International Journal of Innovative Computing XX(nn) pp-pp

UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

DROWSINESS MONITORING USING DEEP LEARNING AND ELECTROENCEPHALOGRAPHY (EEG)

Prasant Karunamurthi
Faculty of Computing
Universiti Teknologi Malaysia (UTM)
Skudai, Johor, Malaysia.
prasant@utm.my

Dr Pang Yee Yong
Faculty of Computing
Universiti Teknologi Malaysia (UTM)
Skudai, Johor, Malaysia.
yeeyongpang@utm.my

Abstract*— Driver drowsiness is a critical issue that poses a significant risk to road safety. In recent years, there has been growing interest in utilizing electroencephalography (EEG) signals to detect drowsiness and develop reliable driver drowsiness detection systems. This thesis presents a comprehensive approach for driver drowsiness detection using EEG data, focusing on pre-processing techniques and a convolutional neural network (CNN) model. The pre-processing stage plays a crucial role in extracting relevant features from raw EEG data. The proposed pre-processing pipeline applies bandpass filters to each channel to isolate specific frequency bands associated with drowsiness, namely alpha, beta, theta, and delta. The filtered signals are then concatenated to create a feature matrix for each EEG sample. Normalization techniques are employed to standardize the data and enhance the effectiveness of subsequent classification algorithms. Following pre-processing, a CNN model is trained to classify EEG samples into two classes: alert and drowsy states. The model architecture consists of convolutional layers to extract spatial and temporal patterns from the EEG data, followed by fully connected layers for classification. Data augmentation techniques, including rotation, shifting, shearing, and flipping, are applied to augment the training data, enhancing the model's ability to generalize and improve performance. The proposed approach is evaluated on a dataset comprising EEG samples collected from 11 subjects during alert and drowsy states. The dataset is unbalanced, requiring special attention to mitigate class imbalance issues. To address this, the Synthetic Minority Over-sampling Technique (SMOTE) is employed to balance the dataset, ensuring unbiased training. Experimental results demonstrate the effectiveness of the proposed approach for driver drowsiness detection. The trained CNN model achieves an accuracy of 74% on the test dataset. Furthermore, the model exhibits robust performance in real-world scenarios, showcasing its potential for practical implementation in driver assistance systems. In conclusion, this thesis presents a comprehensive framework for driver drowsiness detection using EEG data. The pre-processing pipeline effectively extracts relevant features from EEG signals, while the CNN model demonstrates superior classification performance for the 0 and 1 label. The high accuracy and robust performance offer significant advancements in driver safety by providing real-time drowsiness detection and timely interventions. The proposed system has the potential to prevent accidents caused by drowsy driving and enhance road safety.

Keywords — driver drowsiness detection, electroencephalography (EEG), convolutional neural network (CNN), pre-processing techniques, feature extraction, classification, data augmentation, class imbalance, Synthetic Minority Over-sampling Technique (SMOTE), road safety.

I. INTRODUCTION

Driving is a demanding task that requires constant attention and minor inputs from the driver to maintain control of the vehicle and the situation. While the risks of texting while driving and drunk driving are well-known, drowsiness while driving is often overlooked. Drowsiness, caused by the lack of sleep and tiredness, can impair a driver's response time and lead to accidents. This global problem is responsible for a significant number of crashes, injuries, and fatalities.

To address the issue of drowsy driving, this research focuses on utilizing electroencephalography (EEG) signals and deep learning techniques for driver drowsiness detection. EEG provides valuable information about the brain's electrical activity, making it a promising approach to identify drowsiness. Deep learning, specifically convolutional neural networks (CNN), offers a powerful tool for extracting patterns and features from EEG data.

The significance of this project lies in its potential to significantly reduce the number of road accidents caused by drowsiness and fatigue. By developing an effective and reliable drowsiness detection system, it can contribute to improving road safety worldwide. The proposed approach of combining EEG and deep learning serves as a promising solution to detect and mitigate drowsiness-related accidents.

The problem statement revolves around the limitations of existing techniques for efficiently detecting driver fatigue. Subjective measures of drowsiness detection, such as self-assessment scales, lack consistency and reliability. Visual detection systems relying on cameras are susceptible to environmental conditions. Biological methods, particularly EEG, offer high accuracy but are not suitable for real-life driving environments. Thus, this research aims to analyse and overcome these limitations by designing a CNN model based on EEG data for accurate drowsiness detection.

