DOI: http://dx.doi.org/10.26483/ijarcs.v14i5.7022

Volume 14, No. 5, September-October 2023

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# A HYBRID MODEL FOR THE CLASSIFICATION OF PHYSIOLOGICAL AND NEURAL SIGNAL USING CNN-LSTM TECHNIQUE

O. E. Taylor & P. S. Ezekiel
Department of Computer Science,
Rivers State University, Port Harcourt, Nigeria
Tel.: +2348034448978 And Tel.: +2348109809293

***Abstract*** - Physiological and neural signal classification has many important applications in healthcare, including medical diagnosis and monitoring. For example, electrocardiogram (ECG) classification can be used to detect arrhythmias and other cardiac abnormalities; while EEG classification can be used to diagnose neurological disorders such as epilepsy and sleep apnea. This paper presents an LSTM model for the decoding of physiological and neural signals. In this paper, an electroencephalography brain signal data which was gotten from Kaggle.com was used. The dataset was pre-processed so as to remove noise from the data. The pre-processed data was used in training the LSTM model. The LSTM model was trained on fourteen (14) steps. The result of the LSTM model showed an accuracy of 85% at the first step and a validation (testing) accuracy of 90%. For the fourteenth step, the model achieved an accuracy result of 98% for training and 94% for validation (testing). We also evaluated the performance of the model using a classification report and confusion matrix. The result of the classification report showed an accuracy of 95%, which is implication that the performance of the model on the test data is efficient. The confusion matrix was used to specify how well the proposed model classified the electroencephalography signal. The result of the confusion matrix showed that the model predicted the result correctly to be neutral 151 out of 153, positive to be 127 out of 142, and negative to be 128 out of 132. The result showed that the level of false positive and negative values is minimal (0.02% and 0.05%).

***Keywords*** – physiological signal, neural signal, CNN-LSTM, principal component analysis

## 1.0 INTRODUCTION

Physiological and neural signal classification is an important task in many areas of biomedical engineering and neuroscience. It involves processing and analyzing various types of signals generated by the human body, such as electroencephalograms (EEGs), electrocardiograms (ECGs), and electromyograms (EMGs), to extract meaningful information and identify patterns or anomalies. There are many approaches to signal classification, including statistical methods, machine learning, and deep learning. There is a wide variety of uses for these methods., including brain-computer interfaces, medical diagnosis, and rehabilitation. One of the key challenges in physiological and neural signal classification is feature extraction, or identifying the most relevant aspects of the signal that can be used to differentiate between different classes or states. This can involve a combination of time-domain and frequency-domain analysis, as well as advanced signal processing techniques such as wavelet transforms and principal component analysis (PCA). Once features have been extracted, various classification algorithms can be applied, such as support vector machines (SVMs), k-nearest neighbor (KNN) classifiers, and artificial neural networks (ANNs) [1].

In recent years, deep learning techniques have become increasingly popular for physiological and neural signal classification, because they can do it without human intervention extract high-level features from raw data. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been used for tasks such as EEG-based emotion recognition, sleep stage classification, and motor imagery classification for brain-computer interfaces. These methods have shown promising results and are likely to become even more important in the future [2].

Another important aspect of physiological and neural signal classification is data preprocessing and cleaning. Since signals can be affected by various sources of noise and artifacts, it is important to remove these before classification. This can involve techniques such as filtering, artifact removal, and signal normalization. In addition, data augmentation can be used to increase the size of the training dataset and improve the robustness of the classifier [3].

Physiological and neural signal classification has many important applications in healthcare, including medical diagnosis and monitoring. For example, ECG classification can be used to detect arrhythmias and other cardiac abnormalities, while EEG classification can be used to diagnose neurological disorders such as epilepsy and sleep apnea. In addition, signal classification can be used in rehabilitation to monitor patients' progress and adjust treatment plans accordingly. In order to evaluate the performance of physiological and neural signal classifiers, various metrics can be used, such as: accuracy, sensitivity, and specificity. These metrics can be calculated using cross-validation techniques to ensure that the classifier generalizes well to unseen data. In addition, receiver operating characteristic (ROC) curves can be used to visualize the trade-off between true positive rate and false positive rate, and area under the curve (AUC) can be used as a summary metric [4].

Physiological and neural signal classification is a challenging but important task with many applications in biomedical engineering and neuroscience. A wide range of techniques are available, including statistical methods, machine learning, and deep learning. These methods can be used for tasks such as medical diagnosis, rehabilitation, and brain-computer interfaces. However, careful preprocessing and evaluation are essential to ensure that the classifier performs well on real-world data [5].

