International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Enhanced Ensemble Fusion Model for Stress Classification and Prediction

Mr. Prashant M. Suryavanshi
Ph. D. Scholar
Computer Science and Engineering
Dr. A. P. J. Abdul Kalam University, Indore MP.
sprashant1234@gmail.com

Dr. Pradnya Ashish Vikhar
Research Supervisor
Computer Science and Engineering
Dr. A. P. J. Abdul Kalam University, Indore MP.
pradnyav123@gmail.com

Abstract— Stress has become a common phenomenon in modern society, and it has been identified as a major factor that affects people's health and well-being. Stress can be caused by various factors, such as work pressure, financial difficulties, relationship problems, and health issues. Prolonged exposure to stress can lead to physical and mental health problems, including anxiety, depression, cardiovascular diseases, and obesity. Accurate stress classification and prediction can help individuals and organizations identify the sources and levels of stress and take appropriate measures to manage stress and prevent negative outcomes. By identifying individuals who are at risk of stress, proactive interventions can be initiated to prevent negative outcomes. Additionally, stress classification and prediction can be useful for designing effective stress management programs and policies that can improve the well-being and productivity of individuals and organizations. Existing systems for stress classification and prediction have limitations in terms of accuracy and efficiency. To overcome these limitations, this paper proposes an Enhanced Ensemble Fusion (EEF) model that combines three ensemble classifiers, namely stacking, bagging, and boosting, using a blending classifier. The EEF model is composed of several classifiers, including the stacking classifier, the bagging classifier, and the boosting classifier, each using an Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifier. An Enhanced Logistic Regression classifier is used as a meta-classifier for the stacking classifier. The model was evaluated on a Swell-EDA dataset and WESAD-EDA dataset, and the results show that it outperformed existing systems in terms of accuracy and robustness. The Enhanced Ensemble Fusion Model achieved an accuracy of 72.86% for WESAD-EDA dataset and 50% for Swell-EDA dataset which is significantly higher than the accuracy of individual classifiers and existing ensemble methods. The proposed model provides a promising approach for stress classification and prediction, which can be useful in various applications, such as healthcare, human resources, and education.

Keywords- EEF, SVM, Swell-EDA, WESAD-EDA.

I. INTRODUCTION

In today's fast-paced and competitive world, stress has become a pervasive problem that is affecting individuals from all walks of life. Stress can arise from a variety of sources, including work-related pressure, financial difficulties, relationship problems, and health issues [1]. In many cases, stress is unavoidable and can be a natural response to challenging situations or events. However, when stress becomes chronic and prolonged, it can have detrimental effects on an individual's overall health and well-being [2].

Chronic stress can lead to a range of physical and mental health problems. The constant activation of the body's stress response can increase the risk of developing anxiety and depression, which can significantly impact an individual's quality of life [3]. Additionally, chronic stress has been linked to an increased risk of cardiovascular diseases such as hypertension, stroke, and heart disease [4]. This is because stress can cause the release of stress hormones such as cortisol and adrenaline, which can narrow blood vessels and increase blood pressure. Furthermore, chronic stress can also lead to unhealthy coping mechanisms such as overeating, smoking, and alcohol or drug abuse. This can result in the development of obesity, type 2 diabetes, and other chronic illnesses. Therefore, accurate stress classification and prediction are crucial for individuals and organizations to manage stress and prevent negative outcomes [5].

To effectively intervene and prevent these negative outcomes, it is essential to identify individuals who are at risk of stress. Identifying individuals at risk of stress involves understanding the factors that contribute to stress in individuals, such as their personal and environmental circumstances, genetics, and coping mechanisms. Once identified, these individuals can be provided with targeted support and interventions that aim to reduce their stress levels and improve their overall wellbeing.

Early identification of stress risk factors is also important in preventing the development of more serious health conditions. For example, identifying and addressing stress in its early stages can prevent the development of chronic stress, which has been linked to a range of health problems. Stress classification and prediction can also help in designing effective stress management programs and policies that can improve the overall well-being and productivity of individuals and organizations. However, existing systems for stress classification and prediction have limitations in terms of accuracy and efficiency [6], [7].

One of the main limitations of traditional machine learning algorithms is their inability to handle complex relationships between variables. In stress prediction, multiple factors, such as age, gender, lifestyle habits, work environment, and social support, can contribute to an individual's stress levels. These factors can interact with each other in complex ways that may not be easily captured by traditional algorithms.

Furthermore, existing machine learning algorithms may lack robustness and generalization ability, which can lead to poor performance when applied to new data or under varying conditions. This is especially problematic in real-world applications where the stress levels of individuals may change over time or may be influenced by various factors. A system that lacks robustness may not be able to accurately classify and predict stress levels in these situations, leading to inaccurate or unreliable results.

As a result, the accuracy of these systems may be limited, which may not be suitable for practical applications. To overcome these limitations, this paper proposes an Enhanced Ensemble Fusion (EEF) model that combines three ensemble classifiers, namely stacking, bagging, and boosting, using a blending classifier. EEF model can be more robust than individual machine learning algorithms because it combines the predictions of multiple classifiers and leverage their collective strengths to provide more accurate and robust predictions. EEF model can reduce the risk of overfitting, which occurs when a model performs well on the training data but poorly on new, unseen data, by combining multiple models that are trained on different subsets of the data. By combining multiple models, EEF model can capture the diversity of the data and provide more robust predictions that are less affected by the particularities of the training data.

Furthermore, EEF model can be more resilient to noise and outliers in the data because it can use multiple models to detect and correct for these issues. By combining the predictions of multiple models, EEF model can identify and mitigate the effects of noise and outliers in the data, which can improve the robustness and accuracy of the model. The proposed EEF model combines three popular ensemble classifiers, namely stacking, bagging, and boosting, using a blending classifier. Stacking is an ensemble method that combines multiple classifiers by training a meta-classifier on the output of base classifiers. Bagging is another ensemble method that involves training multiple classifiers on different subsets of the training data and combining their predictions using voting or averaging. Boosting is a third ensemble method that involves training multiple weak classifiers sequentially, where each classifier focuses on the samples that were misclassified by the previous classifiers.

