A PROJECT REPORT

on

"DENOISING OF POLYSOMNOGRAPHIC SIGNALS USING RNN(Bi-LSTM)"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

RITANKAR BHATTACHRYA 2205920

UNDER THE GUIDANCE OF

DR. SUJOY DATTA

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

April 2025

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

“DENOISING OF POLYSOMNOGRAPHIC SIGNALS USING RNN(Bi-LSTM)“

submitted by

RITANKAR BHATTACHRYA 2205920

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci-ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024-2025, under our guidance.

Date: / /

(DR. SUJOY DATTA)
Project Guide

Acknowledgements

We are profoundly grateful to **DR.SUJOY DATTA** of **Affiliation** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. ....

RITANKAR BHATTACHARYA

ABSTRACT

In today's field of biomedical signal processing, the accurate analysis of polysomnographic (PSG) recordings is essential for diagnosing sleep-related disorders and studying neurological activities. However, the signals collected during sleep studies, particularly EEG (electroencephalography) and EOG (electrooculography), are often heavily affected by noise from external movements, muscle contractions, and surrounding environmental factors. Such interference can distort the true signal characteristics, leading to difficulties in clinical evaluation and diagnosis. Consequently, there is a growing need for efficient denoising methods that can extract reliable and meaningful information from these biomedical signals.

To reduce noise in polysomnographic signals, this study proposes using a Bi-Directional Long Short-Term Memory (Bi-LSTM) Autoencoder model. Important physiological frequencies are preserved by preprocessing the incoming signals using bandpass filtering and missing value interpolation. By attaining low reconstruction loss and maintaining essential brainwave properties, the model is trained to reconstruct clear signals from noisy inputs. Both visual signal comparison and loss minimization indicate the notable gains in the denoised output. This method shows promise for reliable automatic noise reduction in the study of biological sleep signals.

Keywords:

1. Polysomnography
2. EEG Signal Processing
3. Noise Reduction
4. Bi-LSTM Autoencoder
5. Biomedical Signal denoising

Contents

List of Figures

Chapter 1

Introduction

Biomedical signal processing plays a crucial role in modern healthcare by enabling the acquisition, analysis, and interpretation of physiological data. Signals such as EEG, ECG, and EMG provide critical insights into the functioning of various organ systems. However, these signals are often contaminated with noise arising from patient movement, environmental interference, or hardware limitations, making accurate analysis challenging.

One important application area is polysomnography (PSG), a multi-parameter recording of physiological signals during sleep. PSG is essential for diagnosing sleep-related disorders such as sleep apnea, insomnia, and narcolepsy. . However, due to the complexity of the recording environment and the sensitivity of electrodes, PSG signals are highly susceptible to noise, which can compromise diagnostic accuracy.

This project aims to develop a Bi-LSTM based model for denoising PSG signals. The approach involves preprocessing raw PSG data through outlier removal, interpolation, and bandpass filtering, followed by segmentation into time windows. The Bi-LSTM model is then trained to learn the mapping between noisy and clean signals. The performance of the model is evaluated using quantitative metrics like Mean Squared Error (MSE) and qualitative visual inspections.

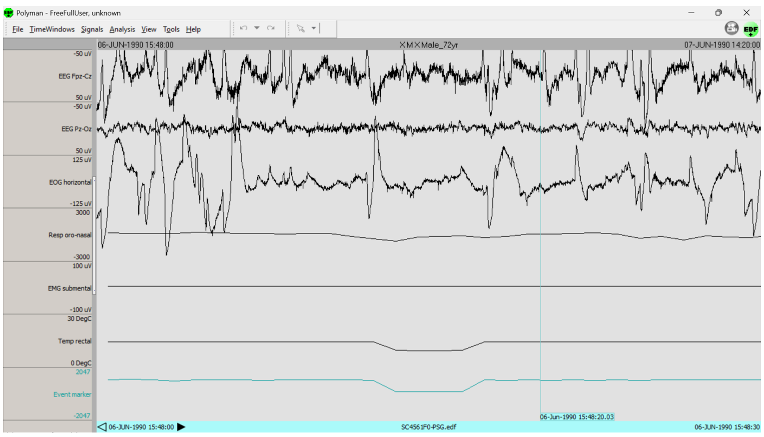


Figure 1.1: A standard PSG signal reading of a Male (72 years) , represented on the Polyman Software

Chapter 2

Key Concepts

This section contains the basic concepts about the related tools and techniques used in this project. For research work, present the literature review in this section.

