In: 2023 AIAA Aviation and Aeronautics Forum and Exposition (AIAA AVIATION Forum), 12-16 June 2023, San Diego, CA
DOI: 10.2514/6.2023-4529

A Comprehensive Analysis of Machine Learning and Deep Learning Models for Identifying Pilots' Mental States from Imbalanced Physiological Data

Ibrahim M. Alreshidi[1], Satendra Yadav[2], Irene Moulitsas[3], and Karl W. Jenkins[4]
School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford, MK43 0AL, UK

This study focuses on identifying pilots' mental states linked to attention-related human performance-limiting states (AHPLS) using a publicly released, imbalanced physiological dataset. The research integrates electroencephalography (EEG) with non-brain signals, such as electrocardiogram (ECG), galvanic skin response (GSR), and respiration, to create a deep learning architecture that combines one-dimensional Convolutional Neural Network (1D-CNN) and Long Short-Term Memory (LSTM) models. Addressing the data imbalance challenge, the study employs resampling techniques, specifically downsampling with cosine similarity and oversampling using Synthetic Minority Over-sampling Technique (SMOTE), to produce balanced datasets for enhanced model performance. An extensive evaluation of various machine learning and deep learning models, including XGBoost, AdaBoost, Random Forest (RF), Feed-Forward Neural Network (FFNN), standalone 1D-CNN, and standalone LSTM, is conducted to determine their efficacy in detecting pilots' mental states. The results contribute to the development of efficient mental state detection systems, highlighting the XGBoost algorithm and the proposed 1D-CNN+LSTM model as the most promising solutions for improving safety and performance in aviation and other industries where monitoring mental states is essential.

I. Introduction

THE ongoing evolution of the aviation industry hinges on maintaining rigorous safety standards, as advancements in aircraft design, endurance, and safety have contributed to a worldwide decrease in aircraft accidents [1,2]. Cognitive tendencies, particularly those related to attentional focus, are common among pilots and can be elicited by various factors such as cockpit alerts, extreme weather turbulence, takeoff, and landing. These situations can potentially impact the pilot's mental state, leading to changes in brain activity and increasing the risk of losing control of the aircraft. Data collected by the International Air Transport Association (IATA) from 2012 to 2021 in [3,4] revealed that 45 plane crashes, resulting in 1,645 fatalities, were caused by pilots losing control of the aircraft. This underscores the vital importance of mental sharpness in aircraft operations. Studies suggest that human factors contribute to over 70% of aviation accidents, making them a key factor in improving flight safety and management [5–7]. The Commercial Aviation Safety Team (CAST) investigated 18 aircraft accidents involving loss of control and

[1] Researcher, Centre for Computational Engineering Sciences, Machine Learning and Data Analytics Laboratory, Digital Aviation Research and Technology Centre (DARTeC), AIAA member.
[2] Student, Centre for Computational Engineering Sciences.
[3] Senior Lecturer and Course Director, Centre for Computational Engineering Sciences, Machine Learning and Data Analytics Laboratory, Digital Aviation Research and Technology Centre (DARTeC).
[4] Professor and Head, Centre for Computational Engineering Sciences.

discovered that flight crew attention deficiencies played a role in 16 of these incidents [8]. Consequently, CAST recommended that the aviation community conduct research to detect and evaluate attention-related human performance-limiting states (AHPLS), focusing on channelized attention (CA), diverted attention (DA), and startle/surprise (SS) mental states. CA occurs when pilots become absorbed in a puzzle-based video game, such as Tetris, to the detriment of other tasks, while DA involves pilots solving math problems that sporadically appear while monitoring displays. Pilots in the SS mental state experience unexpected disruptions to the primary flight display in the simulator. Addressing these attention-related mental states is crucial for enhancing flight safety and reducing the risk of accidents caused by cognitive limitations. By developing and implementing strategies to detect and manage these mental states, the aviation industry can further bolster its safety measures and ensure more secure flights for both passengers and crew.

Scholars and professionals in both academia and industry have devoted considerable effort to exploring the identification of pilots' mental states using physiological signals and machine learning (ML) techniques. These studies have employed quantitative sensors to record biological signals from the human body, with electroencephalography (EEG) receiving particular attention due to its capacity to capture short-lived changes in brain activity [9]. Nevertheless, EEG comes with certain drawbacks, as it is prone to artifacts arising from environmental factors and physiological events, which can adversely affect signal quality. To counteract these limitations, researchers frequently enhance their data collection by concurrently acquiring additional non-brain signals, such as electrocardiogram (ECG), galvanic skin response (GSR), and respiration (R), together with EEG from pilots. This multimodal approach offers a more comprehensive understanding of pilots' physiological responses, which can be leveraged to develop robust ML systems capable of detecting pilots' mental states [10]. The ultimate objective of this research is to enhance aviation safety by reducing the likelihood of accidents related to pilots' mental states.

However, data imbalance poses a significant challenge for researchers developing mental state detection systems using physiological signals. These detection systems depend on ML algorithms that require substantial amounts of labeled data for training. In real-world situations, some mental states occur less frequently than others, leading to unequal instance distributions in the dataset. This imbalance can result in biased models with poor performance in detecting underrepresented mental states, as the algorithms tend to focus on the majority class, ultimately undermining the detection system's overall effectiveness and utility [11].

The aim of this study is to investigate the potential for identifying mental states associated with AHPLS using an imbalanced, publicly available physiological dataset. The research offers two primary contributions. Firstly, it develops a comprehensive deep learning (DL) architecture, consisting of one-dimensional Convolutional Neural Network (1D-CNN) and Long Short-Term Memory (LSTM) models, designed to effectively combine EEG and non-brain signals. Secondly, the study addresses the data imbalance issue by employing data resampling techniques, such as downsampling and oversampling, to create more balanced datasets for improved model performance. In addition to the 1D-CNN and LSTM fusion model, the study also incorporates and critically analyzes the performance of other ML and DL models, including eXtreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost), Random Forest (RF), Feed-Forward Neural Network (FFNN), standalone 1D-CNN, and standalone LSTM. This comprehensive analysis aims to provide a more robust understanding of the strengths and weaknesses of each model when dealing with imbalanced physiological data and detecting pilots' mental states.

The performance results of ensemble learning models and DL models, along with the impact of data resampling techniques like Synthetic Minority Over-sampling Technique (SMOTE) and the combination of Cosine Similarity (CS) and SMOTE, are reported and compared. This comparison sheds light on the effectiveness of different models and resampling techniques in handling data imbalance and improving mental state detection in the context of AHPLS.

The remainder of the research paper is structured as follows: Section II offers an overview of the relevant literature. Section III delineates the utilized dataset, the preprocessing methods, feature extraction methods, data imbalance approaches, and the classification methods utilized in this study. Section IV presents and discusses the experimental findings. Finally, Section V. concludes the investigation and suggests future research directions.

II. Related Work

The literature on previous studies that have attempted to detect mental states or related tasks, such as emotion recognition and mental states detection, using ML and DL models, specifically 1D-CNNs and LSTMs, is reviewed.

A. Emotion Recognition

Various studies have explored the use of both DL models and traditional ML techniques to classify emotions using physiological signals [12–15]. For example, Tripathi et al. [12] utilized a 1D-CNN+LSTM model for accurate emotion classification on the Dataset for Emotion Analysis using Physiological and Audiovisual Signals (DEAP) [16], which

contains EEG and peripheral physiological signals. Similarly, Zheng and Lu [13] investigated critical frequency bands and channels for EEG-based emotion recognition using a 1D-CNN+LSTM model. On the other hand, Bhardwaj et al. [17] employed Support Vector Machines (SVM) and Linear Discriminant Analysis (LDA) classifiers to classify human emotions from EEG signals. Although these studies employed DL and ML models for emotion recognition using EEG signals, they did not specifically focus on predicting pilots' mental states or incorporate other physiological signals such as ECG, GSR, and respiration [18]. Roza et al. [15] employed a Multilayer Perceptron (MLP) model to identify emotions through analyzing physiological signals. The accuracy of the MLP model's performance varied between 55% and 100%, depending on the different sets of features used.

B. Mental States Detection

Numerous attempts have been made to classify individuals' cognitive states by combining EEG signals with a variety of ML and DL approaches. Previous research by Lal et al. [19], Jap et al. [20], Kar et al. [21], and Trejo et al. [22] investigated statistical alterations in EEG during driving simulation tasks to determine fatigue levels. Johnson et al. examined algorithms independent of probes for classifying three degrees of task complexity in an EEG-based flight simulator experiment. Binias et al. [23] implemented spatial pattern characteristics extracted from EEG signals and diverse ML methods to differentiate between specific brain activity states related to idle but focused visual cue anticipation and the following response. Sonnleitner et al. [24] applied regularized LDA to study the predictive power of EEG for detecting distraction in single-trial analyses. Chaudhuri et al. [25] focused on SVM classification of typical and fatigued states in a simulated setting using the source localization technique. Dehais et al. [26] used frequency features derived from shrinkage LDA to classify mental workload and typical states. Nevertheless, these investigations mainly depended on manually crafted EEG features for creating classifiers.

Deep learning techniques have been increasingly adopted for identifying cognitive states without external support. For instance, Patel et al. [27] implemented a neural network to detect early signs of driver fatigue using ECG data. Bashivan et al. [28] proposed a deep recurrent CNN for identifying workload states from multi-channel EEG signals. Hajinoroozi et al. [29] introduced a channel-wise CNN and a variation with restricted Boltzmann machine for determining suboptimal driver performance. Jiao et al. [30] demonstrated a deep CNN method for detecting mental workload levels from EEG data, integrating a fusion strategy with a pointwise gated Boltzmann machine for various EEG inputs. Zhang et al. [31] used a recurrent 3D CNN to learn spatial-spectral-temporal EEG features for assessing mental workload across tasks. Wu et al. [32] suggested a deep stacked contractive autoencoder network to learn fatigue-related features from raw EEG data for fatigue recognition. Gao et al. [33] developed an EEG-based spatial-temporal CNN for accurate fatigue state detection. However, it is essential to recognize that these studies focused solely on one type of signal for cognitive state detection.

Merging data from various biosignal sensors has proven to be a successful strategy for enhancing detection performance in comparison to single-sensor recognition. For instance, Hogervorst et al. [34] explored combined features from physiological signals such as EEG, ECG, and eye blinks for mental workload assessment. Ahn et al. [35] collected EEG, ECG, and Functional near-infrared spectroscopy (fNIRS) data simultaneously to examine the neurophysiological correlates of subjects' fatigue levels. Liu et al. [36] combined EEG, fNIRS, and physiological measures for workload classification, showcasing improved performance when fusing these modalities. Han et al. [10] developed a multimodal neural network architecture comprising CNN and LSTM models to identify distraction, workload, fatigue, and normal mental states. Nevertheless, to the best of researchers' knowledge, no current study combines multimodal biosignal datasets with the 1D-CNN approach.

C. Mental States Detection in the Context of AHPLS

Prior research has delved into the detection and evaluation of AHPLS. For instance, Harrivel et al. [37] employed RF, XGBoost, and Deep Neural Network (DNN) classifiers in a sophisticated flight simulator environment to predict CA, DA, and low workload states using various sensing modalities. In subsequent research, Harrivel et al. [38] utilized RF, gradient boosting, and two SVM classifiers to discern CA and SS states. Terwilliger et al. [39] aggregated CA, DA, and SS mental states into an "event" category and introduced a convolutional autoencoder method to differentiate the event class from the normal state (NE). In earlier investigations, the impact of two preprocessing techniques on SVM and Artificial Neural Network (ANN) models using EEG data from a pilot exposed to CA, DA, SS, and NE states was examined [40]. However, there were certain limitations to these studies: 1) the performance was not optimal, and 2) no study conducted a multiclass classification categorizing CA, DA, SS, and NE. Notably, the curse of dimensionality restricted the accuracy of predicting DA and SS states, despite the potential of merging data from two distinct scenarios.

