Feature Extraction and Classification of EEG Signals Using Machine Learning Algorithms for Biometric Systems

Bhawna kaliraman¹, Manoj Duhan²

¹ Department of ECE, DCRUST, Murthal, (India)
Bhawna.kaliraman5@gmail.com

² Department of ECE, DCRUST, Murthal, (India)
Duhan_manoj@rediffmail.com

Abstract: EEG (electroencephalogram) based biometrics systems are used in very high-security areas due to its several advantages over traditional biometric systems. This paper presents an approach for extracting features and classification of EEG signals acquired from users for authentication purposes. The Autoregressive (AR) model with order three features is calculated because the AR model features reveal the signal’s intrinsic characteristics. An experiment is performed on many classifiers to classify the extracted features. Classifiers are tested with different kernels and optimizers to accomplish good accuracy for the system. Machine learning algorithms such as support vector machines (SVM), k-nearest neighbor (k-NN), multilayer perceptron (MLP), XGBoost are used as classifiers to classify the signals for authentication. Cross-validation is used for splitting data in the train and test set so that more accurate results were obtained on unseen data. 10-fold cross-validation is used in the proposed work. Obtained results show that mean accuracy values up to 99.7% is achieved; in some trials, accuracy up to 100% is achieved with few classifiers. A comparison table is shown, which compares the accuracy values obtained by different classifiers using different kernels and optimizers.

Keywords: AR models, Classification, Feature extraction, k-NN, Machine learning, MLP, SVM, XGBoost.

I. Introduction

Electroencephalogram (EEG) signals are used by a good number of researchers in their work. In 1985 R. Carton showed an electrical activity on the surface of the brain [1]. EEG signals can be easily acquired by placing electrodes on the scalp.

Earlier EEG was used only in the clinical and research environment, but now it is used in several other fields. One of the applications is the EEG based biometric system. With the rapid development of technology, security issues are a significant concern in these fields. However, there are old security measures available, such as passwords, radio frequency identification (RFID), and personal identification number (PIN codes), but these are insecure or hacked easily. So, biometrics systems can be used in such places to provide better security.

Biometric systems identify individuals by analyzing their physiological or behavioral characteristics. There are many physiological and behavioral characteristics present, which can be used as biometric traits such as fingerprint, face, retina, palm print, iris, voice, signature, DNA, hand geometry, etc. Generally, biometrics systems use physiological characteristics because behavioral characteristics are considered weak and can be replicated, such as signature, keystroke dynamics, etc. For any biometric system it should follow some characteristics [2]:

- Uniqueness Any parameter used for the biometric system must be unique for all users[3].
- Universality Any biometric system must be universal, i.e., it must present in all human beings. So, EEG is a good fit for this category.
- Permanence Any parameter used for biometric trait should be constant for a long time, i.e., it should not change with respect to time.
- Collectability Any parameter to be used as a biometric trait should be easily collected. As EEG signals are collected non-invasively from the scalp, nowadays, EEG signals are collected using portable headsets. So, these can act as a good trait for the biometric system. However, still much is required to be done in this area.
- Performance Any biometric system must have excellent performance in terms of an acceptable accuracy score used in high-security areas. In this paper, the main focus is on this parameter.
- Acceptability Any biometric parameter used must be accepted readily by the applicant.

However, with the advancement of technology, these are still prone to forgery. Such as fingerprint can be forged using a gummy finger, face print can be faked using photos, like these all other biometrics have several pros and cons. So, to meet the need for advanced security, much robust biometrics are
required that cannot be easily forged. In such a case, EEG can be used as a promising biometrics trait [4]. EEG signals are acquired non-invasively from the scalp, and these are also robust against spoofing attacks. No one can replicate the EEG of a person because these are difficult to copy and steal. So, these can be used in very highly confidential areas.

This article is divided into five sections: the first section includes an introduction about EEG based biometrics systems; in the second section literature review is presented, which provides research work done in the area of EEG based biometric systems; in third section description of the database used and discussion of research methodology used in this article including feature extraction and classification techniques; the fourth section consists of results and analysis section which analyze the results obtained from various classifiers and comparison of accuracies obtained from all classifiers; in the last section includes conclusion and references part.