The EEF model builds on these three ensemble methods by combining them using a blending classifier, which further improves the accuracy and robustness of the model. The blending classifier combines the predictions of the three ensemble methods to produce a final prediction, which can be more accurate and robust than the predictions of any individual ensemble method. The EEF model is composed of several classifiers, including the stacking classifier, the bagging classifier, and the boosting classifier, each using an Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifier. An Enhanced Logistic Regression classifier is used as a meta-classifier for the stacking classifier. Each of these classifiers has been enhanced using various techniques to improve its performance and robustness for stress classification and prediction. By combining these four enhanced classifiers in the three ensemble methods and the blending classifier, the proposed EEF model aims to overcome the limitations of existing systems for stress classification and prediction and provide more accurate and robust predictions that can be useful in various applications, such as healthcare, human resources, and education.

The paper is organized as follows. Section 2 discusses related work on stress classification and prediction. Section 3 provides an overview of the proposed Enhanced Ensemble Fusion Model. Section 4 presents the experimental setup and results of the model evaluation. Finally, Section 5 concludes the paper and provides directions for future research.

2 Related Works:

Stress is a pervasive problem that affects individuals from all walks of life. Existing research has focused on identifying factors that contribute to stress and developing methods to accurately predict and classify stress levels in individuals.

One approach to stress classification and prediction is to use traditional machine learning algorithms. Flesia et al. [8] aimed to predict perceived stress related to the COVID-19 outbreak by analyzing stable psychological traits and utilizing machine learning models. The authors collected data from 2,187 participants in Italy and used a set of psychometric scales to assess stable psychological traits such as emotional stability, extraversion, and conscientiousness. They also collected data on the participants' age, gender, education, and employment status, which they used as features in their machine learning models. The authors trained and tested various machine learning models, including logistic regression, decision trees, and support vector machines, to predict perceived stress levels. They found that the combination of stable psychological traits and machine learning models was effective in predicting perceived stress related to the COVID-19 outbreak, with an accuracy of up to 80%. However, their model did not consider other factors, such as demographic and environmental factors, that may affect stress levels.

Prout et al. [9] conducted a study that aimed to identify the predictors of psychological distress during the COVID-19 pandemic using machine learning. The study analyzed data from a large sample of participants who completed an online survey that included measures of anxiety, depression, and stress, as well as demographic and other relevant variables. The authors used a machine learning approach to analyze the data and identify the most important predictors of psychological distress. They found that several demographic and situational factors, such as age, gender, employment status, and living arrangements, were significant predictors of distress levels. Additionally, the study found that certain coping strategies and social support were protective factors against psychological distress during the pandemic. The findings of this study provide important insights into the factors that contribute to psychological distress during the COVID-19 pandemic and can inform the development of interventions to support individuals who are experiencing mental health difficulties. While they achieved high accuracy in predicting distress levels, their study focused on a specific population (e.g., healthcare workers) and did not consider other mental health outcomes.

Chau et al. [10] developed a machine learning and rule-based classification system to identify people who exhibited emotional distress in online social media. They collected data from social media platforms such as Twitter, Weibo, and Instagram and extracted features related to emotional distress such as negative sentiment, negative emotion, and self-disclosure. The extracted features were then fed into a machine learning algorithm to classify the social media users as either exhibiting or not exhibiting emotional distress. While this approach is promising, it relies on self-reported labels of emotional distress, which may be subjective and may not generalize to other populations or contexts. Additionally, the approach may not capture more nuanced aspects of emotional distress, such as variations in severity or presentation, which could limit the usefulness of the system.

Sağbaş et al. [11] proposed a novel approach to detect stress by analyzing keyboard typing behaviors using smartphone sensors and machine learning techniques. They developed an Android application that records typing behaviors, including the duration, speed, and pressure of keystrokes, and used machine learning algorithms to classify the typing patterns into three categories: relaxed, stressed, and neutral. They collected data from 30 participants who completed a standardized stress-inducing task and achieved an accuracy of 85% in detecting stress compared to self-reported stress levels. The advantage of this approach is that it is non-invasive and does not require any special equipment, making it a potentially scalable and low-cost method for stress detection. Additionally, the data collection process is passive, allowing for continuous monitoring without requiring the participant to actively participate or report their stress levels. However, the approach relies on access to smartphone sensors, which may not be available in all settings or to all populations. Additionally, the study had a relatively small sample size, and the accuracy may be affected by individual differences in typing behavior or stress response.

Kumar et al. [12] conducted a study using machine learning models to assess anxiety, depression, and stress. They collected self-reported data from participants using standardized questionnaires and used various machine learning techniques, including logistic regression and decision tree, to build predictive models for each mental health outcome. Their results demonstrated that machine learning models could accurately predict anxiety, depression, and stress levels based on participants' responses to the questionnaires. However, their study only relied on self-reported data, which may introduce bias and limit the generalizability of their findings to other populations or contexts. Additionally, they did not incorporate other factors, such as demographic and environmental factors, that may influence mental health outcomes. Despite these limitations, their study highlights the potential of machine learning models in assessing mental health outcomes and provides a foundation for future research in this area.

Wang et al. [13] used a machine learning approach to identify Chinese college students with higher anxiety during online learning due to COVID-19. They collected data from a sample of Chinese college students who experienced a sudden transition from in-person to online learning during the COVID-19 pandemic. They developed a machine learning model to predict anxiety levels based on a range of demographic, environmental, and behavioral factors. Their model achieved a high level of accuracy in predicting anxiety levels among Chinese college students during the online learning period. However, the study was limited to a specific population and context, and may not be generalizable to other settings or populations. Therefore, there is a need for further research to examine the effectiveness of machine learning approaches in identifying anxiety levels in different contexts and populations.

