An Efficient Approach for Lie Detection using EEG Signals and Ensemble Classification

Submitted in partial fulfilment of the requirements of the degree of
Bachelor of Technology

By

Naveen Shankar	**187239**
Naman Mittal	**187238**
Hemant Kumar Singh	**187220**

Supervisor:
Dr. Ramalingaswamy Cheruku
Assistant Professor
NIT Warangal

Computer Science and Engineering
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
2021 - 2022

APPROVAL SHEET

The project work entitled by **An Efficient Approach for Lie Detection using EEG Signals and Ensemble Classification** by **Naveen Shankar (187239) Naman Mittal (187238) and Hemant Kumar Singh (187220)** is approved for the degree of Bachelor of Technology in ComputerScience and Engineering.

Examiners

____

____

____

Supervisor

____

Dr. Ramalingaswamy Cheruku
Assistant Professor

Chairman

____

Prof. S. Ravi Chandra
Department of Computer Science and Engineering

Date: 14/05/2022
Place: Warangal

DECLARATION

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from where proper permission has not been taken when needed.

Naveen Shankar - 187239

Naman Mittal - 187238

Hemant Kumar Singh - 187220

Date: 14/05/2022

CERTIFICATE

This is to certify that the project work entitled **An Efficient Approach for Lie Detection using EEG Signals and Ensemble Classification** is a bonafide record of work carried out by **Naveen Shankar(187239), Naman Mittal(187238) and Hemant Kumar Singh(187220)**, submitted to the faculty of Computer Science and Engineering Department in partial fulfilmentof the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at National Institute of Technology, Warangal during the academic year 2021-2022.

Dr. S. Ravi Chandra	**Dr. Ramalingaswamy Cheruku**
Head of the Department	Assistant Professor
Department of CSE	Department of CSE
NIT Warangal	NIT Warangal

ABSTRACT

Electroencephalography (EEG) has proven to be one of the crucial techniques for lie detection. We propose an efficient approach for lie detection using EEG signals and the Ensembling. This approach classifies subjects into guilty and innocent people, using EEG signals and ensemble techniques to achieve higher accuracy. EEG data was collected from ten subjects by placing an emotive headset on their scalp to record brain waves. A component of EEG, known as P300, is used to detect whether a person is telling a lie. During each trial, subjects are shown a specified image and a yes or no option under some time frame. If the image is recognized by the subject, then a P300 event-related potential will be elicited in the brain wave signals. The EEG data is collected and processed into a ten-fold dataset, where each fold contains train and test data. On this dataset, for each subject, several classification models are applied with hyperparameter tuning that is done using k-fold cross-validation. From those models, the best models are selected using the classification ranking technique and, on these models, voting classification has been applied to produce output with an accuracy. Lastly, an average of all the accuracies is taken to get the final accuracy. The proposed solution effectively classifies subjects as guilty or innocent with an accuracy of 86.06%.

Keywords - Electroencephalography, Ensembling, Hyperparameter Tuning, K-fold Cross Validation, Classification, Event related potential.

Contents

List of Figures

List of Table

Chapter 1

Introduction

Lie detection has been a hot topic for quite some while, and its applications have expanded to multiple fields and have proved to be significant. Some physical activities, such as psychology, respond when a subject is faced with a question-and-answer interview. Previously polygraphs were used to measure several physical activities, such as blood pressure, heart rate, respiration, and skin conductance. After the experiment, the examiner compares the measured physiological values to the expected normal values to indicate the subject's degree of honesty. The fact that subjects show anxiety for reasons, but lying beats the test by controlling physiological variables, has long been declared and so it can be employed to detect lies.

Many states have ruled out polygraph test results, stating it is not acceptable legally and unreliable. However, during the past decade, some new technologies have been studied and developed to bring lie detection beyond polygraphs based on electroencephalography (EEG). The EEG signal detects changes in brain waves, which are used to predict lies and verify truths. Although used primarily in medical applications, primarily for epilepsy patients, this technique is also used for forensic purposes to identify a sudden change in brain activity that would signal a highly possible lie due to the changed mental state of the subject being interrogated. There are many modern techniques in lie detection; Among them, the method of detecting lies based on brain electrical activity was first proposed in 2001 by Farwell. The method uses one of the brain's most important EEG components, called the P300. The P300 is a positive peak lasting about 300ms due to the infrequent stimulation of the crank series. In simple terms, P300 is triggered by the subject's brain after encountering some meaningful stimulus. As a result, if the subject was previously aware of the stimulus, the P300 could be reflected and detected in the EEG signal. Our objective is to study the lie detection technique based on human brain signal response during question-answer interviews and design a model to resolve the issue of lie detection without the effects of controlling emotions and physiological expression.

1.1 Electroencephalography

A non-invasive physiological method to record the electrical activity generated by the brain via electrodes placed on the scalp surface is known as Electroencephalography, or EEG. Electrodes are mounted in adjustable caps similar to bathing caps, ensuring that data can be collected from identical scalp positions across all respondents; this is done for faster application.

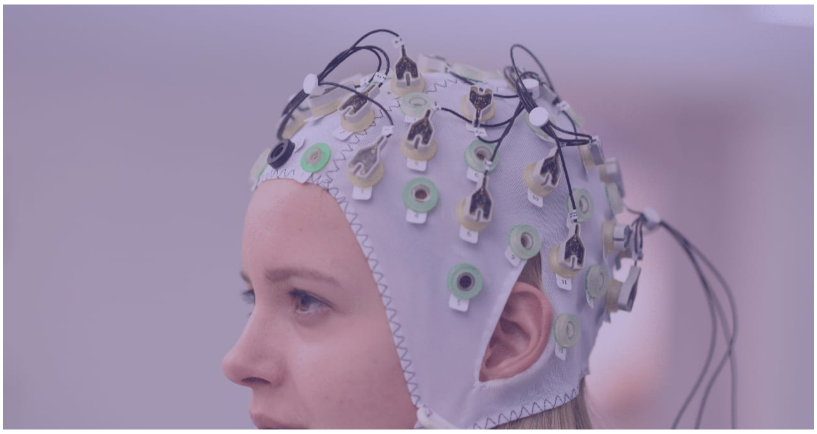


Figure 1.1: The Anatomy of an EEG cap

The brain syncs with thousands of neurons to make a decision, and there is some electrical activity generated due to this. These signals are recorded as EEG. It provides excellent time resolution, allowing you to detect activity within cortical areas even at sub-second timescales.

Since the measurement of voltage recorded by the electrodes is minimal, we store that in digital format and amplify it which would then be used for display in a sequence of voltage values.

1.1.1 Lie Detection Using EEG Signals

Lie detection is the evaluation of oral statements to identify possible deliberate deception. Lie detection can refer to the cognitive process of detecting lies by evaluating the content of messages as well as non-verbal cues. It can also refer to interrogation techniques used in conjunction with techniques to record physiological functions and determine truth and falsehood in response.

When EEG signals are used for lie detection, there is a component called P300 which gets elicited when a person has to make a decision or during recognition. So, if an image has any sort of connection with the subject, then the P300 event-related potential (ERP) response will be seen 300-800ms after the stimuli, i.e., a positive deflection in the EEG signal scan can be seen.

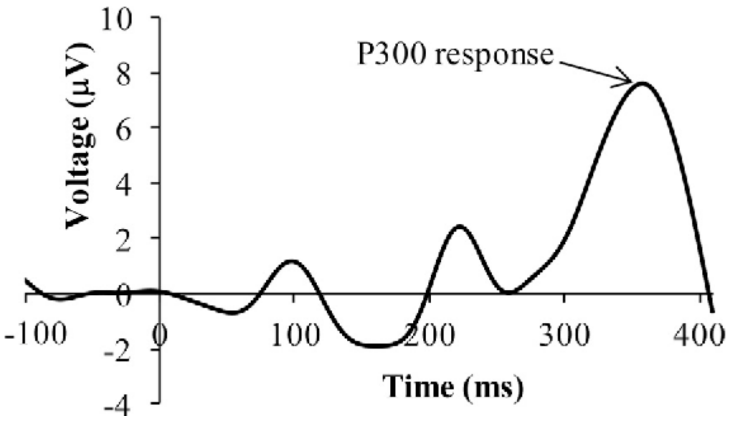


Figure 1.2: P300 Response

1.2 Why Machine Learning?

As discussed earlier, we have a host of traditional methods for lie detection that include the use of multiple devices such as a polygraph, blood pressure monitor, and heartbeat sensor while also monitoring the sweat and respiratory rate. The most widely used method, the polygraph, can reach an accuracy as high as 91%. But most of the techniques in the traditional approach are invasive and require an experienced human interviewer to conduct the interrogation and interpret the results. Moreover, most of those techniques are not always reliable, as it has been found that trained people can trick the system such as the polygraph.

Whereas EEG is a method that doesn't require a professional to conduct the test, it would also be convenient as only one device is needed instead of multiple arrangements. This method is also trick proof, as any subject will not be able to control his brain signals while he can control his emotions and physiological reactions. Although we can get a sense of truth or lie based on the P300 response of EEG, it would require manual testing each time we want some results. That is why we use machine learning as we construct a model and once that is validated, we can use the same to predict whether it is true or lie in future instances. This means we might simply record a bunch of datasets at once and run our model for results, which would be very convenient compared to all the previous methods. For these reasons, we preferred machine learning on EEG for our project.