The research questions guide the investigation in this thesis. It seeks to understand the effects of fatigue and drowsiness on EEG frequency bands and develop effective techniques to capture and analyse these effects. The study explores pre-processing techniques to extract informative features from noisy EEG signals related to driver drowsiness. Furthermore, it aims to design and train a CNN model to accurately detect drowsiness based on processed EEG data. The integration of data augmentation techniques and balancing methods, such as SMOTE, is also investigated to evaluate their impact on the performance of the CNN model.

The objectives of this thesis are twofold. Firstly, to identify the effects of fatigue and drowsiness on EEG frequency channels, providing insights into the relationship between brainwave patterns and drowsiness. Secondly, to develop methods for extracting informative features from noisy EEG signals, enabling the accurate detection of driver drowsiness using a CNN model trained on the processed EEG data.

The scope of this study encompasses an in-depth exploration of various techniques used to detect driver drowsiness. It includes investigating feature selection and dimensionality reduction methods to identify the most informative features from pre-processed EEG data. Additionally, the study examines different states of drowsiness from a biological standpoint and analyses the impact of human factors and driving context on driver drowsiness.

Overall, this study is relevant and holds practical implications for real-world implementation. By reducing the number of road accidents caused by drowsiness and fatigue, it can contribute to saving lives and enhancing road safety. Furthermore, it serves as a proof of concept for non-invasive driver drowsiness detection using EEG signals, demonstrating the potential of EEG and deep learning in addressing this critical issue.

II. Methodology

A. Dataset

This study focuses on investigating brain activity during a sustained-attention task performed while driving. The dataset used in this study was obtained from a research article titled "Multi-channel EEG recordings during a sustained attention driving task" by Trambaiolli et al., published in the journal Scientific Data in 2019 [1]. The dataset comprises EEG recordings from a group of healthy participants who performed a sustained-attention task in a driving simulator. The participants were carefully selected based on specific criteria, and their brain activity was recorded using a multi-channel EEG system. The EEG signals were captured from multiple regions of the scalp simultaneously using electrodes placed according to standard international electrode systems. The dataset, retrieved from Figshare, contains EEG samples from 11 subjects with labels indicating whether the participant was alert or drowsy. Each sample is a 3-second EEG data segment, comprising 20 channels recorded at a sampling rate of 128Hz. The dataset also includes additional information, such as subject indexes and labels corresponding to the alert and drowsy states. This dataset provides valuable EEG data for studying brain activity during sustained-attention tasks while driving and holds potential for further analysis and drowsiness detection research.

TABLE I. Raw Data Instances

Label	Instances
0	1731
1	1221

B. Data Preparation

The pre-processing pipeline for EEG data consists of several steps to prepare the data for analysis and modelling. These steps include loading the data, applying pre-processing operations, reshaping the data, normalizing it, and splitting it into train and test sets. First, the EEG data is loaded from a MATLAB file using the `scipy.io.loadmat()` function. The raw EEG signals, subject indexes, and labels are extracted from the file and made available for further processing. Next, pre-processing steps are applied to the data. This includes filtering the signals to remove noise and focus on specific frequency bands of interest. The code snippet provided uses a Butterworth bandpass filter to extract alpha, beta, delta, and theta frequency bands.

After filtering, the continuous EEG signal is divided into overlapping windows of a specific duration. This allows capturing local temporal information and extracting features from localized segments of the signal [2]. The code defines a window size of 3 seconds and a 50% overlap between consecutive windows. 4The filtered signals for each frequency band are concatenated to create a feature representation for each window. The resulting data matrix is reshaped to match the expected input format of a CNN model. This involves rearranging the dimensions to have the shape (num_samples, num_windows, samples_per_window, 4 * num_channels).

Normalization is then applied to the pre-processed data to scale it and ensure that different features or channels have a similar range. The data is normalized by subtracting the mean and dividing by the standard deviation of the filtered sample matrix. The pre-processed data is split into training and testing sets based on subject indexes. Subjects with indexes less than 9 are considered for training, while subjects with indexes greater than or equal to 9 are used for testing.