II. Literature Review

Research in the human-computer interface (HCI) area booms nowadays. Many researchers are going on in this realm. Stassen H. used communication theory to recognize a person by their EEG spectral patterns. Then Fourier transform was applied to the EEG time series. After that, the multivariate statistical procedure was used like linear discriminant functions and cluster analysis. Results show that a confidence interval of around 90% is acquired for the identification of subjects. Eighty-two subjects with four different psychiatric diagnostic groups were used for the study [5].

Amanda S. et al. worked on brain signals and used them as a passport, i.e., biometric verification. They estimate the efficiency of existing EEG devices as identification with a biometric [6].

Macros D. et al. worked in this area and provided a frequency domain thorough explanation of EEG as biometric identification. Ten experiments were performed on six publicly available datasets. They performed ten experiments. The results focused on the uniqueness of EEG information found in frequency below 40 Hz [7].

Campisi P. and Rocca D. provide reviews on the EEG based biometric recognition system. A detailed survey was taken up for different acquisition protocols, feature extraction algorithms, database structure, and classification algorithms employed in EEG based research [8].

Marcel at el. [9] recorded EEG signals with the Biosemi system, which has 32 electrodes located using a 10-20 electrode placement system with a sampling rate of 512 Hz. No artifacts rejection and correction were employed. This paper considers nine healthy subjects for the dataset, and data is recorded in twelve non-feedback periods for three days. Where subjects had to execute three different tasks, subjects perform each task for 15 seconds, and then on the operator’s request, subjects can switch their task. They used PSD as features and classify using Gaussian Mixture Model, and the results show half a total error rate of 7.1%.

Palaniappan and Ravi [10] used VEP signals from 20 subjects. They used a 61-channel system to acquire their data. They use PCA for noise removal and use PSD features. They use three classifiers for better classification performance, i.e., simplified fuzzy ARTMAP, Linear Discriminant, and k-nearest neighbor. Here, to locate the nearest neighbors, they use two distances, i.e., Euclidean and Manhattan. Their study found that by applying PCA, the classification performance was improved for LD and SFA, but for k-NN, the performance was degraded.

Yazdani et al. [11] used VEP signals from gamma-band, i.e. (GBVEP), from 20 subjects using a 61 channel sensor system. For feature extraction, they use AR and PSD coefficients. They employ k-NN as a classifier for their work and got 100% accuracy when k=5 is used.

Karthikeyan and sabarigiri [12] used the Naive bayes classifier in their work to classify feature set, which is composed of AR and PSD coefficients. They randomly select four recordings as training and leftover for testing. Posterior probability was used as a similarity match. They got 4.16% EER using the proposed method.

III. Research Methodology

The research methodology used in this research is explained with the flow chart shown in Fig. 1.

To identify individuals based on their EEG signals, first, their features are calculated and stored in some files. The extracted features are passed to a classifier that tells the class of each signal. Then based on the decision of the classifier, the identity of individuals can be predicted. This section contains information about the database, feature extraction, and classification algorithm used in the proposed work.

A. Database

The database is obtained from the UCI KDD EEG database, an open-source dataset [13]. This database consists of recording from 64 channels, sampled at a frequency of 256 Hz for one second, and signals were passed through a hardware passband filter of 0.1-50 Hz.

Electrodes are located using standard electrode position nomenclature. The dataset contains two groups of subjects; one is an alcoholic group, and another is non-alcoholic. Data is acquired using short latency (300ms) visual stimuli where
either single stimuli (S1) or two stimuli (S1 and S2) were shown to each subject. Each stimulus was taken from the picture set given in the article [14]. Subjects were asked to recognize the pictures shown on the screen, which is one meter away from their eyes and separated by blank screen intervals of 5.1 seconds. One second EEG signal was recorded after every picture displayed.

This study includes data from 11 subjects. Recording from each 64 channel is considered. Each subject has 50-70 trials, further divided into train and test set using a k-fold cross-validation approach.