1.3 Motivation

Generally, known lie detection techniques, like polygraphs, uses physiological activities, such as blood pressure, heart rate, respiration, and skin conductance. When we conduct a lie detection experiment, we analyze the change in values from the normal values specifically for the subject and if a significant change is found, then the person is found guilty. But it has been known that these techniques are not reliable. It has been observed that the subject is able to control their physiological behaviour to an extent, making it difficult for the analyst to know about these details. Hence, the emergence of using brain waves as a measure of detecting lies using EEG signals made such life-defining experiments more robust. The proposed solution improves the efficiency of detecting lies using EEG signals and will help to know whether the subject is guilty or innocent with more precision, hence helping society to reduce crime.

1.4 Problem Statement

We model the task of Lie Detection as a classification problem.

Let *D* be the dataset containing EEG signals captured for each subject and where x_i is the data collected by $\boldsymbol{i^{th}}$ channel where $\boldsymbol{i \in \{1, 2, 3 \ldots . 16\}}$ and y be the output class label where if $\boldsymbol{y} = \mathbf{1}$ then the subject is guilty and if $\boldsymbol{y} = \mathbf{2}$ then subject is innocent.

Let $\boldsymbol{y_{act}}$ be the actual output class label and let $\boldsymbol{y_{pred}}$ be the predicted output class label using the proposed model. Then the Accuracy A can be defined as:

$$\boldsymbol{A} = \frac{(\boldsymbol{y_{act}} - \boldsymbol{y_{pred}} \; is \; 0)}{total \; class \; labels}$$

Design an optimum model that classifies $\boldsymbol{y}$ using the dataset *D* containing input feature $\boldsymbol{x_i}$ such that the A is maximized.

Chapter 2

Review of Literature

There is a lot of work that has been done on lie detection using EEG signals. Apart from lie detection, EEG signals have also been proven to be effective in mental state analysis [9], the stress level classifier achieved an accuracy of 94% hence proving the significance and applications of EEG signals.

There are several papers which have used the raw EEG data. Using the data, time and frequency domain information was extracted using techniques like wavelet transform, BAT algorithm, Binary BAT Algorithm, etc. [1] [3] [4] [5] [6] [8] [11] that helped to get optimal feature extraction from the dataset and hence the accuracy increased. The raw EEG dataset contains numerical data in the time series format then deep neural networks techniques were used to classify EEG signals with an accuracy of 95% [11]. Using deep belief networks instead of the deep neural network did not work well, leading to an average accuracy of 85% even after applying the wavelet transform technique for feature extraction [8]. Also, using K means clustering and feeding forward neural networks after wavelet transform produces only accuracy of 83.1% [6]. Another work considers wavelet transformation for feature extraction and then support vector machine for classification achieved an accuracy of 84.29% [3]. Another work that has used frequency and time-domain representation of the data and uses LDA for classification instead of SVM achieved an accuracy of 86% [1]. However, our project scope is limited to the potential (or amplitude) information of the EEG data, which means each cell in the data represents the maximum peak value achieved at any instant of time for that recorded signal.

2.1 Related Work

The major work related to the proposed work has included feature extraction using the wavelet transform technique to find out the optimal features and then passed on to several classification algorithms including LDA, SVM, MLFNN, Naïve Bayes, KNN. Then the accuracy and g-measure are calculated for each model. Based on the g-measure values, classification ranking was done, and the best models produced were LDA, SVM, and MLFNN. Now in the combiner, voting classifiers are used (weighted voting, majority voting, and unanimous voting) to check which of them gives the best results. The performance using voting classifiers and ensemble framework applied to the potential value is 80.4% [4]. The contributions to the proposed work are as follows:

- Use of different classification models – The models used in the proposed approach are SVM, Decision Trees, Logistic Regression, Random Forest, KNN, and Boosting. The models take into consideration the different work done on EEG signals and which models are performing for general classification purposes.

- The way of training EEG dataset – The way dataset is trained in the proposed approach has a drastic difference compared to the related work. The models used to train EEG data are tuned using hyperparameter tuning, where hyperparameters are found using k-fold cross validation.
- Classification ranking technique – The classification ranking technique used in the proposed approach is by using f-measure value as compared to the related work that uses the g-measure value. Here f-measure values observed are higher than the g-measure values.

Chapter 3

Proposed Solution

This chapter describes the ensemble technique that is used to resolve the issue of lie detection without the effects of controlling emotions and physiological expression.

3.1 Proposed Architecture

An overview of the architecture is shown in the figure 3.1.
In the proposed approach, for each fold in the ten-fold EEG dataset we do the following in each classifier model:

For each fold i:

- Test data = Subject i data
- Train data = Subject j data, for all j where $1 \leq j \leq 10$ and $j \neq i$.
- The hyperparameters, cross validation accuracy and cross validation f-measure for the above training data are found using k-fold cross validation.
- Now the models are trained with train data and tested with the test data to get the metrics (accuracy, sensitivity, specificity, f-measure) for all the models.
- Then all the models are ranked in descending order of their cross-validation f-measure values.
- From those models only the top three models are passed to the majority voting classifier.
- Finally, the majority voting classifier produce the output class label with the final accuracy.

This whole process is repeated for each fold, and we will get the accuracies for all ten subjects. The final accuracy is achieved by taking average of each subject's accuracy.

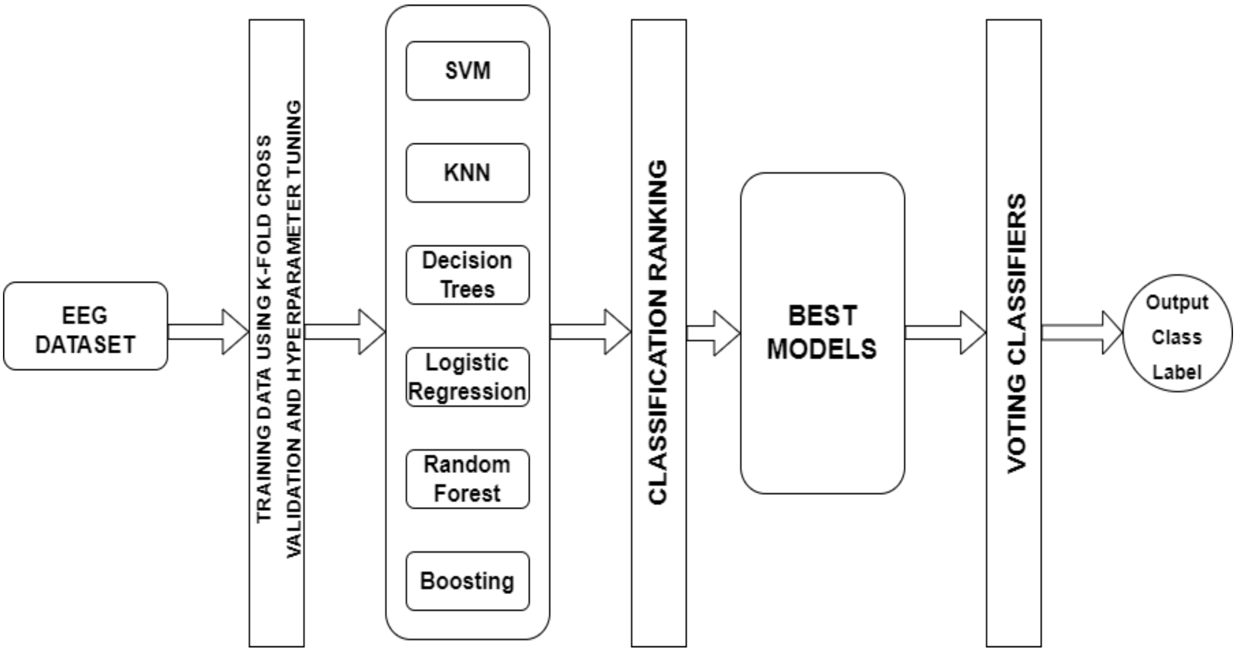


Figure 3.1 Architecture

3.2 Classification Algorithms

3.2.1 Boosting

Boosting is a special ensemble technique that takes several weak learners (that produce poor accuracy) into strong learners (that produce good accuracy).

There are two types of Boosting:

- Adaptive boosting
- Gradient Boosting

Gradient Boosting

Gradient Boosting iteratively learns from weaker models and tries to build a stronger model. Gradient boosting models are becoming popular because of their effectiveness at classifying complex datasets and have recently been used to win many Kaggle data science competitions.

The gradient boosting classifier architecture is shown in figure 3.2. There are three main components of Gradient booting:

Loss Function: The loss function determines how well the model is working in predicting the given data.

Weak Learner: These are the models that performs poorly, perhaps no better than random guessing. For example, decision trees are generally used as weak learners in gradient boosting.

Additive Model: This is a progressive approach to adding weaker models one at a time. The goal is to reach as close as possible to the final model, which is minimizing the loss function.