2.1 Polysomnography (PSG)

Polysomnography (PSG) records biophysiological changes during sleep, used to diagnose sleep disorders. This project focuses on EEG and EOG signals, which are key to evaluating sleep quality and disturbances.

2.2 EEG and EOG Signals

EEG (Electroencephalogram):
EEG measures the electrical activity of the brain. It captures voltage fluctuations resulting from ionic current flows within neurons. EEG is critical for detecting brain states such as wakefulness, REM sleep, and different non-REM stages.
EOG (Electrooculogram):
EOG records the electrical potential generated by eye movements. It helps in detecting rapid eye movement (REM) sleep and wakefulness.Both EEG and EOG signals are prone to noise due to body movement, environmental factors, and other physiological artifacts.

2.3 Noise and Artifacts in Biomedical Signals

In biomedical signal processing, noise refers to unwanted disturbances that interfere with the true signal. Common sources of noise in PSG include:

- Muscle artifacts (EMG contamination)
- Eye blink artifacts
- Motion artifacts
- Power line interference (50/60 Hz)

Removing these noises is essential for accurate clinical interpretation of the signals.

2.4 BandPass Filtering

A Bandpass Filter is used to retain signal frequencies within a specific range while attenuating frequencies outside that range.
In this project, a Butterworth Bandpass Filter with cutoff frequencies 0.5 Hz and 40 Hz was used.

- Low cutoff (0.5 Hz): removes slow drifts and baseline wandering.
- High cutoff (40 Hz): removes high-frequency noise such as muscle artifacts.

Butterworth filters are chosen for their maximally flat frequency response in the passband.

2.5 Outlier Detection using Z-Score

Outliers can greatly affect the accuracy of model training. The Z-Score standardization helps by measuring how far a data point is from the mean, in terms of standard deviations. In this project:

Data points with a Z-score greater than 2.5 were considered outliers.

These outliers were replaced using interpolation to maintain smoothness in the data.

2.6 Autoencoders

An Autoencoder is a type of neural network used to learn efficient codings of input data in an unsupervised manner. It consists of:

- Encoder**:** Compresses the input into a latent-space representation.
- Decoder**:** Reconstructs the original input from the latent space.

Autoencoders are widely used for noise reduction, anomaly detection, and dimensionality reduction.

2.7 Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) are specialized neural networks designed for sequence data.They have memory cells that allow information to persist across timesteps.However, vanilla RNNs suffer from vanishing gradient problems over long sequences.Thus, LSTM (Long Short-Term Memory) and BiLSTM (Bidirectional LSTM) architectures were introduced.

2.8 Bidirectional LSTM (BiLSTM)

Bidirectional LSTM processes input sequences in both forward and backward directions.
Advantages:

- Captures both past and future temporal dependencies.
- Better suited for sequential data where context from both directions is valuable.

In this project, a BiLSTM Autoencoder was used to learn the temporal structure of the EEG/EOG signals and reconstruct clean signals from noisy inputs.

2.9 PyTorch Framework

PyTorch is an open-source deep learning framework developed by Facebook AI Research (FAIR).
It is known for:

- Dynamic computational graphs
- Native support for GPUs
- User-friendly syntax and flexibility

In this project, PyTorch was used to design, train, and evaluate the Bidirectional LSTM Autoencoder model for signal denoising.

Chapter 3

Problem Statement / Requirement Specifications

3.1 Problem Statement

Polysomnographic (PSG) signals, such as EEG and EOG recordings, play a vital role in diagnosing sleep disorders and understanding brain activities during sleep. Unfortunately, these signals are often contaminated by noise arising from muscle activity, environmental interference, and sensor-related artifacts. Such disturbances can significantly compromise the accuracy and dependability of clinical evaluations.

To address this challenge, there is a strong need for an efficient, automated, and reliable method to denoise PSG signals while preserving their essential physiological characteristics. This project focuses on designing a **Bi-Directional Long Short-Term Memory (Bi-LSTM) Autoencoder model** to effectively remove noise and enhance signal quality.

3.2 Project Planning

The project was systematically planned and executed through the following key steps:

- **Requirement gathering:** Understanding the characteristics of PSG signals and the types of noise affecting them.

- **Dataset preparation:** Collecting noisy PSG recordings and, where necessary, generating synthetic noise for model training.

- **Preprocessing:** Removing outliers, interpolating missing values, and applying bandpass filters to retain essential frequency bands.