B. Feature extraction

The autorregressive burgs model is used for feature extraction. It is a time-domain representation of the type of random process. Here, it forecasts interested variables using the previous values of the variable [15]. It is given by (1):

\[ X_t = c + \sum_{i=1}^{p} \theta_i X_{t-i} + \varepsilon_t \]  

Where, model coefficients\(= (\theta_1, \theta_2, ..., \theta_p)\), c is constant and \(\varepsilon_t\) = white noise. Equation (1) is named the AR(p) model, where p is the autoregression model's order. The autoregression model's order is the number of nearly previous values in the sequence used to predict the current value as in (2):

\[ y_t = c + \theta_1 y_{t-1} + \theta_2 y_{t-2} + \varepsilon_t \]  

Equation (2) is called AR (2) because values at time t are predicted from time t-1, t-2. Equation (2) can be written as (3) when lag operator B is introduced:

\[ X_t = c + \sum_{i=1}^{p} \theta_i B^i X_{t-i} + \varepsilon_t \]  

So, the \(X_t\) it can be represented as a series of AR coefficients \(\theta_i\) and white noise \(\varepsilon_t\). In this research, AR model coefficients are used as features of the biometric system because they may reveal various intrinsic characteristics.

C. Classification

1) Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning model used for classification and regression tasks. Its main motive is to find the hyperplane in N-D space (where N is the number of features) that classifies the data points that are closer to the hyperplane are called support vectors; this hyperplane is a decision boundary. Kernel functions are used by support vector machines to find support vector classifiers in higher dimensions systematically. The support vector machine uses a kernel trick to reduce the amount of computation. In the kernel trick, kernel functions do not transform lower dimension points in a higher dimension, but it only calculates the relationship between every pair of points as if they are in a higher dimension. So, this trick of calculating a higher dimensional relationship without actually transforming the data to a higher dimension is called the kernel trick.

There are many types of the kernel which can be used in support vector classification such as:

A polynomial kernel with a parameter \(d\) (\(d=\)degree of a polynomial), This value of \(d\) defines the dimension for a support vector classifier. A good value of \(d\) can be found using cross-validation. The polynomial kernel can be represented by (4):

\[ (a \cdot b + r)^d \]  

Where, \(r = \) coefficient of the polynomial, \(d = \) degree of a polynomial, \(a,b = 2\) different observations of dataset.

Radial Basis Function (RBF) kernel is also commonly used in SVM. It finds support vector classifiers in higher dimensions. RBF kernel is given by (5):

\[ e^{-\gamma (a-b)^2} \]  

\(\gamma\) is determined by cross-validation. Apart from that, there are other kernels like the linear kernel and sigmoid kernels are also used.

2) k-Nearest Neighbor

k-nearest neighbor (k-NN) is a supervised machine learning algorithm used in both classification and regression tasks. k-NN assumes that similar things are present nearby. It uses feature similarity to predict data points. It predicts the data based on the training sample, which is close to the new data point. Alternatively, it can be said that a new data point is predicted based on the majority of its neighbors’ votes. A distance measure is used to find which k instances in a dataset are similar to new data points such as Euclidean, Hamming, Manhattan and Minkowski Distance, etc. Here distances between the test vector and all other vectors in the database are calculated and based on distances, k number of nearest vectors are chosen. The value of k can be found by algorithm tuning.

3) Multilayer perceptron

A multilayer perceptron (MLP) is a supervised learning algorithm. Perceptron contains a fully connected input and output layer. MLP has the same input and output layer, but it can have multiple hidden layers. The multilayer perceptron can have more than one linear layer, shown in Fig. 2.

![Figure 2: Neural network](image-url)

Data is given to the input layer, and final results are obtained from the output layer. Hidden layers can be increased based on the requirement. Weights are multiplied to input (using dot product) before passing through the hidden layer. The activation function is used at the hidden layer to process input coming from the previous layer. This output of the activation function is passed through the next layer by taking the dot product.
product with corresponding weights. Many activation functions can be used, such as ReLU, Sigmoid, SoftMax, Tanh, etc. Optimizers are also used in neural networks to train weights such as Adam optimizer, SGD optimizer, and lbfgs optimizer. Adam is a stochastic gradient-based optimizer, sgd is stochastic gradient descent, and lbfgs is Limited-memory BFGS (Broyden-Fletcher-Shanno). Finally, the output is taken from the output layer, and the backpropagation algorithm is used at the output layer.