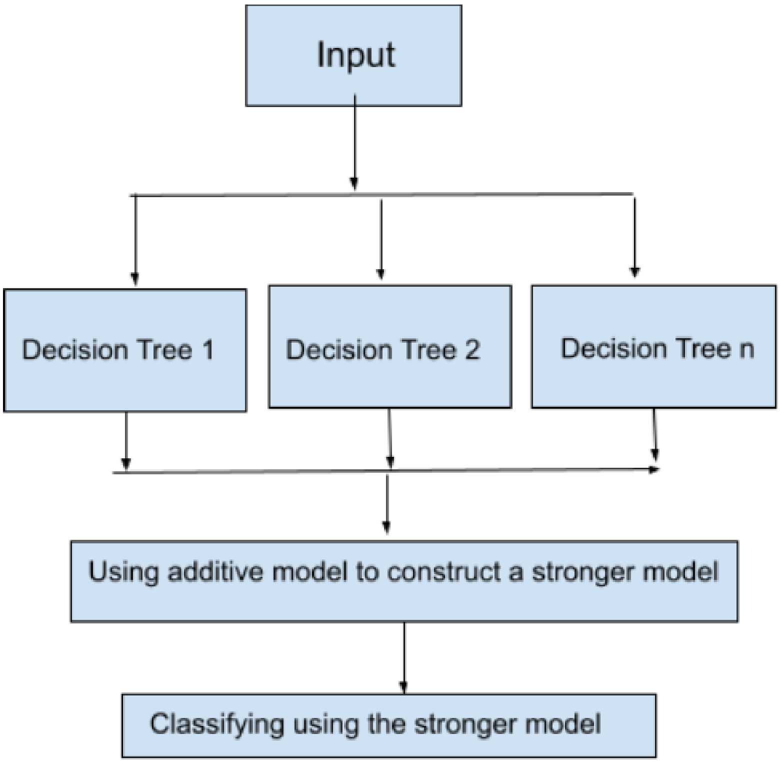

Input
Decision Tree 1
Decision Tree 2
Decision Tree n
Using additive model to construct a stronger model
Classifying using the stronger model

Figure 3.2 Gradient Boosting

3.2.2 Random Forest Classifier Algorithm

Random Forest is a famous machine learning algorithm. It is a part of the supervised learning technique. In Random Forest, we divide the dataset into multiple subsets. In each subset, we apply the decision tree and finally take the average accuracy of all trees to improve the predictive accuracy of that dataset. Random forest uses majority voting for classification. It can be used for Classification and Regression problems in ML. It uses ensemble learning, in which multiple classifiers are combined to solve a complex problem and thus improve the performance of the model. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. The Random Forest architecture is shown in figure 3.3. Random Forest has the following major advantages:

- It takes lesser time compared to other ensemble algorithms for training.
- Accuracy is high even for large datasets.

- Provides good accuracy even in case some of the data is missing during training.

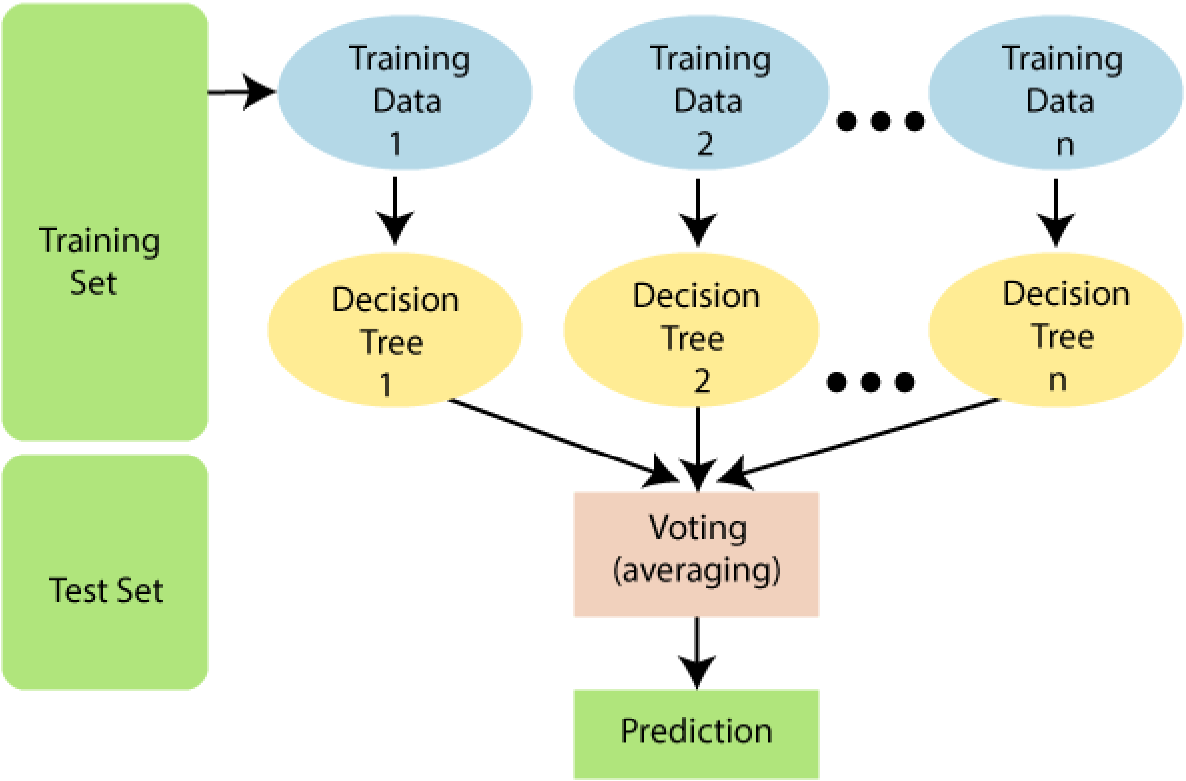


Figure 3.3 Random Forest

3.2.3 Logistic Regression

Logistic regression is a popular supervised learning algorithm. In a classification problem, the output variable can take only discrete values for a given feature or input, X. The output can either be produced and gives a probability value that tells whether it is true (1) or false (0).

In Logistic regression, we do not fit a line; unlike linear regression, we fit a "S" shaped logistic function known as the sigmoid function, which predicts one of the two maximum values (0 or 1). The sigmoid function used in logistic regression is shown in figure 3.4:

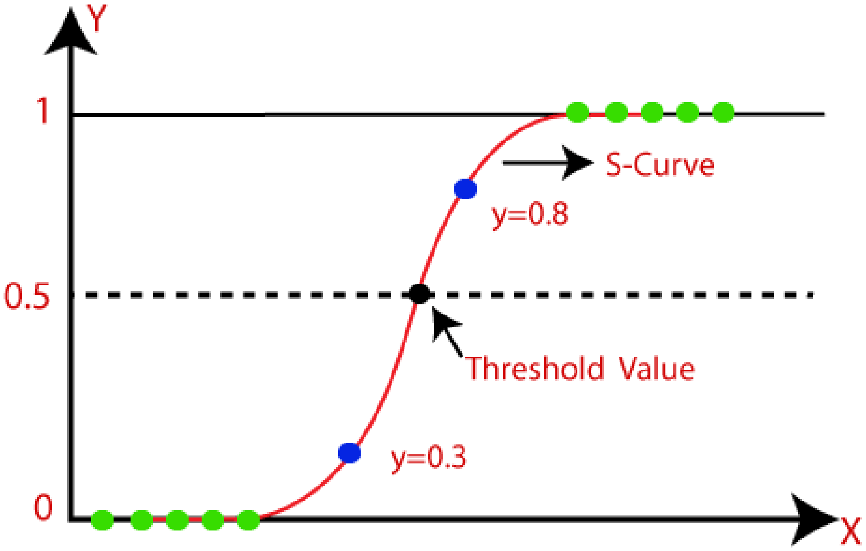


Figure 3.4 Logistic Regression

Following is the equation of Logistic regression

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

Here y represents the output variable and $x_1 \ldots x_n$ represent input features. Logistic Regression is used because of the following reasons:

- Logistic regression is easier to implement, interpret, and very efficient to train. For unknown records, it is very fast.
- If data is linearly separable it performs well.
- It can interpret model coefficients as indicators of feature importance.

3.2.4 Decision Tree Classifier

The decision tree is a supervised learning algorithm. It consists of a tree structure; internal nodes of the tree represent features and the branches of the tree represent decision rules, while the leaf nodes represent the outcome. The decision tree consists of two types of nodes, namely decision nodes (internal nodes) and output nodes (leaf nodes). It is called a decision tree because based on the decision nodes it makes the decisions and produces an outcome. To build a decision tree, we use the CART Algorithm. CART stands for Classification and Regression Algorithm. The architecture of the Decision Tree is shown in figure 3.5.