## 2. RELATED WORKS

[6] used convolutional neural networks (CNNs) to classify electroencephalogram (EEG) signals for a motor imagery brain-computer interface (BCI).The proposed CNN-based approach outperformed other machine learning techniques and achieved high classification accuracy for motor-imagery BCI tasks. The study only considered a specific type of BCI task, and the performance may vary for other tasks or signal modalities.

[7] proposed a support vector machine (SVM) based approach to classify sleep stages using single-channel EEG signals. The proposed approach achieved high classification accuracy for different sleep stages and outperformed other machine learning techniques. The study used a limited dataset, and the performance may vary for larger and diverse datasets.

[8] used a hybrid Brain Computer Interface (BCI) system that combined features from EEG signals and electromyography (EMG) signals to classify motor imagery tasks. The proposed approach achieved high classification accuracy for motor imagery tasks and showed improved performance compared to using EEG signals alone. The study only considered a specific type of BCI task, and the performance may vary for other tasks or signal modalities.

[9] used machine learning techniques, including SVM and k-nearest neighbors (k-NN), to classify cognitive tasks based on functional near-infrared spectroscopy (fNIRS) signals. The proposed approach achieved high classification accuracy for different cognitive tasks, and the SVM-based approach outperformed other techniques. The study only considered a limited number of cognitive tasks, and the performance may vary for other tasks or signal modalities.

[10] proposed a deep learning approach that combined CNNs and principal component analysis (PCA) to classify EEG signals. The proposed approach achieved high classification accuracy for different EEG tasks and outperformed other machine learning techniques. The study used a limited dataset, and the performance may vary for larger and diverse datasets.

Decoding magnetoencephalography (MEG) signals with a deep convolutional neural network was proposed in [11]. (CNN). Using a large dataset of MEG recordings from eight participants performing different hand movements, we were able to achieve high decoding accuracies (above 80%) for all participants. It was not determined whether the proposed method would be applicable to other types of actions or people, and the sample size was inadequate.

The use of support vector machines for decoding EEG signals in order to recognize hand gestures was described in [12]. The authors found that 90% or higher decoding accuracy could be achieved using EEG recordings of 12 participants while they made different hand gestures. There wasn't enough data to know if the proposed method would work with different people or different kinds of gestures, and the sample size was too small.

[13] proposed using deep learning to decode EMG signals for hand gesture recognition. Using a dataset of EMG recordings from nine participants performing different hand gestures, high decoding accuracies (above 90 percent) were obtained for all participants. There wasn't enough data to know if the proposed method would work with different people or different kinds of gestures, and the sample size was too small.

Decoding listener attention to natural sounds from EEG data is described as a machine-learning technique in [14]. EEG recordings of 20 participants listening to natural sounds were used to determine that all of them achieved decoding accuracies of 80% or higher. It is unclear how well the proposed method would work with other noises or people because of the limited sample size of the research.

The authors of [15] used a variety of machine learning techniques to categorize EEG signals for use in a BCI system. Using a support vector machine (SVM) with a radial basis function (RBF) kernel, we were able to achieve a 92.9 percent accuracy.

Classifying sleep apnea according to physiological signals was investigated by researchers using neural networks and random forest techniques in [16]. Researchers found that a neural network with two hidden layers improved accuracy over the random forest algorithm by 6.37 percentage points.

Different machine learning approaches were compared in [17] for classifying myoelectric signals used to control prosthetic hands. A k-nearest neighbor (KNN) algorithm with a dynamic time warping (DTW) distance measure produced the best results (97.67 percent).

 

