



Human Stress Detection in and Through Sleep Patterns Using Machine Learning Algorithms

R. Geetha¹ · S. Gunanandhini¹ ·
G. Umarani Srikanth² · V. Sujatha³

Received: 23 October 2023 / Accepted: 14 May 2024
© The Institution of Engineers (India) 2024

Abstract Stress has a remarkable impact on various cognitive functions, demanding timely and effective detection using strategies deployed across interdisciplinary domains. It influences decision-making, attention, learning, and problem-solving abilities. As a result, stress detection and modeling have become important areas of study in both psychology and computer science. This study links the fields of psychology and machine learning to deal with the urgent requirement of accurate stress detection methodologies and highlights sleep patterns as a key indicator for stress detection, discussing a novel approach to understand and determine stress levels. Psychologists use affective states to measure stress, which refers to a sense of feeling an underlying emotional state. However, most stress classification work has been limited to user-dependent models, which new users cannot use without additional training. This can be a significant time burden for new users trying to predict their affective states. Therefore, it is critical to address basic mental health issues in children and adults to prevent them from developing more complex problems on account

of undergoing stress. The medical field processes vast amounts of medical data; the machine learning algorithms sift through patterns that might escape the human eye. The machine learning algorithms act as detectives, able to spot correlations and bring out a sense of complex information. The machine learning algorithms reveal fine correlations and patterns, aiding in more precise and prompt diagnoses particularly to focus fundamental mental health issues in individuals of all ages. This research work deploys an enhanced Multilayer Perceptron (MLP), exhibiting an extensive feature analysis for processing medical datasets, resulting in improved effectiveness in predicting stress levels. This helps us to diagnose issues more accurately and swiftly which improves the patient outcomes. The proposed and enhanced MLP model undergoes stringent evaluation and its performance metrics are measured as Accuracy 99%, Precision 98.6%, Recall 99%, and F1-Score 99.5% compared against existing competent machine learning algorithms that include Adaboost, Random Forest, Gradient Boosting, and Decision Tree for different stress levels undertaken. The results show that MLP provides best results of accuracy compared with existing machine learning techniques in identifying stress detection via sleep patterns.

✉ R. Geetha
geetha@saec.ac.in
S. Gunanandhini
gunanandhinis@saec.ac.in
G. Umarani Srikanth
umaranisrikanth@gmail.com
V. Sujatha
sujatha@saec.ac.in

- ¹ Department of Computer Science and Engineering, S.A. Engineering College, Chennai 600077, India
- ² Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India
- ³ Department of Master of Computer Applications, S.A. Engineering College, Chennai 600077, India

Keywords Stress detection · Machine learning · Multilayer perceptron (MLP) · Random forest · Decision tree · Gradient boosting

Introduction

The technique for detecting stress based on physiological cues is a popular approach which is used in research and clinical settings for many years [16–18]. This technique is referred as biofeedback, which involves measuring

physiological parameters and providing feedback to the individual. Galvanic Skin Response (GSR), pulse rate, body temperature, muscle tone, and blood pressure are all commonly used parameters to measure stress. GSR is a measure of skin electrical conductivity and increases when a person is under stress. As part of the body's "fight or flight" reaction to stress, heart rate rises. Due to decreased blood supply to the extremities, temperature may decrease. The body's reaction to stress includes an increase in muscle tone, which makes the individual to feel uncomfortable. The body's response to stress can increase blood pressure. By measuring these parameters, it is possible to provide feedback to the individual on their state of mind and level of stress [19–21]. For example, an individual may be asked to relax their muscles or take deep breaths to lower their HR and muscle tension.

Stress is said to be occurred as facing rough and unpredictable challenges which demands the need for effective coping tools to navigate through the difficult times. Training the individual about the relaxation approaches like meditation or progressive muscle relaxation are the considered to be the best and effective tools enable them to navigate the stormy seas of stress. It entitles individuals to take an active role in managing their stress levels. Meditation helps in calming the mind, while progressive muscle relaxation alleviates body strain. It's a holistic strategy that addresses both the mental and physical aspects of stress [22–24]. Overall, persons who want to regulate their stress levels can benefit from using physiological indicators to identify stress, along with using other forms of stress management, like counseling, exercise, and a healthy lifestyle.

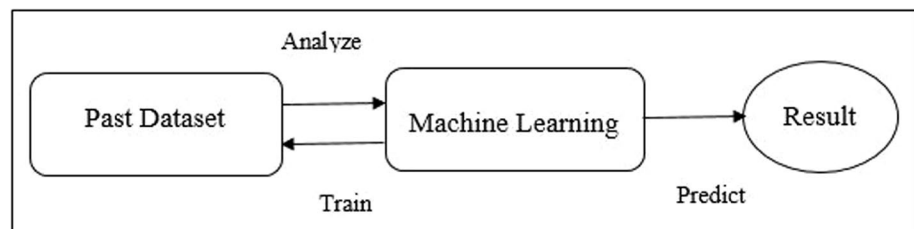
Artificial intelligence (AI) [27] is termed as the intelligence to carryout intended tasks which naturally require intelligence par with humans. The objective of AI is to imitate cognitive abilities of human in machines, empowers computing machines to execute complex tasks and adjust to changing environments. The advanced algorithms of AI are widely exploited in machine learning [28] expert systems, natural language processing, speech recognition and machine vision. It mirrors growing perception of real artificial intelligence as technology gets progressed, being evolved into a common, prominent and inevitable too. Optical Character Recognition (OCR) [29], which involves the ability of machines to distinguish and interpret text from

images or scanned documents, is one of the cutting-edge AI applications and it has demonstrated outstanding success, assisting in the resolution of a number of difficult issues in both industry and academics. AI provide enterprises with valuable findings into their functionalities, unveiling patterns and trends that may have gone unobserved using conventional methods. This analytical capability can be considered as an innovative solution for strategic decision-making. Tasks such as analyzing large scale sets of legal documents require meticulous attention, and AI excels in this domain. Its competence to process large volumes of data swiftly, reduces the likelihood of errors, making it more reliable and efficient tool for types of tasks. AI can be termed as precise assistant in handling the analysis of complex workflows.

Machine learning uses [26] historical data to forecast the future. ML enables the computers that can learn from data without explicitly programming. This concept is similar to teach processing machines to observe patterns and draw conclusions based on past experience. The emphasis on developing programs that adapt to new data, reinforce the dynamic and evolving nature of machine learning applications. This leads to creating systems that can continuously improve and optimize their performance as such systems encounter new information dynamically. Python highlights the practical aspect of implementing ML. Python is a popular programming language for ML due to its simplicity and versatility where specialized algorithms are deployed using ML in training and prediction processes.

Unsupervised, reinforced, and supervised learning are the three different types of learning [25]. The input data and necessary tagging are sent to a supervised learning system such as artificial Neural Networks [30] which enables it to learn the material, which must first be tagged by a person. Learning without supervision has no labels, well standard algorithms are designed for training the machines to act intellectually and it needs the classification algorithms in order to achieve the same. Reinforcement learning gets associated with its environment, learns from positive/negative response to improve performance. Data scientists used to handle classical machine learning approaches to identify novel patterns in python that result in insights as shown in Fig. 1. The data used for classification can be multi-class or binary, depending on the task at hand, such as identifying

Fig. 1 Machine learning model



the gender of a person or detecting spam messages. Classification problems are prevalent in various sectors such as speech recognition, handwriting recognition, biometric identity verification, medical document analysis, stress detection and so on.

Supervised learning is the most remarkable strategy, corresponding algorithm learns from a labeled dataset, where it's supplied with typical input-output pairs (X and Y), and objective is to learn mapping function (f) which exactly predicts the output variable (Y) for quite new and unseen input data (X). The intention is to carry-out the task of designing an optimal mapping function which predicts the output data(Y) when an input data (X) is presented. Supervised learning models, such as logistic regression, multi-class classification, decision trees, and support vector machines, are commonly used supervised training models, suitable for different types of problems. The training data is labeled with the exact outcomes, and this method is proven to build predictions by finding suitable patterns and relationships with this labeled data.

Furthermore, the proposed framework has several potential applications. Identifying stress levels in individuals by analyzing their sleep patterns thoroughly which enables healthcare providers to apply preventative measures to alleviate the impact of stress. The framework developed in this research can also be integrated into wearable devices or smart phone applications to provide real-time monitoring of stress levels of individuals. With the advent of telemedicine, the proposed framework can enable remote monitoring of patients' stress levels. Employers can also utilize the research findings to initiate workplace wellness programs to mitigate stress among employees.

The remainder of the sections is arranged as follows. Section "[Related Work](#)" discusses about relevant survey work carried out to find stress detection using machine learning algorithms. Section "[Methodology](#)" portrays architecture of the proposed method; Section "[Performance Evaluation](#)" illustrates outcomes and performance evaluation with empirical results and concludes the findings of the proposed solution presented in this article.

Related Work

Hatoon Alsagri et al [1] used machine learning techniques to identify Twitter users who may be experiencing depression by observing their behavior and keywords patterns in their tweets. Social media sites such as Facebook, Twitter, and Instagram seem to yield remarkable and significant influence on society. While social networking has its benefits, there are also significant downsides. Researchers have observed that frequent social media usage results higher rates of depression among the users. The authors

developed and tested classifiers to analyze a person's network activity and tweets to determine whether the individual depressed. The results show that accuracy and F-measure scores for spotting depressed users improve as more features are included. This data-driven method used as a predictive strategy for early identification of depression and other mental illnesses. Key contribution of the work highlighted in this work is the investigation of the traits and impact on analyzing the severeness of depression.

In this study, Meera sharma et al [2], the authors worked with unknown datasets to find whether individuals are seeking treatment for mental health issues by employing range of deep learning, machine learning classifiers and predictive techniques to ensure accurate predictions through statistical analysis to overcome both issues. The study conducted in the year 2017 revealed that, more than 792 million individuals, which is around 10% of the world's population, suffered with mental disorders, led 78 million suicides. Previous efforts to predict suicidal tendencies using data science have been unsuccessful. Additionally, the authors employed extensive variety of deep learning and machine learning classifiers to make exact, optimal predictions using statistical analysis.

Sandhiya et al [3] handled a dataset of questionnaire posted to IT employees to assess their mental health status. Several machine learning approaches were applied to study the outcome, which highlighted the importance of consistent mental health screenings for IT workers to monitor their well-being. Although mental health is a popular research topic nowadays, but it is less discussed in everyday life, despite the fact that one's level of well-being is an indicator of their mental health. Due to the increasing use of technology, individuals in various industries, including IT, may experience mental health issues, such as stress, worry, and depression. Companies should provide medical care in the workplace and offer benefits to affected employees. Detecting and treating common childhood mental health issues early can greatly improve patients' quality of life. Machine learning techniques have been designed and proved well in analyzing medical data and aiding in diagnosis.

Sumathi et al [4] validated performance of eight distinct machine learning strategies in identifying five common mental health issues. The techniques were been trained and experimented on a dataset consisting of 60 cases, with 25 characteristics identified as crucial for determining the issue. Feature selection approaches were exploited to minimize the features and correctness of the classifiers was measured using entire attribute set and condensed features set. Multilayer Perceptron, Multiclass Classifier, and LADTree classifiers were found to produce most accurate results with little variation between using overall attributes set and condensed attributes set. It is important to continue developing

and improving these techniques to effectively diagnose and treat childhood mental health issues.