D. Addressing Data Imbalance Issue

A critical challenge in mental state detection using biosignals is the data imbalance issue, where certain mental states may be underrepresented in the dataset [41,42]. This issue can lead to biased model predictions and poor generalization to real-world scenarios [43]. However, previous studies have not adequately addressed this problem in the context of mental state detection using DL models, traditional ML techniques, and multimodal biosignals [44,45].

The novelty of the proposed research lies in its application of a 1D-CNN+LSTM architecture to predict pilots' mental states (CA, DA, SS, and NE) using EEG signals and non-brain signals, such as ECG, GSR, and R. Furthermore, this study addresses the data imbalance issue by employing resampling strategies, including downsampling using the CS method and oversampling using the SMOTE method. To the best of our knowledge, no previous study has combined these specific mental states, DL architecture, multimodal biosignals, and data balancing techniques with traditional ML methods to predict pilots' mental states, making this research a unique contribution to the field.

III. Materials and Methods

A. AHPLS Dataset

The AHPLS dataset, collected by Harrivel et al. [37], is publicly released on the NASA open portal website. It comprises psychophysiological data gathered from 18 pilots during various scenario events designed to induce CA, DA, SS, and NE states. These data were recorded using the Advanced Brain Monitoring X24 EEG and Mind Media B.V. Nexus Mark II systems. For each pilot, four sets of data, including EEG, ECG, GSR, and Respiration, were provided. Three of the four sets were collected in a non-flying environment, while the fourth set was obtained in a high-fidelity flight simulator, featuring approximately one hour of labeled benchmark data. This set consists of 25 columns, which include a time stamp, 20 EEG channels, an ECG channel, an R channel, a GSR channel, and an event label.

In this research, the fourth set was utilized as it was collected in a flight simulator and contains labeled benchmark data that induced the states of interest (NE, SS, CA, and DA). The NE, SS, CA, and DA states are annotated as Class 0, Class 1, Class 2, and Class 3, respectively. The dataset exhibits significant class imbalance; for each pilot, Class 0 constitutes approximately 83% of the data, followed by Class 2 at about 14%, Class 3 at around 2%, and Class 1 comprising only 1% of the data.

B. Signal Preprocessing

The EEG, ECG, GSR, and R signals were preprocessed using open-source libraries, specifically MNE-Python [46] and BioSSPy [47]. MNE-Python was employed to implement advanced preprocessing techniques for cleaning artifacts from the EEG data, ensuring the highest quality signal for subsequent analysis. Initially, the dataset was transformed into a compatible format to facilitate the use of MNE-Python functions [40]. For the EEG signal, an automated preprocessing pipeline was employed to identify and eliminate artifacts [48], ensuring the data's integrity and reliability. In parallel, the ECG, GSR, and R signals were filtered using BioSSPy, a specialized library for biosignal processing. With the aid of BioSSPy, one distinctive feature was extracted from each of these channels, providing a comprehensive representation of the physiological data. The combination of MNE-Python and BioSSPy allowed for effective preprocessing and feature extraction, setting the foundation for accurate and reliable analysis of the pilots' psychophysiological states.

C. Features Extraction

For the EEG signals, the Power Spectral Density (PSD) features were extracted using Welch's method [49], a widely recognized technique for spectral estimation. Welch's method employs the Fast Fourier Transform (FFT) algorithm to estimate power spectra, providing an accurate representation of the signals' frequency domain characteristics. The parameters utilized for extracting the PSD values using the MNE-Python library are summarized in Table 1.

Table 1 Parameters utilized for PSD values extraction.

Parameters	Description	Value
sfreq	The sampling frequency	256
fmin	The lower frequency of interest.	1
fmax	The upper frequency of interest	50
n_fft	The length of FFT used	1280
n_overlap	The number of points of overlap between segments	255
n_per_seg	Length of each Welch segment	1280
window	Windowing function to use	boxcar

The EEG signals were recorded at a sampling frequency rate of 256 Hz. Consequently, the 'sfreq' parameter was set to 256, matching the sampling frequency rate. The 'fmin' and 'fmax' parameters were set to 1 and 50, respectively, generating 50 periodograms (i.e., features) for each channel within each epoch. These parameters define the range of periodograms and yield an equal number of PSD values for each epoch. The default length of FFT and the Welch segment is 256, equivalent to 1 second. Both the length of FFT and the Welch segment can adopt values that are multiples of the sampling frequency. In this study, the length of FFT and the Welch segment was set to 1280, corresponding to 5 seconds with an overlap of one second. The key equations associated with Welch's method [50] are outlined below:

Let $x_m(n) \triangleq w(n)x(n+mR)$ represent the m^{th} windowed segment of the signal, where $n = 0,1,\dots,M-1$ and $m = 0,1,\dots,K-1$. R denotes the window hop size, and K indicates the total number of segments. The periodogram of the m^{th} block is calculated as:

$$P_{x_m,M}(\omega_k) = \frac{1}{M}\left|FFT_{N,k}(x_m)\right|^2 \triangleq \frac{1}{M}\left|\sum_{n=0}^{N-1} x_m(n)\, e^{-\frac{j2\pi nk}{N}}\right|^2 \quad (1)$$

The Welch estimate of the power spectral density is given by the average of periodograms across all segments:

$$S_x^W \triangleq \frac{1}{K}\sum_{m=0}^{K-1} P_{x_m,M}(\omega_k). \quad (2)$$

This method computes an average of periodograms derived from non-overlapping successive blocks of data when $w(n)$ is a rectangular window.

Welch's method produces 50 features for each channel, totaling 1000 for each epoch. Fig. 1 illustrates the Welch's periodogram for a single epoch and channel.

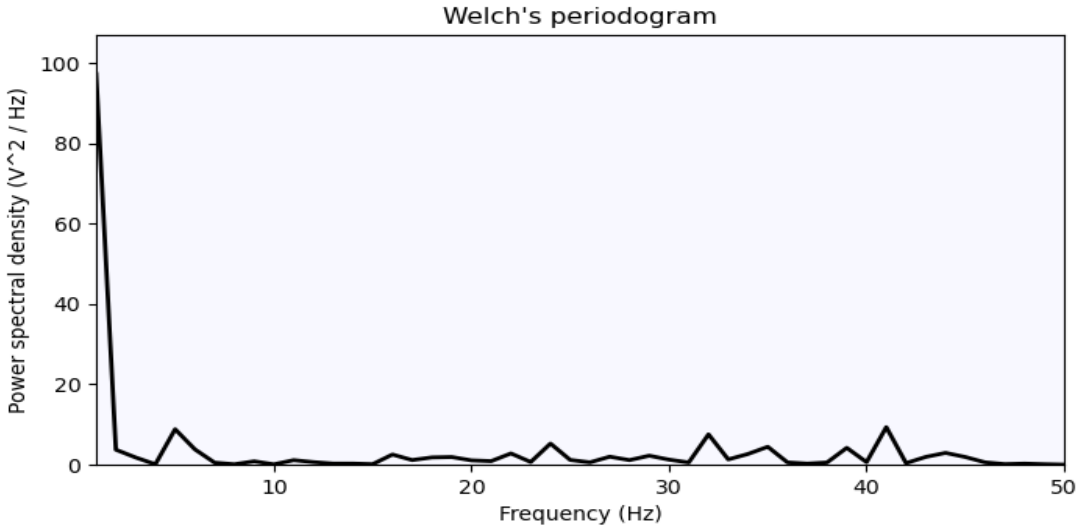

Fig. 1 Welch's periodogram for a single epoch and channel

To reduce the dimensionality of the feature space from 1000 to 100 features, the absolute PSD values for five distinct frequency bands were computed. These frequency bands include delta (0-4 Hz), theta (4-8 Hz), alpha (8-13 Hz), beta (13-20 Hz), and gamma (20-50 Hz). For instance, to extract the PSD values intersecting the delta band, the 'logical AND' operation from the NumPy library was employed. As illustrated in Fig. 2, the absolute PSD values for the delta band were determined using the Area Under Curve (AUC) method. Due to the curve's indefinite shape, the Composite Simpson's Rule (CSR) was utilized to compute the AUC. The absolute PSD values for other bands were calculated similarly. The CSR operates on the principle of dividing the larger area into smaller parabolic segments and subsequently calculating the sum of the area under each parabola.

The total number of EEG features generated per epoch was 5 × 20. In addition to these 100 features, ECG, R, and GSR signals were incorporated into the dataset after filtering and feature extraction using the BioSSPy library,

contributing one feature per channel. Consequently, the total number of features generated per epoch amounted to 103.

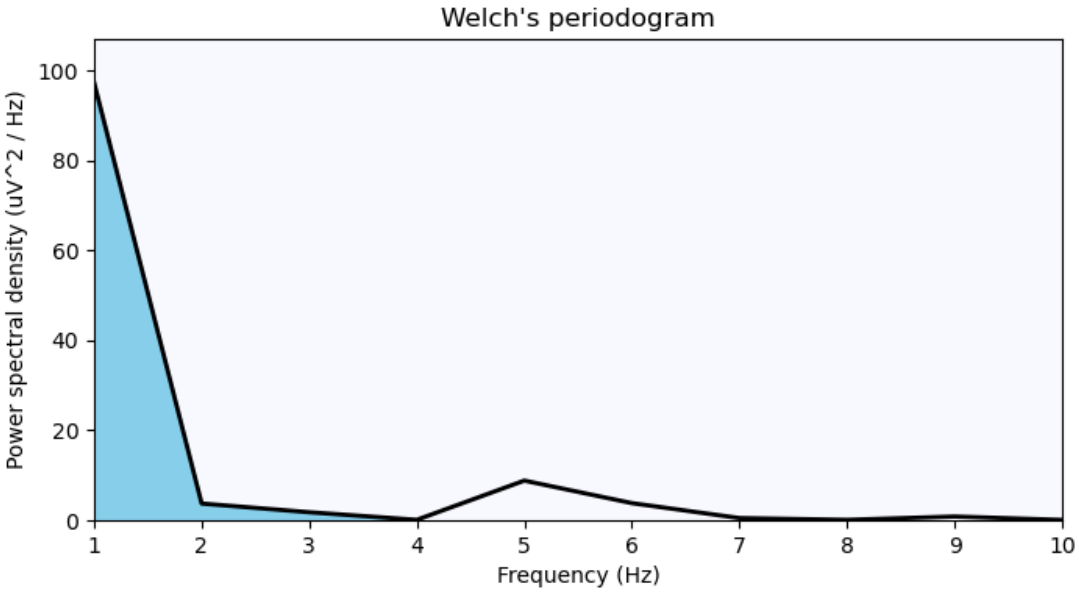

Fig. 2 Delta Band's Absolute PSD

D. Data Balancing

In this subsection, two resampling techniques, namely Cosine Similarity and Synthetic Minority Over-sampling Technique (SMOTE), are introduced and explained in detail. These techniques are employed in the study to address the data imbalance issue, which is a critical challenge in mental state detection using multimodal biosignals.

1. Cosine Similarity (CS)

The CS is a widely used similarity metric to measure the angular distance between two vectors in a multi-dimensional space, providing a value between -1 and 1. It is particularly effective in high-dimensional datasets, as it is less sensitive to the size of the vectors compared to Euclidean distance. The CS between two non-zero vectors A and B is calculated using the following formula:

$$Cosine\ Similarity(A,B) = \frac{(A.B)}{(\| A \| \| B \|)} \tag{3}$$

where A and B are two non-zero vectors, $A.B$ denotes the dot product of A and B, and $||A||$ and $||B||$ represent the magnitudes of the vectors A and B, respectively.

In this study, the CS method is utilized as a downsampling technique to identify and remove similar instances within the majority class (i.e., Class 0). By computing the similarity between instances in the majority class, the most representative samples can be retained, thus reducing the data imbalance and mitigating the impact of duplicate or highly similar instances on the model's performance.