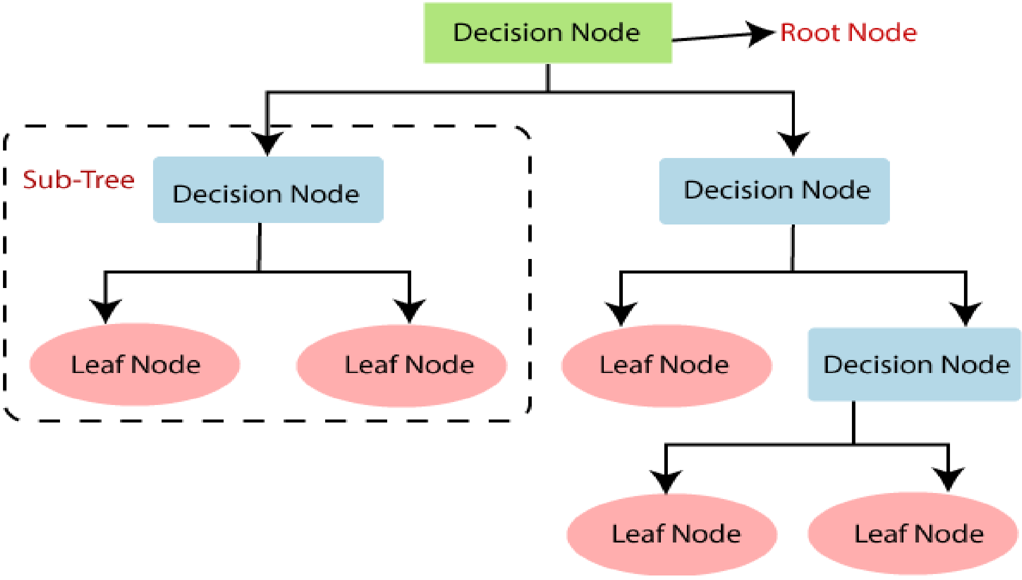


Figure 3.5 Decision Trees

3.2.5 K-Nearest Neighbor (KNN)

KNN is a simple supervised learning technique. It is a non-parametric algorithm that simply puts the new data point into the category which matches the most with the available category. It can be used for regression as well as classification, but it is mostly used for classification problems. Based on the similarity, the new data point is being put into the available category. The use of the KNN algorithm is shown in figure 3.6. KNN is used for the following reasons:

- Simple to implement
- Remove noise easily
- More effective for large dataset

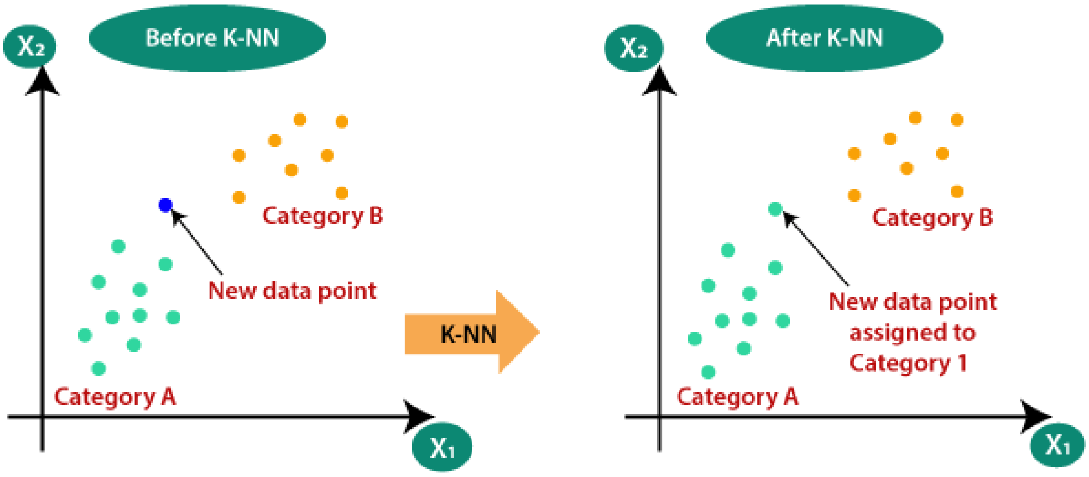


Figure 3.6 KNN

3.2.6 Support Vector Machine (SVM)

Support vector machine is a popular supervised learning algorithm technique that is used for both regression and classification algorithms. The major usage is for classification problems. In SVM, we try to fit a decision boundary into the n-dimensional space to segregate the data into classes so that we can easily put the new data point in the correct category in the future. The decision boundary is called a hyperplane. The vectors that help in creating the hyperplane are called support vectors, hence the name support vector machine. An example of an SVM hyperplane is shown in figure 3.7. There are two types of SVM: Linear and Non-Linear SVM. Linear SVM is used to separate data linearly, that is we can separate the data using a single straight line into two classe Non-Linear SVM is used for data that cannot be linearly separated using the kernel functions where the data is transformed to higher dimensions where classification is then done.

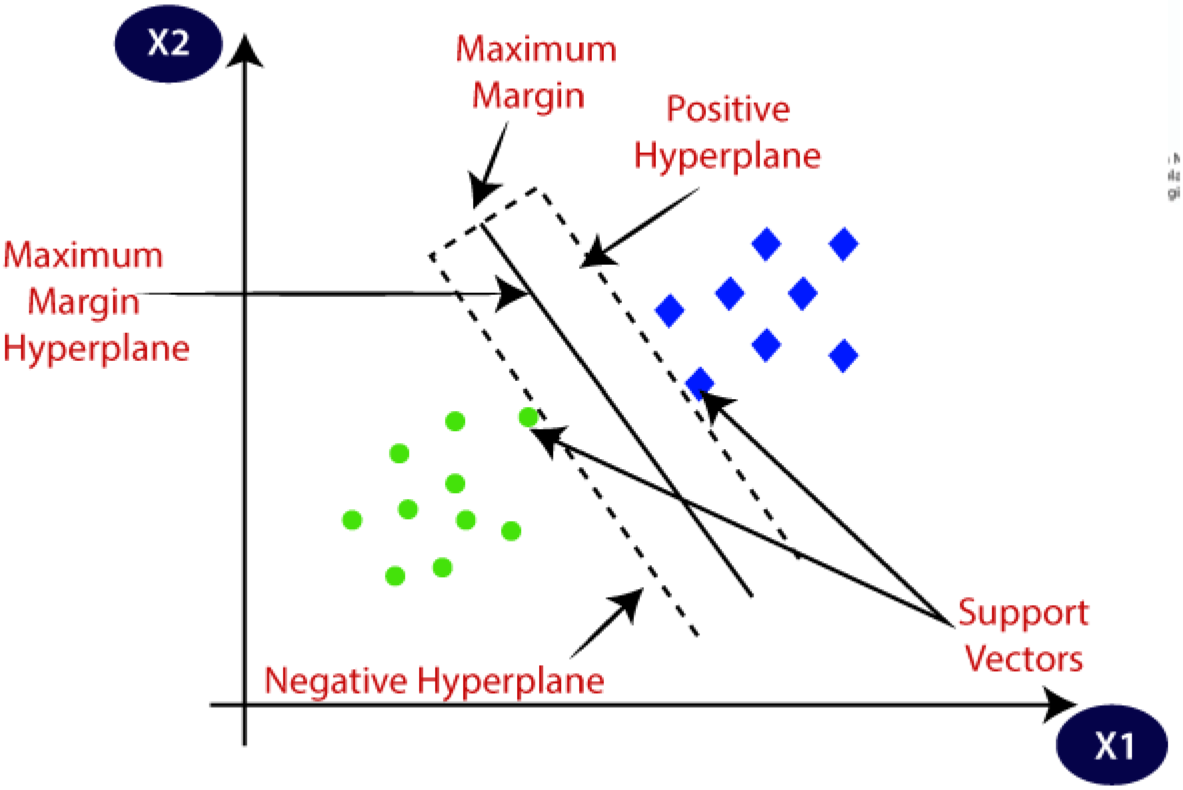


Figure 3.7 SVM

3.3 Voting Classifiers

A voting classifier is a classification machine learning algorithm. For training purposes, it takes on an ensemble of several models and then the results of all classifiers are merged and the class label that gets the highest voting is produced as the output. In this section, different voting techniques are discussed that are used in the proposed approach.

3.3.1 Weighted Voting Classifier

Weighted voting works on the ideology of giving more weightage to the stronger model for prediction. In weighted voting, weights are assigned to a classifier based on some specific characteristics. In this report, the weight is assigned based on the validation accuracy of the classifier. The below equation can be used to assign weights to each classifier:

$$\boldsymbol{Weight\ of\ the\ classifier} = \frac{\boldsymbol{Accuracy\ of\ the\ classfier}}{\boldsymbol{summation\ of\ Accuracy\ of\ all\ the\ classifiers}}$$

After the weights are assigned, the class which gets the highest vote is produced as the final output.

3.3.2 Majority Voting Classifier

In this classifier, the results of several classifiers that produce the class labels are taken and the class label that's chosen the maximum number of times is taken as the final prediction. Figure 3.8 clearly explains the working of majority voting classifiers.

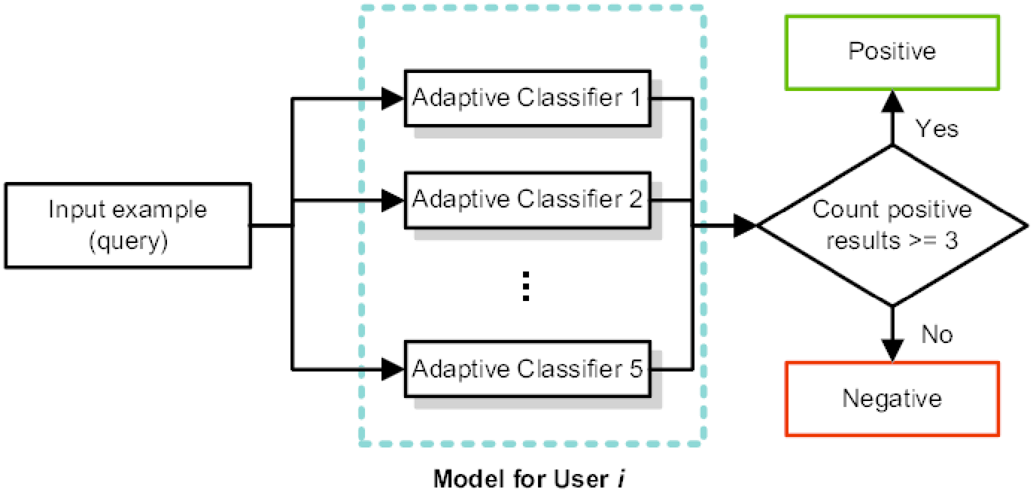


Figure 3.8 Majority voting classifier

3.3.3 Unanimous Voting Classifier

Unanimous voting classifier is a unique classifier where all classifiers need to agree on a particular class label to produce output as that class label. It means that if any classifiers disagree, then the decision will be overturned. In this report, if any classifier output any subject as guilty, then the output label will be guilty.