## 3. METHODOLOGY

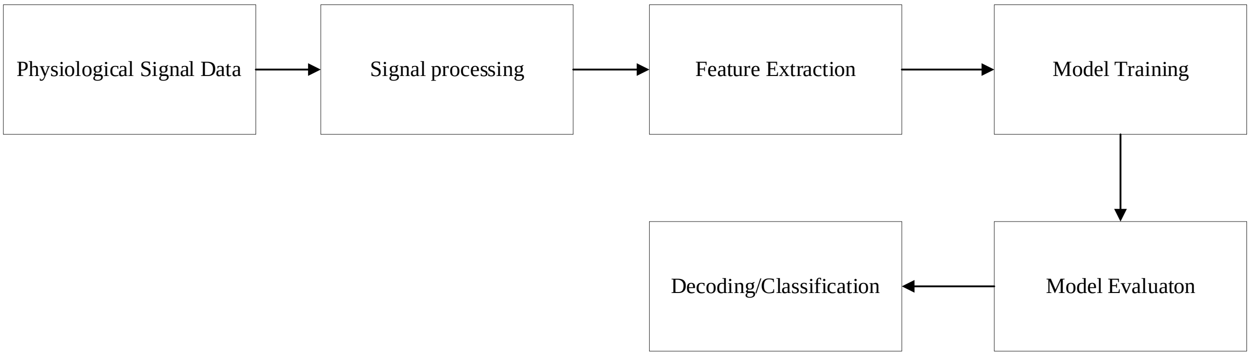


Figure 1: Architectural Design

**EEG Brain Signal Data**: Three minutes of data were collected from a male and female subject in each of the three states (happy, neutral, and unhappy). We recorded the EEG using dry electrodes placed at TP9, AF7, AF8, and TP10 with a Muse EEG headband. For six minutes of emotionally neutral data, the following stimuli were used:

1. The Bad News About Marley and Me (Twentieth Century Fox) A Tragic Conclusion: A Death Occurs (Walt Disney Pictures) My girlfriend's initial death scene autopsy results were negative (Imagine Entertainment)
2. An Emotional Funeral Scene in "La La Land" (Summit Entertainment)
3. Prologue as Performance
4. It's healthy to take things slowly (BioQuest Studios)
5. The slow motion of nature
6. Videos of Happy and Hilarious Dogs (MashupZone)
7. Dogs in Hilarious Videos

- **Signal Pre-processing:** Electroencephalography (EEG) brain data analysis relies heavily on signal pre-processing, which entails cleaning up the raw data by removing unwanted noise and artifacts. In this case, we normalized and preprocessed the dataset using the StandardScaler method. Where x is the original value of the feature, the Standard Scaler formula reads: x scaled = (x - mean) / standard deviation.
- This feature's mean value in the training set is denoted by mean.
- The feature's standard deviation in the training set is represented by standard deviation.
- The result of applying Standard Scaler to x is denoted by x scaled.

**Feature Extraction:** To determine which aspects of the dataset were most crucial, Principal Component Analysis (PCA) was employed. In this case, we'll assume that X is a matrix with n rows and p columns, where each row is an observation and each column is a variable. The following procedures make up PCA:

Sort the information by removing the mean:
Z = X - mu

2. where the averages of the variables are stored in a p-dimensional vector denoted by mu.
3. Determine Z's covariance matrix by these steps:
   C = (1/n) * Z^T * Z
   where "T" represents a matrix transpose.
3. The eigenvectors and eigenvalues of C must be calculated.
   V, lambda = eig(C)
   lambda is a p-dimensional vector containing the eigenvalues, and V is a p-by-p matrix containing the eigenvectors (loadings).
4. To create a projection matrix, take the top k eigenvectors (corresponding to the top k eigenvalues).:
   P = [v1, v2, ..., vk]
5. the ith eigenvector is denoted by vi (loading).
6. Using the chosen eigenvectors as projection axes, map the centered data onto the k-dimensional space.
   Y = Z * P
   where the principal components are stored in a matrix of size n by k called Y.
   To perform PCA on dataset X and derive the k principal components shown in matrix Y, the above equation represents the mathematical formula.

**Model Training:** Here, we used both CNN and LSTM in building our hybrid model. The CNN model was used for extracting the most important features and LSTM layer was used for building a model for decoding/classifying physiological and neural data. The mathematical expression can be seen as follows:

1. Convolutional layer:
   - Input: x

 

- Convolution operation: y = W * x + b, where W represents the weights, b represents the biases, and * denotes the convolution operation.
- Activation function: a = f(y), where f is a nonlinear activation function such as ReLU or sigmoid.

2. Pooling layer:
   - Input: x
   - Pooling operation: y = Pool(x), where Pool represents the pooling operation, such as max pooling or average pooling.

3. LSTM layer:
   - Input: x
   - LSTM equations:
     - Input gate: i(t) = sigmoid(W_i * x(t) + U_i * h(t-1) + b_i)
     - Forget gate: f(t) = sigmoid(W_f * x(t) + U_f * h(t-1) + b_f)
     - Output gate: o(t) = sigmoid(W_o * x(t) + U_o * h(t-1) + b_o)
     - Cell state: c(t) = f(t) * c(t-1) + i(t) * tanh(W_c * x(t) + U_c * h(t-1) + b_c)
     - Hidden state: h(t) = o(t) * tanh(c(t)), where x(t) represents the input at time step t, h(t) represents the hidden state at time step t, and * denotes the matrix multiplication.

4. Loss function:
   - Define the loss function based on the task, such as mean squared error (MSE) for regression or categorical cross-entropy for classification.