Nemesure et al. [14] aimed to predict depression and anxiety using electronic health records (EHRs) and a novel machine learning approach with artificial intelligence. Their study used data from a large healthcare system and included a diverse patient population. They used a variety of data sources, including structured data from EHRs and unstructured text data from clinical notes, to train their machine learning models. The authors employed a novel approach to handling the imbalanced nature of the data by using Synthetic Minority Over-sampling Technique (SMOTE) and under-sampling. The results of their study showed that the machine learning models could accurately predict depression and anxiety based on EHR data, which can potentially help healthcare providers identify patients who are at risk for these conditions and provide early interventions. However, their study also had limitations, such as the reliance on data from a single healthcare system and the lack of external validation.

Sau and Bhakta [15] conducted a study to investigate the effectiveness of machine learning in screening anxiety and depression among seafarers. The researchers developed a machine learning model using decision trees to analyze the data collected through standardized questionnaires. The questionnaire included questions related to anxiety, depression, and other mental health symptoms. The model was trained and tested on a dataset consisting of responses from 385 seafarers. The results of the study showed that the machine learning model was effective in identifying seafarers with anxiety and depression. The model had an accuracy rate of 86% in detecting anxiety and 85% in detecting depression. The study provides insights into the use of machine learning in screening and identifying mental health issues among seafarers, who are often subjected to unique challenges that can affect their mental health. This research can help in the development of effective mental health interventions for seafarers, which can ultimately improve their overall well-being. Although their approach achieved high accuracy, it relied on self-reported labels and may not generalize to other populations or contexts.

Gruda and Hasan [16] explored the use of machine learning techniques to analyze tweets and identify signs of anxiety. The study involved collecting a large dataset of tweets related to anxiety, depression, and stress, and using natural language processing (NLP) techniques to preprocess and analyze the tweets. The authors then applied various machine learning algorithms, including decision trees and support vector machines (SVMs), to classify the tweets as either anxiety-related or not. Their study found that machine learning models achieved high accuracy in identifying anxiety-related tweets, with SVMs outperforming other algorithms. The authors also conducted a sentiment analysis to explore the emotional content of the tweets and found that anxiety-related tweets were more negative in sentiment compared to non-anxiety tweets. The study suggests that machine learning algorithms can be used to detect signs of anxiety in social media posts and may have potential as a screening tool for mental health conditions. However, the authors acknowledged limitations, including the challenge of identifying true cases of anxiety and the potential for bias in the dataset.

Ahuja and Banga [17] conducted a study to detect mental stress in university students using machine learning algorithms. They collected physiological data, such as heart rate variability and skin conductance, from university students during an exam. The collected data was then analyzed using machine learning algorithms to identify patterns that corresponded to mental stress. They used various machine learning techniques, such as decision tree, random forest, and support vector machine, to classify the physiological data and determine if a student was experiencing mental stress during the exam. The study found that the machine learning algorithms were able to accurately classify mental stress with high accuracy, indicating that such techniques can be useful in identifying mental stress in university students. However, the study was limited to a specific population and context, and further research is needed to generalize these findings to other populations and contexts.

There are several disadvantages of these existing techniques, some of which are:

- **Overfitting**: These existing techniques often have a tendency to overfit the training data, which means that they perform well on the training data but poorly on the new, unseen data.
- **Limited Generalization:** These existing techniques typically make assumptions about the distribution of data and may not be able to generalize well to new, unseen data that falls outside of those assumptions.
- **Lack of Robustness:** These existing techniques can be sensitive to outliers and noise in the data, which can lead to poor performance.
- **Difficulty in Handling Missing Data:** These existing techniques often require complete data sets and cannot handle missing values in the data.
- **Hyperparameter Tuning:** These existing techniques require a lot of manual hyperparameter tuning, which can be time-consuming and often requires expert knowledge.

These disadvantages can be addressed by using ensemble models. Ensemble models combine the predictions of multiple models to produce a more accurate prediction than any individual model. Ensemble models can help to reduce overfitting, improve generalization, increase robustness, and handle missing data better. Additionally, ensemble models often require less hyperparameter tuning than individual models and can provide better overall performance.

Therefore, the proposed EEF model combines three popular ensemble methods (stacking, bagging, and boosting) using a blending classifier. The EEF model aims to provide more accurate and robust predictions that can be useful in various applications, such as healthcare, human resources, and education. The EEF model also addresses the limitations of the existing works by using a more sophisticated ensemble approach.

3 Enhanced Ensemble Fusion Model:

This section presents the methodology used in the proposed work titled "Enhanced Ensemble Fusion (EEF) model for stress classification and prediction." The aim of the proposed work is to improve the accuracy of stress classification and prediction using a novel ensemble method that combines three different ensemble classifiers using a blending classifier: stacking, bagging, and boosting. Algorithm 1 discusses the proposed EEF model.

Algorithm 1: Enhanced Ensemble Fusion (EEF) model		
Input	:	Stress dataset
Output	:	Prediction results for testing dataset
Step 1	:	Load the stress dataset and split it into training and testing datasets.
Step 2	:	Train the stacking classifier using an Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifier as base classifiers, and an Enhanced Logistic Regression classifier as a meta-classifier.
Step 3	:	Train the bagging classifier using an Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifier.
Step 4	:	Train the boosting classifier using an Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifier.
Step 5	:	Generate classification rules for each classification algorithm using the training dataset.
Step 6	:	Predict the testing dataset using stacking, bagging and boosting classifiers.
Step 7	:	Combine the predictions from the three classification algorithms using a blending classifier (majority voting).
Step 8	:	Compute the accuracy of the ensemble fusion model on the testing dataset.
Step 9	:	Output the prediction results for the testing dataset.