4) XG-Boost

XG-Boost is a machine learning algorithm based on an ensemble decision tree and uses a gradient boosting framework. It was developed by Tianqi Chen and Carlos Guestrin [16] as their research project at the University of Washington. XGBoost is called an extreme gradient boosting technique. It is based on the principle which uses gradient descent to boost weak learners, or it can be said that it is the implementation of gradient boosted decision trees.

Several gradient tree boosting algorithms are there; one of them is a regularized learning objective. Which explains that for a given dataset having n number of examples and m features given by (6) & (7):

\[ D = \{(x_i, y_i)\} | D| = n, \ x_i \in \mathbb{R}^m, y_i \in \mathbb{R} \] (6)

\[ \tilde{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} \] (7)

Where \( \mathcal{F} = \{ f(x) = w_{q(x)} \} | q: \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T \} \) is the space of regression trees. In this q = structure of each tree, T gives the number of leaves, w = leaf weight, k = additive function used in the tree ensemble method.

The tree ensemble model can give minimized regularized objective, given by (8)

\[ \mathcal{L}(\phi) = \sum_{i} l(\tilde{y}_i, y_i) + \lambda \Omega(\phi) \] (8)

\[ \Omega(\phi) = \gamma T + \frac{1}{2} \lambda \| w \|^2 \]

and \( l \) calculate the difference between predicted and target variables, i.e., \( \tilde{y}_i, y_i \) respectively. These regularization equations help in avoiding overfitting by smoothing the final learned weights.

IV. Results and Analysis

Biometrics application requires more secured and authentic parameters to increase the security and privacy of any system. In order to enhance the correctness of the biometric data, the best combination of feature extraction, data classification, and optimization is required through this work a sincere effort to increase the efficiency of a model by using a most optimal combination of feature extraction and classification, to come out with a perfect biometric parameter, i.e., EEG signals.

Various features in the autoregressive model with order three are used in the present work. As there are 64 channels for which the total number of features calculated are 64*3, i.e., 192 features for a single user, which in turn have 763*192 feature vector as each user has to pass through 70-76 trials. After getting the required feature vectors, an effort is being made to select an optimal classifier, which can classify the AR model’s extracted features.

Different classifiers are used for classification such as SVM, k-NN, MLP, and XG Boost to classify extracted features. Accuracy values up to 99.7% are achieved using different optimizers for each classifier, as shown in Fig. 17.

Instead of randomly splitting tests and training data, the data has been divided using a cross-validation approach. So that results will be more accurate, and the problem of overfitting will not occur. In this work, ten-fold cross-validation has been used.

A. Cross-validation

In this approach, data has been divided into train, test, and validation. Train and test can be used for final calculation, but the validation set is not used. In this, training data is divided into k number of smaller sets. For training, k-1 sets are used, and the remaining part of k-folds is used to validate the resulting model. Similarly, training is done for each k-1 set, and performance is calculated using the average of all values computed using a loop. This approach is known as k-fold cross-validation. These training and validation set is given to the classifier model using a loop. Then this classifier model classifies the data, i.e., it predicts the users based on their features. Confusion matrix and classification reports are used to check the performance of the classification model.

B. Confusion matrix (CM)

Confusion matrix helps in understanding the quality of predictions. It provides the text summary of precision, recall, F1-score, and accuracy for each class.

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>No</td>
<td>FP</td>
<td>TN</td>
</tr>
</tbody>
</table>

Table 1. Confusion Matrix

TP (True positive) - When the prediction is true.
TN (True negative) - Observation that is negative and is predicted to be negative.
FP (False positive) - Observation that is negative but predicted as positive.
FN (False negative) - Observation that is positive but predicted as negative.