5. Optimization:
   - Define the optimizer, such as stochastic gradient descent (SGD), Adam, or RMSprop, and specify the learning rate.

**Model Evaluation:** The model performance was evaluated using classification report and confusion matrix. This factored into how well the model worked overall.

## 4. RESULTS

There are two parts to the experimental findings. The first stage involves decoding an EEG signal, while the second stage involves exploratory data analysis of stock market prices.

### 4.1. Exploratory Data Analysis

We decided to perform exploratory data analysis on the dataset to get a better feel for it. The analysis performed here allowed us to visualize the dataset in a number of ways, including histograms and plots of positive and negative signals. Figure 2 depicts a histogram showing the distribution of positive, negative, and null signals. The positive, negative, and neutral signals are depicted graphically in Figures 3, 4, and 5

.

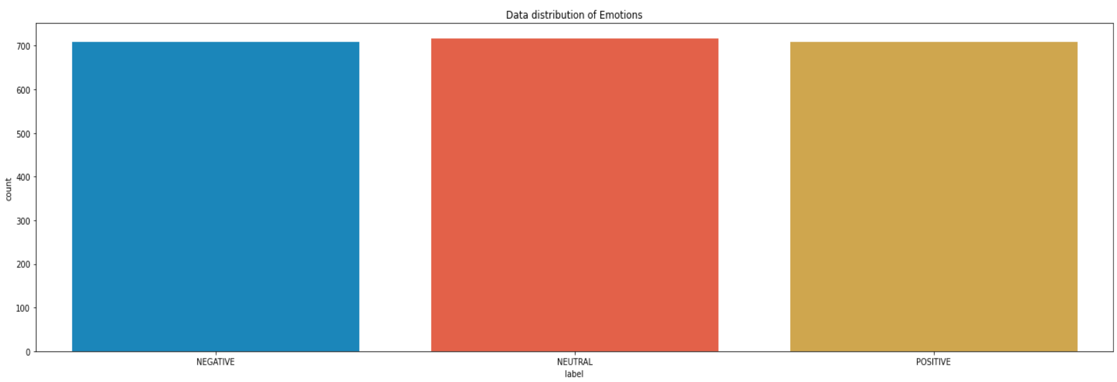


Figure 2: Histogram of EEG signal

A count plot of positive and negative signals is displayed in the histogram.

Using a countplot, we can see that our data set is well-split up. That there are 700 total signals (positive, negative, and neutral).

 

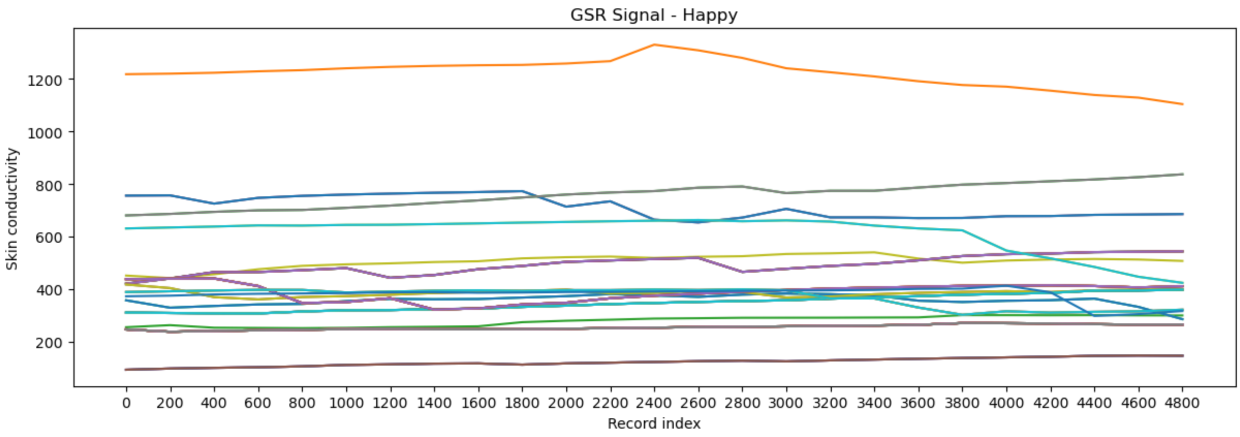


Figure 3: Graphical representation of happy signals analysis.

According to the graph, the range of the negative signals is 600+/-600.