Sarah Graham et al [5] provided an overview of the potential benefits and drawbacks of AI technology in mental healthcare. Recent original research on AI and its current uses in healthcare was also examined. The review analyzed various studies, utilized diverse methods using e-health records; brain imaging data, monitoring systems, and social media platforms. The objective is to categorize diseases pertaining to mental illnesses. Although promising, authors caution against premature conclusions and emphasize the need for bridging gap between clinical treatment and research about mental health using artificial intelligence. Amir Mohammed Mohammadi et al [6] described a stress detection model that uses four signal types, including body temperature, respiration, Electro Cardio Gram (ECG), and Electro Dermal Activity (EDA), extracts 65 features from a public dataset. The study found that 43 of the 65 features significantly differ between stressed and relaxed states using Kruskal-Wallis analysis. The K-Nearest Neighbor (KNN) technique was exploited to classify the states, achieving an accuracy of 96.024%. The system is advantageous as it requires fewer sensors and less power, relying on ECG and EDA signals, which provide excellent accuracy. Additionally, a high-performance sensor was devised which measures ECG and EDA signals from 18 strong individuals aged 16-40, who are exposed to stress using the Stroop Color-Word Test and an arithmetic mental exercise. This sensor achieves an accuracy of 94.425% and can operate for up to 70 hours on a single battery charge.

Samriti sharma et al [7] aimed to construct a simple pre-surgery stress detection method using Electrodermal Activity (EDA) measured through a minimally invasive wrist bracelet. The study recruited 41 participants from Sri Ramakrishna Hospital in Coimbatore, India, who underwent various surgical procedures. Using the EDA data collected, a supervised machine learning algorithm was developed to detect motion artifacts, achieving 97.83% accuracy on a new user dataset. Stress can have detrimental effects on individuals undergoing surgery, both physically and mentally, highlighting the importance of identifying preoperative stress levels. The findings emphasize the potential of this approach in detecting preoperative stress levels and mitigating negative impacts on surgical outcomes.

Ravinder Ahuja et al [8] focused on investigating influence of stress on candidates who are pursuing degree in an Institution, during different phases of their academic periods, specifically in a week ahead of examinations and the time intervals when using the internet. Mental stress, particularly among young individuals, is a significant problem in today's world. The supposed carefree period of life is now fraught with increased stress levels, leading to various issues like depression, suicide, heart attacks, and stroke. The

study highlighted mental stress due to "overlooked" impact of exam and recruitment process, and the authors observed that there is a connection between this type of stress and student's frequent internet usage. The authors collected a dataset from 206 candidates studying at a university, used categorization methods to measure sensitivity, specificity, and accuracy and it was proved that Support Vector Machines exhibit highest accuracy rate of 85.71%.

Shruti gedan et al [9] provided an in-depth review of stress identification using wearable sensors along with machine learning approaches. Stress is an elevated state of both body and mind, arises in situations which challenging or demanding. Stressors are the environmental factors that trigger stress. If someone is exposed to multiple stressors simultaneously over an extended period, it can lead to chronic health problems. Wearable technology allows for constant and real-time data collection, enabling individuals to monitor their own stress levels. This paper also suggests the construction of a multimodal stress identification architecture which was designed in association with wearable sensor-based deep learning techniques. Future research studies are expected to examine the stresses, methods, outcomes, benefits, limitations, and concerns for each study. Can et al [10] have devised an approach for stress detection which utilizes smart bands to collect physiological data. The novel architecture was adopted to monitor the stress levels of 216 individuals over an eight-day training session for an EU work. The study collected 2780 self-report questions from participants of various nationalities, as well as 1440 hours of physiological data. The system captured environmental information and various forms of physiological data to calculate each participant's subjective stress levels. The proposed system could be effectively utilized to determine perceived stress levels over sessions, days, and time.

Gjoreski et al [11] introduced a system that can continuously detect stressful events using a commercially available wrist device. Long-term exposure to stress indeed results detrimental effects both on physical and mental health. It attributes various health issues, such as cardiovascular diseases, weakened immune system and mental health disorders. Hence it is imperative to detect stress early for preventing the negative impacts. The proposed architecture has three components: a stress detector device assesses short-term stress periodically; an activity monitor that keeps track of user activity consistently records contextual data, and context-based stress detector captures outcome of stress detector and user context to make a decision every 20 minutes. This proposed device was measured in both laboratories and in real-world settings, achieved 92% for a two-class problem and launched as Smartphone app for managing physical and mental health issues.

Can et al [12] discussed about the widespread use of Smart phones, smart watches, and smart wristbands taking over people's lives. Stress has become a prevalent issue among common people, and this has led to a discussion about the potential for wearable sensors and Smart phones to detect and prevent stress. In this study, the researchers examined current research on the use of wearable technology and Smart phones for detecting stress in various daily life settings, including office, campus, transportation, and unrestricted daily living situations. Ayten Ozge Akmandor et al [13] focused on stress is that a common and widespread psychological disorder that inevitably affects people's mood and behavior. If left unchecked, chronic stress will create serious impacts on an individual's physical and mental health. There is potential for the application of various nature-inspired computing techniques and deep learning methods, such as Deep-Belief Network, Convolutional-Neural Network, and Recurrent-Neural Network, to analyze multimodal data gathered from behavioral testing, electroencephalogram signals, finger temperature, respiration rate, pupil diameter, galvanic-skin-response, and blood pressure readings.

Furthermore, Kim et al [14] designed a hybrid model incorporating several computational approaches, adaptation, parameter adjustment, utilizing chaos, levy, and Gaussian distribution, to express issues related to stress. Prolonged exposure to stress can create negative impacts on immune, cardiovascular, and endocrine systems. In order to deal this problem, a team of researchers have devised a Stress Detection and Alleviation system called SoDA. System makes use of Wearable Medical Sensors (WMSs), including ECG, GSR, respiration rate, blood pressure, and blood oximeter to consistently examine stress levels. The system's effectiveness was evaluated by analyzing data obtained from 32 individuals who experienced four stressors and were subjected to three stress reduction techniques. SoDA uses a mixture of both supervised feature selection and unsupervised dimensionality reduction to identify stress with 95.8% accuracy. Nath et al [15] created a stress prediction model for elderly people using a smart wristband that measures Electro-Dermal Activity (EDA), Blood Volume Pulse (BVP), and Heart Rate Variability (HRV) were gathered from 40 individuals during an analysis process known for inducing stress, measured through salivary cortisol. A supervised method was adopted to select 27 out of 47 features extracted from the signals.

Accumulating information from multiple signal streams proved to have remarkably escalated the model's performance in distinguishing between stressed and not-stressed states. Achieving accuracy of 94% is quite substantial and recommends that the model is effectively capturing and

leveraging the relevant features from each signal stream. It's a great example of how a holistic approach can improve the capabilities of a model in executing complex tasks like stress detection. This novelty made the model to achieve an accuracy of 94% and a macro-average F1-score of 0.92 when using features from all four signals. The study lasted for a year with an average age of 73.625 ± 5.39 .

Methodology

The proposed method is a new approach to identify stress in the decision-making process, which was evaluated using dataset collected from stressful situations in kaggle website. Unlike prior research that only assessed stress levels generally, this method aims to detect stress specifically in the decision-making process, providing insight for identifying stress in future decision-making scenarios. Stress can impact decision-making, making early recognition of stress vital to enhance clinical performance. Although the existing methods have demonstrated potential in detecting stress, previous studies used only individual-level features for classification, without considering the inter-channel correlations in the brain that could reveal distinctive features for stress detection. The disadvantages include that (a) this is a complex process because some instrument type material was deployed to detect the stress level (b) performance metrics were not measured (c) Deployment is not implemented.

Data about stress from numerous sources is combined to form the dataset. Data is downloaded, verified as accurate, cleaned and trimmed. The acquired dataset is separated as training and testing datasets. Test dataset and testing dataset are created based on the accurateness of results. The system model pre-processes outliers, irrelevant data, and a combination of continuous, categorical, and discrete variables, the ML prediction model proved successful in predicting stress. The training set plays a critical role in the machine learning process with a Multi Layer Perceptron (MLP) classifier, random forest, decision tree classifier, and gradient boosting algorithms, along with test set prediction is made in accordance with the accuracy of the test results. The advantages include accuracy of the work improvised and performance metrics of each algorithm are compared which provide better results. The various phases involved in the proposed methodology as shown in Fig. 2 are as follows:

Data Analysis and Model Deployment

Data Pre-processing

Validation procedures are very useful to access the percentage of errors of machine learning models, which is normally

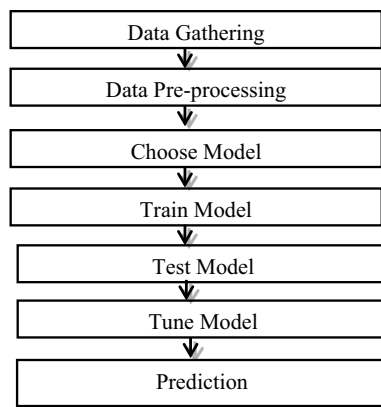


Fig. 2 Phases of the process flow

close to the actual error rate of the dataset. However, when working with data samples that are not representative of the population, validation becomes necessary. This involves identifying missing or duplicate values and data types to ensure data quality and accuracy. Incorporating information from the validation dataset into the model setup can lead to biased evaluations, and adjusting hyper-parameters based on the validation set should be done carefully. Therefore, understanding your data and its characteristics during the data identification phase can assist in choosing the appropriate method for constructing our model. Python's Pandas module can be used for various data cleaning tasks, particularly for handling missing values, which is one of the most significant data cleaning tasks. It is essential to realize the various types of missing data from statistical perspective analysis. Ultimately, more time should be spent on modeling and analysis, and less on data cleaning.

Data Collection

Separating the given dataset is an intelligent approach to validate outcome of models at hand and algorithms like Random Forest, MLP, Decision Trees, Gradient Boosting, and Adaboost were adopted to design the data model. Each algorithm has its potential, and ensemble methods like Random Forest and Boosting can often enhance overall performance. It is advisable to maintain 7:3 ratios for training and testing to keep the balance between training and testing on unseen data.

Data Manipulation

Data is loaded, checked for delicacy, and trimmed and gutted for analysis. Make sure to precisely validate the cleaning opinions and give defense.

Data Visualization

Data visualization provides a powerful set of tools for gaining a qualitative understanding of a dataset, helping to identify patterns, outliers, and other key relationships. By presenting data visually through charts and graphs, it can become more understandable to stakeholders. Visualizing data is also imperative for fast analysis in both applied statistics and machine learning, where various plot types are used to explore and analyze data samples and other objects in Python. Some common data visualization tools and libraries in Python include Matplotlib, Seaborn, Plotly, and Bokeh. These libraries provide an interactive visualization option, allowing for a more engaging and informative presentation of data.

Building the Classification Model

The following factors make the robust and high accuracy prediction model for human stress effective: It produces satisfactory and reliable outputs in classification problems, has the ability to handle well the preprocessing outliers, different types of variables, managing the combination of continuous, categorical, and discrete variables for addressing real-world complexity. It also generates unbiased out-of-bag estimate errors which add impartial in numerous tests.

Construction of a Predictive Model

It is known that machine learning often demands a large amount of data for training; it is not always necessary to use raw, unprocessed information and is a process of cleaning and altering data into a recommended format for machine learning algorithms. Preprocessing can involve several steps, such as removing outliers, normalizing data, and encoding categorical variables. This process explains about how preprocessing steps are tailored to specific needs of the data and basic requirements of machine learning tasks. Regarding accurately predicting human stress levels, there are various machine learning models that can be trained on preprocessed data to achieve this goal. Some popular models for regression problems include linear regression, decision trees, random forests, and neural networks. It is highly inevitable to notice that correctness of the model depends on quality and quantity of data being used for training.