2. Synthetic Minority Over-sampling Technique (SMOTE)

The SMOTE method is an oversampling approach that generates synthetic samples for the minority classes such as DA and SS to balance the class distribution. Unlike simple oversampling techniques that replicate minority class epochs, SMOTE generates synthetic epochs that lie along the line segments joining the minority class instances and their k-nearest neighbors in the feature space.

The process of generating synthetic epochs using SMOTE involves identifying the k-nearest neighbors in the feature space for each epoch in the minority classes. Then, select a random epoch from the minority class and one of its k-nearest neighbors. Finally, generate a synthetic epoch by interpolating between the chosen epoch and its neighbor using the following formula:

$$Synthetic\ Epoch\ =\ Epoch\ +\ \lambda\ *\ (Neighbor\ -\ Epoch) \tag{4}$$

where $Epoch$ is the randomly selected epoch from the minority class, $Neighbor$ is one of its k-nearest neighbors, and λ is a random number between 0 and 1.

In this study, the SMOTE method is applied to generate synthetic samples for the underrepresented mental states (i.e., CA, DA, and SS) in the dataset. By employing both the CS and SMOTE methods, the methodology effectively addresses the data imbalance issue in the dataset, which is essential for improving the performance and generalization of all the adopted models for predicting pilots' mental states.

E. Classification Methods

In the present study, following the completion of data cleaning, feature extraction, and data balancing, four DL models and three ensemble learning models were employed to perform a multiclass classification task. The DL models included LSTM, 1D-CNN, a combined 1D-CNN and LSTM architecture, and FFNN, while the ensemble learning models encompassed AdaBoost, XGBoost, and RF. These models were trained on a dataset composed of combined pilot data. In comparison to other algorithms, such as Logistic Regression and SVM, the ensemble learning algorithms demonstrated superior performance due to their ability to derive hyper-rectangles in the feature space.

For ensemble learning models, the hyperparameters tuning was performed using the *GridSearchCV* function from the scikit-learn library. This function takes a dictionary of hyperparameters and their values as input and constructs a grid of all possible combinations of hyperparameters using the k-fold cross-validation method. In this study, the learning rate, sub-sample, algorithm, bootstrap, n_estimators, and max_depth hyperparameters were fine-tuned using a 3-fold cross-validation method. To fine-tune the DL models developed in this study, a trial-and-error approach was adopted. The hyperparameters of the FFNN, 1D-CNN, LSTM, and CNN+LSTM models that were fine-tuned include learning rate, batch size, and epochs. Fine-tuning these hyperparameters can lead to overfitting or underfitting, which in turn affects the performance of the DL models. Each of the aforementioned models is briefly described in the following subsections, providing an overview of their structure and function in the context of this multiclass classification task.

1. Adaptive Boosting (AdaBoost)

The AdaBoost algorithm is a powerful ensemble method that combines multiple weak classifiers, each trained on different subsets of the training data, to form a robust and accurate strong classifier. The primary objective of this approach is to create a more efficient classifier by capitalizing on the strengths of the individual weak classifiers while minimizing their weaknesses. Initially, the algorithm assigns equal weights to each sample in the training set. Subsequently, a weak classifier is trained on this training set, and its error rate is computed. Based on the error rate, the algorithm calculates the weight of the weak classifier and updates the weights of the samples in the training set. This iterative process continues for a predetermined number of iterations or until a specified threshold is achieved. Upon completion of the iterative process, the weak classifiers are combined by weighting their individual outputs based on their calculated weights, thus forming a strong classifier. The final prediction is made using this combined classifier, which is expected to exhibit improved performance compared to its constituent weak classifiers. By continuously updating the weights of the samples in the training set and retraining the weak classifiers, AdaBoost effectively focuses on the samples that are challenging to classify, thereby enhancing the overall performance of the final classifier. In the present study, several hyperparameters are optimized to achieve the best performance for the AdaBoost classifier. The learning rate, max depth, number of estimators, and loss function parameters are set to 0.6, 5, 200, and 'SAMME', respectively.

2. Extreme Gradient Boosting (XGBoost)

The XGBoost algorithm is a state-of-the-art ensemble learning method that iteratively trains a sequence of weak decision trees and combines their predictions to form a powerful and accurate model. It employs a gradient boosting framework, which involves fitting a model on the residual errors of the preceding iteration. In each iteration, the algorithm calculates the gradient of the loss function with respect to the predicted values, subsequently updating the weights of the decision trees to minimize the loss. XGBoost incorporates regularization techniques, such as L1 and L2 regularization, to prevent overfitting, ensuring a more robust model capable of generalizing well to unseen data. Additionally, it includes a feature selection method that evaluates the importance of each feature, contributing to a more efficient and interpretable model. By integrating these techniques, XGBoost produces highly accurate models that can effectively handle complex datasets with numerous features, making it a popular choice for various ML tasks and applications. In the current study, several hyperparameters are fine-tuned to achieve optimal performance for the XGBoost classifier. The learning rate, max depth, number of estimators, and subsample parameters are set to 0.6, 2, 200, and 0.9, respectively.

3. Random Forest (RF)

RF is an ensemble learning technique that constructs multiple decision trees on distinct subsets of the training data, subsequently integrating their predictions to form a robust model. Each decision tree within the forest is trained on a

unique subset of the training data, and at each node, a random subset of features is chosen for splitting. This strategy serves to mitigate overfitting and enhances the model's generalization capabilities. During the prediction phase, each decision tree in the forest independently forecasts the outcome. The final prediction is then determined by aggregating the individual predictions, typically through a majority vote mechanism. This methodology yields highly accurate models capable of managing intricate datasets characterized by a multitude of features. Additionally, it allows for the assessment of the relative importance of each feature within the dataset. In the present study, the hyperparameters for the RF model are configured as follows: the maximum depth is set to 5, limiting the extent of tree growth and complexity; the number of trees is established at 600, providing a large enough ensemble to capture diverse patterns in the data; and the bootstrap parameter is set to True, enabling the usage of bootstrapped samples for training each individual tree.

4. Feed-Forward Neural Network (FFNN)

FFNN is a type of multi-layer ANN wherein the information flow proceeds unidirectionally, transitioning from the input layer through one or more hidden layers, ultimately reaching the output layer. Each neuron in the network receives a weighted sum of inputs from the preceding layer, applies an activation function to this sum, and conveys the outcome to the subsequent layer. Throughout the training process, the weights and biases of the neurons are adjusted using an optimization algorithm to minimize the discrepancy between predicted and actual outputs. Activation functions can be either linear or nonlinear, and the number of layers and neurons in the network can be fine-tuned to enhance performance. FFNNs are particularly well-suited for complex problems involving large datasets, as they can learn to extract meaningful features from input data. In the present study, hyperparameters such as learning rate, batch size, and epochs are set to 0.0001, 32, and 150, respectively. The FFNN architecture is configured with 103 perceptron units in the input layer, 50 in the hidden layer, and 4 in the output layer as shown in Fig. 3. The Rectified Linear Unit (ReLU) activation function is employed for both the input and hidden layers, while the Softmax activation function is utilized in the output layer to provide class probabilities.

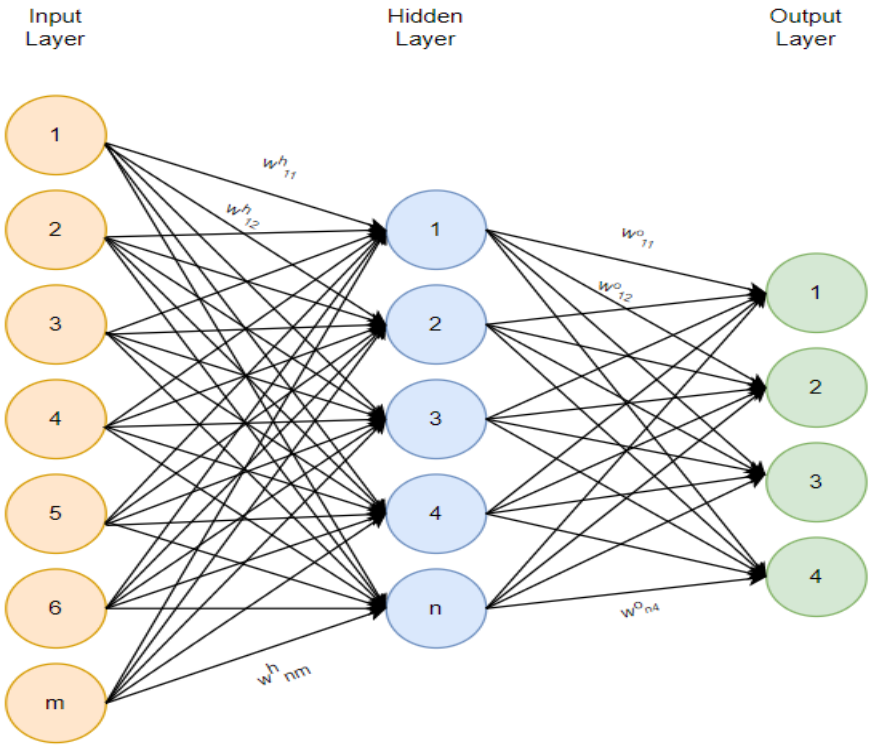

Fig. 3 The FFNN architecture

5. One-Dimensional Convolution Neural Network (1D-CNN)

The 1D-CNN is a specialized type of neural network designed for processing time-series data. The architecture typically consists of one or more convolutional layers, followed by one or more fully connected layers. Convolutional layers apply a set of filters to the input data to extract relevant features, such as changes in frequency or amplitude over time. During the training process, the filter weights are adjusted to minimize the difference between the predicted

and actual output. The fully connected layers then combine the features extracted by the convolutional layers to make a final prediction. 1D-CNNs are especially useful for detecting patterns in sequential data and can accommodate data with variable lengths. Fig. 4 depicts each convolutional stage as a collection of learnable convolutional filters.

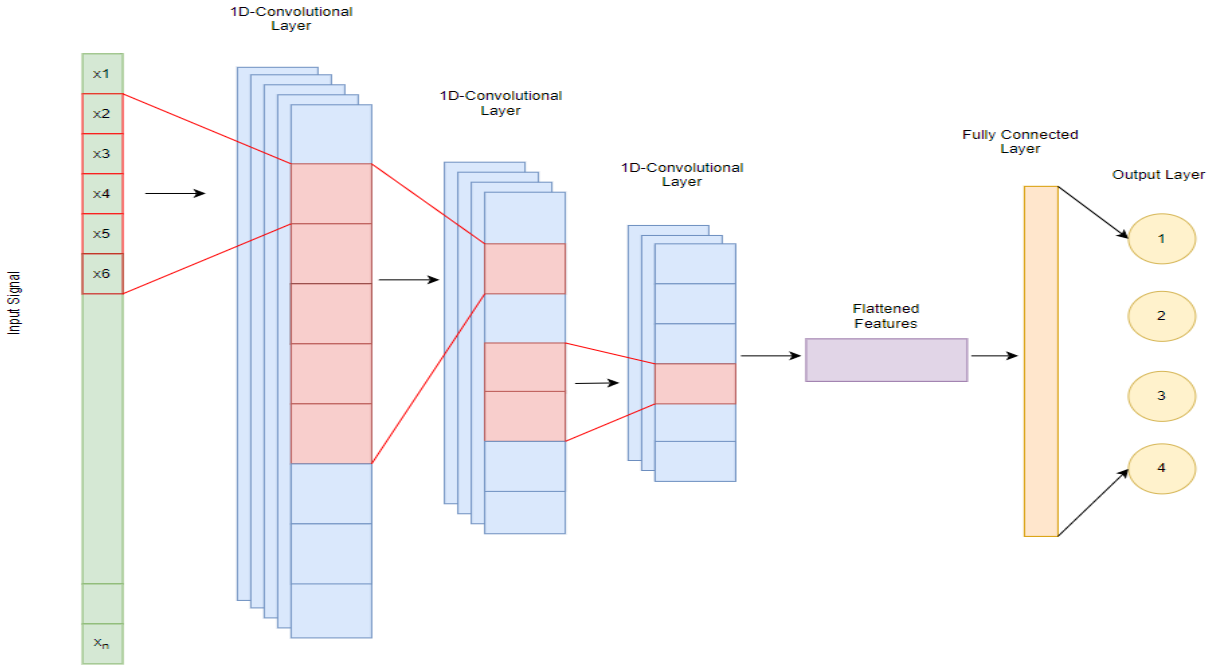

Fig. 4 One-Dimensional Convolution Neural Network

A set of input signals $x2, x3, x4, x5\ and\ x6$, corresponding to the filter size, is chosen for the application of convolution. This process involves the utilization of convolutional filters, which are assigned specific weights. These filters are designed to extract high-level features from a given input signal by applying ReLU activation function. Given that there are xn features in the input signal, the output features of the first layer can be calculated using the following formula, taking into account the filter size (k) and stride (s):

