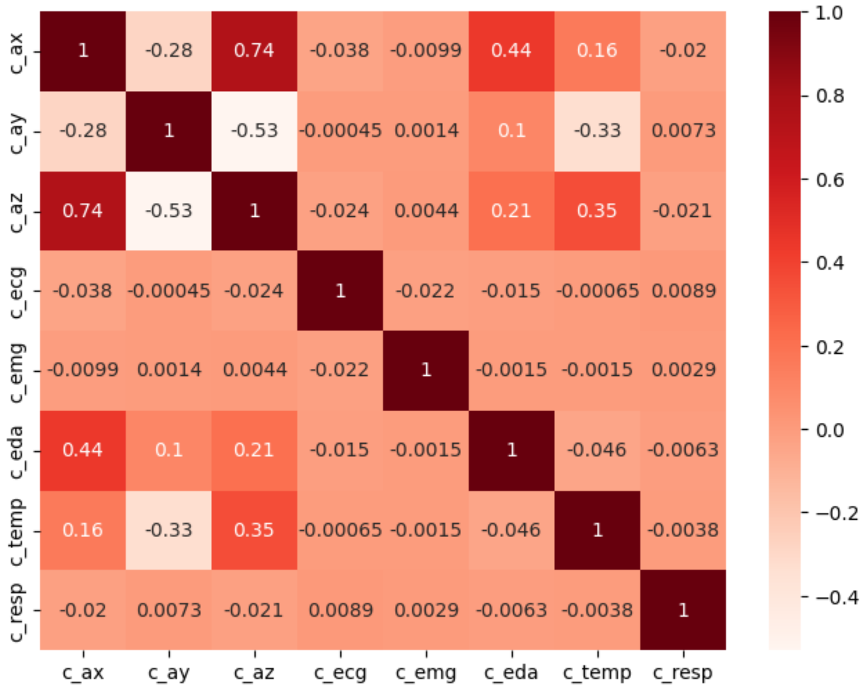


Figure 3: Displays the heatmap on data correlation of normalized dataset.

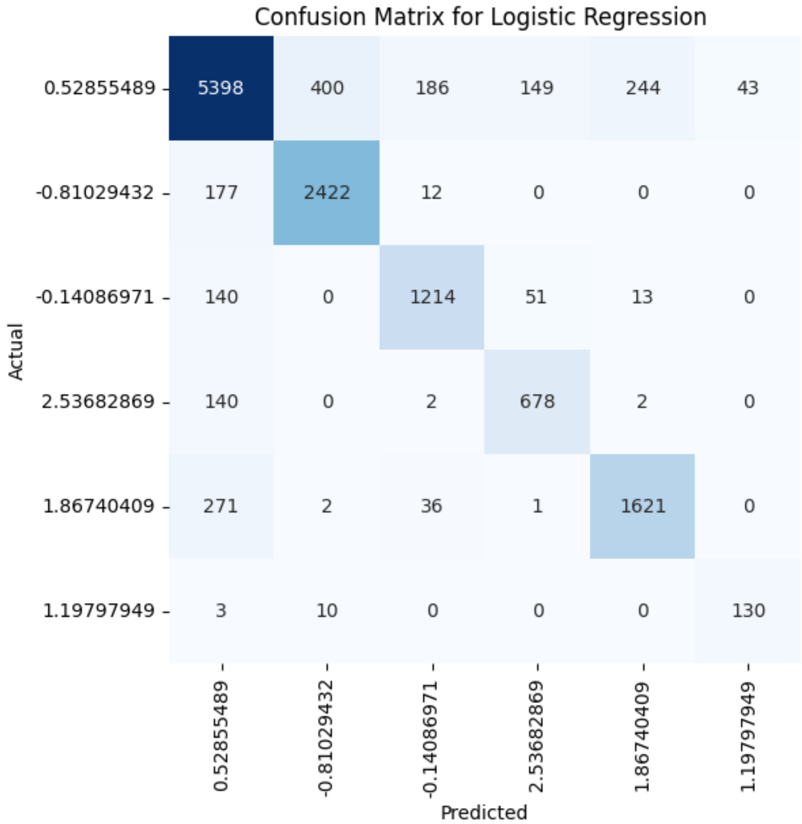


Figure 4: Displays the plot of Confusion matrix for Logistic Regression model.