3.4 Hyperparameter Tuning and K-fold Cross Validation

3.4.1 Hyperparameter Tuning

Hyperparameters cannot be directly trained from the data as they are not model parameters. Model parameters are learned during training when we optimize a loss function using something like gradient descent.

The hyperparameters define how our model is structured whereas the model parameters specify how to transform the input data into the desired output. Unfortunately, there's no way to calculate "which way should the hyperparameter be updated to reduce the loss?" (i.e., gradients) to find the optimal model architecture; thus, we generally try out different experiments to figure out what works best. In general, this process includes:

- Start by defining a model
- Then define the range for the hyperparameter with all possible values
- Go on to define a method for sampling hyperparameter values
- Now define some evaluation criteria to judge our model
- Finally, define a cross-validation method

In the proposed approach, hyperparameter tuning has been done in the similar way as discussed above. The model-specific attributes that are 'fixed' before we train and test our model on data are known as hyperparameters. The task of finding the right hyperparameters is known to be somewhat of a difficult one, and it currently involves either a random search or a grid search. In the upcoming sections, hyperparameter tuning in SVM, KNN, and Logistic regression are explained in detail.

3.4.2 Hyperparameters Tuning in SVM

In the proposed approach, there are two hyperparameters on which tuning is being done; the first is C and the other is gamma. Using these parameters, the best decision is found, and the data is classified. C signifies the penalty for wrongly classified points which means if C is small, the penalty is less as the number of misclassified points is decreased, the decision boundary is fitted more accurately, while if c is large, the penalty gets increased, and the decision boundary is fitted very poorly. If radial basis function (non-linear SVM) is used, then a low gamma value will give more points to being grouped together due to a large similarity radius, while a high gamma value needs data points to be very close to being grouped together. Large gamma values might lead to overfitting. Hence, it is important to find the perfect combination of C and gamma values to predict the best decision boundary, hence achieving better accuracy.

Hyperparameter tuning in SVM is generally done using the grid search method. Grid search takes a dictionary of entire possible values of different hyperparameters that we would want to test, and then it feeds the different combinations through the algorithm for us, and finally, it reports back to us with the one having the highest accuracy. The following steps are used in the proposed approach:

- Possible values of C are taken from 0.1, 1, 10 … till 1000.
- Possible values of gamma are taken from 1, 0.1, 0.01 … till 0.0001.
- All the combinations of C and gamma values are taken and put into model training and the accuracy and the f-measure for each combination is stored in a dictionary.
- C and gamma are chosen of that combination which gives the best f-measure.

3.4.3 Hyperparameter Tuning in KNN

In the proposed approach, hyperparameter tuning in KNN is done using random search. Random search is done by evaluating an n uniformly random points in the hyperparameter space and selecting the one producing the best performance. The following steps are used in the proposed approach:

- N_neighbors – this parameter is used to select the number of neighbors to be used. Possible value of n_neighbors is taken from 5, 10, 20, … till 160.
- Weights – this signifies the way weights have been considered of the neighborhood. Possible values of weights are uniform, distance.
- Algorithm – this signifies the algorithm used to calculate the nearest neighbors. Possible value of algorithm is auto, ball_tree, kd_tree, brute.
- Leaf_size – it is passed to ball_tree and kd_tree algorithm to increase construction speed and query speed. Possible values of leaf_size are 15,30,45… till 100.
- All the combinations of these parameters' values are taken at **random** and put into model training and the accuracy and the f-measure for each combination is stored in a dictionary.
- Parameters are chosen of that combination which gives best f-measure.

3.4.4 Hyperparameter Tuning in Logistic Regression

In the proposed approach, hyperparameter tuning in Logistic Regression is done using random search. The following steps are used in the proposed approach:

- Penalty – It is used to specify the norm of the penalty. Possible values are l1 and l2.
- C – This value signifies the strongness of regularization just like in SVM. Possible values are logspace from -4 to 4 until 20.
- Fit_intercept – This ask if the constant value has to be added to the decision function or not.
- All the combinations of these parameters' values are taken at **random** and put into the model training and the accuracy and the f-measure for each combination is stored in a dictionary.
- Parameters are chosen of that combination which gives best f-measure.

3.4.5 K-fold Cross Validation

In K-fold cross-validation, from the whole data first, the test data is kept aside. The remaining data is divided into K number of folds, where each one of the k-folds is taken as the validation set and the remaining k-1 folds are used as the training set. This will be repeated k times until every fold is made and validation set exactly once. The 5-fold cross-validation is shown in figure 3.9. K-fold cross-validation helps to increase the accuracy of the data that it has not seen before as it applies k number of times on the same dataset with subsets of data.

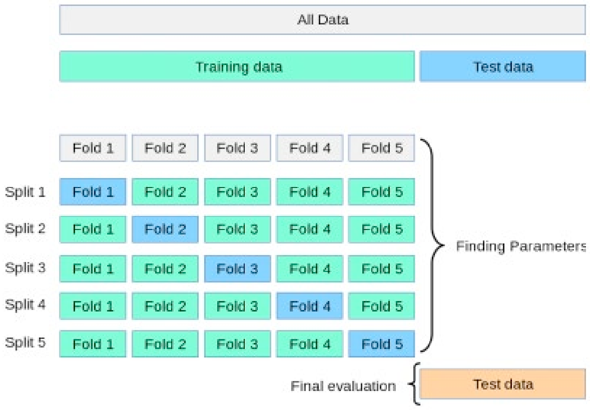


Figure 3.9 5-fold cross validation

Chapter 4

Results

4.1 Datasets

The dataset was recorded from 10 healthy subjects. The recording was performed for two sessions: Guilty and Innocent. During both sessions, the stimuli were presented by performing a concealed information test. The EEG data recording was done by placing 16 Ag/AgCl electrodes at Fz, FC1, FC2, C3, Cz, C4, CP5, CP1, CP2, CP6, P3, Pz, P4, O1, Oz and O2 sites following 10-10 international system for electrode placement. Three types of stimuli (target, irrelevant and probe) are presented to the subject. Instead of asking questions, we show certain images to subjects. The images are categorized into three types of stimuli:

- Target: These are images of celebrated personalities familiar to all the participating subjects (both innocent and guilty).
- Irrelevant: These are certain anonymous person images shown to all the participants.
- Probe: These are rarely occurring images, the image of sufferer in this mock crime scenario.

The lie detection experiment starts by displaying the stimuli images on a 15.4-inch display screen for 31 seconds to the subject. The subject has to recognize those images and has to answer with a “yes” or a “no”. Each image will be displayed for 1.1 seconds followed by a blank image for 2 seconds. So, a total of ten images are presented to the subject, out of ten images; seven are irrelevant images; two are target images, and one is the probe image. These images are randomly presented to the subject. The experiment was performed for two sessions, where each session had been conducted for 30 trials for a single subject. So, for ten subjects we have 600 trials or samples (2 × 30 × 1 × 10), which are sampled at 250 Hz.

- Guilty Session: During the guilty session, subjects are instructed to respond with “yes” for the target images, whereas “no” for the probe and irrelevant images. The probe is an image of the victim known to the guilty person. Here, the response generated in the guilty brain for target and probe is the same, as the probe is known to the guilty person, but he/she is intentionally denying that. As soon as the probe and target images are presented, a P300 response is elicited for the guilty subject.

- Innocent Session: During the innocent session, subjects are instructed to respond with “yes” for the target images, whereas “no” for the probe and irrelevant images. Here, the response generated in the innocent brain for irrelevant and probe is the same, as both the stimuli generated are due to the subject having no knowledge of the either images. Only target stimuli will elicit P300 in the innocent subject's brain.

Format of Data:

The Dataset is processed into 10 folds representing data collected for each subject. Each fold has two sections:

- Train data
- Test data

4.2 Observations and Analysis

On applying the weighted voting classifier to all the subjects, the results are shown in figure 4.1. Figure 4.1 shows the comparison of the results for all subjects. Average accuracy for the subjects using three-weighted voting was observed to be 85.06% and using five-weighted it was observed to be 86.04%. Hence, it can be observed that the five-weighted voting classifier performs better than the three-weighted voting classifiers.