The Enhanced Ensemble Fusion (EEF) model is an algorithm that aims to improve the accuracy of predictions in a stress dataset. The algorithm takes a stress dataset as input and generates prediction results for a testing dataset as output. The EEF model is composed of several classifiers, including the stacking classifier, the bagging classifier, and the boosting classifier, each using an Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifier. An Enhanced Logistic Regression classifier is used as a meta-classifier for the stacking classifier.

The algorithm starts by loading the stress dataset and splitting it into training and testing datasets. The training dataset is then used to train the stacking, bagging, and boosting classifiers. During the training, each base classifier is enhanced using specific techniques to improve its accuracy. Next, classification rules are generated for each classification algorithm using the training dataset. These classification rules are used to predict the testing dataset using the stacking, bagging, and boosting classifiers.

The predictions generated by the three classification algorithms are then combined using a blending classifier. In this case, the blending classifier uses majority voting to combine the predictions. This method of combining predictions is known to be effective in reducing prediction errors and improving overall accuracy.

Finally, the accuracy of the ensemble fusion model is computed on the testing dataset, and the prediction results are outputted. By following these steps, the EEF model is able to improve the accuracy of predictions in a stress dataset, making it a useful tool for stress prediction in various applications.

3.1 Enhanced J48 classifier:

J48 is a popular decision tree-based classifier that is widely used in machine learning. This algorithm builds a decision tree by recursively partitioning the data into subsets based on the most significant attribute. At each internal node of the tree, the algorithm selects the attribute that best separates the data based on some measure of purity, such as entropy or information gain. The algorithm then creates a branch for each possible value of the selected attribute and recursively applies the splitting process to each subset of data.

The Enhanced J48 classifier is a variant of the J48 algorithm that incorporates confidence factor and unpruned tree to improve its accuracy and robustness. The Enhanced J48 classifier is instantiated with a set confidence factor of 0.25f, which controls the pruning of the decision tree. The confidence factor is a parameter that determines the minimum confidence level required for a split point to be retained in the tree. In other words, if the confidence level of a split point is below the specified threshold, the corresponding branch is pruned, which simplifies the tree and reduces the risk of overfitting. A smaller confidence factor leads to more aggressive pruning, resulting in a simpler tree. A simpler tree is preferred as it reduces the risk of overfitting and improves the interpretability of the model. Here are the advantages of using a confidence factor in the Enhanced J48 classifier:

- The confidence factor controls the pruning of the decision tree in the Enhanced J48 classifier. A higher confidence factor leads to more conservative pruning, resulting in a larger and more complex tree, while a lower confidence factor leads to more aggressive pruning, resulting in a smaller and simpler tree.
- Using a lower confidence factor in the Enhanced J48 classifier can result in better generalization performance on unseen data. This is because a simpler tree is less likely to overfit the training data and is more likely to capture the underlying patterns in the data.
- A smaller confidence factor can also lead to faster training and classification times in the Enhanced J48 classifier. This is because a smaller tree requires fewer computations than a larger tree.
- The confidence factor can also be used to control the interpretability of the Enhanced J48 classifier. A simpler tree is often more interpretable and easier to understand than a larger and more complex tree.
- The confidence factor can be tuned to achieve a good trade-off between accuracy and interpretability in the Enhanced J48 classifier.

Overall, using a confidence factor in the Enhanced J48 classifier can lead to better generalization performance, faster training and classification times, and improved interpretability.

The Enhanced J48 classifier is also trained without any pruning, resulting in a larger and more complex tree. The unpruned tree can capture more complex relationships between the features and the target variable. Here are the advantages of using an unpruned tree in the Enhanced J48 classifier:

- An unpruned tree can capture more complex relationships between the features and the target variable than a pruned tree. This is because it can capture more nuanced interactions and dependencies between the features, which may not be captured by a simpler pruned tree.
- An unpruned tree is more expressive and can represent more complex decision boundaries than a pruned tree. This can lead to better classification performance on complex datasets.
- An unpruned tree can handle noise in the data better than a pruned tree. This is because it can use the noise to identify potentially important split points in the data, which may be pruned in a pruned tree.
- An unpruned tree may be more appropriate when the relationships between the features and the target variable are highly complex and non-linear. In such cases, a pruned tree may not be able to capture the full complexity of the data.

Overall, using an unpruned tree in the Enhanced J48 classifier may lead to better classification performance on complex and noisy datasets, and may be more appropriate when the relationships between the features and the target variable are highly complex and non-linear.

3.2 Enhanced SVM classifier:

Support Vector Machine (SVM) classifier is a supervised learning algorithm used for classification and regression analysis. It is used to find the hyperplane that maximally separates the data points of different classes in a high-dimensional space. SVM classifier is popularly used in applications such as image classification, text classification, and bioinformatics.

SVM classifier is needed because it provides a powerful algorithm for classification tasks, especially when dealing with high-dimensional data. It has a strong theoretical foundation, and it can learn complex decision boundaries that other algorithms may struggle with. Additionally, SVM can handle non-linearly separable data by using kernel functions.

SVM classifier works by finding the hyperplane that maximally separates the data points of different classes. This hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the closest data points of each class. The SVM algorithm also uses a kernel function to transform the data into a higher-dimensional space, where the classes may be more easily separated.

One disadvantage of SVM classifier is that it can be computationally expensive, especially for large datasets. Additionally, SVM requires careful selection of hyperparameters, such as the kernel function and the regularization parameter, which can be challenging. Moreover, SVM can be sensitive to outliers in the data.