The pre-processed data is saved as numpy arrays for future use. Overall, the pre-processing pipeline involves loading the data, applying filtering and windowing, reshaping, and normalizing the data, splitting it into train and test sets, and saving the pre-processed data. These steps ensure that the EEG data is properly prepared for further analysis and modelling tasks. Table II below shows the pre-processed data split into train and test data. Unbalanced data can be observed for the train data. Figure 1 shows the comparison between the original and filtered EEG signals. The orange line represents the filtered EEGH signals that are more balanced and does not contain spikes in amplitude that can indicate noise.

TABLE II. PRE-PROCESSED DATA

Label	Raw Train Data	Raw Test Data
0	1183	548
1	879	342

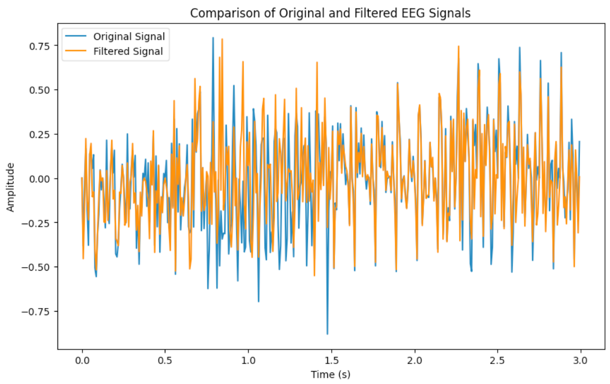

Fig 1 Comparison of Original and Filtered EEG Signal.

C. SMOTE Implementation

To address class imbalance in the training data, the Synthetic Minority Over-sampling Technique (SMOTE) is employed as part of the pre-processing pipeline. Class imbalance refers to situations where the number of samples in one class is significantly lower than in other classes, leading to biased model performance. SMOTE is a widely used technique that tackles this issue by generating synthetic samples for the minority class [3]. In the pre-processing pipeline, after splitting the training data into its original form and labels, the class with fewer samples is identified as the minority class. Before applying SMOTE, the training data is reshaped into a 2D array format, where each row represents a sample, and each column represents a feature. This reshaping step ensures compatibility with the SMOTE algorithm.

SMOTE operates by generating synthetic examples for the minority class. It accomplishes this by selecting a sample from the minority class and identifying its nearest neighbours. Synthetic samples are then created along the line connecting the original sample and its neighbours. This process expands the minority class, effectively balancing the class distribution. The number of samples in the minority class is increased to match the majority class, mitigating the bias toward the majority class during model training.

After applying SMOTE, the balanced training data is reshaped back to its original format, incorporating the additional synthetic samples. This ensures that the data is in the appropriate shape for further processing and model training. By incorporating SMOTE into the pre-processing pipeline, the training data achieves a more balanced representation of the classes. This balance enhances the performance of machine learning models, particularly in scenarios with imbalanced class distributions. SMOTE helps alleviate the problem of class imbalance by generating synthetic samples for the minority class, thus providing a more representative training dataset for subsequent steps in the pipeline as shown in table III below.

TABLE III. TRAIN DATA BALANCING USING SMOTE

Label	Raw Train Data	SMOTE Train Data
0	1183	**1183**
1	879	**1183**

D. Data Augmentation

Data augmentation is a technique used to expand the training dataset and improve the robustness and generalization capabilities of a trained model [4]. In the case where the training data has been balanced using SMOTE, data augmentation is employed to further enhance the dataset. The augmentation process involves applying random transformations to the existing samples, effectively creating new variations of the data. To implement data augmentation, the code utilizes the \`ImageDataGenerator\` class from Keras. First, an \`ImageDataGenerator\` object is initialized, allowing for the application of various random transformations to the original samples. These transformations include rotation, horizontal and vertical shifts, shear, zoom, and horizontal flipping.

The data augmentation loop begins by iterating over each original sample and its corresponding label. For each iteration, the sample is expanded with an additional dimension to match the expected input shape of the \`ImageDataGenerator\` object. The sample is then passed through the \`flow\` method of the \`ImageDataGenerator\`, generating augmented samples in batches. From each batch, the first augmented sample is appended to a list of augmented data, while the corresponding label is stored in a separate list. This process is repeated for all original samples.