$$Output\ size = \frac{input\ features - k}{s} + 1 \tag{5}$$

The outputs generated by the first layer are subsequently fed into a second convolution layer. This layer extracts features for the subsequent layer using the same formula as before. This iterative process reduces the spatial scale of the features extracted by the convolutional filters, while simultaneously emphasizing the salient features learned by each filter. The output of the second layer is then passed through a third convolution layer to generate the final convolution output features. As the input signals progress through the convolutional layers, the network becomes increasingly adept at learning problem-specific characteristics. Upon reaching the final stage, the extracted features are flattened and passed through a densely connected hidden layer. This layer is connected to an output layer consisting of four nodes, which ultimately yields the final output.

In this study, the 1D-CNN architecture is configured with three convolutional layers, each followed by a dropout of 0.5 to prevent overfitting. The first convolutional layer consists of 128 filters, each with a kernel size of 5 and a ReLU activation function. The second convolutional layer comprises 64 filters, each with a kernel size of 5 and a ReLU activation function, followed by a dropout of 0.5. The third and final convolutional layer has 32 filters, each with a kernel size of 2 and a ReLU activation function, followed by a dropout of 0.5. After the convolutional layers, the features are flattened and passed through a fully connected layer with 128 nodes. The output layer consists of 4 nodes, corresponding to the four classes, and employs the Softmax activation function to yield class probabilities. The learning rate, batch size, and epochs hyperparameters are set to 0.0001, 32, and 150, respectively.

6. Long Short Term Memory Network (LSTM)

LSTM networks are designed to address the vanishing gradient problem and process sequential data with long-term dependencies. They use a memory cell that can store information over extended periods, a set of input, output, and forget gates to control the flow of information, and a set of cell state transformations to manipulate the stored information. The input gate controls the addition of new information to the memory cell, the forget gate determines

the discarding of old information, and the output gate controls the information exposure to the subsequent layer. During training, backpropagation through time adjusts the weights of the gates and transformations to minimize the difference between the predicted and actual output.

To train the data using LSTM, the sequence of input data $x1, x2, x3, \ldots\ldots, xm$ is fed to the input gate of the LSTM layer in the network, as illustrated in Fig. 5. The features generated by a single LSTM cell $(a1)$ are stored in the cell memory and then passed to the next cell $(a2, a3, \ldots\ldots, am)$. The output of each cell is computed using the features passed on by the previous cell, and each cell provides the output through the output gate. The outputs provided by the output gate of each cell $(h1, h2, h3, \ldots, hm)$ are then multiplied by the weights of each cell. This stored data in memory is then used to derive new features or to observe the pattern in time series. After the LSTM layer, the data proceeds to the hidden layer by multiplying the weights of the hidden layer $(w_{11}^{h}, w_{12}^{h}, \ldots\ldots, w_{nm}^{h})$ with the output of each cell. The weighted output values of each cell are then combined to obtain a sum on each node of the hidden layer.

The LSTM network uses the same architecture and number of nodes as the previously discussed FFNN model. Instead of using the dense layer for input, the LSTM layer is employed for the LSTM model. The LSTM model offers an advantage over the FFNN because the biosignal data is in time-series format, allowing it to generalize results more effectively. In this study, the learning rate, batch size, and epochs hyperparameters are set to 0.0001, 32, and 150, respectively. Additionally, the LSTM architecture is configured by setting the LSTM layer, hidden layer, and output layer to 103, 50, and 4, respectively.

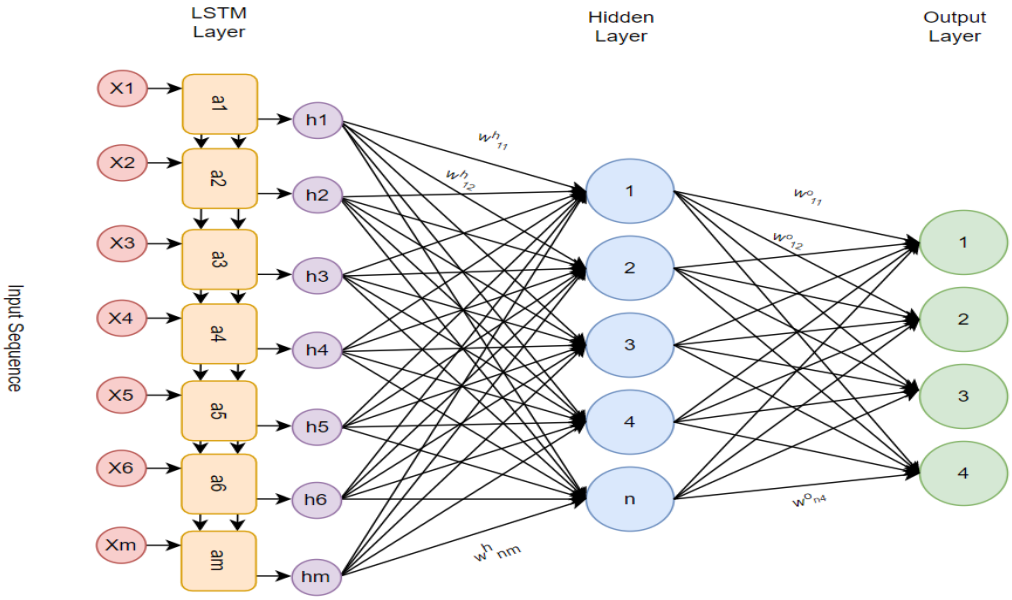

Fig. 5 The LSTM Neural Network

7. The 1D-CNN+LSTM Architecture

This study presents a custom network that combines 1D-CNN and LSTM architectures. The 1D-CNN performs well with graphical and sparse data, while the LSTM demonstrates superior performance with time-series data. Consequently, EEG data features are employed to train the 1D-CNN network, while the remaining three features (ECG, R, and GSR) are used to train the LSTM network.

The input sequences $(x1, x2, x3, \ldots\ldots, xn)$ are fed to the 1D-CNN part of the model, as shown in Fig. 6, and processed in a manner similar to that described in Subsection 5. The remaining features $(x_{n+1}, x_{n+2}, \text{and } x_{n+3})$ are input to the LSTM portion of the model and processed as detailed in Subsection 6. The output features generated by both models are then concatenated and passed through a fully connected layer before reaching the output layer, consisting of 4 nodes to classify the four classes with the 'softmax' activation function.

For the 1D-CNN+LSTM model, the 1D-CNN architecture only processes the EEG features, as it has demonstrated better performance with EEG data. The LSTM part of the model handles only three features: separate time series data for ECG, Respiration, and GSR features, respectively.

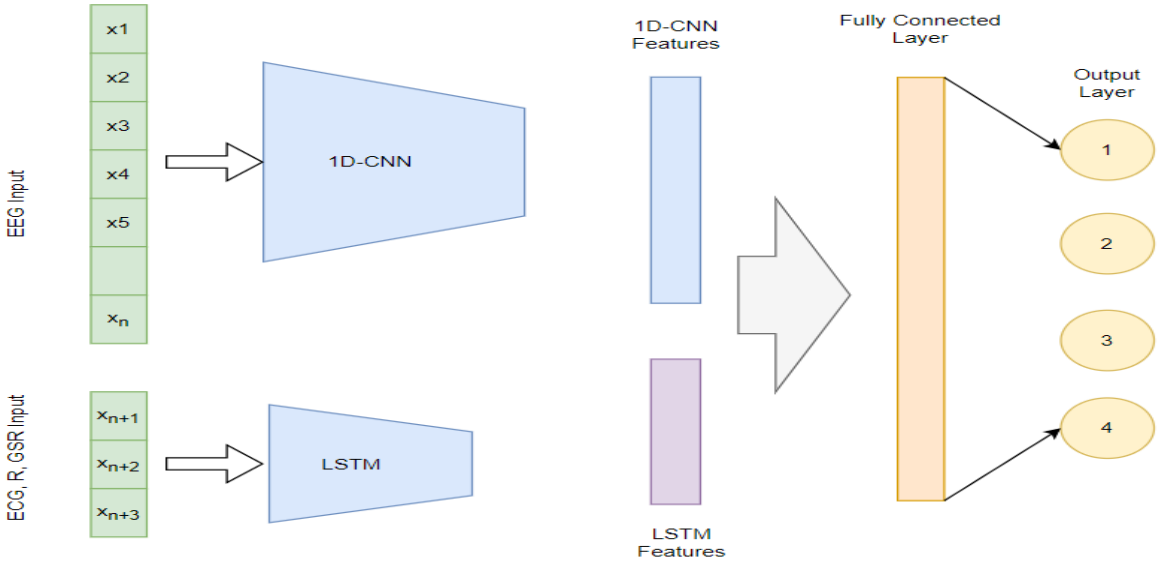

Fig. 6 Overview of the proposed 1D-CNN+LSTM architecture

F. Evaluation Metrics

The confusion matrix, sometimes referred to as the error matrix or contingency table, is vital for assessing the overall performance of the proposed models. It consists of four key elements in multiclass classification tasks: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). TP represents instances accurately classified as positive for a specific class, while FP indicates instances wrongly classified as positive but belonging to another class. FN refers to instances incorrectly classified as negative for a specific class when they belong to it, and TN denotes instances accurately classified as negative for a specific class. These elements are employed to calculate evaluation metrics such as precision, recall, and F1-score for each class, aiding in the assessment of a multiclass classification model's performance. The proposed model's classification performance on the testing set was evaluated using four confusion matrix-based metrics: accuracy, precision, recall, and F1-score. These performance measures are defined as follows:

- **Accuracy:** This metric measures the ratio of correctly classified instances to the total number of instances, essentially quantifying how many instances in the dataset are accurately classified by the model.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

- Precision: This metric gauges the ratio of true positive predictions to the total number of positive predictions, essentially determining how many instances classified as positive are genuinely positive.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

- Recall: This measure calculates the proportion of true positive predictions out of the total number of actual positive instances, indicating how many of the real positive instances are accurately identified as positive by the model.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

- F1-score: Serving as a balance between precision and recall, the F1-score is the harmonic mean of these two metrics and offers a single score that reflects the model's overall performance.

$$F1 - score = 2 * (Precision * Recall)/(Precision + Recall) \tag{9}$$

IV. Results and Discussion

In this section, the results of the proposed multimodal DL architecture are presented, and its performance is evaluated in comparison to various ensemble learning and DL models. Furthermore, the effectiveness of integrating CS with SMOTE to address data imbalance issues in the dataset is assessed.

The results and discussion are organized into three subsections. In subsection A, the performance outcomes of the proposed architecture alongside other ensemble learning and deep learning models, both before and after incorporating CS, are presented. This comparison will provide a comprehensive understanding of each model's strengths and weaknesses.