The dataset was obtained from Kaggle and then goes through data-preprocessing to eliminate duplicate and null values. Then the data is represented in graph by data visualization. The algorithms are implemented and the highest accuracy is shown in the model. The model is deployed using the input given by users as shown in Figs. 3 and 4.

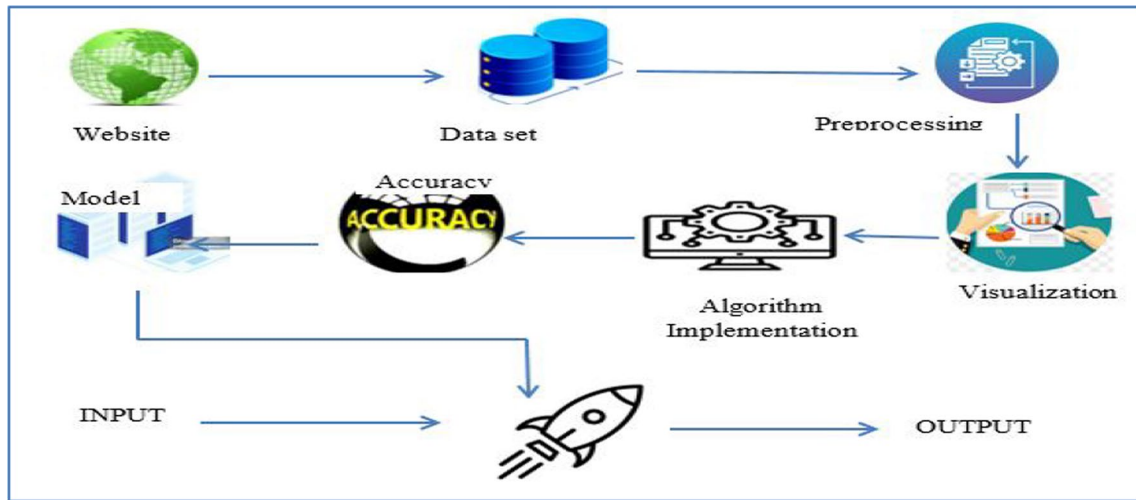
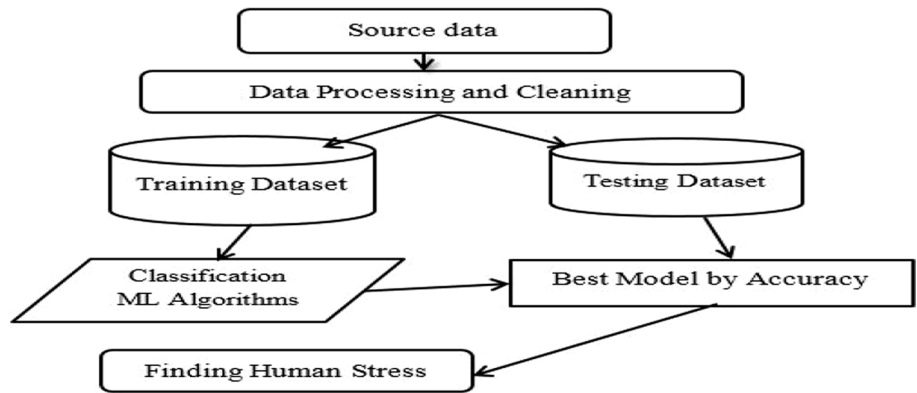


Fig. 3 Architecture diagram

Fig. 4 Workflow diagram



In the initial step, data related to sleep and stress is assembled. This data may include physiological signals such as heart rate, respiration rate, snoring rate, etc that are recorded during sleep. The collected data needs to be preprocessed in order to filter noise or artifacts that may be found. This may include filtering, artifact removal, and normalization of the data. The next step is to collect relevant features from preprocessed data, may include statistical metrics such as mean, standard deviation, skew and spectral features such as power spectral density and spectral entropy. The extracted features may be high-dimensional and contain redundant or irrelevant information. Hence, feature selection strategies such as Principal Component Analysis (PCA) / Mutual Information resorted to choose most identical features.

Machine Learning Model Training

Once the relevant features are selected, a machine learning algorithm is employed on the labeled dataset to predict stress levels. Some of the commonly used algorithms include Multi-Layer Perceptron (MLP), Decision Tree Classifier, Random Forest, Gradient Boosting and Adaboost Classifier.

Model Evaluation

Trained model is validated on a test dataset to estimate its performance. Metrics includes accuracy, precision, recall, and F1-score depend on task being solved. Accuracy gives a measure of correctness, while precision and recall speak about how well the model is performing on specific classes.

Deployment

After the designed model is trained, tested and evaluated, to detect stress levels during sleep. This may involve monitoring physiological signals in real-time and making predictions based on the trained model. The deployment process may also involve testing and validating the overall performance of the model in real-world conditions to ensure that it can handle variations in signal quality, environmental noise, and user variability. Overall, deploying a machine learning model for real-time stress detection during sleep is a difficult and challenging task, but one with the potential to improve the understanding and management of stress-related disorders.

Modified Multilayer Perceptron

In this work we propose a modified Multilayer Perceptron which involves the dropout layers, which randomly deactivate a percentage of neurons during training, preventing overfitting and enhancing generalization. Additionally, we explored the variations in activation functions and the number of hidden layers to optimize MLPs for specific tasks, contributing to the ongoing evolution of neural network architectures in the human stress level detection.

MLP is a type of artificial neural network that consists of multiple layers of interconnected nodes, each layer contributing to the learning and abstraction of complex patterns. By employing hidden layers and activation functions, MLPs can effectively model non-linear relationships, making them versatile for various machine learning tasks such as image recognition and natural language processing.

The process begins with the input layer, which receives the raw data or features to be processed as shown in Fig. 5. Each node in this layer represents a feature of the input data. Every connection between nodes in adjacent layers is associated with a weight, representing the strength of the connection. Additionally, each node has a bias term, allowing for greater flexibility in modeling. Following the input

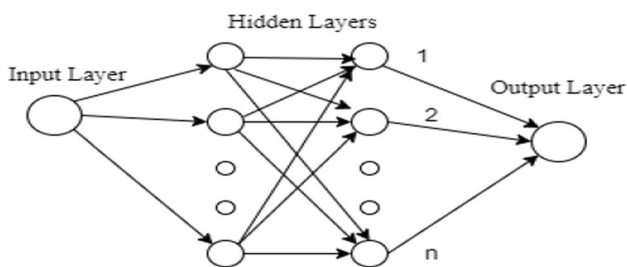


Fig. 5 Modified multilayer perceptron

layer are one or more hidden layers. These layers perform transformations on the input data using weighted sums and activation functions. Nodes within each hidden layer apply an activation function to the weighted sum of inputs and biases. Common activation functions include sigmoid, tanh, ReLU, and softmax. These functions introduce non-linearity, enabling the network to learn complex patterns in the data. The input data is propagated forward through the network layer by layer, with each layer's output serving as the input to the next layer. The final layer, known as the output layer, produces the network's predictions or outputs. The number of nodes in this layer depends on the nature of the problem (e.g., classification, regression).

A loss function measures the difference between the network's predictions and the actual target values. The goal during training is to minimize this loss by adjusting the network's parameters (weights and biases).

To update the network's parameters, an optimization algorithm such as stochastic gradient descent (SGD) is used. Backpropagation, a key concept in training neural networks, calculates the gradients of the loss function with respect to the network's parameters. The gradients obtained from backpropagation are used to update the weights and biases in the direction that minimizes the loss function.

The learning rate determines the size of these updates. Training typically occurs over multiple iterations called epochs. In each epoch, the entire dataset is passed through the network. Batch training involves dividing the dataset into smaller batches to update the parameters more frequently. Techniques such as dropout and L2 regularization are commonly employed to prevent overfitting, where the model performs well on training data but poorly on unseen data. Once training is complete, the model's performance is evaluated on a separate validation set to assess its generalization ability. Finally, the trained model can be used to make predictions on new, unseen data by passing it through the network and obtaining output values.

MLP first layer is input layer, which takes raw input data (such as images or text) and forwards it to the next layer. The next layers, known as hidden layers, perform a series of nonlinear transformations on the input data to capture complex patterns from input features. The final layer, called output layer, constructs classification output based on patterns observed in the previous layers. The training MLP uses a process called back propagation to update values of weights and bias of the neurons found in each layer to improve similarity between predicted and actual outputs.

Utilizing a non-linear kernel function, the outcomes, as denoted in Equation (1), are computed where ' w ' denotes the vector weights, ' y ' represents the input combination, and ' b ' signifies the bias, with the kernel function denoted as ' Φ '.

$$y = \phi \left(\sum_{i=1}^n w_i x_i + b \right) = \phi(w^r x_i + b) \tag{1}$$

The training process of the multilayer perceptron (MLP) involves a two-phase back-propagation approach. In the initial.

forward phase, Eq. (1) is employed to compute categorized outputs based on the provided input data. Subsequently, in the backward phase, partial derivatives of the kernel function concerning parameter adjustments are computed and.

propagated back through the network. Following this, a gradient boosting algorithm is applied to update the network's.

weights, and the entire procedure iterates until the weights converge.

Hyperparameter Optimization

Training an MLP involves updating hidden layer weights to maximize performance, where hyperparameters play a significant role. Hence, fine-tuning through hyperparameter optimization is crucial due to the substantial impact on model performance. Even with the same MLP architecture, accuracy can vary greatly based on hyperparameter combinations. We have chosen 4 hyperparameters for optimization for yielding better results. Among the four optimized hyperparameters in this study, the first two were the number of hidden layers and nodes. Increasing these can effectively capture complex features, but excessive complexity risks overfitting, necessitating careful adjustment. The third hyperparameter, learning rate, determines weight updates during training; extremes can hinder convergence or cause slow progress. Dropout, the fourth hyperparameter, limits training node participation to prevent overfitting, although potentially extending training time.

To ensure the prediction model's accuracy and prevent overfitting, we conducted the aforementioned four hyperparameter optimization and defining the tuning sets for the hyper parameters through trial and error. Twenty percent of the total data was allocated as the validation dataset (SD_SL_{test}). Unlike the training set, this portion wasn't directly involved in training but served to monitor and evaluate the model's predictive accuracy during the training process.

During training, network is supplied with data (stress level data inputs L0, L1,L2, L3,L4 and corresponding outputs-SD_SL_{train}) and weights are updated to reduce this error. This process is continued for numerous epochs until the error is minimized or a predefined stopping criterion is met. Increasing the number of hidden layers can lead to a proliferation of unnecessary features, hindering accurate predictions. Hence, for this model, we opted for two hidden layers each with 10 neurons to balance complexity and performance. Additionally, we experimented with 20 hidden nodes and found improved accuracy. The maximum number of iterations the solver can perform is set to 1000. The random mode is set to 42, which ensures that the MLP is reset every time it runs with the same random weight. This can be beneficial in terms of repeatability. The activation parameter is set to "Relu", which means that the MLP uses a rectified linear unit activation function in its hidden layers. We determined that a learning rate of 0.01 yielded better average performance, avoiding suboptimal solutions or local minima. Setting the dropout value to 0.5 further enhanced average performance, despite the model not necessitating regularization.. It is well suitable for large datasets and deep neural networks.

The pseudocode of the proposed algorithm is explained below that detects the various levels of human stress from the dataset collected from kaggle.