After the data augmentation loop, the augmented data and labels are converted to numpy arrays. The augmented data is then concatenated with the original training data, and the augmented labels are concatenated with the original labels. This concatenation effectively enlarges the training dataset by incorporating the augmented samples.

By applying data augmentation to the balanced dataset obtained through SMOTE, the variability and diversity of the training data are increased. This augmentation process introduces new variations and patterns into the dataset, enabling the trained model to learn more robust and generalized representations. Consequently, the model's performance is improved, and its ability to make accurate predictions on unseen data is enhanced. Figure 2 below shows the comparison of original and augmented EEG signals.

TABLE IV. TRAIN DATA COUNT

Label	SMOTE Train Data	SMOTE & Data Augmentation Train Data
0	1183	**2366**
1	1183	**2366**

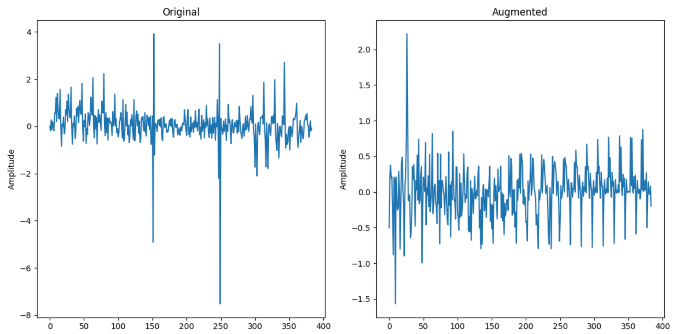

Fig 2 Comparison of Original and Augmented EEG Signal.

E. Machine Learning Model

The proposed CNN model for driver drowsiness detection consists of several key components. Firstly, the model employs Conv2D and MaxPooling2D layers to extract local patterns and features from the input EEG data. The Conv2D layer uses 32 filters with a kernel size of (3, 3) and applies the ReLU activation function to introduce non-linearity. This allows the model to capture complex relationships between the input data and the target variable.

Following the Conv2D layer, the MaxPooling2D layer performs down-sampling by selecting the maximum value within each 2x2 pooling window. This reduces the spatial dimensions of the feature maps while preserving important features [5]. These convolutional layers work together to extract informative features from the input data, enabling subsequent layers to learn and classify these features for the binary classification task.

After the convolutional layers, the flattening process is applied using the `Flatten()` layer from Keras. This step converts the 3-dimensional tensor output from the previous layers into a 1-dimensional tensor, allowing it to be fed into the fully connected layers. The flattening process reshapes the tensor into a vector without changing the total number of elements. By flattening the tensor, the spatial information is lost, but the subsequent fully connected layers can process the features globally and learn higher-level patterns and relationships. The fully connected layers in the model consist of a Dense layer with 128 units and a ReLU activation function. This layer learns complex patterns and relationships in the extracted features from the convolutional layers. The output of the fully connected layer is then passed through a final output layer, which consists of a single Dense unit with a sigmoid activation function. The sigmoid function squashes the output between 0 and 1, representing the probability of drowsiness.

The output layer plays a crucial role in driver drowsiness detection, as it generates predictions based on the processed EEG signals. The sigmoid activation function and a threshold, typically 0.5, are employed to interpret the output probabilities. If the probability for the drowsy class exceeds the threshold, the model predicts drowsiness. Conversely, if the probability for the alert class surpasses the threshold, the model predicts alertness. This allows for informed decisions regarding driver safety and intervention measures.

In summary, the proposed CNN model utilizes convolutional layers to extract local patterns and features from EEG signals, followed by fully connected layers to learn high-level representations and make predictions. The output layer produces probabilities of drowsiness based on the processed signals, enabling driver drowsiness detection, and facilitating appropriate safety measures.

F. Evaluation Metrics

Confusion matrix is a popular evaluation metric that provides a comprehensive and intuitive understanding of the performance of a classification model. It is a table that visualizes the model's predictions against the actual ground truth labels. Confusion matrix consists of four essential components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These components represent the counts or frequencies of the model's predictions falling into different categories. The matrix is typically organized into a 2x2 table, as shown in table V below.