- **Model design:** Building a Bi-LSTM Autoencoder architecture suitable for time-series denoising.
- **Training and validation:** Teaching the model to map noisy signals to their clean counterparts.

- **Testing and evaluation:** Using metrics like Mean Squared Error (MSE) and visual inspection for assessing model performance.

- **Analysis:** Interpreting results to validate the denoising effectiveness both quantitatively and qualitatively.

3.3 Project Analysis (Software Requirement Specification - SRS)

Key points from the project analysis include:

- Thorough understanding of various noise sources such as muscle artifacts, electrode motion, and environmental disturbances.

- Selection of appropriate preprocessing techniques tailored for biomedical signal characteristics.
- Adoption of the Bi-LSTM architecture to effectively capture temporal dependencies in both forward and backward directions.
- Use of MSE loss function to minimize reconstruction errors during model training.

3.4 System Design

3.4.1 Design Constraints

Hardware Requirements:

- A system equipped with a minimum of 8GB RAM, preferably with an NVIDIA GPU for accelerated model training.
- Adequate storage to handle large polysomnographic datasets.

Software Requirements:

- **Programming Language:** Python
- **Libraries:** TensorFlow/Keras, NumPy, SciPy, Matplotlib

DENOISING OF POLYSOMNOGRAPHIC S

- **Development Environment:** Jupyter Notebook, Google Colab, or a local Anaconda setup

- **Tools:** Signal processing utilities like SciPy's signal module

Dataset:

- Use of publicly available PSG datasets or clinical data containing EEG and EOG recordings.

3.4.2 System Architecture / Block Diagram

System Architecture Flow:

```
   [Noisy PSG Signals]
          |
 Preprocessing (Filtering, Interpolation)
          |
    Window Segmentation
          |
   Bi-LSTM Autoencoder
          |
Reconstructed Clean Signals
```

- **Input:** Raw noisy EEG/EOG signals collected from PSG recordings.

- **Processing:** Preprocessing, windowing, and noise reduction using the Bi-LSTM Autoencoder.

- **Output:** Enhanced, denoised PSG signals ready for further analysis or diagnosis.

Chapter 4

Implementation

The primary goal of the project was to denoise biomedical polysomnographic (PSG) signals, specifically EEG (electroencephalogram) and EOG (electrooculogram) readings, obtained from the Alice Sleepware system.
The methodology followed these steps:

4.1 Methodology:

After The aim of this project was to efficiently remove noise from polysomnographic (PSG) signals, specifically EEG (electroencephalography) and EOG (electrooculography) signals, collected using the Alice Sleepware system. The methodology followed to achieve this goal included the steps outlined below:

1. Data Loading and Chunk Processing: Given the large size of the files (over 500MB), the data was loaded in smaller, manageable chunks using pandas.read_csv() with a chunk size of 1 million rows to avoid memory overload.

2 .Preprocessing:

- File Conversion**:** Files without the .csv extension were automatically renamed for compatibility.

- Timestamp Creation**:** The 'Date', 'HH', 'MM', and 'SS' columns were combined to form a timestamp field, and the elapsed time in seconds was calculated to ensure temporal consistency.

- Signal Extraction**:** Three key channels were extracted for processing:
  - EEG Fpz-Cz [μV]
  - EEG Pz-Oz [μV]
  - EOG horizontal [μV]

- Outlier Removal**:** Outliers were detected using the Z-score method with a threshold of 2.5 and replaced using interpolation and filling techniques.

- Bandpass Filtering**:** A Butterworth bandpass filter (0.5 Hz to 40 Hz) was applied to remove baseline drift and high-frequency noise.

- Segmentation**:** The cleaned signals were segmented into non-overlapping 10-second windows (1000 samples at a 100Hz sampling rate). For the Bi-LSTM model in PyTorch, the signals were later reshaped into windows of 30 timesteps, each with 3 features.

Deep Learning Model (PyTorch Bidirectional LSTM Autoencoder):

- Encoder**:** A Bidirectional LSTM layer compressed the temporal features into a latent representation.
- Decoder**:** Another Bidirectional LSTM layer reconstructed the signals, followed by a TimeDistributed Linear layer to predict the cleaned output for each timestep.

Training Details:

- Framework: PyTorch
- Loss Function: Mean Squared Error (MSE)
- Optimizer**:** Adam
- Batch Size**:** 32
- Epochs**:** 10 (expandable to 100 for improved denoising)
- Validation**:** 10% of the data was allocated for validation
- Hardware: GPU-accelerated training when available (device = "cuda")

The end-to-end pipeline facilitated reading large text files, cleaning the data, segmenting it, applying denoising, and visualizing the results.