Input: Dataset

Output: SD_SL Stress Detection _Stress Level Prediction (L0, L1 ,L2, L3,L4)

1. Divide the data set into training(SD_SL_{train}) and testing(SD_SL_{test}) sets
 2. For each individual data in SD_SL_{train} do
 3. If SD_SL_{train} has missing or null entries then
 4. apply the pre-processing steps
 5. Else
 6. Train the model for Classification using MLP
 7. Repeat the above step until all the data in SD_SL_{train} are trained
 8. Test the model with each entry of the SD_SL_{test}
 9. Classify the Stress Level (L0, L1 ,L2, L3,L4)
 10. Return the Classification
-

Proposed Algorithm: Human Stress Level Detection

The provided pseudocode outlines a workflow for stress detection and prediction using an MLP model. Initially, the dataset is split into training and testing sets to facilitate model training and evaluation. For each individual data entry in the training set, a conditional check is performed to ascertain if there are any missing or null entries. If such inconsistencies are detected, preprocessing steps are applied to ensure data integrity. Conversely, if the data is complete, the MLP model is trained for stress level classification. This iterative process continues until all data entries in the training set are utilized for training. Following model training, each entry in the testing set undergoes evaluation using the trained model. The model predicts the stress level for each entry, categorizing it into one of the predefined stress levels (L0 to L4). Finally, the classification results are returned, providing insights into the stress levels present in the dataset. This approach enables the automated detection and prediction of stress levels based on input data, facilitating proactive intervention and support strategies. The proposed model is evaluated against the other machine learning algorithms such as AdaBoost Classifier, Random Forest Classifier, Gradient Boosting Classifier, and Decision Tree Classifier for accuracy, precision, recall and F1 score.

Other Machine Learning Algorithms

Decision Tree Classifier

This classifier adopts a supervised approach to segment training data based on specific parameters. The segmentation process produces decision nodes and leaves, which are used to construct a tree-like structure. Decision trees are useful in various machine learning applications, such as classification, regression for their resemblance to real-world scenarios and represent decisions formally and graphically in decision analysis. In classification and regression problems, decision trees are a non-parametric method that aims to create a model using decision rules derived from data attributes to predict the target variable's value.

Gradient Boosting Classifier

When the decision trees are poor learners, the resulting method, known as gradient-boosted trees, typically beats random forests. Building gradient-enhanced model follows same step-by-step process as the previous enhancement technique, but generalizes the other techniques by allowing a differentiable loss function to be optimized. Gradient boosting classifier's primary premise is to fit a series of decision trees to the training data, where each tree tries to fix the mistakes produced by the preceding tree. The algorithm learns how

to give the incorrectly categorized samples more weight throughout the succeeding iterations during the training process. Gradient boosting classifiers are hence very adept at managing unbalanced datasets. It has a high degree of accuracy and the capacity to manage very vast and intricate datasets. Because it emphasizes fixing the errors made by the prior models, it is also less prone to over fitting than other ensemble approaches, such Random Forests.

Random Forest Classifier

This classifier creates several random samples from the training data and randomly selected features for each split. Each tree is trained on a different sample, and their performance is evaluated during training to select the best tree. The algorithm uses the majority vote of the individual tree forecasts to produce a prediction for a new data point after it has been processed by all the decision trees in the forest. The class with the higher votes decides final prediction. Comparing Random Forest to other classification methods, there are various benefits. The method is strong, resists over-fitting, and can handle noisy or missing data. Moreover, it can handle high-dimensional datasets and trains rather quickly. Applications for random forests include text classification, image classification, and prediction in the financial and medical fields.

AdaBoost Classifier

Adaptive Boosting, is an ensemble learning method combines predictions of multiple weak classifiers to create a strong classifier. A weak classifier is a model that performs slightly better than random chance. Initially it assigns equal weights to all training data samples, trains a weak classifier on the data, and evaluates its performance. Later it calculates the error of the weak classifier and weight of the error is used to identify misclassified examples. Later weights of the misclassified examples are increased, making them more important for the next classifier. This process is repeated till a perfect classifier is achieved.

In this work, deployment is done in Jupyter Notebook in Anaconda Navigator, with Django acting as middleware. The frontend consists of HTML and CSS. Human stress can be detected from numerical values such as snore rate, breathing rate, body temperature, limb movement, blood oxygen, eye movement, sleep time, and heart rate. Django is web framework for developing web applications quickly and easily. It has a built-in administrative interface that can be customized to manage the data in the application. Django's templating engine allows developers to create reusable templates for building consistent user interfaces across the application. The framework

also includes a URL routing system that maps URLs to appropriate views, making it easy to organize and manage the application’s logic. After training a machine learning model, a pickle data format file (known as a.pkl file) is received which is deployed to enhance the user interface and improve accuracy of predictions. By doing so, the trained model can be readily accessed and used for real-time decision-making.

Performance Evaluation

The performance of the proposed algorithm against the existing machine learning models is evaluated using various performance metrics such as true positive, true negative, false positive, false negative, accuracy, precision, recall, F-score, Confusion matrix as discussed below.

False Positive (FP): It occurs when the model identifies a positive outcome, but the real outcome is found to be negative. Diminish FP is difficult, especially in scenarios where the consequences of false alarms are remarkable. It’s a balance between sensitivity and precision in our model. **False Negative (FN):** FN occurs when the model predicts a negative outcome, but the recorded outcome is positive, it’s a situation where model is unable to fail to identify a true positive. FP is crucial to reduce in situations where missing a positive case has severe consequences. **True Positive (TP):** TP occurs when the model correctly predicts a positive outcome, and the actual outcome is positive. It’s a win for the model when it successfully predicts a positive event. It represents instances where the model and reality are coherent, correctly recognizing the positive class. **True Negative (TN):** TN occurs when the model correctly identifies a negative result, and the actual outcome is indeed negative. In simple terms, it accurately identifies

a negative event. It represents instances where the model is said to find the absence of the positive class.

$$\text{True Positive Rate (TPR)} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{False Positive Rate (FPR)} = \text{FP} / (\text{FP} + \text{TN})$$

Accuracy

It is the most common evaluation metrics that provides an overall measure of how well a classification model perform.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

Precision

The proportion of successfully predicted favorable outcomes is known as precision. It is the proportion of all positively predicted observations to those that were correctly predicted.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Recall

Is a metric which measures the ability of a classification model to record all the relevant positive instances.

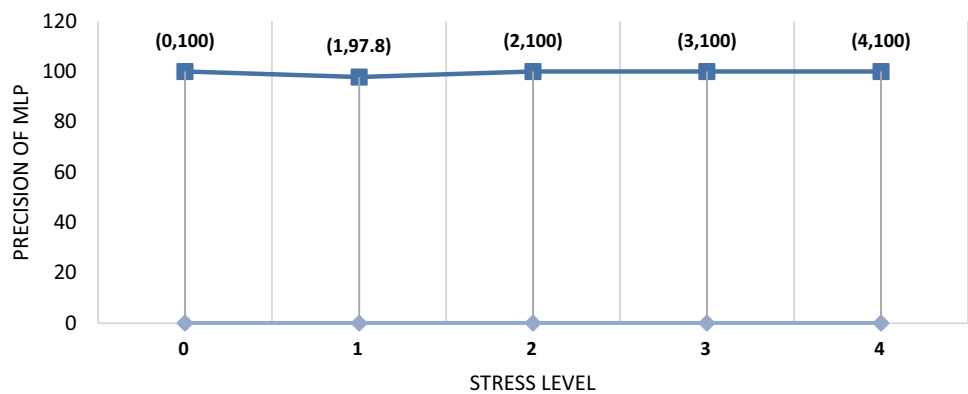
$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

F1 Score

The F1-score is a metric, combination of precision and recall into a single value, providing a balanced measure of a model’s performance.

$$\text{F1 score} = 2\text{TP} / (2\text{TP} + \text{FP} + \text{FN})$$

Fig. 6 Precision graph of MLP



Results and Discussion

In this work, it has some genuine estimation to assess how well the different classification algorithms performed in tests. Many evaluation techniques, such as accuracy, sensitivity, specificity, and precision as well as the F1 measure were used to gauge the effectiveness of the categorization systems.

Multi Layer Perceptron (MLP)

MLP is capable of providing accurate results by leveraging multiple layers of interconnected neurons. By fine-tuning the model, the MLP can achieve high sensitivity, ensuring that it detects a large portion of positive instances correctly. It also exhibits good specificity, meaning it can correctly identify negative instances. Additionally, the MLP's precision is noteworthy, as it delivers precise predictions by minimizing false positives. Overall, the MLP's performance can be evaluated using the F1 measure, which combines both precision and sensitivity, providing a balanced assessment of its predictive capabilities.

Precision

It measures the proportion of positive instances out of all instances that the model predicted as positive. The value of precision measured for the stress levels fall into five categories 0, 1, 2, 3, 4 shown in Fig. 6. From the graph shown, it is observed that the MLP algorithm correctly identified stress levels 0, 1, 2, 3, and 4 with maximum precision for each category and performed exceptionally well in categorizing stress levels. It is demonstrated that high level of effectiveness is arrived which is the indication of the model's ability to accurately predict stress levels based on the provided data.

Recall

It is the measure that correctly identifies True Positives. A perfect recall rate for each stress level category is shown as 100, 97.8, 100, 100, and 100 in Fig. 7. It is observed that the algorithm can successfully identify and recall all viable instances of each stress level category from the given dataset applied during testing. In other words, we can say that the MLP algorithm has a high sensitivity to each stress level category, ensures that it can identify all the appropriate

Fig. 7 Recall graph of MLP

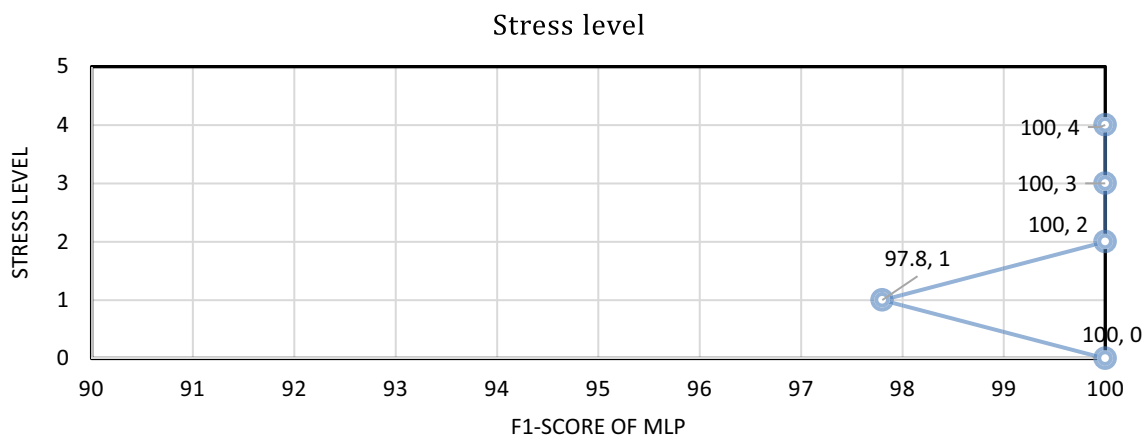
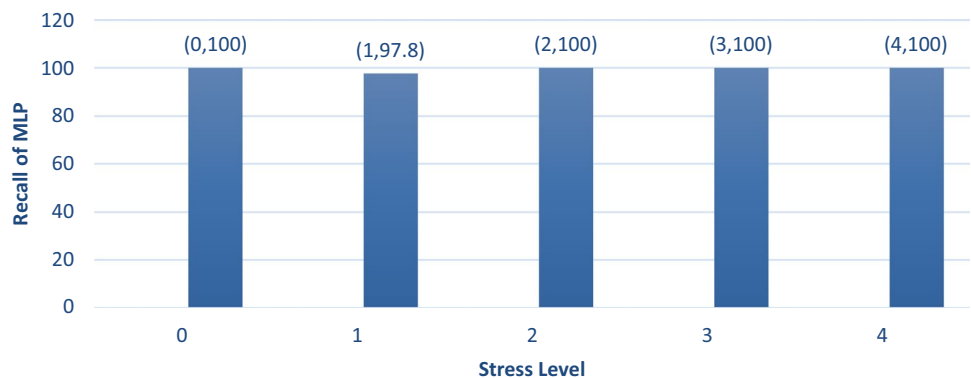
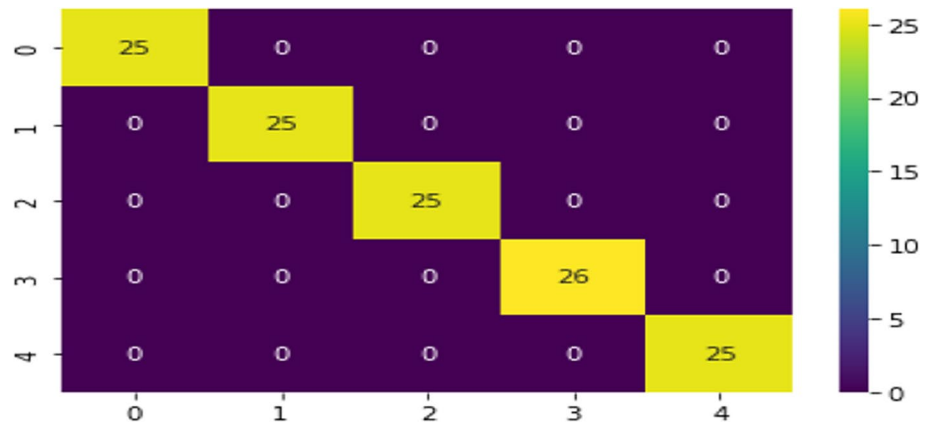