Subsection B focuses on evaluating the training and validation performance of the deep learning models, considering the impact of the sampling techniques on their performance. The convergence and generalization capabilities of these models before and after the utilization of CS are discussed.

In subsection C, the overall impact of CS on the performance of all the trained models is assessed. The influence of the combined approach of SMOTE and CS on the models' performances are analyzed using the confusion matrix. This will provide a deeper understanding of the benefits and potential limitations of using this combined sampling technique.

A. Performance Comparison of Models with and without the CS Method

In this subsection, the classification performance of ensemble and deep learning models is evaluated, using features extracted from EEG, ECG, respiration, and GSR signals. Welch's method and FFT were employed to generate 100 EEG features per channel, which were subsequently reduced to five features per channel using absolute PSD. The resulting combined dataset consisted of 32,867 epochs, each containing 103 features.

For model evaluation, the dataset was divided into 80% for training and 20% for testing for ensemble learning models (i.e., XGBoost, AdaBoost, and RF). Meanwhile, the DL models (i.e., FFNN, 1D-CNN, and CNN+LSTM) used a 60% training, 20% validation, and 20% testing split. SMOTE was applied to address class imbalance in the training data for all models. The performance results of these models, evaluated using the unseen testing dataset, are presented in Table 2. The testing dataset comprised 6,574 epochs, including 5,913 epochs of the NE class, 43 epochs of the SS class, 538 epochs of the CA class, and 80 epochs of the DA class.

Table 2 Classification performance of the pilots' mental states using SMOTE method.

Model	Mental Class	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Support
XGBoost	NE		95.87	98.95	97.38	5913
	SS		100	6.97	13.04	43
	CA		83.84	71.37	77.10	538
	DA		40	5	8.88	80
	Macro Avg		79.92	45.57	49.10	6574
	Weighted Avg	94.94	94.23	94.94	94.09	6574
AdaBoost	NE		95.16	96.53	95.84	5913
	SS		50	9.30	15.68	43
	CA		64.33	63.38	63.85	538
	DA		13.15	6.25	8.47	80
	Macro Avg		55.66	43.86	45.96	6574
	Weighted Avg	92.15	91.34	92.15	91.63	6574
RF	NE		92.28	67.39	77.90	5913
	SS		2.11	51.16	4.05	43
	CA		32.47	23.42	27.21	538
	DA		24.18	25	4.41	80
	Macro Avg		32.32	41.74	28.39	6574
	Weighted Avg	63.17	85.70	63.17	72.37	6574
FFNN	NE		92.46	85.74	88.97	5913
	SS		11.11	4.65	6.55	43
	CA		20.33	36.43	26.09	538
	DA		0.91	1.25	1.05	80
	Macro Avg		31.20	32.01	30.67	6574

	Weighted Avg	80.14	84.91	80.14	82.22	6574
1D-CNN	NE		92.61	79.90	85.79	5913
	SS		2.59	4.65	3.33	43
	CA		17.42	42.19	24.66	538
	DA		4.34	5	4.65	80
	Macro Avg		29.24	32.93	29.60	6574
	Weighted Avg	75.41	84.79	75.41	79.26	6574
LSTM	NE		91.58	88.11	89.81	5913
	SS		1.38	2.32	1.73	43
	CA		19.64	24.90	21.96	538
	DA		3.81	6.25	4.73	80
	Macro Avg		29.10	30.39	29.56	6574
	Weighted Avg	81.38	84.03	81.38	82.64	6574
1D-CNN+LSTM	NE		91.79	92.84	92.31	5913
	SS		6.25	2.32	3.38	43
	CA		27.49	27.13	27.31	538
	DA		2.17	1.25	1.58	80
	Macro Avg		31.92	30.88	31.15	6574
	Weighted Avg	85.76	84.87	85.76	85.31	6574

As illustrated in Table 2, the XGBoost algorithm displayed the best performance among the evaluated models, followed by AdaBoost, 1D-CNN+LSTM, LSTM, FFNN, 1D-CNN, and RF. Both XGBoost and AdaBoost achieved high mean accuracies of 94.94% and 92.15%, respectively, while the RF model lagged behind with a mean accuracy of 63.17%. These results indicate that while ensemble methods employ multiple weak learners to create a more powerful model, they rely on distinct mechanisms and configurations, which lead to differences in performance. Regarding DL models, all of them demonstrated strong performance. The proposed 1D-CNN+LSTM model achieved the highest mean accuracy of 85.76%. Although it was outperformed by XGBoost and AdaBoost, the incorporation of 1D-CNN in this domain is a contribution. The 1D-CNN has proven effective in other areas such as speech recognition and provides the advantage of computational efficiency.

While the SMOTE method was employed to balance the dataset, the majority of the trained models struggled to accurately detect the SS, CA, and DA classes. A closer examination of the precision, recall, and F1-score metrics for the NE class reveals that the models demonstrated exceptional detection performance for this class. However, performance for the other classes was considerably lower, as evidenced by the macro average values. Among the remaining classes, the CA class exhibited the second-best detection performance. This observation suggests that if the dataset were not as imbalanced, the models may have achieved better overall performance across all classes.

To investigate this hypothesis, cosine similarity was applied to the NE epochs. The CS method aims to reduce the number of epochs with high similarity between the rows of the dataset containing the CA and NE classes. This process resulted in a reduction of NE epochs from 29,561 to 6,327, leaving the dataset with a total of 9,633 epochs. The modified dataset was then divided into 80% training and 20% testing for the ensemble learning models (i.e., XGBoost, AdaBoost, and Random Forest), and 60% training, 20% validation, and 20% testing for the deep learning models (i.e., FFNN, 1D-CNN, and CNN+LSTM). Table 3 presents the performance results of these models, evaluated using the unseen testing dataset.

It is important to note that the updated testing dataset consists of 1,927 epochs, including 1,266 NE class epochs, 43 SS class epochs, 538 CA class epochs, and 80 DA class epochs. This modified dataset allows for a more balanced evaluation of the models' performance across all classes.

Table 3 Classification performance of the pilots' mental states using SMOTE and CS methods.

Model	Mental Class	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Support
XGBoost	NE		94.48	97.47	95.95	1266
	SS		77.41	55.81	64.86	43
	CA		87.34	91.07	89.17	538
	DA		65.51	23.75	34.86	80
	Macro Avg		81.19	67.02	71.21	1927
	Weighted Avg	91.69	90.90	91.69	90.08	1927
AdaBoost	NE		92.42	94.47	93.43	1266
	SS		90.90	46.51	61.53	43

	CA		80.31	84.94	82.56	538
	DA		35.71	18.75	24.59	80
	Macro Avg		74.84	61.16	65.53	1927
	Weighted Avg	87.59	86.65	87.59	86.83	1927
RF	NE		85.57	91.86	88.60	1266
	SS		14.13	60.46	22.90	43
	CA		89.95	39.96	55.34	538
	DA		17.24	31.25	22.22	80
	Macro Avg		51.17	55.88	47.27	1927
	Weighted Avg	74.15	82.36	74.15	75.09	1927
FFNN	NE		86.92	92.41	89.58	1266
	SS		26.92	16.27	20.28	43
	CA		70.75	66.54	68.58	538
	DA		18.36	11.25	13.95	80
	Macro Avg		50.74	46.62	48.10	1927
	Weighted Avg	80.12	78.22	80.12	79.03	1927
1D-CNN	NE		84.75	91.31	87.90	1266
	SS		14.54	18.60	16.32	43
	CA		70.72	61.52	65.80	538
	DA		20	10	13.33	80
	Macro Avg		47.50	45.36	45.84	1927
	Weighted Avg	77.99	76.58	77.99	77.04	1927
LSTM	NE		85.99	92.65	89.20	1266
	SS		30	6.97	11.32	43
	CA		70.05	69.14	69.59	538
	DA		13.63	3.75	5.88	80
	Macro Avg		49.92	43.13	44	1927
	Weighted Avg	80.48	77.29	80.48	78.53	1927
1D-CNN+LSTM	NE		83.06	92.57	87.56	1266
	SS		26.08	13.95	18.18	43
	CA		70.10	59.29	64.24	538
	DA		13.15	6.25	8.47	80
	Macro Avg		48.10	43.01	44.61	1927
	Weighted Avg	77.94	75.27	77.94	76.22	1927

Using the same hyperparameters and configuration settings, the performance of the ensemble and deep learning models trained on the new dataset is displayed in Table 3. Once again, the XGBoost algorithm achieved the highest performance, followed by AdaBoost, LSTM, FFNN, 1D-CNN, CNN+LSTM, and RF. These results indicate that XGBoost is particularly suitable for this specific task, outperforming the other models. Interestingly, the proposed 1D-CNN+LSTM model did not perform as well on the new dataset as it did on the original dataset. This can be attributed to the fact that DL models typically perform better when trained with larger datasets.

The application of CS method considerably improved the detection performance for each mental state, as evidenced by the macro average values shown in Table 3. This improvement is further corroborated by examining the precision, recall, and F1-score of each model. These findings confirm the hypothesis that the skewed distribution of the dataset was one of the factors impacting the models' performance. Notably, employing CS to remove the NE epochs with similar row data as the CA class substantially enhanced the detection performance of the CA mental state, especially for the DL models.

It could be argued that the performance of the models trained on the original dataset, as displayed in Table 2, is superior to that of the models trained on the modified dataset shown in Table 3. However, this comparison is complicated by the different testing dataset sizes, as indicated in the support column. To accurately evaluate the performance of the models trained with the original dataset, these models were tested on a dataset identical to the testing dataset used for assessing the models trained on the modified dataset. Table 4 displays the classification performance of the models that were trained using the original dataset and evaluated with the updated testing dataset. This approach allows for a fair comparison between the models, accounting for differences in testing dataset sizes.

Table 4 Classification performance of the pilots' mental states using the updated testing dataset.

Model	Accuracy	Precision	Recall	F1 score
XGBoost	85.21	84.78	85.21	82.6
AdaBoost	81.94	80.84	81.94	79.37
RF	52.93	65.95	52.93	56.08
FNN	65.39	51.99	65.39	52.44
1D-CNN	64.45	61.72	64.45	62.89
LSTM	65.23	60.65	65.23	61.0
1D-CNN+LSTM	68.34	63.69	68.34	63.09

B. Training and Validation Analysis of DL Models

This study developed four distinct DL models to detect the AHPLS states. In addition to presenting the performance metrics of the DL models in Tables 2 and 3, the learning curves (i.e., accuracy and loss curves) for each model are also provided. Figure 7 (A), (B), (C), and (D) display the accuracy and loss curves of the FFNN, 1D-CNN, LSTM, and 1D-CNN+LSTM models, respectively, prior to the application of CS. In general, all the DL models demonstrated strong performance, as evidenced by the increasing training and validation accuracies and the decreasing training and validation losses as the models learned.