C. Classification report

It is used to find the quality of prediction from any classification model i.e. out of all predictions, how many are correct and how many are incorrect.

It provides the text summary of precision, recall, F1-score, and accuracy for each class.

- **Recall:** It tells about all the positive samples. The recall is given by:

  \[ \text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \]

- **Precision:** Precision tells about the accuracy of positive predictions. Precision is given by:

  \[ \text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \]
• **F1-Score**: It is the weighted harmonic mean of precision and recall. It’s best score is 1, and the worst is 0.

\[
F1\text{- score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

• **Accuracy**: Accuracy is also known as classification rate and is given by:

\[
\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}
\]

• **Support**: It is the number of occurrences of each class in our actual test set.

### D. Experimental Results

In this research, 10-fold cross-validation is used for train test split and applied five iterations, i.e., repeat the whole process five times, and final accuracy is calculated using the mean of all five accuracies acquired after each iteration.

1) **Support vector machine**

SVM is used as a classifier in this work, and classification is done using different kernel functions, and each kernel function gave us a different value. Three kernel functions are employed: linear kernel, sigmoid kernel, and radial basis kernel. Fig. 3 shows the average accuracy of five iterations using different kernels.

![Accuracy plot for SVM](image)

**Figure 3**: Accuracy for different SVM kernels

Among all three kernels, the highest accuracy values are acquired from the linear kernel it provides accuracy up-to 99.7%, and between the different fold. Fig. 4 shows the confusion matrix for ten folds obtained after applying the SVM model with a linear kernel. Fig. 5 illustrates the confusion matrix for one-fold, which shows the predictions for 11 users where all predictions are correct, i.e., no predictions are false.

![Confusion Matrix](image)

**Figure 5**: Confusion matrix for one-fold using SVM linear kernel

Fig. 6 shows the classification report one-fold using SVM with the linear kernel as a classifier.

![Classification report for SVM linear classifier](image)

**Figure 6**: Classification report for SVM linear classifier

Fig. 4 shows the confusion matrix, representing no false-positive or false-negative results. It shows that there is a total of 76 test trials, out of which 69 are true positive, and seven are true negative, so it can be said that predictions are 100% correct in this trial, which is also shown in the classification report where 100% accuracy is obtained for this trial, which is represented by 1 in Fig. 6.

2) **k-Nearest Neighbor**

k nearest neighbor is used with different k values. Each k value gives a different result. The results show that when k=3, then maximum accuracy, 96.8% is achieved, shown in Fig 7. Fig.8 shows the confusion matrix for k-NN when k=3, which shows misclassification for three users. Similarly, Fig. 8 shows the classification report for k-NN with k=3, which shows that values of precision, recall, and f1-score are 100% except for users 2, 3, and 5, which has less score means there is classification error.
A multilayer perceptron classifier is used to identify individuals. Three different weight optimizers are applied, such as adam optimizer, sgd optimizer, and lbfgs optimizer, and got different accuracy values. Activation function relu is used with all optimizers. Fig. 10 shows the accuracy values obtained after applying MLP with all optimizers.

The accuracy table shows that the maximum accuracy is achieved using the ADAM optimizer. These accuracies values are mean accuracies of five iterations, and each iteration has ten folds. Using the ADAM optimizer, accuracy up to 100% is also achieved in some folds. Figures 11, 12, and 13 show the result obtained after applying MLP classifier with ADAM optimizer.
Figure 12(a),(b): confusion matrix of two folds for MLP with adam optimizer

Fig. 11 shows the confusion matrix for the MLP classifier using the ADAM optimizer. The confusion matrix shows that out of 10 predictions, eight predictions are 100% correct, and only two predictions have a false negative and false positive value. Fig. 12 (a) and (b) show the confusion matrix for two random folds where 1-fold gives one false prediction i.e., user 3 is predicted six times as user 3 but is predicted one time as user 2, and the second fold shows all correct prediction. Fig. 13 shows the classification report obtained after applying MLP classifier with ADAM optimizer, which shows that the average accuracy obtained is 99% and precision, recall, and f1-score for user all users are 100% except user two and six.