Figure 3 displays a heatmap illustrating the correlation between different features in the normalized WESAD dataset. The heatmap can help identify patterns and relationships between variables. High correlation values suggest strong associations, while low values may indicate weaker or no associations. Figure 4 represents a plot of the confusion matrix for the Logistic Regression model. The confusion matrix is a table that outlines the model's performance in terms of true positives, true negatives, false positives, and false negatives. This graphical representation aids in assessing the classification accuracy of the Logistic Regression model. Figure 5 depicts a plot of the confusion matrix for the Decision Tree model. Similar to Figure 4, this visualization provides a detailed breakdown of the model's classification performance, allowing for an evaluation of its effectiveness in making accurate predictions on the WESAD dataset.

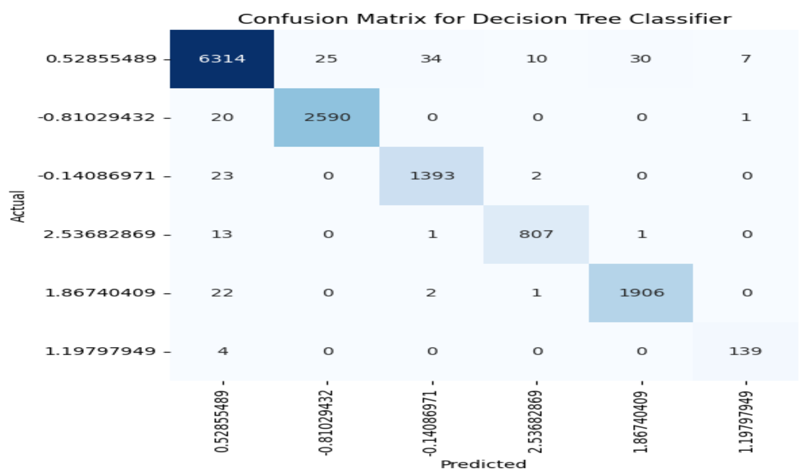


Figure 5: Displays the plot of Confusion matrix for Decision Tree model.

Table 1: Performance comparison of quality metrics obtained using LR and decision tree classifier model.

Model	Logistic Regression	Decision Tree
Accuracy (%)	0.85	0.98
Precision (%)	0.86	0.98
Recall (%)	0.85	0.98
F1-score (%)	0.85	0.98

For the Logistic Regression model:

- The Accuracy is 0.85, indicating the accuracy between the actual and predicted values
- The Precision is 0.86, suggesting that, on average Precision between the actual and predicted values.
- The Recall is 0.85, suggesting that, on average Recall between the actual and predicted values.
- The F1-score is 0.85, representing the average F1-score between the actual and predicted values.

For the Decision Tree model:

- The Accuracy is 0.98, indicating the accuracy between the actual and predicted values.
- The Precision is 0.98, suggesting that, on average Precision between the actual and predicted values.
- The Recall is 0.98, suggesting that, on average Recall between the actual and predicted values.
- The F1-score is 0.98, representing the average F1-score between the actual and predicted values.

5. CONCLUSION

In conclusion, the utilization of machine learning models for stress affect detection, particularly through the analysis of physiological data from wearable sensors like the WESAD dataset, marks a pivotal advancement in addressing the pervasive issue of stress in today's fast-paced society. Traditional methods reliant on self-reported questionnaires or clinical assessments are often subjective and time-consuming. Therefore, the incorporation of machine learning enables the development of efficient and timely stress detection systems, providing individuals with actionable insights for proactive stress management. The primary challenge lies in crafting robust features from raw sensor data and training models capable of generalizing across diverse individuals and scenarios. By leveraging advanced algorithms, this research strives to transcend conventional statistical techniques, contributing to a transformative shift in the paradigm of stress management. The proposed system not only enhances accuracy in stress affect classification but also empowers individuals with valuable information about their stress levels. The significance of integrating machine learning with physiological measures underscores the potential for a more personalized and effective approach to stress management. Through the amalgamation of wearable sensor data and sophisticated algorithms, this system offers individuals the means to make informed decisions about their mental and physical well-being. As we move forward, the future scope of this research involves refining and expanding the capabilities of machine learning models for stress affect detection, ensuring adaptability to diverse real-world scenarios and contributing to the ongoing evolution of stress management strategies.