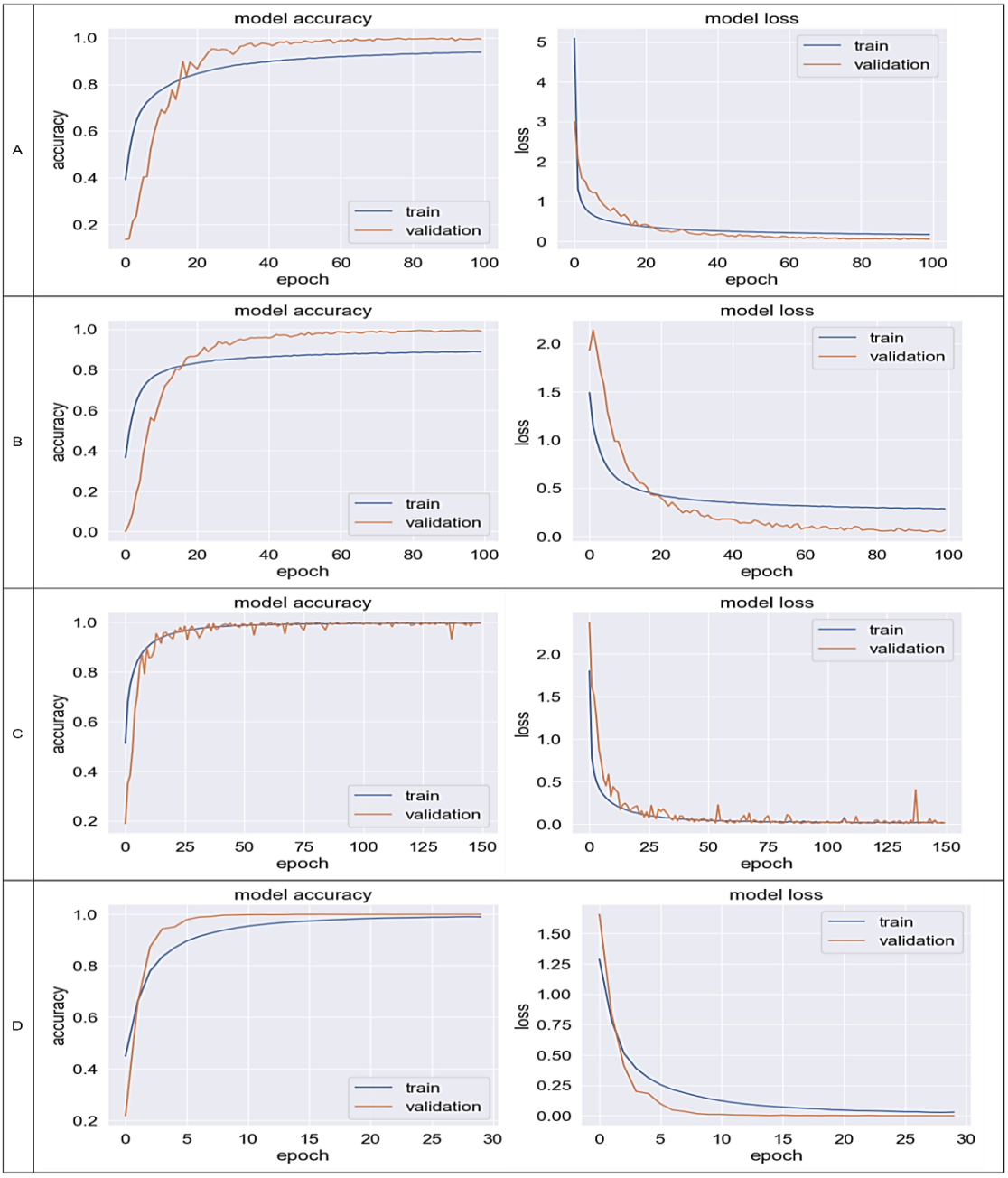

Fig. 7 The DL models' learning curves before incorporating the CS method.

Fig. 7 (A) shows that the validation accuracy and loss are slightly better than the training accuracy and loss, which typically indicates that the training data is somewhat more challenging to model than the validation data. This is likely due to the non-linearity of the dataset. However, this is not the case for the 1D-CNN and 1D-CNN+LSTM models shown in Fig. 7 (B) and (D), as dropout was used during their training. During the training process, a percentage of the features are set to zero, while all features are used during validation. This results in higher validation accuracy, suggesting that the model is more robust. Although the LSTM model's accuracy and loss curves in Fig. 7 (C) display negligible differences between training and validation, as the model is fully converged, the fluctuations in the validation data imply that the model is not generalizing well to the validation data. Consequently, among all the DL models, the proposed 1D-CNN+LSTM model is considered the best for the dataset prior to the incorporation of CS.

The DL models were also trained on the modified dataset after applying the CS method. Fig. 8 (A), (B), (C), and (D) depict the accuracy and loss curves of the FFNN, 1D-CNN, LSTM, and 1D-CNN+LSTM models, respectively,

after training them on the modified dataset. All the DL models demonstrated strong performance, as indicated by the increasing training and validation accuracies and the decreasing training and validation losses as the models learned.

Examining the FFNN model's loss curve in Fig. 8 (A), it is evident that it is a superb curve, as the training and validation losses initially correlated, then diverged slightly, and finally converged again. Similarly, the LSTM model shown in Fig. 8 (C) and the 1D-CNN+LSTM model shown in Fig. (D) displayed good loss curves, as the training and validation curves exhibit minor differences. The fluctuations in the validation data suggest that the models were not generalized enough to work on different data, such as the validation data. Regarding the 1D-CNN model depicted in Fig. 8 (B), the validation data appears unrepresentative compared to the training data; however, they begin to converge at the end. This implies that training the model on more epochs might yield better convergence. The reason behind this trend is likely the decrease in the number of training and validation samples compared to the old dataset, which justifies the observed behavior. It is crucial to highlight that the observed variations are statistical in nature rather than systematic. As a result, it could be argued that the proposed 1D-CNN+LSTM model demonstrates promising and strong performance for the dataset both before and after the application of CS.

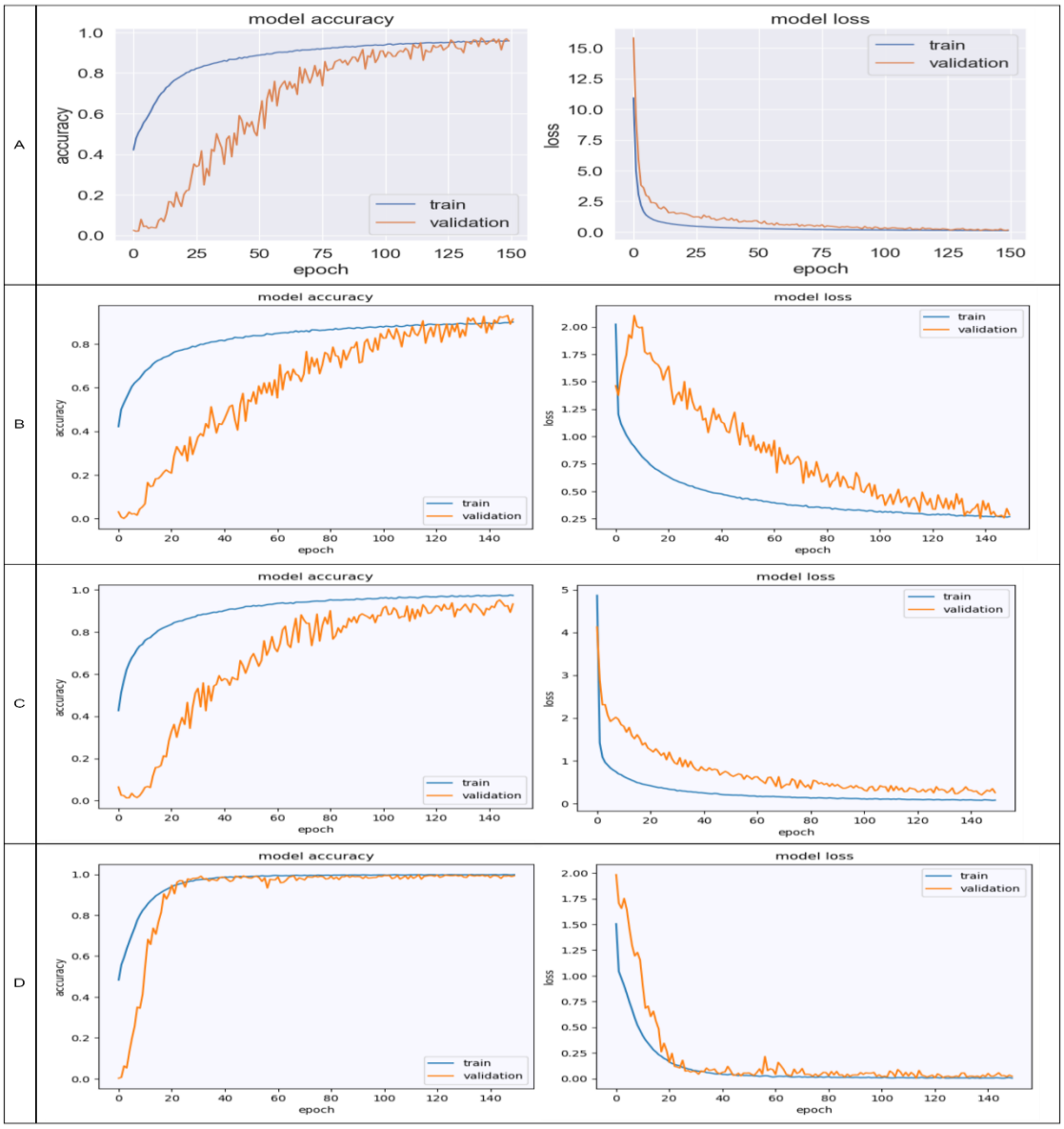

Fig. 8 The DL models' learning curves after incorporating the CS method.

C. Impact of Cosine Similarity on Model Performance: Confusion Matrix Analysis

To gauge the efficacy of the cosine similarity method, an identical testing dataset was utilized to measure the performance of the models trained both prior to and following the application of CS. The level of confusion produced by the models, before and after implementing the CS, was computed.

Fig. 9 displays the confusion matrices for the models before applying CS. Specifically, Fig. 9 (A) to (H) depict the confusion matrices for the XGBoost, AdaBoost, RF, FFNN, 1D-CNN, LSTM, and 1D-CNN+LSTM models, respectively. In contrast, Fig. 10 presents the confusion matrices for the models after incorporating CS, where Fig. 10 (A) to (H) illustrate the confusion matrices for the same models. The values of the diagonal elements in the matrices indicate the percentage of accurately predicted classes. This comparison allows for a more comprehensive understanding of the impact of CS on model performance.

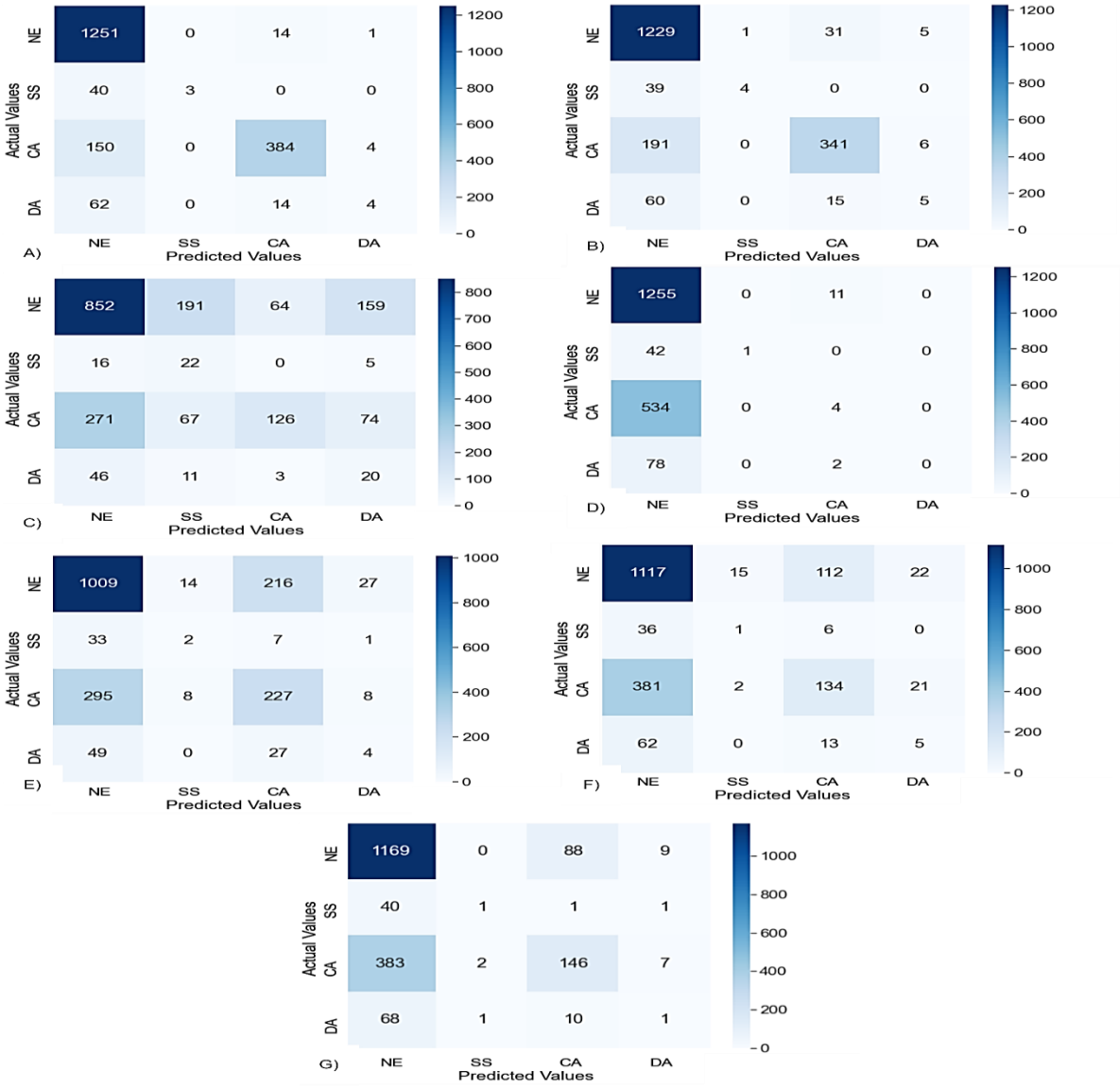

Fig. 9 Confusion matrices for the models before incorporating CS method.