Fig. 8 F1-score graph of MLP

Fig. 9 Confusion matrix graph of MLP



instances accurately, and demonstrates the robustness of the MLP algorithm in accurately identifying stress levels, which is inevitable for applications such as stress monitoring and prediction.

F1-Score

It is one of the machine learning evaluation metrics to measure accuracy and overall performance of a binary classification model and combines both precision and recall scores of a model. It is metric to deal with imbalanced classes. The model is trained to classify data into five different categories representing stress levels ranging from 0 to 4. The outcome for these categories is plotted as a scatter graph shown in Fig. 8. F1-Score of 1 indicating that perfect precision and recall are achieved for the five categories of stress levels 0,1,2,3,4. This is achieved due to high-quality training data and the model is capable of learning complex patterns in the data, allowing it to make accurate predictions across all stress level categories.

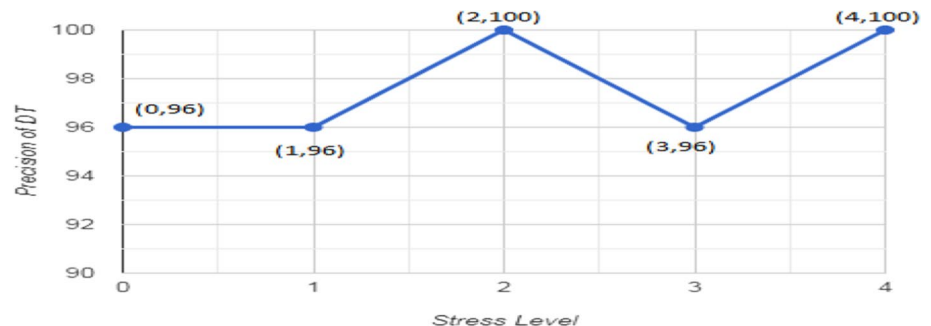
Confusion Matrix

It is one of the performance evaluation tools in machine learning, representing the accuracy of a classification model. It is the N*N matrix compares the actual target values with the predicted values generated by the machine learning model. In the Fig. 9, the confusion matrix of MLP is drawn using the matrix value. The result of the matrix came diagonally; it shows the overall support for the algorithm using stress levels. From the diagonal values it is implied that this algorithm has good support for correctly predicting stress levels across the dataset. The provided confusion matrix disclosed that the MLP algorithm is making predictions exclusively in favor of certain classes while completely neglecting other.

Decision Tree Classifier

It exhibits good sensitivity, allowing it to effectively detect positive instances. The decision tree classifier also demonstrates high specificity, enabling it to accurately identify negative instances. Moreover, precision is an important metric for the decision tree classifier, as it emphasizes the reduction of false positives. The F1 measure can be used to evaluate

Fig. 10 Precision graph of decision tree



its overall performance by considering both precision and sensitivity.

Precision

The Fig. 10, describes the precision scores achieved by a Decision Tree Classifier algorithm for different stress levels that fall into five categories 0, 1, 2, 3, 4. The precision scores 96, 96, 100, 96, 100 achieved by the Decision Tree Classifier for each stress level respectively. Each percentage represents

the proportion of correct predictions for the respective stress level. For example, precision of 96% indicate that the algorithm correctly classified instances of the respective stress level. For example, if the stress level is 0, the algorithm predicted instances as stress level 0 with 96% accuracy out of all instances it predicted. This algorithm understood the 0, 1, 3 level of stress is 96%, means it works correctly for 96% of trained datasets, while 4% were incorrectly classified.

Fig. 11 Recall graph of decision tree



Fig. 12 F1-score graph of decision tree

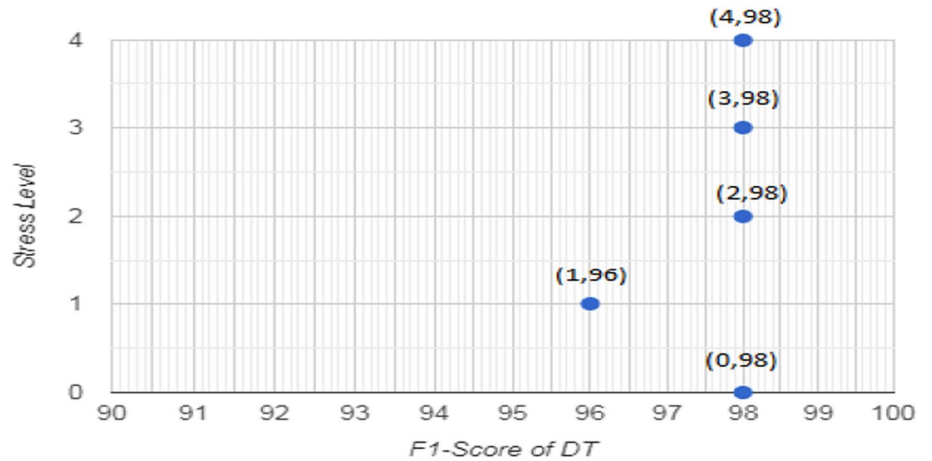
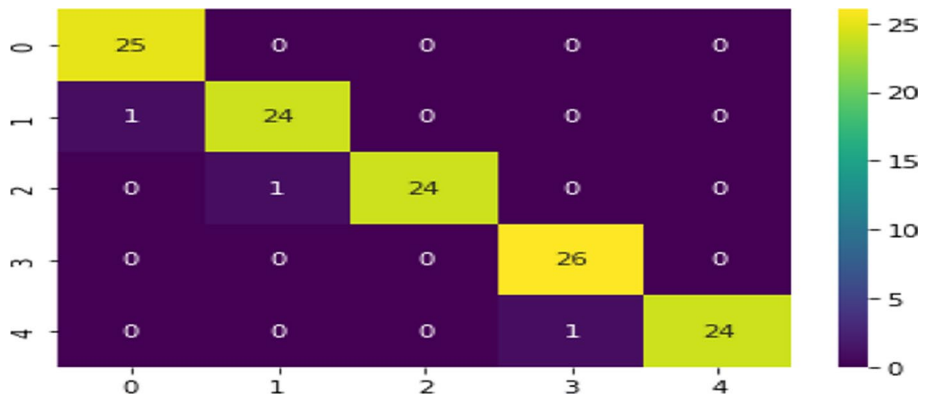


Fig. 13 Confusion matrix graph of decision tree



Recall

Figure 11 refers to the graphical representation of the recall scores for the Decision Tree Classifier algorithm for each stress levels 0, 1, 2, 3, 4 resulted as 100%, 96, 96, 100, 96 respectively. When it's stated that the algorithm understood the levels by 100%, it means that the algorithm correctly identified all instances of that stress level. For example, if the Recall for stress level is 0, the algorithm correctly identified all instances of stress level 0. This suggests that the recall scores obtained from testing the data can be used as an indicator of how well the model will perform on unseen data in the future. High recall scores imply that the model is able to correctly identify a large proportion of instances belonging to each stress level. The algorithm identifies all instances of that stress levels 0 and 3. A recall score of 96% resulted in stress levels 1,2 and 4 means that the model accurately determines 96% of all relevant instances out of the total instances that actually belong to the positive class.

F1-Score

Figure 12 shows the F1-Score of the 0,1, 2,3 and 4 stress levels. F1-Score of 1 indicates that perfect precision and

recall is achieved for the five categories of stress levels 0,1,2,3,4. The F1-score achieved by the Decision Tree Classifier for each stress levels 0, 1, 2, 3,4, resulted as 98, 96, 98, 98, 98% respectively. The results have shown that this model has good balance between precision and recall. Overall, the Decision Tree Classifier seems to perform reasonably well across all stress levels, proved strong performance in identifying instances of each class.

Confusion Matrix

This matrix aids in analyzing model performance, identifying misclassifications, and improving predictive accuracy. It is the NxN matrix used for evaluating the performance of a classification model, where N is the total number of target classes. It compares the actual target values with those predicted by the machine learning model, provides holistic view of how well the classification model is performing well and what kinds of errors it is making. In the Fig. 13, the confusion matrix of Decision Tree is drawn using the matrix values. 25 instances are correctly classified as class 0, 24 instances are correctly classified as class 1 and there is 1 instance of class 1 misclassified as non-class 1, 24 instances are correctly classified as class 2, 1 instance of class 2 misclassified as non-class 2, 26 instances are correctly classified