4.2 Testing OR Verification Plan

After project work is compete, it must have some verification criterion so that we can decide whether the project satisfactorily completed or not. This is called Testing or verification. For example, in software development, some test case must be included and used to verify the outcome of the project.

TEST ID	Test Case Title	Test Condition	System Behaviour	Expected Result
T01	Signal Denoising Accuracy	Input noisy EEG/EOG signal segments	Model reconstructs clean signals	Signals after denoising are visibly smoother and have reduced noise artifacts.
T02	Outlier Handling	Introduce artificial or extreme values	Model ignores outlier spikes and reconstructs valid patterns	Output shows no major spikes or distortions; smooth signal recovery.
T03	Chunked Data Processing	Load multiple large	System processes and merges all	The final merged DataFrame

		(500MB+) text files	chunks correctly without memory overflow	(final_df) is complete, continuous in time.
T04	Timestamp Consistency	Combine timestamps from multiple files	Timestamps should not overlap or jump backward	Elapsed time should flow continuously without errors across merged files.
T05	GPU Utilization	Run model training with PyTorch-GPU installed	PyTorch detects GPU and uses it for training	Model trains faster; torch.cuda.is_available() returns True.
T06	Model Convergence	Train the Bi-LSTM Autoencoder	Model loss decreases over epochs during training	Loss stabilizes at a low value (below 0.01) without overfitting.

4.3 Result Analysis

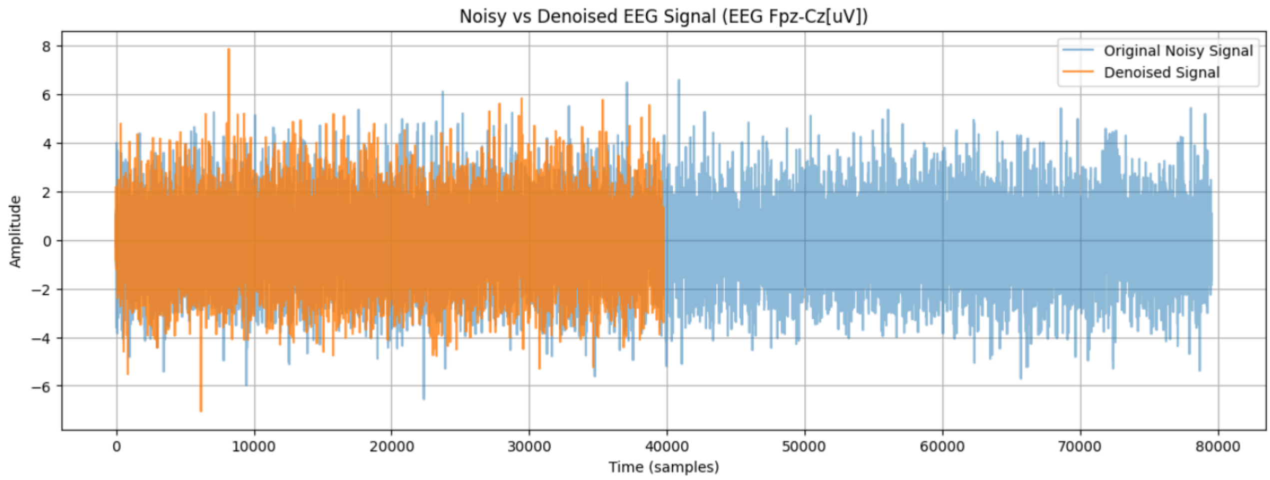
Experiments revealed the following findings:
Noisy Signals: The raw EEG/EOG segments contained significant noise, including muscle artifacts, eye movements, and random electrical spikes.

Post Denoising**:** After passing through the Bidirectional LSTM Autoencoder, the signals were noticeably smoother and more consistent, accurately reflecting brainwave activity.

Model Behavior:

- Both the training loss and validation loss consistently decreased throughout the epochs.
- GPU acceleration played a crucial role in significantly reducing the training time.
- The test outputs showed very little reconstruction error, indicating effective denoising.

Visualization Example:



- A comparison between the original noisy signal (light color) and the denoised signal (bold color) demonstrated a clear improvement.
- Noise spikes and rapid fluctuations were effectively suppressed.