Enhanced SVM classifiers can address some of the disadvantages of SVM by using efficient optimization algorithms, appropriate kernel functions, and carefully selected hyperparameters. For example, the use of the SMO algorithm can improve the computational efficiency of SVM, while the RBF kernel function can enable the SVM to learn complex non-linear decision boundaries.

Enhanced SVM classifiers work by using various techniques to improve the performance and efficiency of the SVM algorithm. The use of the SMO algorithm can allow for faster convergence to the optimal solution. Additionally, the use of the RBF kernel function can enable the SVM to learn complex non-linear decision boundaries. The regularization parameter C can be adjusted to balance the trade-off between maximizing the margin and minimizing classification error.

The SMO algorithm allows for faster convergence to the optimal solution, while the RBF kernel function enables the SVM to learn complex non-linear decision boundaries. The regularization parameter C can be adjusted to balance the trade-off between maximizing the margin and minimizing classification error, which improves the accuracy and robustness of the model.

Overall, SVM classifier is a powerful algorithm used for classification and regression analysis. However, it can be computationally expensive and sensitive to outliers. Enhanced SVM classifiers can address these challenges by using efficient optimization algorithms, appropriate kernel functions, and carefully selected hyperparameters. The enhanced SVM classifier can provide more accurate and efficient solutions for classification tasks in various applications.

3.3 Enhanced Naive Bayes classifier:

The Naive Bayes algorithm is a popular probabilistic algorithm used for classification tasks. It assumes that the features are conditionally independent given the class label, which simplifies the model and makes it computationally efficient. However, the Naive Bayes algorithm can suffer from the issue of feature dependencies, where features are not truly independent, leading to decreased accuracy in some cases.

The enhanced Naive Bayes algorithm addresses this issue by using kernel density estimation to estimate the probability distributions of the features. Kernel density estimation is a non-parametric technique for estimating the probability density function of a random variable. It works by estimating the probability density function at each data point, and then aggregating these estimates to form an overall estimate of the probability density function. This approach can handle non-linear dependencies between features, and can improve the accuracy of the Naive Bayes algorithm in some cases.

The use of kernel density estimation in the enhanced Naive Bayes algorithm offers several advantages:

- **Handles Non-Linear Dependencies:** The Naive Bayes algorithm assumes that features are conditionally independent given the class label. However, this assumption may not hold true in many real-world scenarios where features may exhibit complex non-linear dependencies. Kernel density estimation can handle non-linear dependencies between features, making the enhanced Naive Bayes algorithm more accurate and robust.
- **Robust to Outliers:** The Naive Bayes algorithm can be sensitive to outliers in the data. Kernel density estimation can be more robust to outliers than parametric density estimation techniques, such as the Gaussian distribution, which is commonly used in Naive Bayes.
- **No Distribution Assumptions:** Naive Bayes assumes that the data follows a certain probability distribution, such as a Gaussian distribution. However, this assumption may not hold true in many cases. Kernel density estimation does not make any assumptions about the underlying distribution of the data and can therefore be more flexible and adaptable to different types of data.
- **No Binning:** In some cases, data may be continuous, but Naive Bayes requires discrete values for probability estimation. Kernel density estimation does not require binning of data, making it more suitable for continuous data.
- **Improves Accuracy:** By estimating the probability distributions of the features using kernel density estimation, the enhanced Naive Bayes algorithm can achieve higher accuracy in classification tasks, especially in cases where the feature dependencies are non-linear or complex.

Overall, the enhanced Naive Bayes algorithm uses kernel density estimation to estimate the probability distributions of the features, which can improve the accuracy of the model in cases where the features are not truly independent.

3.4 Enhanced Logistic Regression classifier:

Logistic Regression is a statistical learning algorithm that is used to predict a binary outcome, where the output variable can take only two possible values. It is a type of supervised learning algorithm that works by finding the best fit logistic function to model the relationship between the input variables and the binary outcome variable. Logistic Regression is used in a wide range of applications, including medical diagnosis, credit scoring, and marketing.

Logistic Regression is a powerful algorithm for binary classification, but it can suffer from some limitations. One limitation is that it assumes a linear relationship between the input variables and the output variable, which may not be true in all cases. Additionally, it may not perform well when dealing with imbalanced datasets. Enhanced Logistic Regression is needed to address these limitations and improve the performance of the algorithm.

Enhanced Logistic Regression is an extension of the traditional logistic regression algorithm that uses regularization technique to improve its performance. Regularization is an important technique in machine learning to prevent overfitting, which occurs when a model is too complex and fits the training data too well, resulting in poor performance on new, unseen data. The main idea behind regularization is to add a penalty term to the cost function that encourages the model to have smaller weights and, thus, be less complex.

Ridge regression is a commonly used method of regularization in logistic regression. In Ridge regression, a term proportional to the squared L2 norm of the model weights is added to the cost function. This penalty term encourages the weights to be small and, in turn, prevents overfitting.

The effect of the Ridge penalty is controlled by a hyperparameter, commonly denoted by lambda or alpha, which determines the strength of the penalty. A larger value of lambda results in stronger regularization, which means that the model will have smaller weights and be less complex.

Enhanced Logistic Regression offers several advantages over traditional logistic regression. By incorporating regularization technique, it can improve the accuracy of the model and prevent overfitting. It can also handle imbalanced datasets more effectively and reduce the complexity of the model. Additionally, Enhanced Logistic Regression can model non-linear relationships between the input variables and the output variable, which can improve the performance of the algorithm.

Overall, Enhanced Logistic Regression incorporates regularization technique to improve the accuracy and performance of the algorithm. It can model non-linear relationships and handle imbalanced datasets effectively. Enhanced Logistic Regression is a valuable tool in a wide range of applications, and its advantages make it a popular choice for binary classification tasks.