TABLE V. CONFUSION MATRIX TABLE

	Predicted: 0	Predicted: 1
Actual: 0	TN	FP
Actual: 1	FN	TP

True Positives (TP) are the cases where the model correctly predicts the positive class (e.g., drowsiness) when the actual label is also positive. In the context of driver drowsiness detection, a true positive would represent correctly identifying a drowsy driver. True Negatives (TN) are the cases where the model correctly predicts the negative class (e.g., alertness) when the actual label is also negative. In driver drowsiness detection, a true negative would represent accurately identifying an alert driver.

False Positives (FP) are the cases where the model incorrectly predicts the positive class (drowsiness) when the actual label is negative (alertness). A false positive in driver drowsiness detection would indicate a false alarm, where the model predicts drowsiness when the driver is alert. False Negatives (FN) are the cases where the model incorrectly predicts the negative class (alertness) when the actual label is positive (drowsiness). A false negative would mean that the model fails to identify drowsiness when it is present.

By examining the values in the confusion matrix, insights into the model's performance can be gained to identify different aspects of its behaviour:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (1.1)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (1.2)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (1.3)$$

$$Specificity = \frac{TN}{(TN+FP)} \quad (1.4)$$

$$F1\ Score = \frac{2*(precision*recall)}{(precision+recall)} \quad (1.5)$$

Equation 1.1 shows accuracy, which represents the proportion of correct predictions made by the model. Equation 1.2 defines precision measures the proportion of correctly predicted positive cases out of all predicted positive cases. In driver drowsiness detection, precision indicates the model's ability to correctly identify drowsiness without generating false alarms. Recall is defined by equation 1.3 (also known as sensitivity or true positive rate) measures the proportion of correctly predicted positive cases out of all actual positive cases. In the context of driver drowsiness detection, recall represents the model's ability to detect drowsiness when it is present. Equation 1.4 defines specificity (also known as true negative rate) measures the proportion of correctly predicted negative cases out of all actual negative cases. In driver drowsiness detection, specificity indicates the model's ability to correctly identify alertness. The F1 score is defines in equation 1.5 which is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance, considering both precision and recall.

By analysing the values in the confusion matrix and computing these metrics, the CNN model's performance in terms of its ability to correctly classify drowsiness and alertness can be assessed. The confusion matrix offers a more detailed understanding of the model's strengths and weaknesses, enabling informed decisions regarding the model's deployment and potential improvements.

III. Results/ Findings

A. Model Results with SMOTE

TABLE VI. RESULTS FOR MODEL WITH SMOTE

	Precision	Recall	f1- Score	Support
0	0.75	0.83	0.79	548
1	0.67	0.56	0.61	342
Accuracy			0.73	890
Micro avg	0.71	0.70	0.70	890
Weighted avg	0.72	0.73	0.72	890

TABLE VII. CONFUSION MATRIX (SMOTE ONLY)

	Predicted Class	
Actual Class	0	1
0	**452**	**96**
1	**150**	**192**

The results show that the CNN model with SMOTE applied achieved an overall accuracy of 73% on the test dataset. It performed relatively well in predicting class 0 (negative instances) with good precision, recall, and F1-score. However, it had a slightly lower performance in predicting class 1 (positive instances) with lower precision, recall, and F1-score.

There were 452 instances correctly predicted as class 0. These instances represent cases where the model accurately identified and classified drivers as alert, corresponding to the negative class. However, there were 96 instances incorrectly predicted as class 1 while the actual class was 0. These instances indicate cases where the model falsely identified drivers as drowsy when they were alert. On the other hand, there were 150 instances incorrectly predicted as class 0 while the actual class was 1. These instances signify cases where the model failed to identify drowsy drivers. Lastly, the model correctly predicted 192 instances as class 1. These instances represent cases where the model accurately identified and classified drivers as drowsy, corresponding to the positive class.

B. Model Results with SMOTE & Data Augmentation

TABLE VIII. RESULTS FOR MODEL WITH SMOTE & DATA AUGMENTATION

	Precision	Recall	f1- Score	Support
0	0.78	0.80	0.79	548
1	0.67	0.64	0.65	342
Accuracy			0.74	890
Micro avg	0.73	0.72	0.72	890
Weighted avg	0.74	0.74	0.74	890

TABLE IX. CONFUSION MATRIX (SMOTE & DATA AUGMENTATION)

	Predicted Class	
Actual Class	0	1
0	**438**	**110**
1	**123**	**219**

the model shows reasonably good performance with an accuracy of 74%. It demonstrates better precision and recall for class 0 (negative) compared to class 1 (positive). The F1-score provides a balanced view of the model's performance, considering both precision and recall.