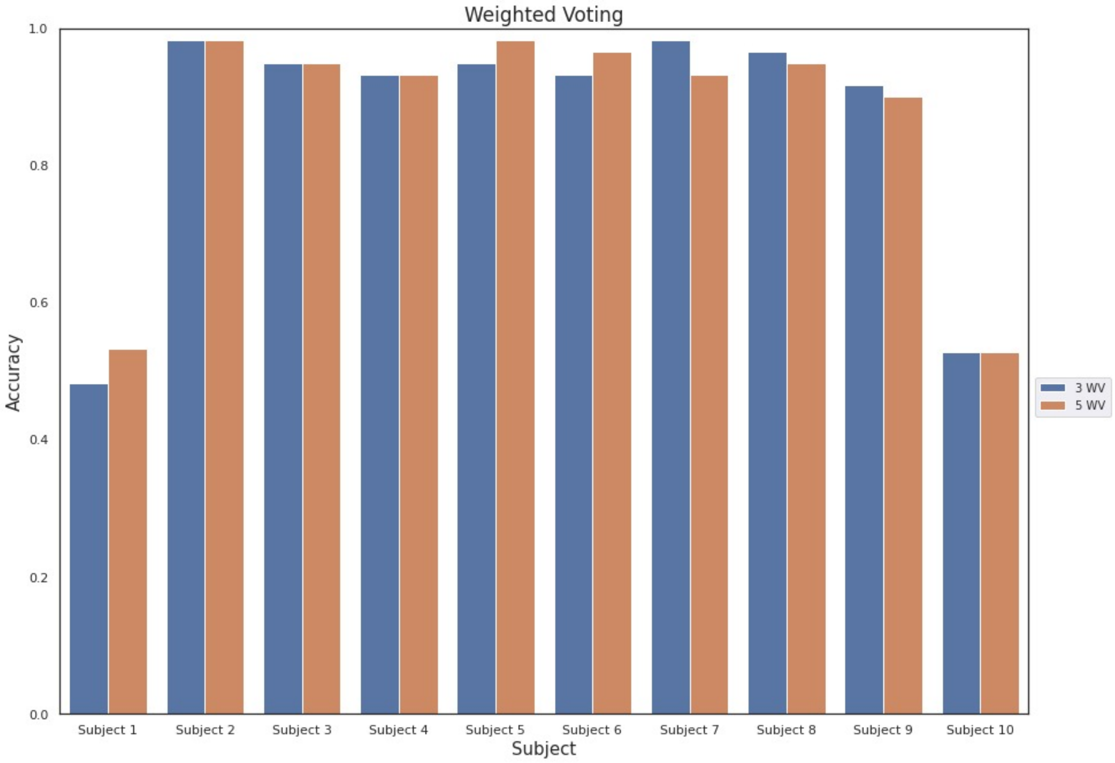


Figure 4.1 Weighted voting for all subjects

On applying the majority voting classifier to all the subjects, the results are shown in figure 4.2. Figure 4.2 shows the comparison of the results for all subjects. The average accuracy for the subjects for the three-majority voting was observed to be 86.06% and for the five-majority voting, it was observed to be 86.04%. Hence, it can be observed that the three-majority voting classifier is performing better than the five-majority voting classifier.

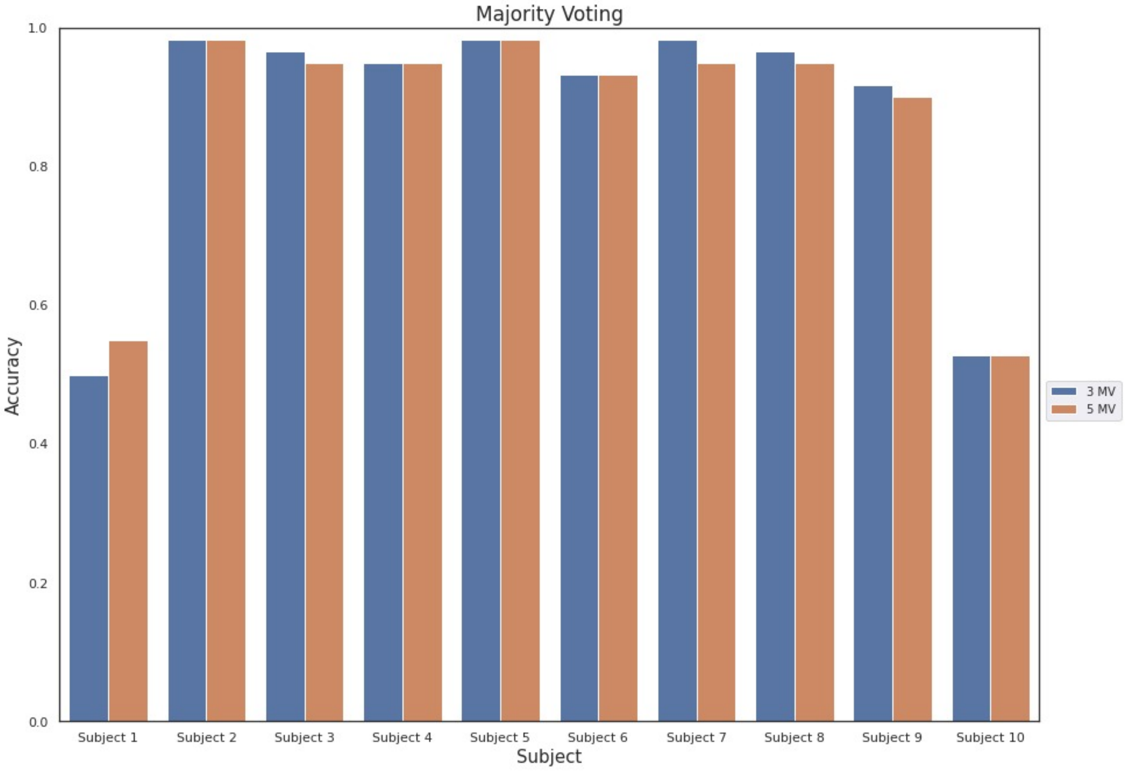


Figure 4.2 Majority voting for all subjects

On applying the unanimous voting classifier to all the subjects, the results are shown in figure 4.3. Figure 4.3 shows the comparison of the results for all subjects. The average accuracy for the subjects for the three-unanimous voting was observed to be 83.71% and for the five-unanimous voting it was observed to be 79.87%. Hence, the three-unanimous voting classifier has better accuracy than the five-unanimous voting classifier.

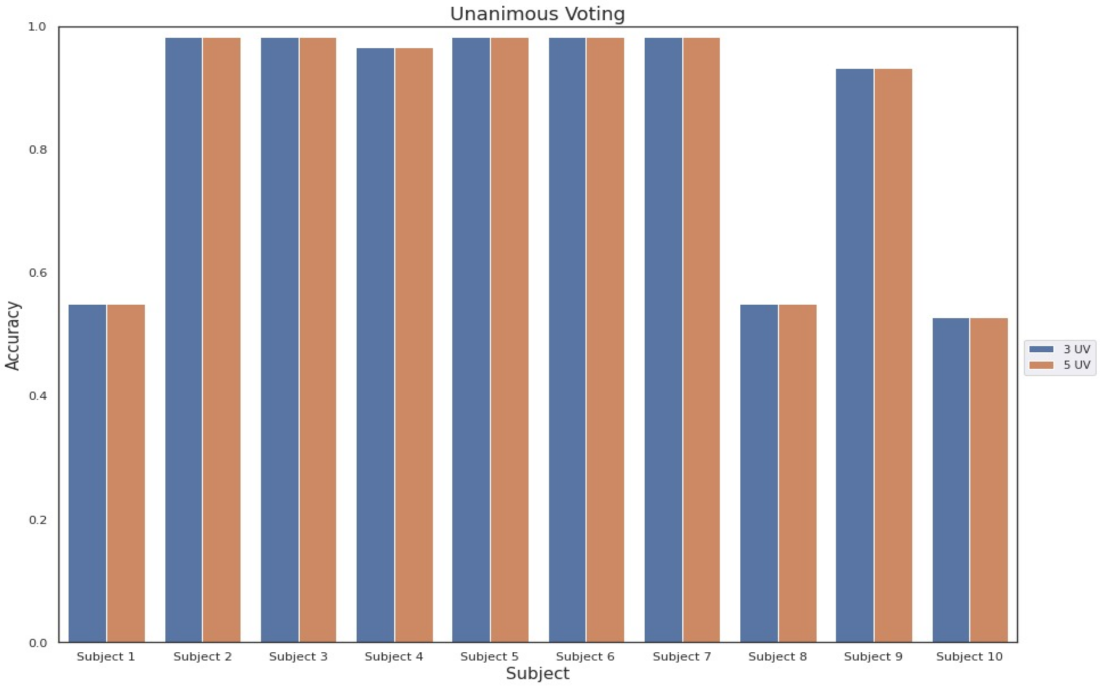


Figure 4.3 Unanimous voting for all subjects

Figure 4.4 shows the comparison of three weighted voting, three majority voting and three unanimous voting classifiers with respect to each subject. This is called as the three-model ensemble.