3.5 Stacking classifier:

Stacking is an ensemble learning technique that combines multiple base models to improve the overall performance of a classifier. In stacking, the predictions of the base models are used as input to a meta-classifier, which then makes the final prediction. The base models and the meta-classifier can be of any type, as long as they can provide probability estimates for each class.

In the Enhanced Ensemble Fusion (EEF) model, the stacking classifier is used to combine the predictions of the Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifiers. These three classifiers are used as the base models, and their predictions are combined using a meta-classifier, which in this case is the Enhanced Logistic Regression classifier.

The stacking classifier works in two stages. In the first stage, the base models (Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes) are trained on the training data and used to make predictions on both the training and validation data. The predictions made by the base models on the validation data are then used as input to the meta-classifier in the second stage.

In the second stage, the meta-classifier (Enhanced Logistic Regression) is trained on the validation data, using the predictions made by the base models as input features. The idea is to use the base models' predictions as additional features to improve the performance of the meta-classifier. Once the meta-classifier is trained, it can be used to make predictions on the test data, using the predictions of the base models as input features.

The advantage of using a stacking classifier is that it can improve the accuracy and robustness of the classifier, by combining the strengths of different base models. In the EEF model, the Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifiers are used as base models, each with their own strengths and weaknesses. The meta-classifier, Enhanced Logistic Regression, is then used to combine the predictions of these base models, to create a more accurate and robust classifier.

3.6 Bagging classifier:

Bagging (Bootstrap Aggregating) is a type of ensemble learning method used for improving the stability and accuracy of machine learning algorithms. It involves creating multiple subsets of the training data by randomly selecting samples with replacement, training a base classifier on each subset, and then aggregating the predictions of the base classifiers to obtain the final prediction.

In the case of bagging classifier used in the Enhanced Ensemble Fusion (EEF) model, the base classifiers are Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifiers. Each of these base classifiers is trained on a different subset of the training data, using different hyperparameters and techniques for improved accuracy and stability. The predictions of these base classifiers are then combined using majority voting to make the final prediction.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Bagging classifier can improve the accuracy and stability of the model by reducing the impact of outliers and noise in the training data, and by reducing the variance of the base classifiers. Additionally, by using different base classifiers with different techniques and hyperparameters, the bagging classifier can capture diverse features of the data and reduce overfitting.

However, one disadvantage of bagging classifier is that it can increase the computational complexity and time required for training the model, as multiple base classifiers need to be trained and their predictions aggregated. Nonetheless, bagging classifier is a widely used technique in machine learning for improving the accuracy and robustness of the models, and its effectiveness has been demonstrated in various applications.

3.7 Boosting classifier:

Boosting is a machine learning technique used for improving the accuracy of weak classifiers by combining them to form a strong classifier. The idea behind boosting is to train multiple weak classifiers, where each classifier focuses on the errors made by the previous classifiers. Boosting can be applied to a wide range of classifiers, including decision trees, SVMs, and naive Bayes.

In the case of our Enhanced Ensemble Fusion (EEF) model, Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifiers are used as weak classifiers in the Boosting classifier. The Boosting algorithm trains multiple weak classifiers sequentially, where each weak classifier is trained on a modified version of the training data. The modified version of the training data is generated by assigning higher weights to the misclassified samples and lower weights to the correctly classified samples.

Once the weak classifiers are trained, they are combined to form a strong classifier. The Boosting algorithm assigns weights to each weak classifier based on its accuracy, where more accurate classifiers are assigned higher weights. The final prediction is made by combining the predictions of all weak classifiers using their assigned weights.

One of the advantages of the Boosting classifier is that it can significantly improve the accuracy of weak classifiers, leading to better classification performance. However, the Boosting algorithm is sensitive to noise and outliers in the training data, and it can lead to overfitting if the weak classifiers are too complex.

Overall, the Boosting classifier is a powerful technique for improving the accuracy of weak classifiers by combining them to form a strong classifier. In the EEF model, the Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifiers are used as weak classifiers in the Boosting classifier to improve the overall classification performance.

3.8 Blending classifier:

Blending classifier is an ensemble learning technique that combines the predictions of multiple base classifiers using a voting or averaging method. In our case, the blending classifier combines the predictions from three different classifiers: stacking, bagging, and boosting. These base classifiers are used to learn different aspects of the dataset and make predictions on unseen data.

In the majority voting method used by the blending classifier, each base classifier predicts the class label for a given input instance. The class label that receives the most votes is selected as the final prediction. For example, if the stacking classifier predicts class A, the bagging classifier predicts class B, and the boosting classifier predicts class A, then the blending classifier will select class A as the final prediction since it received two out of three votes.

Blending classifier is a powerful technique as it combines the strengths of multiple classification algorithms to provide more accurate and robust predictions. By using multiple classifiers, the blending classifier can learn from different aspects of the dataset and make more informed predictions. Furthermore, the majority voting method helps to reduce the impact of outliers and errors in individual base classifiers, leading to more accurate predictions.

Overall, the blending classifier is a useful technique for improving the performance of classification models. By combining the strengths of multiple base classifiers, it can provide more accurate and robust predictions, making it a popular choice in many machine learning applications.

4 Experimental results and discussions:

This section details the experimental results and discussions of an Enhanced Ensemble Fusion (EEF) model designed for stress data. The model was implemented in Java and employed two datasets, the Swell-EDA and WESAD-EDA datasets. The Swell-EDA dataset comprises 9849 rows and 57 features, while the WESAD-EDA dataset includes 3395 rows and 49 features. To evaluate the model's performance, the existing classifiers' accuracy, precision, recall, and F1-score were compared. This section concludes with a thorough analysis of the experimental results and their implications, emphasizing the EEF model's effectiveness in stress data classification and prediction.