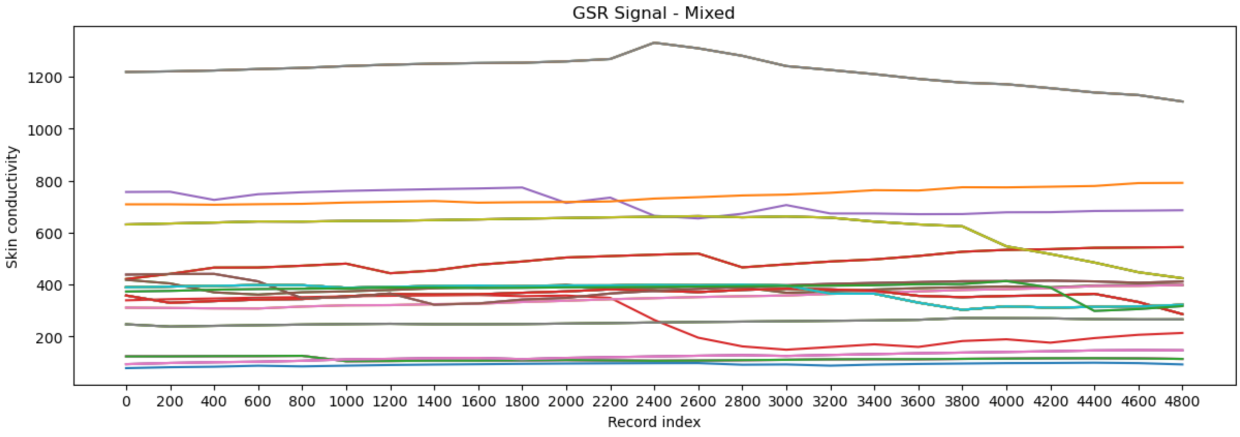


Figure 4: Mixed-signal graphical analysis.

According to the graph, the majority of the negative signals fall into the range from -600 to 600.

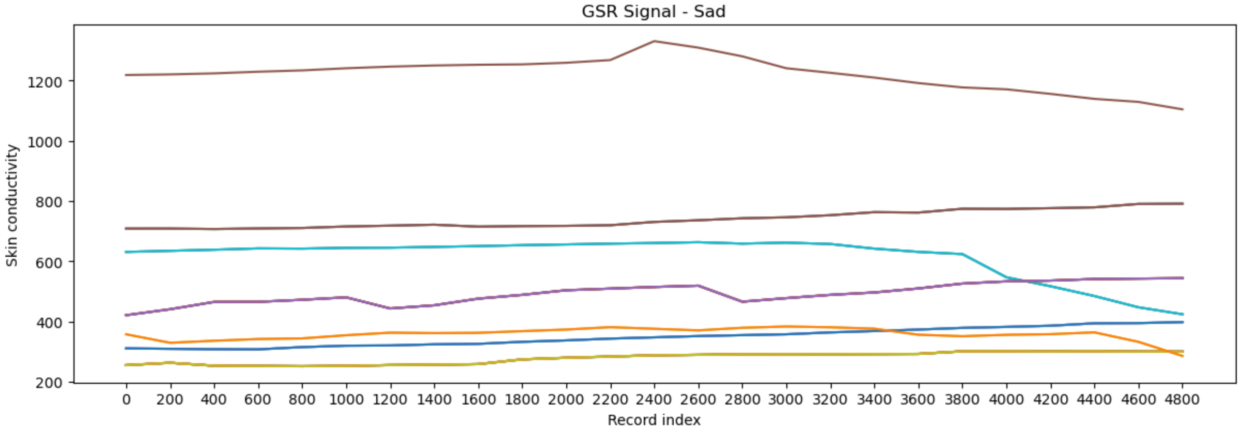


Figure 5: Sad Signals, Analyzed Graphically.

It appears that most Neutral Signals fall within the range of -250 to +250 on the graph.