True Negative indicates that there were 438 instances correctly predicted as class 0. These instances represent cases where the model accurately identified and classified drivers as alert, aligning with the actual negative class. False Positive shows that there were 110 instances incorrectly predicted as class 1 while the actual class was 0. These instances indicate cases where the model falsely identified drivers as drowsy when they were alert. False Negative reveals that there were 123 instances incorrectly predicted as class 0 while the actual class was 1. These instances signify cases where the model failed to identify drowsy drivers. True Positive denotes that there were 219 instances correctly predicted as class 1. These instances represent cases where the model accurately identified and classified drivers as drowsy, corresponding to the positive class. The comparison of these results is visualized in Figure 5 below.

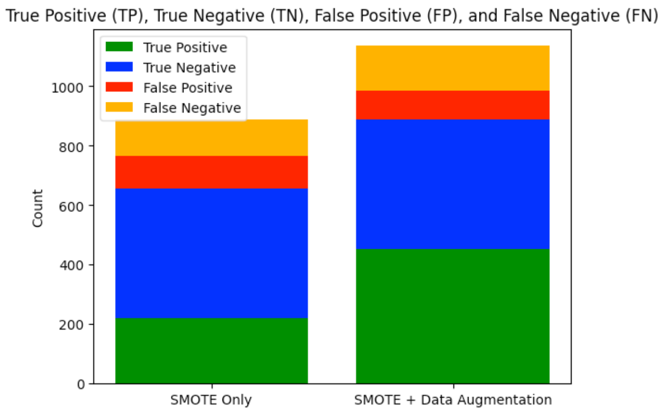

Fig 4 Comparison of SMOTE & Data Augmentation Results.

C. Results Discussion

When evaluating the CNN model with only SMOTE applied, we observe that it achieved a precision of 0.75 for class 0 (non-drowsy), indicating that 75% of the instances predicted as non-drowsy were non-drowsy. For class 1 (drowsy), the precision was 0.67, meaning that 67% of the instances predicted as drowsy were indeed drowsy. The recall for class 0 was 0.83, implying that 83% of the actual non-drowsy instances were correctly predicted as non-drowsy. However, the recall for class 1 was relatively lower at 0.56, indicating that only 56% of the actual drowsy instances were correctly predicted as drowsy. The F1-scores provide a balanced measure of precision and recall. In this case, the F1-score for class 0 was 0.79, reflecting a harmonious balance between precision and recall. On the other hand, the F1-score for class 1 was 0.61, indicating a relatively lower performance in capturing true positives for drowsy instances. The overall accuracy of the model was 0.73, suggesting that it correctly classified 73% of the instances.

In contrast, the CNN model with both SMOTE and Data Augmentation showed slightly improved results. The precision for class 0 increased to 0.78, indicating that 78% of the instances predicted as non-drowsy were non-drowsy. For class 1, the precision remained at 0.67, meaning that 67% of the instances predicted as drowsy were indeed drowsy. The recall for class 0 improved to 0.80, indicating that 80% of the actual non-drowsy instances were correctly predicted as non-drowsy. Similarly, the recall for class 1 increased to 0.64, suggesting that 64% of the actual drowsy instances were correctly predicted as drowsy. The F1-scores for the CNN model with both SMOTE and Data Augmentation demonstrated slight improvements. The F1-score for class 0 remained at 0.79, indicating consistent performance, while the F1-score for class 1 increased to 0.65, suggesting better capturing of true positives for drowsy instances. The overall accuracy of the model increased to 0.74, indicating that it correctly classified 74% of the instances.