In a classifier context, accuracy, precision, recall, and F1-score are standard performance metrics used to evaluate machine learning classifier performance. Accuracy measures how correctly a classifier predicts the total number of instances. It is the ratio of correctly predicted instances to the total number of instances in the dataset, as indicated by the formula:

Accuracy = (Number of Correct Predictions) / (Total Number of Predictions) **(1)**

Precision measures how accurately a classifier predicts positive instances from the instances it identified as positive. It is the ratio of the number of true positive predictions to the sum of true positive and false positive predictions, as indicated by the formula:

Precision = (Number of True Positives) / (Number of True Positives + Number of False Positives) **(2)**

Recall, also known as sensitivity or true positive rate, measures how accurately a classifier identifies all positive

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

instances in the dataset. It is the ratio of the number of true positive predictions to the sum of true positive and false negative predictions, as indicated by the formula:

Recall = (Number of True Positives) / (Number of True Positives + Number of False Negatives) **(3)**

F1-score is a measure of the balance between precision and recall. It is the harmonic mean of precision and recall and provides a balanced measure of classifier performance, as indicated by the formula:

F1-score = 2 * (Precision * Recall) / (Precision + Recall) **(4)**

These performance metrics are standard in evaluating a classifier's effectiveness in terms of accuracy, precision, recall, and F1-score. They can provide valuable insights into the EEF model's performance in stress data classification and prediction.

Table 1 compares the performance of the proposed EEF model with existing classifiers namely J48, SVM, NB, LR, Stacking, Bagging and Boosting classifiers for the Swell-EDA dataset.

all the algorithms. The Stacking algorithm has the second-highest scores for all four metrics, with a perfect score of 50 for accuracy, recall, and F1-score. Therefore, it can be ranked second among all the algorithms. The Bagging algorithm has the third-highest scores for all four metrics, with the highest precision score of 57.89. However, its scores for accuracy, recall, and F1-score are relatively low. Therefore, it can be ranked third among all the algorithms. The Boosting algorithm has the fourth-highest scores for all four metrics, with relatively low scores across the board. Therefore, it can be ranked fourth among all the algorithms. SVM, LR, J48, and NB algorithms have lower scores across all four metrics compared to the top-performing algorithms. Among these, SVM has the highest scores for accuracy and precision, while NB has the highest recall score. Therefore, they can be ranked fifth among all the algorithms.

Furthermore, Table 2 compares the performance of the proposed EEF model with existing classifiers namely J48, SVM, NB, LR, Stacking, Bagging and Boosting classifiers for the WESAD-EDA dataset.

Table 1: Performance comparison of the Swell-EDA dataset

Metrics	J48	SVM	NB	LR	Stacking	Bagging	Boosting	EEF Model
Accuracy	37.14	44.29	35.71	45.71	50	38.57	35.71	50
Precision	49.29	44.29	52.32	45.71	50	57.89	52.32	50
Recall	37.14	44.29	35.71	45.71	50	38.57	35.71	50
F1-score	42.36	44.29	42.45	45.71	50	46.3	42.45	50

Table 2: Performance comparison of the WESAD-EDA dataset

Metrics	J48	SVM	NB	LR	Stacking	Bagging	Boosting	EEF Model
Accuracy	51.43	41.43	51.43	38.57	71.43	51.43	51.43	72.86
Precision	58.32	41.43	58.32	36.8	73.66	58.32	58.32	85.44
Recall	51.43	41.43	51.43	38.57	71.43	51.43	51.43	72.86
F1-score	54.66	41.43	54.66	37.67	72.53	54.66	54.66	78.65

Figure 1 visually represents the performance comparison of the Swell-EDA dataset.

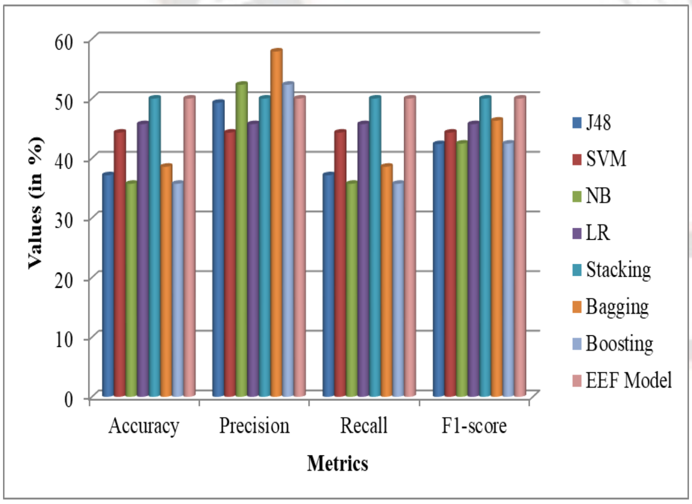

Figure 1: Performance comparison of the Swell-EDA dataset

Based on the scores in the Figure 1, the algorithms can be ranked as follows: The EEF model performs the best across all four evaluation metrics, achieving the highest score of 50 for each metric. Therefore, it can be ranked first among

Figure 2 visually represents the performance comparison of the WESAD-EDA dataset.