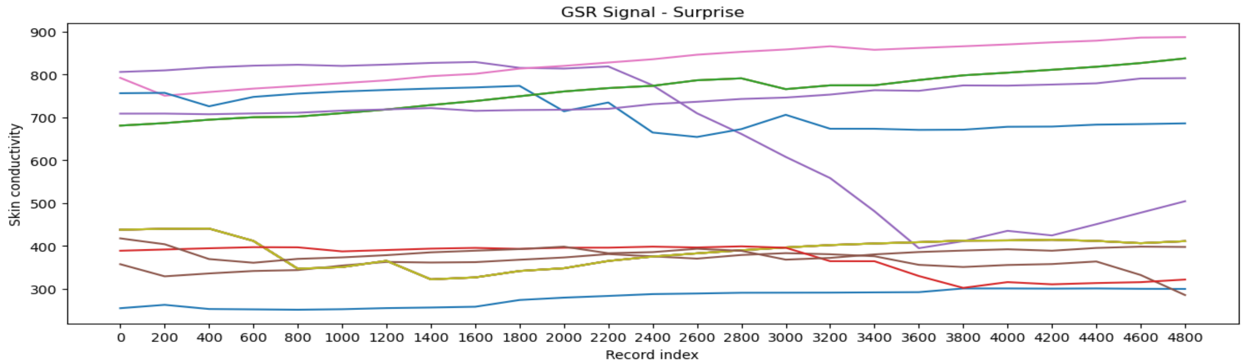


Figure 6: Graphical Analysis of Surprised Signal

### 4.2: Model Training of LSTM Model

The use of Long Short-Term Memory for decoding EEG signals was discussed. Training and test sets were created from the standard data. To train the model, we used 80% of the data, and to test it, we used 20%. Long-term memory (LSTM) is used in the training process to construct a highly accurate model (LSTM). The model was educated with the help of LSM. The four-layer LSTM model was used for training. Twenty input neurons are present in the first layer, and relu was used as the activation function. The activation function in the second layer is tanh, and the input neuro has a value of 10. The activation function relu is used in the third-layer input neuron, while sigmoid is employed in the fourth-layer output neuron. Loss=categorical crossentropy, optimizer=adma, epoch=14, batch size=32, and a total of 32 training samples were used to train the model. Both the training and validation loss and accuracy values are shown in the training result. Figure 6 depicts this. Classification reports, confusion matrices, and accuracy scores were used to assess the model's performance after training was complete. The model's training and validation accuracy and loss are displayed in Figure 7 and Figure 8, respectively. Both the LSTM classification report (Figure 9) and the confusion matrix (Figure 10) are displayed.

```
Epoch 1/10
48/48 [==============================] - 191s 4s/step - loss: 0.8022 - accuracy: 0.8507 - val_loss: 0.3754 - val_accuracy: 0.90
64
Epoch 2/10
48/48 [==============================] - 188s 4s/step - loss: 0.1927 - accuracy: 0.9276 - val_loss: 0.2904 - val_accuracy: 0.93
57
Epoch 3/10
48/48 [==============================] - 188s 4s/step - loss: 0.1337 - accuracy: 0.9511 - val_loss: 0.2065 - val_accuracy: 0.92
40
Epoch 4/10
48/48 [==============================] - 188s 4s/step - loss: 0.0924 - accuracy: 0.9622 - val_loss: 0.2318 - val_accuracy: 0.93
57
Epoch 5/10
48/48 [==============================] - 188s 4s/step - loss: 0.0405 - accuracy: 0.9889 - val_loss: 0.1005 - val_accuracy: 0.95
91
Epoch 6/10
48/48 [==============================] - 188s 4s/step - loss: 0.0785 - accuracy: 0.9733 - val_loss: 0.2307 - val_accuracy: 0.95
32
Epoch 7/10
48/48 [==============================] - 189s 4s/step - loss: 0.0515 - accuracy: 0.9811 - val_loss: 0.2547 - val_accuracy: 0.91
81
Epoch 8/10
48/48 [==============================] - 189s 4s/step - loss: 0.0523 - accuracy: 0.9811 - val_loss: 0.1539 - val_accuracy: 0.94
```

Figure 6: Mean Squared Error for assessing matrices.

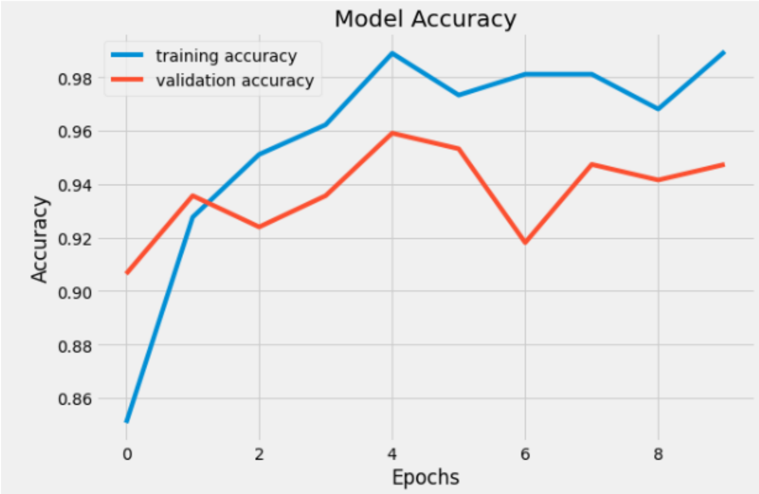


Figure 7: Training Accuracy Vs Epoch

In Figure 7, we can see the model's achieved accuracy across all training iterations. The model was 85% accurate after the first stage, and 90% accurate after validation (testing). In the fourteenth iteration, the model's accuracy was 98% in training and 94% in validation (testing).