Fig. 14 Precision graph of gradient boosting

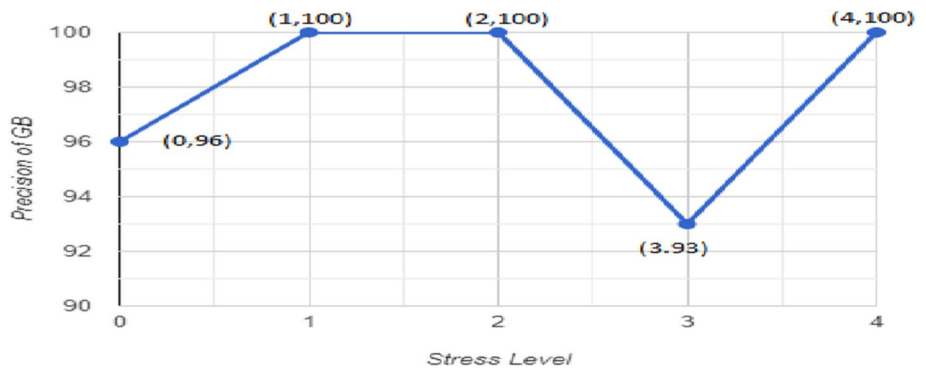


Fig. 15 Recall graph of gradient boosting

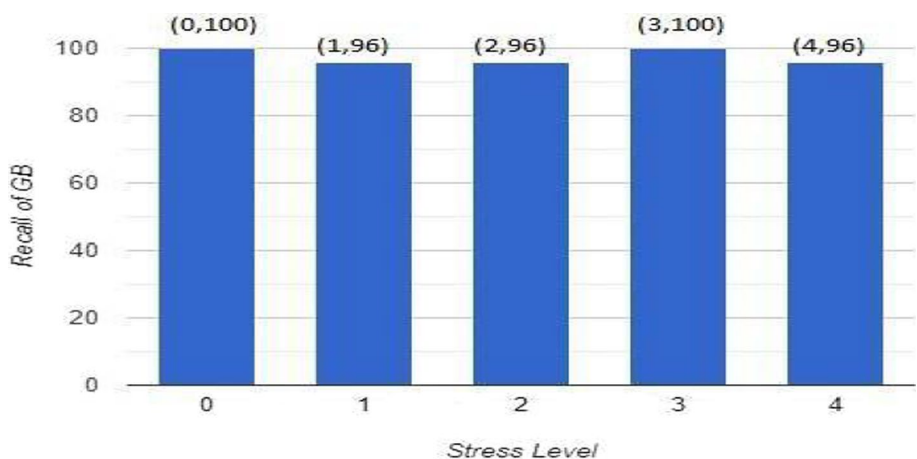
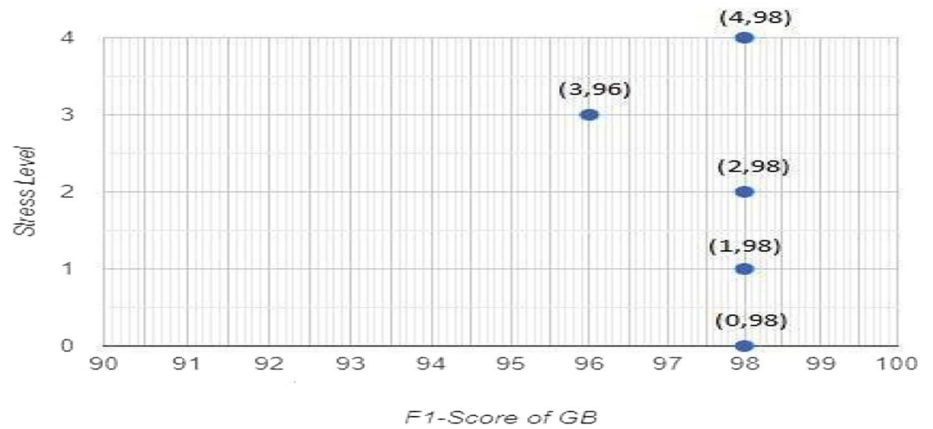


Fig. 16 F1-score of gradient boosting



as class 3 and 24 instances are correctly classified as class 4. The classifier demonstrates high overall accuracy as it correctly classifies the majority of instances across all classes.

Gradient Boosting Classifier Algorithm

Precision

A gradient boosting classifier algorithm was trained to classify stress levels into five categories:0,1,2,3 and 4. From the Fig. 14, it is clear that for stress level 0, the precision is measured as 96%, means when the algorithm predicts a stress level of 0, it is correct 96% of the time. For stress level 1, the precision is measured as 100%, which implies that the algorithm predicts a stress level of 1, it is correct 100% of the time. For stress level 2, the precision is 100%, level 3, the precision is 96% and for stress level 4, the precision is 100%. This indicates that when the algorithm predicts a stress level of 4, it is correct 100% of the time. The consistently high precision scores across all stress levels indicate the classifier’s robustness and reliability in making accurate predictions, with particularly

noteworthy performance for stress levels 1,2 and 4, where the precision scores reach 100%.

Figure 15 shows the recall scores measured for different stress levels, fall into five categories 0,1,2,3,4. For each stress levels of recall the gradient boosting classifier algorithm understood the levels by 100, 96, 96, 100, 96. There may be a small number of instances of stress levels 1, 2 and 4 that are incorrectly classified as belonging to other stress levels. The scores provide insights into how well the Gradient Boosting Classifier performs in correctly identifying instances belonging to each stress level. In this case, the recall scores range from 96% to 100%, indicating a high level of accuracy in identifying instances across different stress levels. It also demonstrated that it exhibits consistent, high-performance levels across all stress levels and proved its effectiveness in stress level classification tasks.

F1-Score

Figure 16 shows that the F1-Score of gradient boosting algorithm. The algorithm is trained to classify data into five different categories representing stress levels ranging

Fig. 17 Confusion matrix graph of gradient boosting

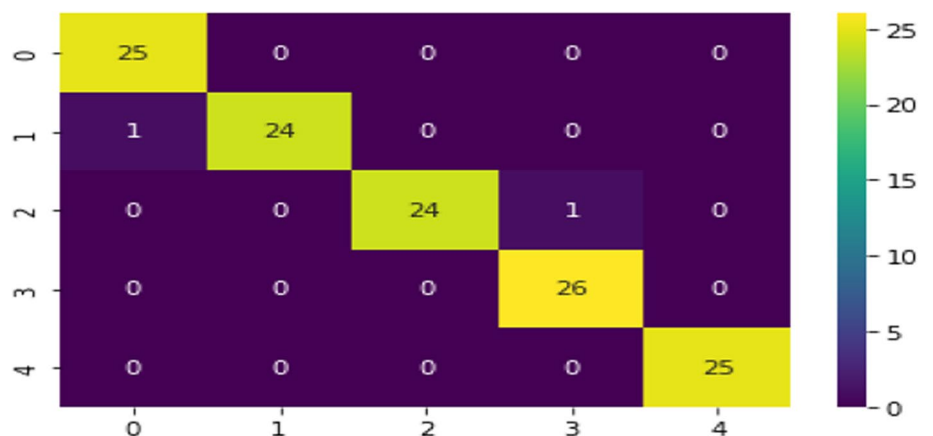


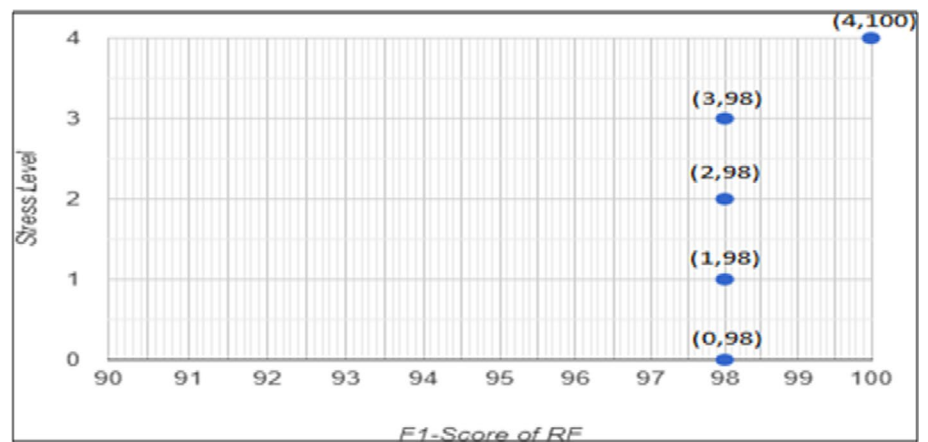
Fig. 18 Precision graph of random forest



Fig. 19 Recall graph of random forest



Fig. 20 F1-score graph of random forest classifier



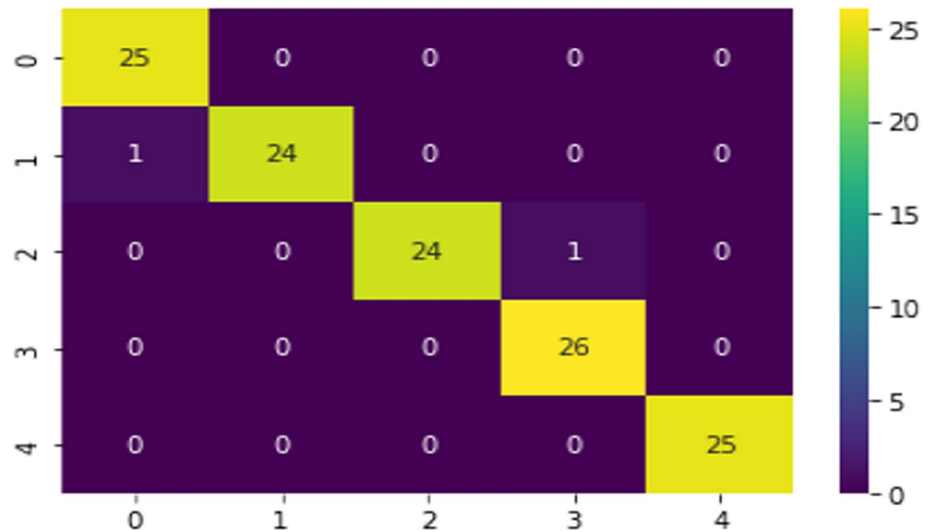
from 0 to 4. The outcome for these categories is plotted as scatter graph shown in Fig. 15. F1-Score of 1 indicates that perfect precision and recall is achieved for the five categories of stress levels 0, 1, 2, 3, 4. The F1-score achieved for each stress levels 0, 1, 2, 3, 4 and resulted as 98, 98, 98, 96, 98 respectively. The results have shown that this model has good balance between precision and recall for the corresponding stress level. Overall, the gradient

descent algorithm performs reasonably well across all stress levels, proved strong performance in identifying instances of each class.

Confusion Matrix

In the Fig. 17, the confusion matrix of gradient boosting algorithm is drawn using the matrix values. 25 instances

Fig. 21 Confusion matrix graph of random forest



are correctly classified as class 0, 24 instances are correctly classified as class 1 and there is 1 instance of class 1 misclassified as non-class 1, 24 instances are correctly classified as class 2, 1 instance of class 2 misclassified as non-class 2, 26 instances are correctly classified as class 3 and 24 instances are correctly classified as class 4. The classifier demonstrates high overall accuracy as it correctly classifies the majority of instances across all classes.

Random Forest Classifier

It demonstrates good sensitivity, effectively capturing positive instances in the dataset. With high specificity, it can accurately classify negative instances. Precision is an important metric for the random forest classifier, focusing on minimizing false positives. The F1 measure combines precision and sensitivity, providing a comprehensive assessment of its overall performance.

Precision

A random forest classifier algorithm was trained to classify stress levels into five categories: 0, 1, 2, 3, and 4. From the Fig. 18, it is clear that for stress level 0, the precision is measured as 96%, for stress level 1, the precision is measured as 100%, which implies that the algorithm predicts a stress level of 1, it is correct 100% of the time. For stress level 2, the precision is 100%, level 3, the precision is 96% and for stress level 4, the precision is 100%. The algorithm understood the 0, 3 levels of stress are 96%, means it works correctly for 96% of training datasets. The consistently high precision scores across all stress levels indicate the classifier’s robustness and reliability in making accurate predictions, with particularly noteworthy performance for stress levels 1 and 4, where the precision scores reach 100%.