Upon comparing the confusion matrices of the models before and after incorporating the CS method, as illustrated in Fig. 9 and 10, a significant improvement in model performance was observed. The NE and CA states were detected with relatively higher performance in almost all the models before employing the CS method, compared to the SS and DA states. After incorporating CS, more SS and DA samples were identified.

A few exceptions were noted, however. For the NE class, the XGBoost, AdaBoost, and FFNN models performed slightly better before using cosine similarity. This can be attributed to the reduction in the number of NE class samples, which affected the performance of these three models. Similarly, the LSTM model performed marginally better before applying cosine similarity, detecting 5 out of 80 samples correctly, while only 3 samples were correctly predicted after

using cosine similarity. This decrease in the identification of the DA state is hypothesized to be due to the configuration of the LSTM model.

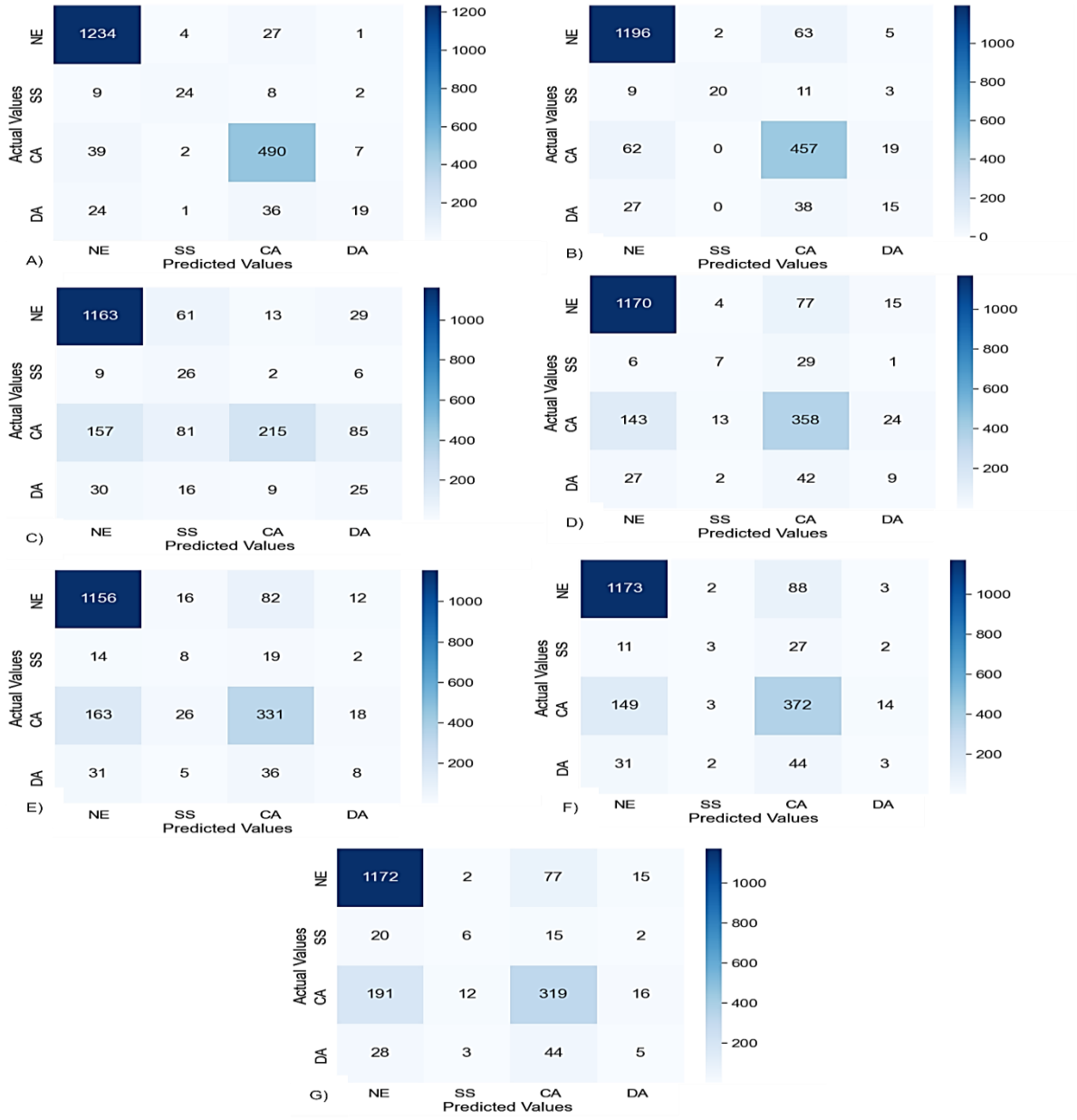

Fig. 10 Confusion matrices for the models after incorporating CS method.

In summary, this study presented a comprehensive evaluation of ensemble and DL models for detecting mental states using multimodal physiological signals. The performance of these models was assessed before and after the application of CS in conjunction with SMOTE to address the data imbalance issue. The results revealed that the XGBoost algorithm consistently outperformed other models, while the proposed 1D-CNN+LSTM model demonstrated considerable potential as a DL solution.

Upon analyzing the performance metrics, it was evident that the data imbalance issue had a significant impact on the models' ability to detect specific mental states. The implementation of CS led to a considerable improvement in the detection performance of each mental state, particularly for the CA class. This finding confirmed the hypothesis that data skewness was a major factor affecting the models' performance.

The learning curves of the DL models, both before and after the application of CS, displayed robust performance, with the proposed 1D-CNN+LSTM model deemed suitable for the given dataset. When comparing the confusion matrices before and after the use of CS, an overall improvement in model performance was observed. However, some

models, such as XGBoost, AdaBoost, FFNN, and LSTM, exhibited slightly better performance in detecting certain mental states before applying CS, which could be attributed to the reduction in the number of samples for specific classes.

The incorporation of CS and SMOTE proved to be effective in addressing data imbalance and improving the performance of the models in detecting mental states. Among the considered models, the XGBoost algorithm and the proposed 1D-CNN+LSTM model emerged as the most promising solutions for the given dataset.

V. Conclusion

This research has made significant strides in addressing the critical need for detecting pilots' mental states, particularly AHPLS, to enhance aviation safety and reduce the likelihood of accidents. By employing a multimodal approach that combines EEG with non-brain signals such as ECG, GSR, and respiration, the study has developed a robust DL architecture that effectively fuses 1D-CNN and LSTM models.

The research has also tackled the challenge of data imbalance, which is prevalent in real-world datasets and often results in biased models with poor detection performance for underrepresented mental states. By incorporating data resampling techniques, including downsampling using the CS method and oversampling using the SMOTE method, the study has successfully created more balanced datasets, which led to improved model performance.

As part of future work, further refinement of the models' performance will be carried out, and the training dataset will be enlarged to enhance the generalization capability of the models. Additionally, the possibility of extracting other meaningful features from the multimodal sensor data will be explored to further enhance the accuracy and robustness of the classification model.

Overall, this study highlights the potential of using multimodal sensor data and the proposed 1D-CNN+LSTM model for classifying pilots' mental states. The findings contribute to the growing body of literature on human factors in aviation and have implications for the development of real-time mental state monitoring systems for aviation safety applications.

References

[1] Pan, T., Wang, H., Si, H., Li, Y., and Shang, L. “Identification of Pilots’ Fatigue Status Based on Electrocardiogram Signals.” *Sensors*, Vol. 21, No. 9, 2021. https://doi.org/10.3390/s21093003.

[2] Oehling, J., and Barry, D. J. “Using Machine Learning Methods in Airline Flight Data Monitoring to Generate New Operational Safety Knowledge from Existing Data.” *Safety Science*, Vol. 114, No. May 2018, 2019, pp. 89–104. https://doi.org/10.1016/j.ssci.2018.12.018.

[3] International Air Transport Association. *2021 Safety Report Edition*. 2022.

[4] International Air Transport Association. *Loss of Control In-Flight Accident Analysis Report 2019 Edition*. 2019.

[5] Yen, J. R., Hsu, C. C., Yang, H., and Ho, H. “An Investigation of Fatigue Issues on Different Flight Operations.” *Journal of Air Transport Management*, Vol. 15, No. 5, 2009, pp. 236–240. https://doi.org/10.1016/j.jairtraman.2009.01.001.

[6] Hankins, T. C., and Wilson, G. F. “A Comparison of Heart Rate, Eye Activity, EEG and Subjective Measures of Pilot Mental Workload during Flight.” *Aviation, space, and environmental medicine*, Vol. 69, No. 4, 1998, pp. 360–7.

[7] Boksem, M. A. S., and Tops, M. “Mental Fatigue: Costs and Benefits.” *Brain Research Reviews*, Vol. 59, No. 1, 2008, pp. 125–139. https://doi.org/10.1016/j.brainresrev.2008.07.001.

[8] Commercial Aviation Safety Team. SE211: Airplane State Awareness - Training for Attention Management. http://www.skybrary.aero/index.php/SE211:_Airplane_State_Awareness_-_Training_for_Attention_Management_(R-D). Accessed Dec. 25, 2022.

[9] Khanna, A., Pascual-Leone, A., Michel, C. M., and Farzan, F. “Microstates in Resting-State EEG: Current Status and Future Directions.” *Neuroscience & Biobehavioral Reviews*, Vol. 49, 2015, pp. 105–113. https://doi.org/10.1016/j.neubiorev.2014.12.010.

[10] Han, S. Y., Kwak, N. S., Oh, T., and Lee, S. W. “Classification of Pilots’ Mental States Using a Multimodal Deep Learning Network.” *Biocybernetics and Biomedical Engineering*, Vol. 40, No. 1, 2020, pp. 324–336. https://doi.org/10.1016/j.bbe.2019.12.002.

[11] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. “SMOTE: Synthetic Minority Over-Sampling Technique.” *Journal of Artificial Intelligence Research*, Vol. 16, 2002, pp. 321–357. https://doi.org/10.1613/jair.953.

[12] Tripathi, S., Acharya, S., Sharma, R., Mittal, S., and Bhattacharya, S. “Using Deep and Convolutional Neural Networks for Accurate Emotion Classification on DEAP Data.” *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31, No. 2, 2017, pp. 4746–4752. https://doi.org/10.1609/aaai.v31i2.19105.