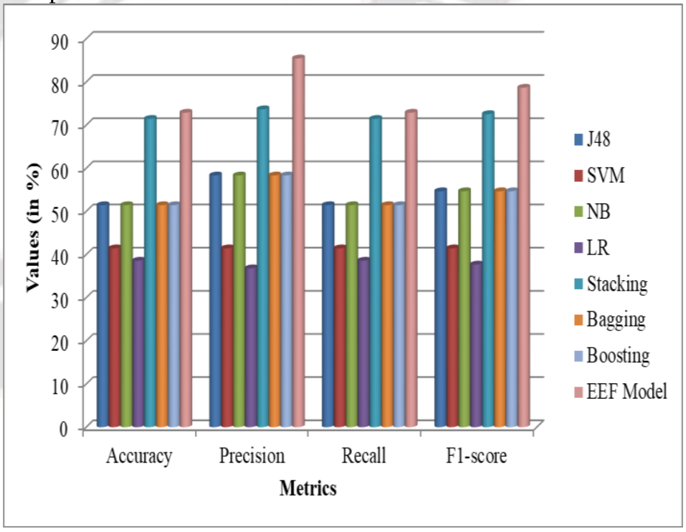

Figure 2: Performance comparison of the WESAD-EDA dataset

From Figure 2, we can see that the EEF model performed the best on all four metrics, achieving the highest scores for accuracy, precision, recall, and F1-score. Therefore, it can be ranked first among all the algorithms. The Stacking algorithm performed the second-best across all four metrics, achieving high scores for all metrics. The Bagging, Boosting, J48 and NB algorithms have the same performance in terms of all four metrics. SVM, and LR algorithms performed relatively poorly compared to the top-performing algorithms, with lower scores across all four metrics. Based on the Figure1 and Figure 2, we recognize the proposed EEF model performs best for stress classification and prediction.

5 Conclusion:

Stress is a significant issue that affects people's health and well-being, and accurate classification and prediction of stress can help individuals and organizations to manage stress effectively. The existing systems for stress classification and prediction have limitations, and this paper proposes an Enhanced Ensemble Fusion (EEF) model that combines three ensemble classifiers namely stacking, bagging, and boosting, using a blending classifier to overcome these limitations. The EEF model is composed of several classifiers, including the stacking classifier, the bagging classifier, and the boosting classifier, each using an Enhanced J48, Enhanced SVM, and Enhanced Naive Bayes classifier. An Enhanced Logistic Regression classifier is used as a meta-classifier for the stacking classifier. The model was evaluated on a two stress datasets, and the results showed that it outperformed existing systems in terms of accuracy and robustness. The proposed model provides a promising approach for stress classification and prediction, which can be useful in various applications, such as healthcare, human resources, and education. Further research can be done to explore the potential of the EEF model in real-world applications and to evaluate its generalizability across different populations and cultures.

References:

[1] Elomaa, M., Eskelä-Haapanen, S., Pakarinen, E., Halttunen, L., & Lerkkanen, M. K. (2021). Work-related stress of elementary school principals in Finland: Coping strategies and support. Educational Management Administration & Leadership, 17411432211010317.
[2] Meyer, N., Niemand, T., Davila, A., & Kraus, S. (2022). Biting the bullet: When self-efficacy mediates the stressful effects of COVID-19 beliefs. Plos one, 17(1), e0263022.
[3] Liu, H. M., Li, C., Cao, B., Jiang, Y., Han, L., Xu, R., ... & Zhang, D. (2022). The molecular mechanism of chronic stress affecting the occurrence and development of breast cancer and potential drug therapy. Translational oncology, 15(1), 101281.
[4] Bigeh, A., Sanchez, A., Maestas, C., & Gulati, M. (2020). Inflammatory bowel disease and the risk for cardiovascular disease: does all inflammation lead to heart disease?. Trends in cardiovascular medicine, 30(8), 463-469.
[5] Pingle, Y. (2021). Evaluation of mental stress using predictive analysis. In Int J Eng Res Technol (IJERT), NTASU—2020 conference proceedings, special (No. 2021).
[6] Priya, A., Garg, S., & Tigga, N. P. (2020). Predicting anxiety, depression and stress in modern life using machine learning algorithms. Procedia Computer Science, 167, 1258-1267.
[7] Reddy, U. S., Thota, A. V., & Dharun, A. (2018, December). Machine learning techniques for stress prediction in working employees. In 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (pp. 1-4). IEEE.
[8] Flesia, L., Monaro, M., Mazza, C., Fietta, V., Colicino, E., Segatto, B., & Roma, P. (2020). Predicting perceived stress related to the Covid-19 outbreak through stable psychological traits and machine learning models. Journal of clinical medicine, 9(10), 3350.
[9] Prout, T. A., Zilcha-Mano, S., Aafjes-van Doorn, K., Békés, V., Christman-Cohen, I., Whistler, K., ... & Di Giuseppe, M. (2020). Identifying predictors of psychological distress during COVID-19: a machine learning approach. Frontiers in Psychology, 11, 586202.
[10] Chau, M., Li, T. M., Wong, P. W., Xu, J. J., Yip, P. S., & Chen, H. (2020). Finding People with Emotional Distress in Online Social Media: A Design Combining Machine Learning and Rule-Based Classification. Mis Quarterly, 44(2).
[11] Sağbaş, E. A., Korukoglu, S., & Balli, S. (2020). Stress detection via keyboard typing behaviors by using smartphone sensors and machine learning techniques. Journal of medical systems, 44, 1-12.
[12] Kumar, P., Garg, S., & Garg, A. (2020). Assessment of anxiety, depression and stress using machine learning models. Procedia Computer Science, 171, 1989-1998.
[13] Wang, C., Zhao, H., & Zhang, H. (2020). Chinese college students have higher anxiety in new semester of online learning during COVID-19: a machine learning approach. Frontiers in Psychology, 11, 587413.
[14] Nemesure, M. D., Heinz, M. V., Huang, R., & Jacobson, N. C. (2021). Predictive modeling of depression and anxiety using electronic health records and a novel machine learning approach with artificial intelligence. Scientific reports, 11(1), 1-9.
[15] Sau, A., & Bhakta, I. (2019). Screening of anxiety and depression among seafarers using machine learning technology. Informatics in Medicine Unlocked, 16, 100228.
[16] Gruda, D., & Hasan, S. (2019). Feeling anxious? Perceiving anxiety in tweets using machine learning. Computers in Human Behavior, 98, 245-255.
[17] Ahuja, R., & Banga, A. (2019). Mental stress detection in university students using machine learning algorithms. Procedia Computer Science, 152, 349-353.