In the Fig. 19, the stress levels fall into five categories 0,1,2,3,4. For each stress levels of recall, the random forest classifier algorithm understood the levels by 100, 96, 96, 100, 100. This recall levels can be predicted by testing the

Fig. 22 Precision graph of adaboost

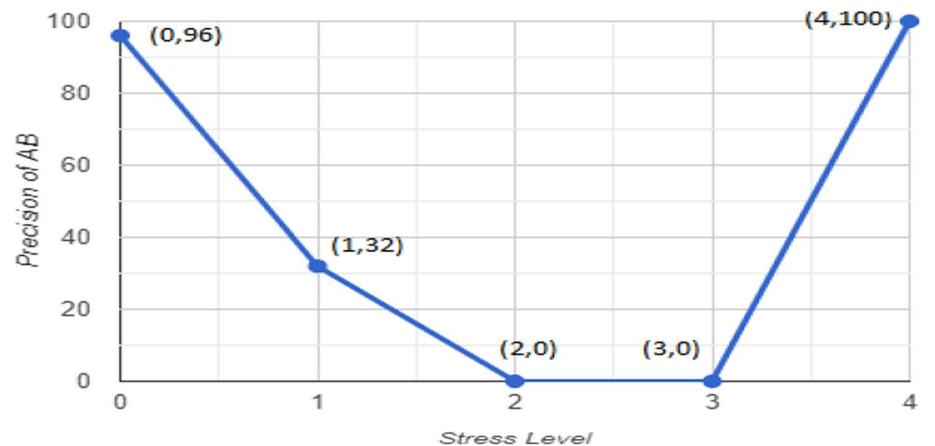
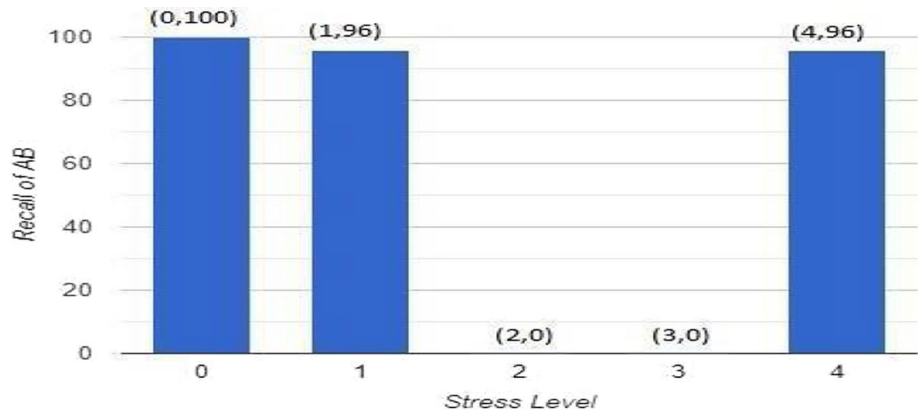


Fig. 23 Recall graph of ada-boost



data in the future process. The algorithm understood the 1,2 levels of stress are 96%, means it works correctly for 96% of training datasets. The random forest classifier algorithm accomplishes recall scores of 100% for stress levels 0, 3, and 4, indicating perfect recall for these stress levels. The results also showed that the algorithm exhibits commendable performance, producing high recall scores across stress levels 0, 3, and 4 and 96% accuracy for stress levels 1 and 2.

F1-Score

F1-Score of Fig. 20 indicates that perfect precision and recall is achieved for the five categories of stress levels 0, 1, 2, 3, 4 using Random Forest classifier. The F1-score achieved for each stress levels 0, 1, 2, 3,4 and resulted as 98, 98, 98, 98, 100% respectively. With F1-scores of 98% for stress levels 0, 1, 2, and 3, and a perfect F1-score of 100% for stress level 4, the classifier has produced exceptional performance across all stress levels. These high F1-scores imply that the classifier achieved a balance between precision and recall, effectively classifying both the positive and negative instances for each stress level.

Confusion Matrix

In the Fig. 21, the confusion matrix of Random Forest is drawn using the matrix value. 25 instances are correctly classified as class 0, 24 instances are correctly classified as class 1 and there is 1 instance of class 1 misclassified as non-class 1, 24 instances are correctly classified as class 2, 1 instance of class 2 misclassified as non-class 2, 26 instances are correctly classified as class 3 and 24 instances are correctly classified as class 4. The classifier demonstrates high overall accuracy as it correctly classifies the majority of instances across all classes.

Adaboost Classifier

It achieves high accuracy by iteratively adjusting the weights of misclassified instances. It demonstrates good sensitivity, effectively capturing positive instances in the dataset. With high specificity, it can accurately classify negative instances. Precision is an important metric for the AdaBoost classifier, focusing on minimizing false positives. The F1 measure combines precision and sensitivity, providing a comprehensive assessment of its overall performance.

Fig. 24 F1-score graph of adaboost

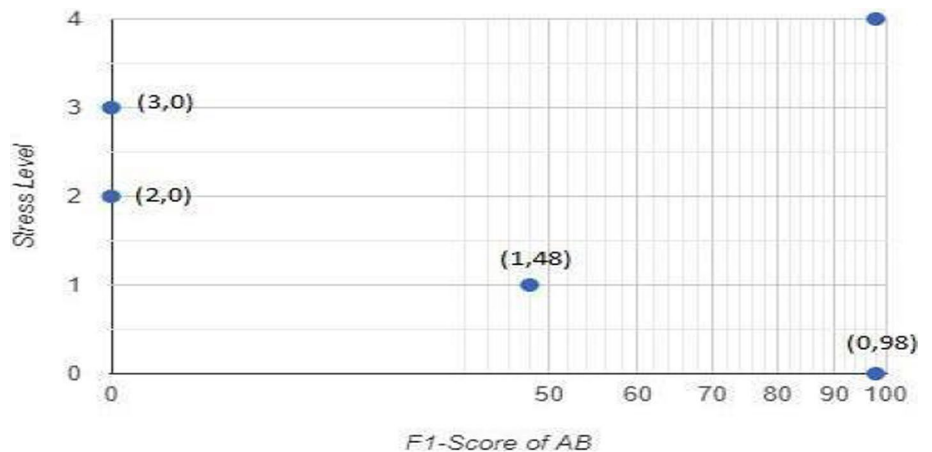
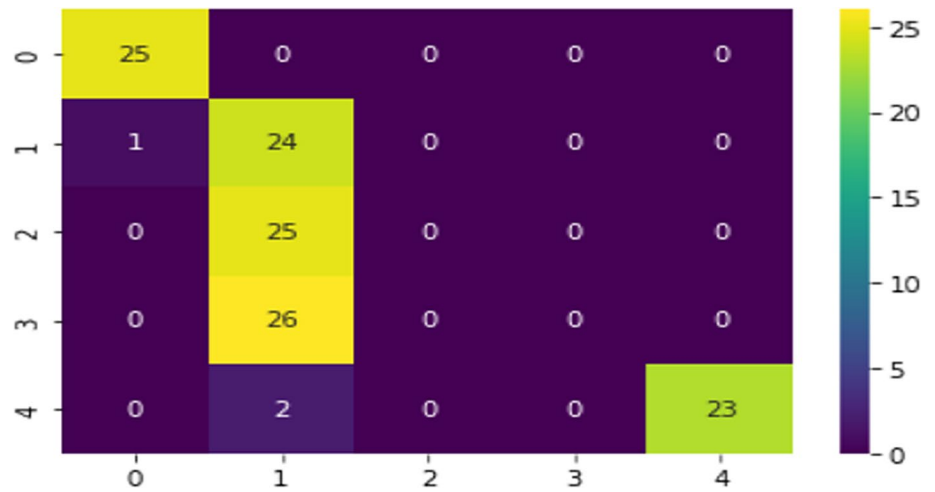


Fig. 25 Confusion matrix graph of adaboost



Precision

In the Fig. 22 the stress levels of Adaboost Classifier is shown by using precision values as 0,1,2,3,4. For each stress levels of precision the AdaBoost algorithm understood the levels by 96%, 32, 0, 0, 100 by training the data. The algorithm achieves precision scores of 96% for stress level 0 and 100% for stress level 4, which concludes that the algorithm performs well and produces high degree of accuracy in correctly identifying instances belonging to these stress levels. At the same time, for stress levels 2 and 3, the algorithm results lower precision scores 0%. This suggests that the algorithm is unable to identify instances belonging to stress levels 2 and 3, resulting in a higher rate of false positives.

Recall

The recall metric of the stress levels has shown in the Fig. 23 fall into five categories 0, 1,2,3,4. For each stress levels of recall the AdaBoost classifier algorithm

understood the levels by 100, 96, 0, 0, 96. This recall levels can be predicted by testing the data in the future process. The recall scores of 100% and 96% for stress levels 0,1 and 4 indicate that the algorithm accurately identifies all instances belonging to these stress levels. The algorithm fails to correctly identify instances belonging to these stress levels 2 and 3 during training which shows that a significant limitation in the algorithm’s ability to discriminate instances that belong to the stress levels 2 and 3. Hence it is vital to adopt the process of continuous refinement and optimization of machine learning algorithms to ensure reliable and accurate classification across all stress level categories.

F1-Score

F1-Score for the Adaboost algorithm is shown in Fig. 24 indicates that perfect precision and recall is achieved for the five categories of stress levels 0,1,2,3,4. The F1-score

Fig. 26 Support graph

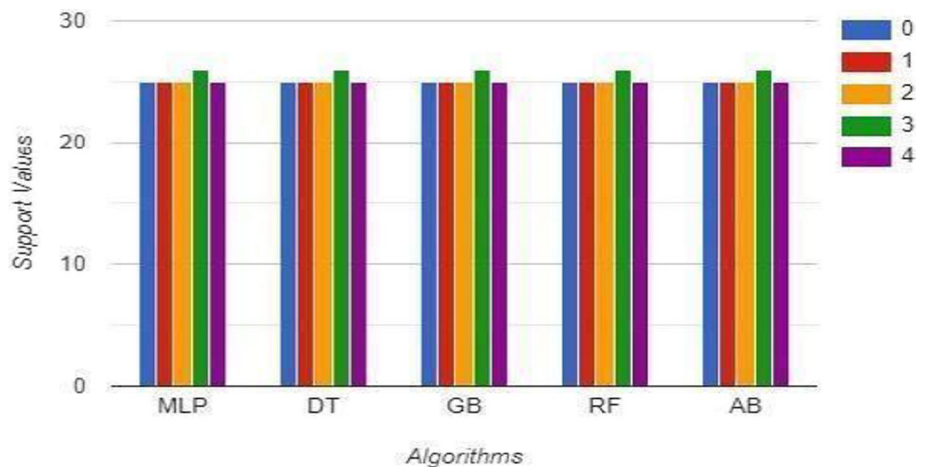
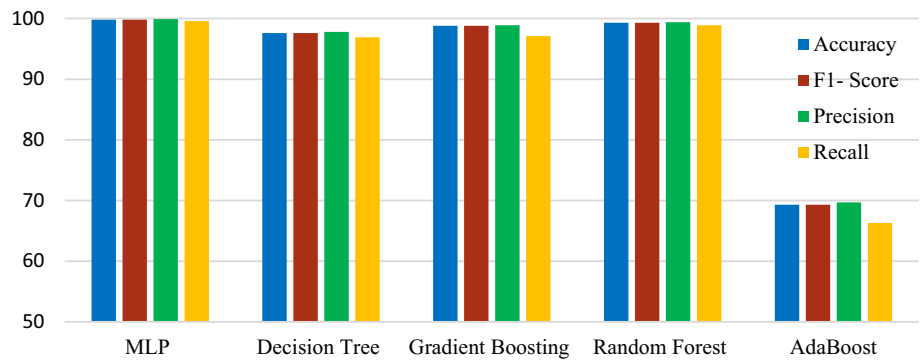


Fig. 27 Performance comparison graph



achieved for each stress levels 0, 1, 2, 3,4 and resulted as 98, 48, 0, 0, 98 respectively. From these results, it's evident that the classifier performs exceptionally better for stress levels 0 and 4, achieving F1-scores of 98%. However, it demonstrates poor performance for stress level 1 with F1-scores of 48%. Additionally, for stress levels 2 and 3, the F1-score is 0%, indicating a complete failure to accurately classify instances belonging to this stress levels.