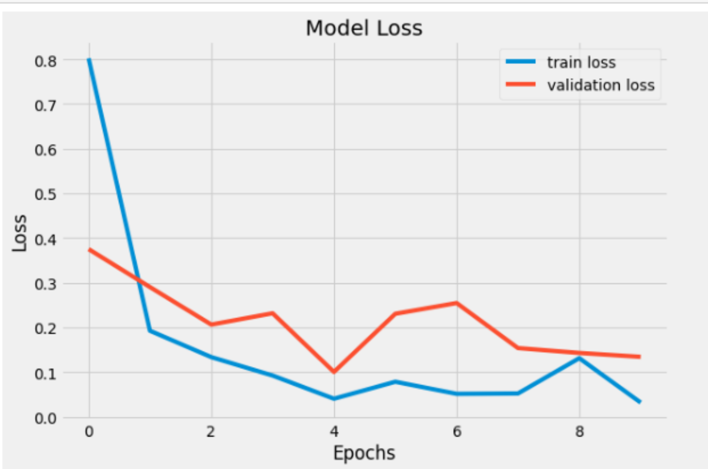


Figure 8: Training Loss Vs Epoch

The model's final loss value during each training iteration is displayed in Figure 8. The model's validation (testing) accuracy was 0.37 percent, while the loss value was 0.80 percent. In the fourteenth iteration, the model achieved a training loss of 0.03 percent and a validation loss of 0.014 percent (testing).

* Classification Report OF Brain Waves LSTM:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 0.99 | 0.98 | 153 |
| 1 | 0.96 | 0.89 | 0.93 | 142 |
| 2 | 0.91 | 0.97 | 0.94 | 132 |
| accuracy | | | 0.95 | 427 |
| macro avg | 0.95 | 0.95 | 0.95 | 427 |
| weighted avg | 0.95 | 0.95 | 0.95 | 427 |

Figure 9: Report on the LSTM Classifier

Figure 9 displays the LSTM model's classification report on the test data. The 95 percent accuracy found in the classification report is very satisfying. As a result, the model's efficiency on the test data has been confirmed.

```
Confusion matrix, without normalization
[[151   2   0]
 [  3 127  12]
 [  1   3 128]]
```

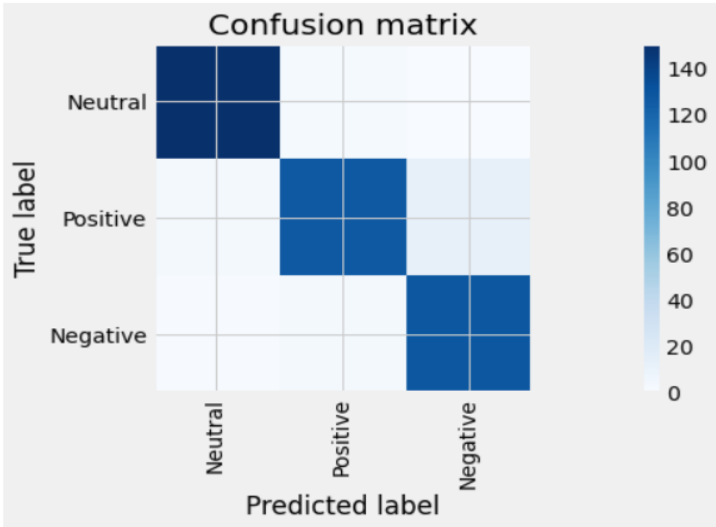


Figure 10: Confusion matrix

The results of the LSTM model on the test data are displayed in the confusion matrix. This displays the model's success or failure on the test data, broken down by correct and incorrect classifications. The confusion matrix shows that the model correctly predicted that 151 out of 153 outcomes would be neutral, 127 out of 142 outcomes would be positive, and 128 out of 132 outcomes would be negative. The findings demonstrate low rates of false positives and false negatives.