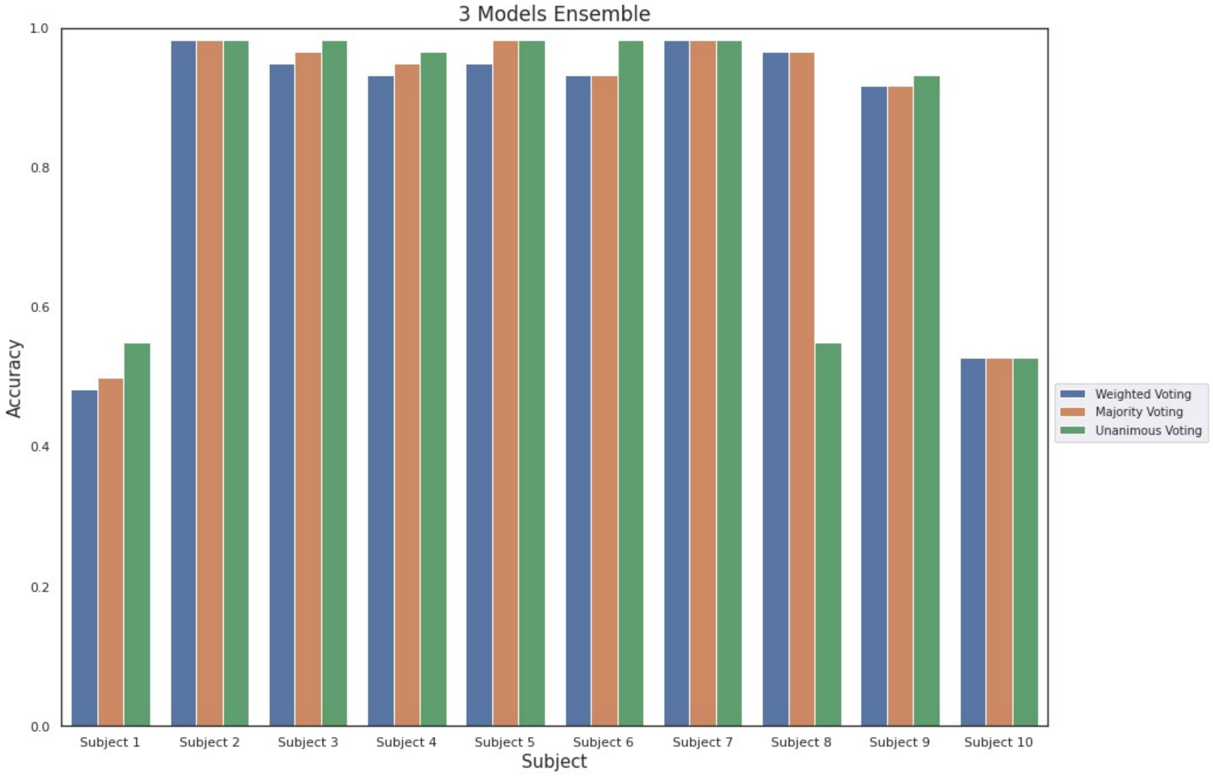


Figure 4.4 3 model ensemble for all subjects

Figure 4.5 shows the comparison of five weighted voting, five majority voting and five unanimous voting classifiers with respect to each subject. This is called as the five-model ensemble.

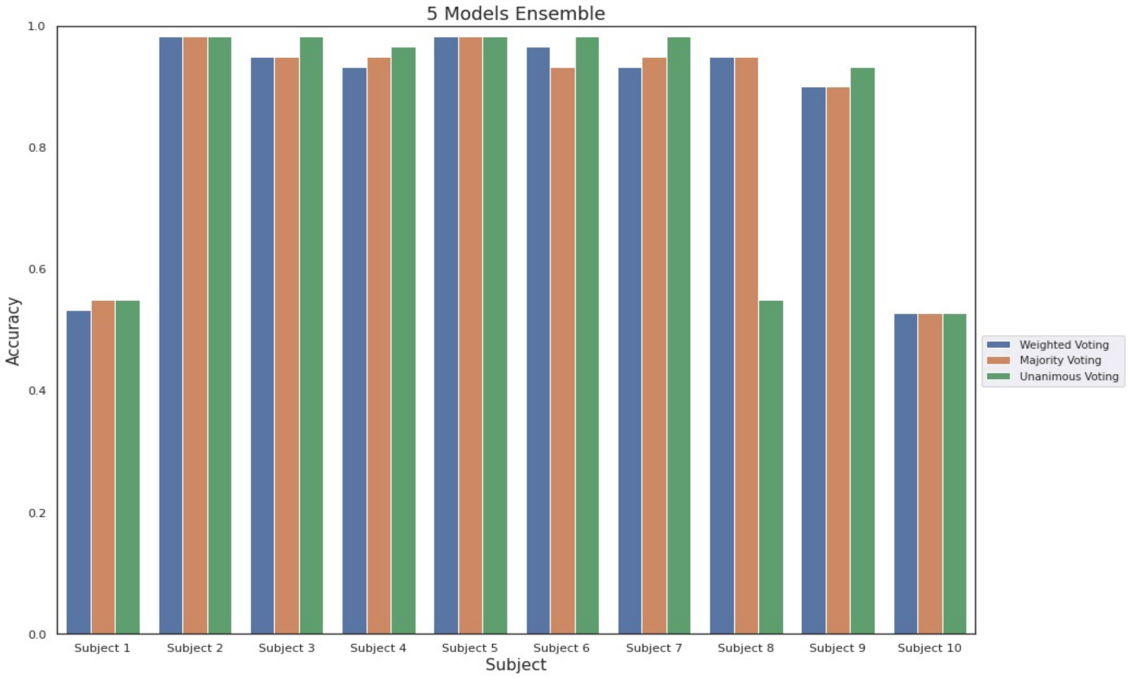


Figure 4.5 5 model ensemble for all subjects

4.3 Results

From the observations and analysis of each voting classifier considering different numbers of best models, the best accuracy was achieved by three majority voting classifiers with an accuracy of 86.06%.

Other measures like specificity, sensitivity and f-measure values are also compared for different voting classifiers and finally they are compared to the related work [4].
The comparison is shown in Table 1.

Specificity refers to the number of data points that are correctly classified as negative as to how many points are actually negative.

Sensitivity refers to the number of data points that are correctly classified as positive as to how many points are actually positive.

F-measure is the harmonic mean of precision and recall. Precision refers to the point correctly classified as positive as opposed to total positives. While sensitivity and recall are one and.

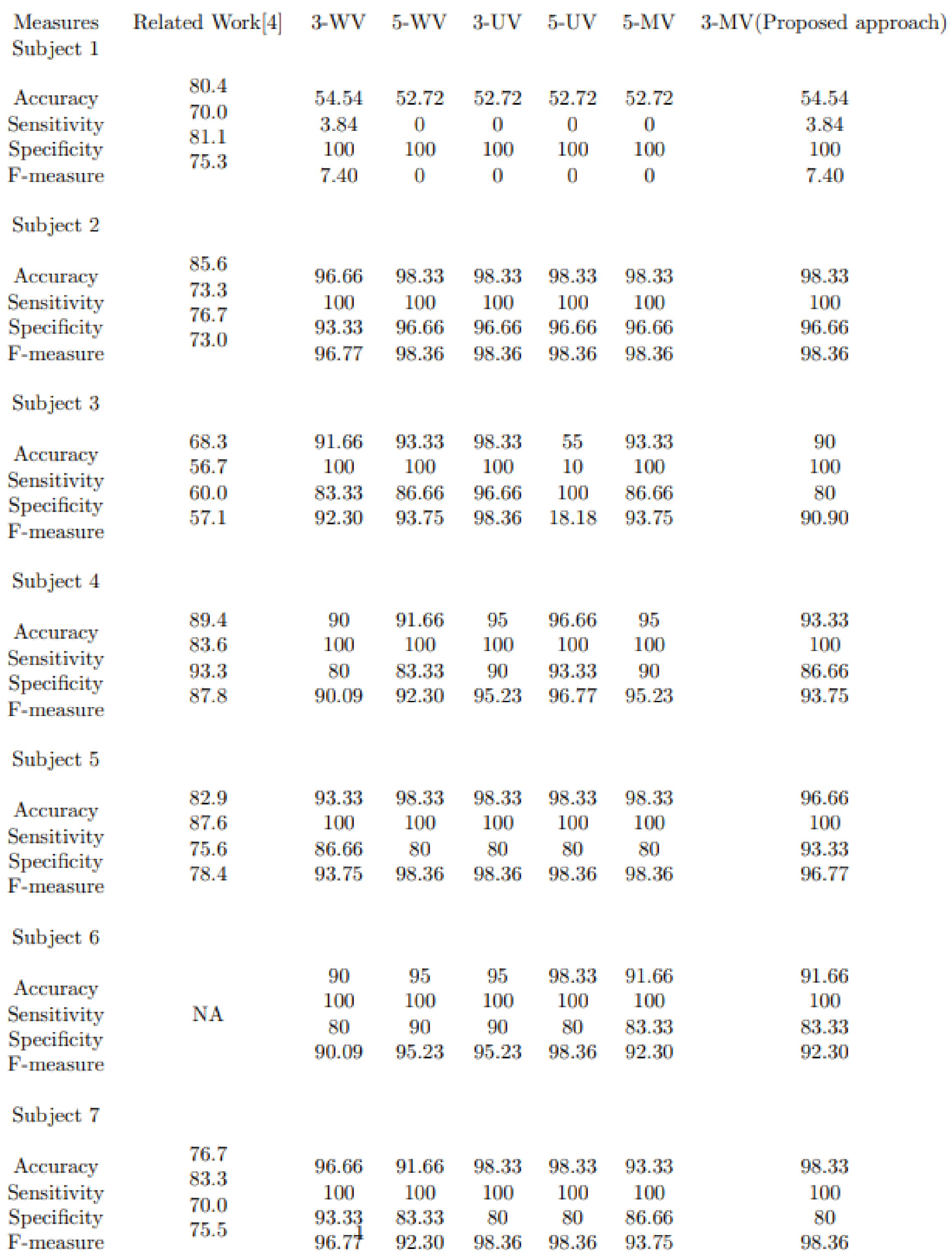
$$Sensitivity = \frac{TP}{FN + TP}$$

$$Specificity = \frac{TN}{FP + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F - Measure = 2 * \frac{precision * recall}{precison + recall}$$

Table 1 Subject wise performance using various voting classifiers and comparison of related work [4] with proposed approach.