REFERENCES

[1]. Li, C. From Involution to Education: A Glance to Chinese Young Generation. In Proceedings of the 2021 4th International Conference on Humanities Education and Social Sciences (ICHESS 2021), Xishuangbanna, China, 29–31 October 2021; Atlantis Press: Amsterdam, The Netherlands, 2021; pp. 1884–1887.

[2]. Ponzini, A. Educating the new Chinese middle-class youth: The role of quality education on ideas of class and status. J. Chin. Sociol. 2020, 7,

[3]. Pascoe, M.C.; Hetrick, S.E.; Parker, A.G. The impact of stress on students in secondary school and higher education. Int. J. Adolesc. Youth 2020, 25, 104–112.

[4]. College Student Suicide: Failures and Potential Solutions. Available online: (accessed on 19 February 2023).

[5]. Wagh, K.P.; Vasanth, K. Performance evaluation of multi-channel electroencephalogram signal (EEG) based time frequency analysis for human emotion recognition. Biomed. Signal Process. Control 2022, 78, 103966.

[6]. Vijayakumar, S.; Flynn, R.; Corcoran, P.; Murray, N. CNN-based Emotion Recognition from Multimodal Peripheral Physiological Signals. In Proceedings of the IMX'22: ACM International Conference on Interactive Media Experiences, Aveiro, Portugal, 22–24 June 2022

[7]. Miao, M.; Zheng, L.; Xu, B.; Yang, Z.; Hu, W. A multiple frequency bands parallel spatial–temporal 3D deep residual learning framework for EEG-based emotion recognition. Biomed. Signal Process. Control 2023, 79, 104141.

[8]. Montero Quispe, K.G.; Utyiama, D.M.; Dos Santos, E.M.; Oliveira, H.A.; Souto, E.J. Applying Self-Supervised Representation Learning for Emotion Recognition Using Physiological Signals. Sensors 2022, 22, 9102.

[9]. Tang, Y.; Wang, Y.; Zhang, X.; Wang, Z. STILN: A Novel Spatial-Temporal Information Learning Network for EEG-based Emotion Recognition. arXiv 2022, arXiv:2211.12103.

[10]. Choi, J.; Lee, J.S.; Ryu, M.; Hwang, G.; Hwang, G.; Lee, S.J. Attention-LRCN: Long-term Recurrent Convolutional Network for Stress Detection from Photoplethysmography. In Proceedings of the 2022 IEEE International Symposium on Medical Measurements and Applications (MeMeA), Messina, Italy, 22–24 June 2022; IEEE: New York, NY, USA, 2022; pp. 1–6.

[11]. Hu, X. Wenjuanxing Official Website. Available online: (accessed on 19 December 2022).

[12]. Electro, P. Polar Verity Sense. Available online: (accessed on 21 December 2022).

[13]. Xinweilai. BMD101 ECG Detection Package. Taobao. Available online: (accessed on 19 June 2023).

[14]. NeuroSky. MindWave Mobile Setup Kit. Available online: (accessed on 8 August 2022).

[15]. Mekruksavanich, S.; Hnoohom, N.; Jitpattanakul, A. A Deep Residual-based Model on Multi-Branch Aggregation for Stress and Emotion Recognition through Biosignals. In Proceedings of the 2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Prachuap Khiri Khan, Thailand, 24–27 May 2022; IEEE: New York, NY, USA, 2022; pp. 1–4

[16]. Fawaz, H.I.; Lucas, B.; Forestier, G.; Pelletier, C.; Schmidt, D.F.; Weber, J.; Webb, G.I.; Idoumghar, L.; Muller, P.-A.; Petitjean, F. Inceptiontime: Finding alexnet for time series classification. Data Min. Knowl. Discov. 2020, 34, 1936–1962