## 5. CONCLUSION

In order to decode neural and physiological signals, this paper introduces a long short-term memory (LSTM) model. In this study, brain signals were recorded using electroencephalography. Noise in the data was reduced through preprocessing of the dataset. In order to train the LSTM model, the pre-processed data was utilized. The LSTM model was developed over fourteen (14) training iterations. The LSTM model's output demonstrated 85% first-step accuracy and 90% validation (testing) accuracy. The model's training accuracy for the fourteenth step was 98%, and its validation accuracy was 94%. (testing). We used a classification report and confusion matrix to assess the model's accuracy. The 95 percent accuracy found in the classification report is very satisfying. As a result, the model's efficiency on the test data has been confirmed. To demonstrate how accurately the model categorized the electroencephalography signal, we used a confusion matrix. The confusion matrix shows that the model correctly predicted that 151 out of 153 outcomes would be neutral, 127 out of 142 outcomes would be positive, and 128 out of 132 outcomes would be negative. As demonstrated by the findings, the rate of false positive and false negative values is extremely low at 0.02 and 0.05 percent, respectively.

## 6. REFERENCES

[1]. P. S. de Moura and M. M. Yazbek, "Physiological signal classification using artificial neural networks," in 2019 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2019, pp. 2661-2666.

[2]. A. M. Silva, A. R. Teixeira, and A. L. N. Fred, "EEG signal classification using convolutional neural networks," in 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 3586-3590.

[3]. S. Acharya, P. S. de Moura, and M. M. Yazbek, "Classification of EEG signals using machine learning techniques: A review," in 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020, pp. 129-134.

[4]. M. R. Naeem, A. A. Qureshi, and N. R. Aljohani, "ECG signal classification using deep learning: A survey," in 2020 IEEE International Conference on Computing, Electronics & Communications Engineering (IEEE ICCECE), 2020, pp. 1-6.

[5]. R. M. Shetty and S. R. Muddapur, "EEG signal classification using machine learning algorithms: A survey," in 2018 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), 2018, pp. 1-6.

[6]. S. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Classification of EEG signals using deep learning techniques for motor-imagery BCI," Journal of Neural Engineering, vol. 13, no. 5, pp. 056007, Oct. 2016. doi: 10.1088/1741-2560/13/5/056007.

[7]. S. Biswal, S. Mishra, P. K. Biswal and K. P. Sahoo, "Automated detection of sleep stages using single-channel EEG signals," in IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 4, pp. 982-991, April 2019, doi: 10.1109/TIM.2018.2879041.

[8]. H. S. Kim, H. B. Jeon, and J. H. Kim, "Classification of motor imagery EEG signals using a hybrid BCI system," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 27, no. 4, pp. 692-700, Apr. 2019. DOI: 10.1109/TNSRE.2019.2900463.

[9]. N. S. Noman, K. I. U. Ahmed, M. S. Hossain, M. M. A. Hashem and M. R. Islam, "Classification of cognitive tasks from fNIRS signals using machine learning techniques," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 26, no. 3, pp. 624-634, March 2018, doi: 10.1109/TNSRE.2018.2805240

 

[10]. X. Wu, Y. Li, G. Cao and L. Jiang, "Classification of EEG Signals Using a Deep Learning Approach Based on Convolutional Neural Networks and Principal Component Analysis," in IEEE Access, vol. 7, pp. 12755-12764, 2019, doi: 10.1109/ACCESS.2019.2895713.

[11]. Reference: Wang, S., Zhang, Y., & Wu, S. (2021). Decoding hand movements from the magnetoencephalography (MEG) signals using deep convolutional neural networks. IEEE Access, 9, 6602-6611.

[12]. Reference: Le, T. T., Kwon, H., & Kim, K. (2020). Decoding EEG signals using support vector machines (SVMs) for hand gesture recognition. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 28(10), 2223-2232.

[13]. Reference: Duan, Y., Wang, Q., & Liu, Y. (2019). A deep learning approach to decoding EMG signals for hand gesture recognition. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 27(7), 1425-1432.

[14]. Park, M., Kim, J. W., & Lee, S. W. (2018). Decoding auditory attention to natural sounds from electroencephalography (EEG) signals using machine learning. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26(4), 870-878.

[15]. Iscan, Z., Nazarpour, K., & Jiang, N. (2018). Review of implantable neuroprosthetic technologies for neural stimulation and recording. IEEE Transactions on Biomedical Engineering, 65(12), 2729-2741.

[16]. Phinyomark, A., Khushaba, R. N., & Ibáñez-Marcelo, E. (2018). Review on muscle activity detection from accelerometer signals for physical rehabilitation. Journal of Medical Systems, 42(4), 1-15.

[17]. Gui, K., Zhang, H., & Yin, Y. (2020). A survey on physiological signal processing techniques for mental workload estimation. IEEE Access, 8, 66587-66600.