[13] Wei-Long Zheng, and Bao-Liang Lu. “Investigating Critical Frequency Bands and Channels for EEG-Based Emotion Recognition with Deep Neural Networks.” *IEEE Transactions on Autonomous Mental Development*, Vol. 7, No. 3, 2015, pp. 162–175. https://doi.org/10.1109/TAMD.2015.2431497.

[14] Cecotti, H., and Graser, A. “Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 3, 2011, pp. 433–445. https://doi.org/10.1109/TPAMI.2010.125.

[15] Roza, V. C. C., and Postolache, O. A. “Multimodal Approach for Emotion Recognition Based on Simulated Flight Experiments.” *Sensors (Switzerland)*, Vol. 19, No. 24, 2019. https://doi.org/10.3390/s19245516.

[16] Koelstra, S., Lee, J.-S., and Pun, T. “DEAP: A Database for Emotion Analysis Using Physiological Signals.” *IEEE TRANS. AFFECTIVE COMPUTING*, Vol. 3, 2012, pp. 18–31.

[17] Bhardwaj, A., Gupta, A., Jain, P., Rani, A., and Yadav, J. Classification of Human Emotions from EEG Signals Using SVM and LDA Classifiers. 2015.

[18] SUBASI, A. “EEG Signal Classification Using Wavelet Feature Extraction and a Mixture of Expert Model.” *Expert Systems with Applications*, Vol. 32, No. 4, 2007, pp. 1084–1093. https://doi.org/10.1016/j.eswa.2006.02.005.

[19] Lal, S. K. L., Craig, A., Boord, P., Kirkup, L., and Nguyen, H. “Development of an Algorithm for an EEG-Based Driver Fatigue Countermeasure.” *Journal of Safety Research*, Vol. 34, No. 3, 2003, pp. 321–328. https://doi.org/10.1016/S0022-4375(03)00027-6.

[20] Jap, B. T., Lal, S., Fischer, P., and Bekiaris, E. “Using EEG Spectral Components to Assess Algorithms for Detecting Fatigue.” *Expert Systems with Applications*, Vol. 36, No. 2 PART 1, 2009, pp. 2352–2359. https://doi.org/10.1016/j.eswa.2007.12.043.

[21] Kar, S., Bhagat, M., and Routray, A. “EEG Signal Analysis for the Assessment and Quantification of Driver’s Fatigue.” *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 13, No. 5, 2010, pp. 297–306. https://doi.org/10.1016/j.trf.2010.06.006.

[22] Trejo, L. J., Kubitz, K., Rosipal, R., Kochavi, R. L., and Montgomery, L. D. “EEG-Based Estimation and Classification of Mental Fatigue.” *Psychology*, Vol. 06, No. 05, 2015, pp. 572–589. https://doi.org/10.4236/psych.2015.65055.

[23] Binias, B., Myszor, D., and Cyran, K. A. “A Machine Learning Approach to the Detection of Pilot’s Reaction to Unexpected Events Based on EEG Signals.” *Computational Intelligence and Neuroscience*, Vol. 2018, 2018. https://doi.org/10.1155/2018/2703513.

[24] Sonnleitner, A., Treder, M. S., Simon, M., Willmann, S., Ewald, A., Buchner, A., and Schrauf, M. “EEG Alpha Spindles and Prolonged Brake Reaction Times during Auditory Distraction in an On-Road Driving Study.” *Accident Analysis and Prevention*, Vol. 62, 2014, pp. 110–118. https://doi.org/10.1016/j.aap.2013.08.026.

[25] Chaudhuri, A., and Routray, A. “Driver Fatigue Detection through Chaotic Entropy Analysis of Cortical Sources Obtained from Scalp EEG Signals.” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21, No. 1, 2020, pp. 185–198. https://doi.org/10.1109/TITS.2018.2890332.

[26] Dehais, F., Duprès, A., Blum, S., Drougard, N., Scannella, S., Roy, R. N., and Lotte, F. “Monitoring Pilot’s Mental Workload Using Erps and Spectral Power with a Six-Dry-Electrode EEG System in Real Flight Conditions.” *Sensors (Switzerland)*, Vol. 19, No. 6, 2019, p. 1324. https://doi.org/10.3390/s19061324.

[27] Patel, M., Lal, S. K. L., Kavanagh, D., and Rossiter, P. “Applying Neural Network Analysis on Heart Rate Variability Data to Assess Driver Fatigue.” *Expert Systems with Applications*, Vol. 38, No. 6, 2011, pp. 7235–7242. https://doi.org/10.1016/j.eswa.2010.12.028.

[28] Bashivan, P., Rish, I., Yeasin, M., and Codella, N. “Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks.” *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.

[29] Hajinoroozi, M., Mao, Z., Jung, T. P., Lin, C. T., and Huang, Y. “EEG-Based Prediction of Driver’s Cognitive Performance by Deep Convolutional Neural Network.” *Signal Processing: Image Communication*, Vol. 47, 2016, pp. 549–555. https://doi.org/10.1016/j.image.2016.05.018.

[30] Jiao, Z., Gao, X., Wang, Y., Li, J., and Xu, H. “Deep Convolutional Neural Networks for Mental Load Classification Based on EEG Data.” *Pattern Recognition*, Vol. 76, 2018, pp. 582–595. https://doi.org/10.1016/j.patcog.2017.12.002.

[31] Zhang, P., Wang, X., Zhang, W., and Chen, J. “Learning Spatial-Spectral-Temporal EEG Features With Recurrent 3D Convolutional Neural Networks for Cross-Task Mental Workload Assessment.” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 27, No. 1, 2019, pp. 31–42. https://doi.org/10.1109/TNSRE.2018.2884641.

[32] Wu, E. Q., Peng, X. Y., Zhang, C. Z., Lin, J. X., and Sheng, R. S. F. “Pilots’ Fatigue Status Recognition Using Deep Contractive Autoencoder Network.” *IEEE Transactions on Instrumentation and Measurement*, Vol. 68, No. 10, 2019, pp. 3907–3919. https://doi.org/10.1109/TIM.2018.2885608.

[33] Gao, Z., Wang, X., Yang, Y., Mu, C., Cai, Q., Dang, W., and Zuo, S. “EEG-Based Spatio-Temporal Convolutional Neural Network for Driver Fatigue Evaluation.” *IEEE transactions on neural networks and learning systems*, Vol. 30, No. 9, 2019, pp. 2755–2763. https://doi.org/10.1109/TNNLS.2018.2886414.

[34] Hogervorst, M. A., Brouwer, A. M., and van Erp, J. B. F. “Combining and Comparing EEG, Peripheral Physiology and Eye-Related Measures for the Assessment of Mental Workload.” *Frontiers in Neuroscience*, Vol. 8, No. OCT, 2014, pp. 1–14. https://doi.org/10.3389/fnins.2014.00322.

[35] Ahn, S., Nguyen, T., Jang, H., Kim, J. G., and Jun, S. C. “Exploring Neuro-Physiological Correlates of Drivers’ Mental Fatigue Caused by Sleep Deprivation Using Simultaneous EEG, ECG, and FNIRS Data.” *Frontiers in Human Neuroscience*, Vol. 10, No. MAY2016, 2016. https://doi.org/10.3389/fnhum.2016.00219.

[36] Liu, Y., Ayaz, H., and Shewokis, P. A. "Multisubject 'Learning' for Mental Workload Classification Using Concurrent EEG, FNIRS, and Physiological Measures." *Frontiers in Human Neuroscience*, Vol. 11, 2017, p. 389. https://doi.org/10.3389/fnhum.2017.00389.

[37] Harrivel, A. R., Liles, C. A., Stephens, C. L., Ellis, K. K., Prinzel, L. J., and Pope, A. T. "Psychophysiological Sensing and State Classification for Attention Management in Commercial Aviation." *AIAA Infotech @ Aerospace Conference*, No. January, 2016, pp. 1–8. https://doi.org/10.2514/6.2016-1490.

[38] Harrivel, A. R., Stephens, C. L., Milletich, R. J., Heinich, C. M., Last, M. C., Napoli, N. J., Abraham, N. A., Prinzel, L. J., Motter, M. A., and Pope, A. T. "Prediction of Cognitive States during Flight Simulation Using Multimodal Psychophysiological Sensing." *AIAA Information Systems-AIAA Infotech at Aerospace, 2017*, No. January, 2017, pp. 1–10. https://doi.org/10.2514/6.2017-1135.

[39] Terwilliger, P., Sarle, J., Walker, S., Terwilliger, P., Sarle, J., and Walker, S. "A ResNet Autoencoder Approach for Time Series Classification of Cognitive State A ResNet Autoencoder Approach for Time Series Classification of Cognitive State." *MODSIM World*, No. 0053, 2020, pp. 1–11.

[40] Alreshidi, I. M., Moulitsas, I., and Jenkins, K. W. *Miscellaneous EEG Preprocessing and Machine Learning for Pilots' Mental States Classification : Implications*. Association for Computing Machinery, 2022.

[41] Haibo He, and Garcia, E. A. "Learning from Imbalanced Data." *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 9, 2009, pp. 1263–1284. https://doi.org/10.1109/TKDE.2008.239.

[42] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. "Learning from Class-Imbalanced Data: Review of Methods and Applications." *Expert Systems with Applications*, Vol. 73, 2017, pp. 220–239. https://doi.org/10.1016/j.eswa.2016.12.035.

[43] Weiss, G. M., and Provost, F. *The Effect of Class Distribution on Classifier Learning: An Empirical Study*. 2001.

[44] Kim, K. H., Bang, S. W., and Kim, S. R. "Emotion Recognition System Using Short-Term Monitoring of Physiological Signals." *Medical & Biological Engineering & Computing*, Vol. 42, No. 3, 2004, pp. 419–427. https://doi.org/10.1007/BF02344719.

[45] Minguillon, J., Lopez-Gordo, M. A., and Pelayo, F. "Trends in EEG-BCI for Daily-Life: Requirements for Artifact Removal." *Biomedical Signal Processing and Control*, Vol. 31, 2017, pp. 407–418. https://doi.org/10.1016/j.bspc.2016.09.005.

[46] Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., and Hämäläinen, M. S. "MEG and EEG Data Analysis with MNE-Python." *Frontiers in Neuroscience*, Vol. 7, 2013. https://doi.org/10.3389/fnins.2013.00267.

[47] Carreiras, C., Alves, A. P., Lourenço, A., Canento, F., Silva, H., and Fred, A. BioSPPy - Biosignal Processing in Python. https://github.com/PIA-Group/BioSPPy/. Accessed Mar. 27, 2023.

[48] Alreshidi, I., Moulitsas, I., and Jenkins, K. W. "Multimodal Approach for Pilot Mental State Detection Based on EEG." 2023, pp. 1–22 (to be published).

[49] Welch, P. D. "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms." *IEEE Transactions on Audio and Electroacoustics*, Vol. 15, No. 2, 1967, pp. 70–73. https://doi.org/10.1109/TAU.1967.1161901.

[50] Smith, J. O. *Spectral Audio Signal Processing*. W3K Publishing, 2011.

2023-06-08

A comprehensive analysis of machine learning and deep learning models for þÿ i d e n t i f y i n g p i l o t s m e n t a l s t a t imbalanced physiological data

Alreshidi, Ibrahim

AIAA

Alreshidi IB, Yadav S, Moulitsas I, Jenkins KW. (2023) A comprehensive analysis of machine þÿ l e a r n i n g a n d d e e p l e a r n i n g m o d e l s f o r i d e n t i f y i n g p i l o t s m e n t a l s t a t e s f physiological data. In: 2023 AIAA Aviation and Aeronautics Forum and Exposition (AIAA AVIATION Forum), 12-16 June 2023, San Diego, CA

https://doi.org/10.2514/6.2023-4529