Confusion Matrix

In the Fig. 25, the confusion matrix of Adaboost is drawn using the matrix value. The result of the matrix came diagonally; it shows the overall support for the algorithm using stress levels 0, 1, 2, 3, 4. The algorithm correctly predicted all instances of class 0 (25 instances). There were no instances of class 0 that were incorrectly predicted as any other class. The algorithm correctly predicted 24 instances of class 1, while incorrectly predicting 1 instance of class 1 as class 0. The algorithm incorrectly predicted all 25 instances of class 2 as class 1. It did not predict any instance of class 2 correctly. The algorithm incorrectly predicted all 26 instances of class 3 as class 1. It did not predict any instance of class 3 correctly. The algorithm correctly predicted 23 instances of class 4, while incorrectly predicting 2 instances of class 4 as class 1.

Comparative Analysis of the Various Algorithms

Support Graph

Support graph refers to a visual representation that illustrates the performance or effectiveness of different machine learning algorithms. Each algorithm is evaluated based on specific metrics, such as accuracy or error rate, graph provides a comparison of their performance. It helps in identifying the algorithm that performs the best for a given task or dataset, allowing users to make informed decisions when selecting the most suitable algorithm for their needs.

The support graph shown in Fig. 26, compares the support of various machine learning algorithms Multilayer Perceptron (MLP), Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier and Adaboost for stress detection. Each algorithm likely represents a different approach to classification. All the algorithms support the stress detection equally as shown in table which demonstrates that each algorithm has a similar level of support for detecting stress across different stress levels. In other words, these algorithms are trained on datasets with similar distributions of stress levels and all algorithms have comparable support for detecting stress across different stress levels.

Performance Comparison

The performance of the Multilayer Perceptron algorithm is compared against the aforementioned and discussed machine learning algorithms with respect to their performance metrics such as Accuracy, F1 Score, and Precision. It is observed that the proposed MLP performs better when compared to the other algorithms in all aspects in classifying the stress levels of human beings accurately. This allows for informed decision-making in selecting the most suitable algorithm for achieving high prediction accuracy. In the Fig. 27, the metrics of all the algorithms are shown, MLP shows the highest accuracy proving that it has been considered as the model for deployment.

Conclusion

In conclusion, the impact of stress on decision-making, attention, learning, and problem-solving underscores the importance of stress detection and modeling. To achieve this, the work followed a standard analytical process that involved several steps, starting with data preparation and preprocessing. The study presented in this work leverages an enhanced Multilayer Perceptron (MLP) that employs feature analysis to streamline the attribute dataset,

resulting in superior stress level prediction. The proposed model analyzed the algorithm with highest accuracy and was then selected for use in the application. The resulting application can be designed to help healthcare professionals detect stress levels in patients. By using machine learning techniques, the application can accurately predict the stress level of a patient based on a variety of input factors. Overall, the work demonstrates the value of machine learning in healthcare and highlights the importance of following a rigorous analytical process to ensure that the resulting models and applications are accurate and effective. The evaluation of performance metrics, including Accuracy, Precision, Recall, and F1-Score, demonstrates the efficacy of the MLP in outperforming existing machine learning algorithms such as Adaboost, Random Forest, Gradient Boosting, and Decision Tree. The focus on sleep patterns as a key indicator for stress detection showcases the potential of the enhanced MLP and its selected attribute set in advancing our understanding and prediction of stress levels. This research contributes valuable insights into the intersection of machine learning and mental health, offering a promising avenue for future developments in stress detection and intervention strategies.

Funding No funding interest.

Data Availability None.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Disclosure of Potential Conflicts of Interest No Conflict of interest.

Research Involving Human Participants and/or Animals Not applicable.

Informed Consent NA.

References

1. H. AlSagri, M. Ykhlef, Machine Learning-based Approach for Depression Detection in Twitter Using Content and Activity Features, vol. **E103D**, No.8, pp.1825–1832, (2020). <https://doi.org/10.48550/arXiv.2003.04763>
2. M. Sharma, S. Mahapatra, Predicting the utilization of mental health treatment with various machine learning algorithms. *WSEAS Trans. Comput.* **19**, 285–291 (2021). <https://doi.org/10.37394/23205.2020.19.34>
3. P. Sandhya, M. Kantesaria, Prediction of mental disorder for employees in IT industry. Published in *Blue Eyes Intelligence Engineering and Sciences Publication* **8**(6), 374–376 (2019)
4. B.P. Sumathi, Prediction of mental health problems among children using machine learning techniques. In: Proceedings of International Journal of Advanced Computer Science and Applications, (IJACSA), **7**(1) (2016). <https://doi.org/10.14569/IJACSA.2016.070176.2016>.
5. S. Graham, C. Depp, Artificial intelligence for mental health and mental illnesses. *Curr Psychiatry Rep.* **21**(11) (2019) <https://doi.org/10.1007/s11920-019-1094-0>.
6. A. Mohammadi, M. Fakharzadeh, B. Baraieejad, An integrated human stress detection sensor using supervised algorithms. *IEEE Sens. J.* **22**(8), 8216–8223 (2022). <https://doi.org/10.1109/JSEN.2022.3157795>
7. S. Gurvinder Singhm, M. Sharma, A comprehensive review and analysis of supervised learning and soft computing techniques for stress diagnosis in humans, **134**, (2021). <https://doi.org/10.1016/j.combiomed.2021.104450>
8. R. Ahuja, A. Banga, Mental stress detection in university students using machine learning algorithms, **152**, No. C, pp 349–353, (2019). <https://doi.org/10.1016/j.procs.2019.05.007>
9. S. Gedam, S. Paul, A review on mental stress detection using wearable sensors and machine learning techniques. *IEEE*, 84045–84066 (2021). <https://doi.org/10.1109/ACCESS.2021.3085502>
10. S. Gedam, S. Paul, A review on mental stress detection using wearable sensors and machine learning techniques. *IEEE Access* **9**, 84045–84066 (2021). <https://doi.org/10.1109/ACCESS.2021.3085502>
11. M., Gjoreski, H., Gjoreski, M., Lustrek, M., Gams, “Continuous stress detection Using a wrist device: in the laboratory and real life. In: Proceedings of International Joint Conference on Pervasive and Ubiquitous Computing Adjunct, pp. 1185–1193 (2016). <https://doi.org/10.1145/2968219.2968306>
12. Y. S. Can, B. Arnrich, C. Ersoy, Stress Detection in Daily Life Scenarios Using Smart Phones and Wearable Sensors: A Survey, Vol. 92 (2019) <https://doi.org/10.1016/j.jbi.2019.103139>
13. A.O. Akmandor, N.K. Jha, Keep the stress away with SoDA: stress detection and alleviation system. *IEEE Trans. Multi Scale Comput. Syst.* **3**(4) (2017) <https://doi.org/10.1109/TMSCS.2017.2703613>
14. H.-G. Kim, E.-J. Cheon, D.-S. Bai, Y. H. Lee, B.-H. Koo, Stress and heart rate variability: a meta-analysis and review of the literature. *Psychiatry Invest* **5**(3), pp. 235–245 (2018). <https://doi.org/10.30773/pi.2017.08.17>
15. R. K. Nath, and H. Thapliyal, Smart wristband-based stress detection framework For older adults with cortisol as stress biomarker. *IEEE Trans Consumer Electronics* **67**(1) (2021). <https://doi.org/10.1109/TCE.2021.3057806>
16. N. Schneiderman, G. Ironson, S.D. Siegel, Stress and health: Psychological behavioral and biological determinants. *Annu. Rev. Clin. Psychol.* **1**, 607–628 (2005)
17. Z. Hussain, M.D. Griffiths, The associations between problematic social networking site use and sleep quality, attention-deficit hyperactivity disorder, depression, anxiety and stress. *Int J Ment Health Addiction* **19**, 686–700 (2021). <https://doi.org/10.1007/s11469-019-00175-1>
18. R.W. Picard, Automating the recognition of stress and emotion: from lab real- world impact. *IEEE MultiMedia*, **23**(3) (2016). <https://doi.org/10.1109/MMUL.2016.38>
19. S. Betti et al., Evaluation of an integrated system of wearable physiological Sensors for stress monitoring in working environments by using biological markers. *IEEE Trans. Biomed. Eng.* **65**(8) (2018). <https://doi.org/10.1109/TBME.2017.2764507>
20. V. Bijalwan, V.B. Semwal, T.K. Mandal, Fusion of multi- sensor based biomechanical gait analysis using vision and wearable sensor. *IEEE Sens. J.* **21**(13) (2021). <https://doi.org/10.1109/JSEN.2021.3066473>
21. R. Ghosh, S. Phadikar, N. Deb, N. Sinha, P. Das, E. Ghaderpour, Automatic eyeblink and muscular artifact detection and removal from EEG signals using k-Nearest Neighbor Classifier And Long

- Short-Term Memory Networks. *IEEE Sens. J.* **23**(5), 5422–5436 (2023). <https://doi.org/10.1109/JSEN.2023.3237383>
22. X. Du et al., An efficient LSTM network for emotion recognition from multichannel EEG signals. *IEEE Trans. Affect. Comput.* **13**(3), 1528–1540 (2022). <https://doi.org/10.1109/TAFFC.2020.3013711>
 23. R. Ghosh et al., SAM 40: Dataset of 40 subject EEG recordings to monitor the induced-stress while performing stroop color-word test, arithmetic task, and mirror image recognition task. *Data Brief* **40**, p. 107772 (2022). <https://doi.org/10.1016/j.dib.2021.107772>.
 24. R. Geetha, K. Ramyadevi, M. Balasubramaniam, Prediction of domestic power peak demand and consumption using supervised machine learning with smart meter dataset. *Multimedia Tools Appl.* **80**(13), 19675–19693 (2021). <https://doi.org/10.1007/s11042-021-10696-4>
 25. G.U. Srikanth, R. Geetha, Effectiveness review of the machine learning algorithms for scheduling in cloud environment. *Arch Computat Methods Eng* **30**, 3769–3789 (2023). <https://doi.org/10.1007/s11831-023-09921-0>
 26. T.F. Kurnaz, C. Erden, U. Dağdeviren et al., Comparison of machine learning algorithms for slope stability prediction using an automated machine learning approach. *Nat. Hazards* (2024). <https://doi.org/10.1007/s11069-024-06490-8>
 27. S. Bindra, R. Jain, Artificial intelligence in medical science: a review. *Ir. J. Med. Sci.* (2023). <https://doi.org/10.1007/s11845-023-03570-9>
 28. S.R. Zanwar, Y.H. Bhosale, D.L. Bhuyar et al., English handwritten character recognition based on ensembled machine learning. *J. Inst. Eng. India Ser. B* **104**, 1053–1067 (2023). <https://doi.org/10.1007/s40031-023-00917-9>
 29. O.A. Slavin, V.L. Arlazarov, Algorithms of the tiger and cunei-form optical character recognition software. *Pattern Recognit Image Anal.* **33**, 669–684 (2023). <https://doi.org/10.1134/S1054661823040442>
 30. B. Majhi, A modified artificial neural network (ANN)-based time series prediction of COVID-19 cases from multi-country data. *J. Inst. Eng. India Ser. B* **104**, 335–350 (2023). <https://doi.org/10.1007/s40031-022-00849-w>

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.