Measures	Related Work[4]	3-WV	5-WV	3-UV	5-UV	5-MV	3-MV(Proposed approach)
Subject 1							
Accuracy	80.4	54.54	52.72	52.72	52.72	52.72	54.54
Sensitivity	70.0	3.84	0	0	0	0	3.84
Specificity	81.1	100	100	100	100	100	100
F-measure	75.3	7.40	0	0	0	0	7.40
Subject 2							
Accuracy	85.6	96.66	98.33	98.33	98.33	98.33	98.33
Sensitivity	73.3	100	100	100	100	100	100
Specificity	76.7	93.33	96.66	96.66	96.66	96.66	96.66
F-measure	73.0	96.77	98.36	98.36	98.36	98.36	98.36
Subject 3							
Accuracy	68.3	91.66	93.33	98.33	55	93.33	90
Sensitivity	56.7	100	100	100	10	100	100
Specificity	60.0	83.33	86.66	96.66	100	86.66	80
F-measure	57.1	92.30	93.75	98.36	18.18	93.75	90.90
Subject 4							
Accuracy	89.4	90	91.66	95	96.66	95	93.33
Sensitivity	83.6	100	100	100	100	100	100
Specificity	93.3	80	83.33	90	93.33	90	86.66
F-measure	87.8	90.09	92.30	95.23	96.77	95.23	93.75
Subject 5							
Accuracy	82.9	93.33	98.33	98.33	98.33	98.33	96.66
Sensitivity	87.6	100	100	100	100	100	100
Specificity	75.6	86.66	80	80	80	80	93.33
F-measure	78.4	93.75	98.36	98.36	98.36	98.36	96.77
Subject 6							
Accuracy	NA	90	95	95	98.33	91.66	91.66
Sensitivity		100	100	100	100	100	100
Specificity		80	90	90	80	83.33	83.33
F-measure		90.09	95.23	95.23	98.36	92.30	92.30
Subject 7							
Accuracy	76.7	96.66	91.66	98.33	98.33	93.33	98.33
Sensitivity	83.3	100	100	100	100	100	100
Specificity	70.0	93.33	83.33	80	80	86.66	80
F-measure	75.5	96.77	92.30	98.36	98.36	93.75	98.36

Table 1 continued

Subject 8							
Accuracy	81.7	95	95	55	55	95	95
Sensitivity	93.3	100	100	10	10	100	100
Specificity	70.0	90	90	100	100	90	90
F-measure	80.3	95.23	95.23	18.18	18.18	95.23	95.23
Subject 9							
Accuracy	80	90	91.66	93.33	93.33	90	90
Sensitivity	80	90	93.33	90	90	90	90
Specificity	80	90	90	96.66	96.66	90	90
F-measure	74.3	90	91.80	93.10	93.10	90	90
Subject 10							
Accuracy	81.0	52.72	52.72	52.72	52.72	52.72	52.72
Sensitivity	80.0	0	0	0	0	0	0
Specificity	83.3	100	100	100	100	100	100
F-measure	81.0	0	0	0	0	0	0
Average							
Accuracy	**80.7**	85.06	86.04	83.71	79.87	86.04	**86.06**
Sensitivity	78.6	79.38	79.33	79	61	79	79.38
Specificity	76.7	89.66	89.99	92.99	92.66	89.66	89.99
F-measure	75.8(G-Measure)	75.24	75.73	69.51	52.13	75.69	76.30

Chapter 5

Conclusions and Future work

5.1 Conclusions

In this project, we proposed an ensemble approach for lie detection based on EEG signals. The poor model uses hyperparameter tuning, and ensemble models were trained directly. Models used k-fold cross-validation for getting the hyperparameters, validation accuracy and validation f-measure for training the dataset and then choosing the best models out of all the models through classification ranking using validation f-measure, subsequently applying voting classifiers to predict whether the subject is innocent or guilty. In this experiment, we achieved the best accuracy from the three-majority voting classifier with an accuracy of 87.1%. This approach will help reduce crime and increase the security of society to a better extent.

5.2 Future Work

Following are the improvements that can be made to increase the efficiency:

- Refining Dataset – After performing the experiment, our observations have clearly shown that there are some bias/fault in some part of the dataset. The outliers in the dataset must be removed, and the data has to be refined. This process will result in significant improvements in accuracy.

- Channel selection – From the 16 channels that are reading and capturing the brain waves, only those optimal channels can be selected that contain significant information, these channels can be selected with some optimization techniques to produce better results.

- Instead of using the f-measure to choose the best model for the ensemble classification, application-specific measures can be considered.

Bibliography

[1] A. I. Simbolon, A. Turnip, J. Hutahaean, Y. Siagian and N. Irawati, "An experiment of lie detection-based EEG-P300 classified by SVM algorithm," *2015 International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*, 2015, pp. 68-71, doi: 10.1109/ICACOMIT.2015.7440177.

[2] Abootalebi V, Moradi MH, Khalilzadeh MA. A new approach for EEG feature extraction in P300-based lie detection. Comput Methods Programs Biomed. 2009 Apr;94(1):48-57. doi: 10.1016/j.cmpb.2008.10.001. Epub 2008 Nov 28. PMID: 19041154.

[3] Gao J, Lu L, Yang Y, Yu G, Na L, Rao N. A novel concealed information test method based on independent component analysis and support vector machine. Clin EEG Neurosci. 2012 Jan;43(1):54-63. doi: 10.1177/1550059411428715. PMID: 22423552.

[4] Bablani, A., Edla, D.R., Tripathi, D. *et al.* An efficient Concealed Information Test: EEG feature extraction and ensemble classification for lie identification. *Machine Vision and Applications* **30,** 813–832 (2019). https://doi.org/10.1007/s00138-018-0950-y.

[5] A. Bablani, D. R. Edla, D. Tripathi, S. Dodia and S. Chintala, "A Synergistic Concealed Information Test With Novel Approach for EEG Channel Selection and SVM Parameter Optimization," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 11, pp. 3057-3068, Nov. 2019, doi: 10.1109/TIFS.2019.2913798.

[6] Bablani, Annushree & Edla, Damodar & Kuppili, Venkatanareshbabu & Ramesh, Dharavath. (2020). A Multi Stage EEG data classification using k-Means and Feed Forward Neural Network. Clinical Epidemiology and Global Health. 8. 10.1016/j.cegh.2020.01.008.

[7] Damodar Reddy Edla, Shubham Dodia, Annushree Bablani, and Venkatanareshbabu Kuppili. 2021. An Efficient Deep Learning Paradigm for Deceit Identification Test on EEG Signals. ACM Trans. Manage. Inf. Syst. 12, 3, Article 25 (September 2021), 20 pages. https://doi.org/10.1145/3458791.

[8] A. Bablani, D. R. Edla and V. Kuppili, "Deceit Identification Test on EEG Data Using Deep Belief Network," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2018, pp. 1-6, doi: 10.1109/ICCCNT.2018.8494124.

[9] Desai, Rahul & Porob, Pratik & Rebelo, Penjo & Edla, Damodar & Bablani, Annushree.

(2020). EEG Data Classification for Mental State Analysis Using Wavelet Packet Transform and Gaussian Process Classifier. Wireless Personal Communications. 115. 10.1007/s11277-020-07675-7.

[10] A. Bablani, D. R. Edla, V. Kupilli and R. Dharavath, "Lie Detection Using Fuzzy Ensemble Approach With Novel Defuzzification Method for Classification of EEG Signals," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-13, 2021, Art no. 2509413, doi: 10.1109/TIM.2021.3082985.

[11] Damodar Reddy Edla, Shubham Dodia, Annushree Bablani, and Venkatanareshbabu Kuppili. 2021. An Efficient Deep Learning Paradigm for Deceit Identification Test on EEG Signals. ACM Trans. Manage. Inf. Syst.12, 3, Article 25 (September 2021), 20 pages. https://doi.org/10.1145/3458791.

ACKNOWLEDGEMENT

It's our privilege and honor to express our gratitude to all the respected personalities who have guided us, inspired us and supported us in the smooth and successful completion of our project.

We thank our beloved faculty supervisor Dr. Ramalingaswamy Cheruku, Assistant Professor, Departmentof Computer Science and Engineering (CSE), National Institute of Technology, Warangal, for his persistent supervision, guidance, suggestions and encouragement during this project. He has motivated us during the low times and gave us courage to move ahead positively.

We are grateful to Prof. S. Ravi Chandra, Head of the Department, Computer Science and Engineering, National Institute of Technology, Warangal for his moral support to carry out this project. We are also thankful to the Project Evaluation Committee, for their strenuous efforts to evaluate our projects. We wish to thank all the staff members in the department for their kind cooperation and support given throughout our project work. We are thankful to all of our colleague friends who have given valuable suggestions and help in all stages of the development of the project.

Naveen Shankar – 187239

Naman Mittal -187238

Hemant Kumar Singh - 187220