Figure 13: Classification report for MLP classifier with ADAM optimizer

4) XGBoost

Extreme gradient boosting is applied to classify individuals based on their features. XGBoost classifier is used with three different boosting techniques, such as gbtree, gbliner, and dart. All three techniques provide different accuracy values.

Fig. 14 shows that maximum accuracy is obtained using the gbliner method. Fig. 15 and 16 show the confusion matrix and classification report obtained after applying XGBoost classifier with the gbliner boosting method. Fig. 15 shows one false positive prediction for user two and one false negative prediction for user six while all other predictions are correct. The same results are obtained in the classification report shown in Fig. 16 i.e., users 2 and 6 have some mispredictions while all other users had 100% precision, recall, and f1-score.

Figure 14: Accuracies for XGBoost classifier with different boosting techniques.

Figure 15: Confusion matrix for xgboost with gbliner boosting technique.

Figure 16: Classification report for XGBoost with a gbliner boosting technique.

E. Comparison results

Fig. 17 compares each classifier's accuracy values, which is calculated by taking the mean of accuracy values obtained in 5
iterations. Fig. 18 shows the results of all classifiers for each fold. In this work, ten folds are considered for cross-validation. Fig. 17 shows that the highest accuracy is obtained in the SVM classifier with a linear kernel and XGBoost with a gblinear boosting method. Apart from that, all other methods also give pretty good accuracy.

**Figure 17:** Mean accuracy plot for all classifier

**Figure 18:** Accuracy value for ten folds of each classifier

Table 2 shows the comparison with existing research done in the area of an EEG based biometric system using an autoregressive model as a feature extraction technique and different classification techniques.
V. CONCLUSION

The acquisition of biometric signals is a tedious and very precise activity. Also, one may not get a more significant number of users who are willing to provide their EEG signals. There is a scarcity of valid and big sample spaces in the field of EEG. Many machine learning algorithms are employed in this research to process the signals to be used as biometric traits. For feature extraction of EEG signals, an autoregressive model with order three is used. The extracted features are further classified based on different methods i.e., SVM, k-NN, MLP, and XGBoost. K-fold cross-validation with k=10 is used to increase the skills of the classifier. These classifiers are tested with different kernels and optimizers to obtain better accuracy. SVM is tested with three different kernels out of which the linear kernel provides maximum accuracy of 99.7%, among other kernels such as sigmoid (74.5%) and radial basis function (84.2%). k-NN is tested with three different values of k ranging from 3, 4, 5, and maximum accuracy of 96.8% is achieved with k=3 while k=4 gives 96.4%, and k=5 gives 96.5% accuracy. MLP is tested with three different optimizers, out of which adam optimizer gives maximum accuracy as 99.5% among other optimizers such as stochastic gradient descent 90.1%, and lbfgs 97.9%. Similarly, XGBoost is tested with three different boosting techniques in which glinear gives maximum accuracy of 99.7% while gbtree gives accuracy as 97.8%, and dart gives 98% accuracy. Results show that SVM and XGBoost classifiers and the AR model as a feature extraction technique give better results. The major issue in present research work lies with the size of the sample spaces. Suppose the size of the sample space is increased, then the efficiency percentage slashes down by a few points. In future work, efforts will be made to work on large datasets to get similar accurate and efficient values.

Acknowledgment

I would like to thank my Ph.D. supervisor, Dr. Manoj Duhan, for providing many helpful comments during this work and for reviewing various drafts of this paper.

References

[8] P. Campisi, S. Member, D. La Rocca, and S. Member, “Brain Waves for Automatic Biometric-Based User Recognition,” IEEE TRANSACTIONS ON


Author Biographies


Prof. Manoj Duhan is working as a Professor in the department of Electronics & Communication of DCRUST Murthal, Haryana. His research interests include Semiconductor Devices, Communication Systems, DSP, Reliability and Optimization Techniques. He has published 46 papers in International Journals, 13 papers in National Journals, 15 papers in International Conferences and 35 papers in National Conferences. Dr. Duhan has received Best Lecturer Award from Governor of Haryana in 1998.