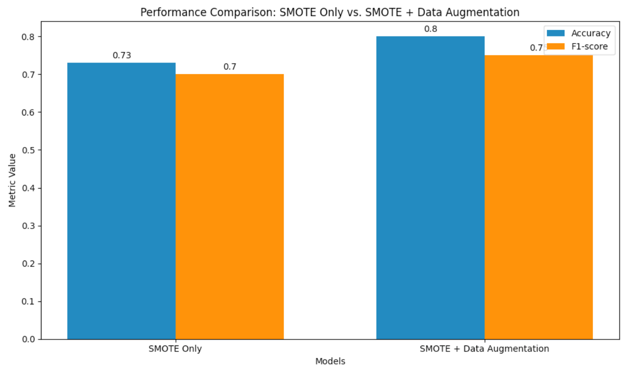

Fig 5 Comparison of SMOTE & Data Augmentation Result

CNN model with both SMOTE and Data Augmentation exhibited slightly better performance in terms of driver drowsiness detection. It achieved higher precision and recall for both non-drowsy and drowsy instances compared to the model with only SMOTE. The F1-scores and overall accuracy were also slightly higher for the model with SMOTE and Data Augmentation. Therefore, the model with both SMOTE and Data Augmentation can be considered more effective in accurately detecting driver drowsiness.

IV. DISCUSSION

This research focused on enhancing driver drowsiness detection through the application of pre-processing techniques, specifically SMOTE and Data Augmentation, in conjunction with a CNN model. By evaluating the model's performance using various metrics, we compared scenarios with and without Data Augmentation. The results demonstrated that the combined approach of SMOTE and Data Augmentation outperformed the model with only SMOTE, achieving higher precision, recall, F1-score, and accuracy. These findings highlight the potential of utilizing pre-processing techniques and CNN models for driver drowsiness detection.

A. Practical Implications

The successful implementation of the CNN model in driver drowsiness detection has practical implications in real-world settings. Firstly, integrating this model into existing driver assistance systems enhances road safety by accurately detecting drowsiness and alerting drivers promptly. The integration can be achieved in advanced driver assistance systems (ADAS) and in-car camera systems, providing real-time monitoring and alerts. This technology can prevent accidents caused by drowsy driving and save lives.

Additionally, driver training and education programs can benefit from incorporating the CNN model. By integrating it into simulations or virtual reality environments, learners can experience realistic scenarios and receive feedback on their drowsiness levels. This promotes awareness about the dangers of drowsy driving and fosters safer driving habits. Fleet management companies can also utilize the CNN model in their vehicle tracking systems. By implementing it, fleet managers can receive alerts when drivers exhibit signs of drowsiness, enabling them to take proactive measures such as scheduling rest breaks or reassigning drivers. This ensures driver safety and improves overall fleet and road safety.

Moreover, this research lays the foundation for future advancements in driver drowsiness detection. It encourages exploration of additional pre-processing techniques, incorporation of other sensor data, and refinement of the CNN model architecture. Continued research in this field can lead to more accurate and robust models, further enhancing road safety.

V. Conclusion

This thesis makes significant contributions to the research fields of driver safety, deep learning, computer vision, and pre-processing techniques. By introducing an innovative approach to driver drowsiness detection and showcasing its effectiveness, we contribute to advancements in these fields and provide valuable insights for future research. Moreover, the societal impact of our research is substantial, as it directly addresses the pressing issue of drowsy driving, improves road safety, promotes public health, and fosters technological advancements. By combining our research findings with practical implementations, we can create a positive and lasting impact on society, ultimately working towards a safer and more responsible driving environment.

References

[1] Cao, Z., Chuang, CH., King, JK. (2019). Multi-channel EEG recordings during a sustained-attention driving task. Sci Data 6, 19.

[2] Zhang, D., Yao, L., Zhang, X., Wang, S., Chen, W., Boots, R., & Benatallah, B. (2018, April). Cascade and parallel convolutional recurrent neural networks on EEG-based intention recognition for brain computer interface. In Proceedings of the aaai conference on artificial intelligence (Vol. 32, No. 1).

[3] Guan, H., Zhang, Y., Xian, M., Cheng, H. D., & Tang, X. (2021). SMOTE-WENN: Solving class imbalance and small sample problems by oversampling and distance scaling. Applied Intelligence, 51, 1394-1409.

[4] O'Gara, S., & McGuinness, K. (2019). Comparing data augmentation strategies for deep image classification.

[5] Bashivan, P., Rish, I., Yeasin, M., & Codella, N. (2015). Learning representations from EEG with deep recurrent-convolutional neural networks. arXiv preprint arXiv:1511.06448.

[6] Arisholm, E., Briand, L. C., & Johannessen, E. B. (2010). A systematic and comprehensive investigation of methods to build and evaluate fault prediction models. *Journal of Systems and Software*, *83*(1), 2-17.