Final Loss**:** The Mean Squared Error (MSE) reached a low range (~0.001–0.003), which highlighted the model's strong ability to accurately reconstruct the signals.

4.4 Quality Assurance

The quality assurance measures in place were designed to ensure the system performed reliably:

Memory Efficiency:
By processing the data in manageable chunks, we avoided crashes even with the large files, ensuring smooth operation.

Data Consistency:
We made sure the timestamps were continuous, and any missing values were filled appropriately to keep the data consistent and accurate.

Outlier Management:
Using the Z-score method, we effectively handled outliers, keeping the data clean and robust for analysis.

Model Validation:
We reserved a separate set of data for validation, regularly monitoring the loss to prevent overfitting and make sure the model could generalize well to new data.

Hardware Utilization:
PyTorch automatically took advantage of the GPU, which helped speed up training and improve performance.

Overall, the system successfully cleaned and denoised the biomedical PSG signals, setting a solid foundation for future research and advancements in sleep analysis.

Chapter 5:

Standards Adopted

5.1 Design Standards

Throughout the development of the project, globally recognized design practices were adhered to for ensuring effectiveness and efficiency. The adopted design standards include:

- Managed large volumes of data through **chunk-based file operations**, optimizing memory usage.
- Generated consistent timestamps using the pandas.to_datetime method, maintaining temporal accuracy.
- Implemented **uniform naming conventions** for dataset variables, utilizing intuitive labels such as elapsed_time_sec.
- Integrated **robust error handling techniques** during file manipulations to promote system resilience.
- Structured the data workflow based on **ETL (Extract, Transform, Load)** models for streamlined processing and organization.

5.2 Coding Standards

The project followed established coding practices to ensure clarity, maintainability, and reliability:

- **Task Segmentation**: Divided operations logically, such as file handling, data parsing, and timestamp management.
- **Meaningful Variable Naming**: Adopted descriptive names like dataset_dir and chunk_size to enhance code clarity.
- **Adherence to PEP 8**: Maintained consistent 4-space indentation and formatting standards as prescribed.
- **Structured Error Handling**: Incorporated try-except blocks to gracefully manage unexpected runtime errors.
- **Memory-Efficient Processing:** Utilized chunked reading techniques (chunksize=1_000_000) to handle large datasets effectively.
- **Focused Simplicity**: Designed each code block to fulfill a singular, clear objective, promoting easy understanding and maintenance.

All coding procedures were executed in strict compliance with the **PEP 8 (Python Enhancement Proposal 8)** standards.

5.3 Testing Standards

Testing protocols followed during the project included:

- **Timestamp and Data Consistency Verification**: Ensured that generated timestamps and dataset structures were accurate post-processing.
- **Systematic Error Logging**: Recorded and tracked errors during the data loading and transformation phases.
- **Manual Output Inspection**: Validated the integrity and structure of the outputs through direct inspection.
- **Graceful Handling of Incomplete Data**: Identified and omitted files with missing or corrupt fields to maintain data quality.

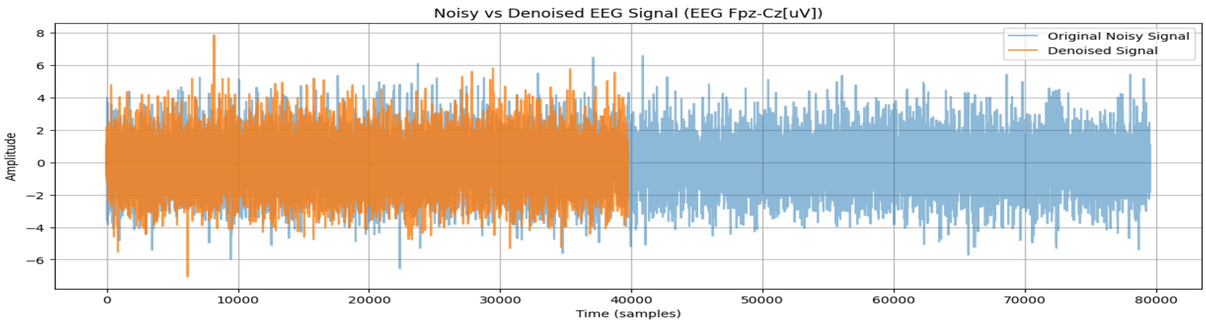
Testing activities were carried out in accordance with the **IEEE 829 Software Test Documentation Standard**, emphasizing thorough validation and systematic error management.

Chapter 6

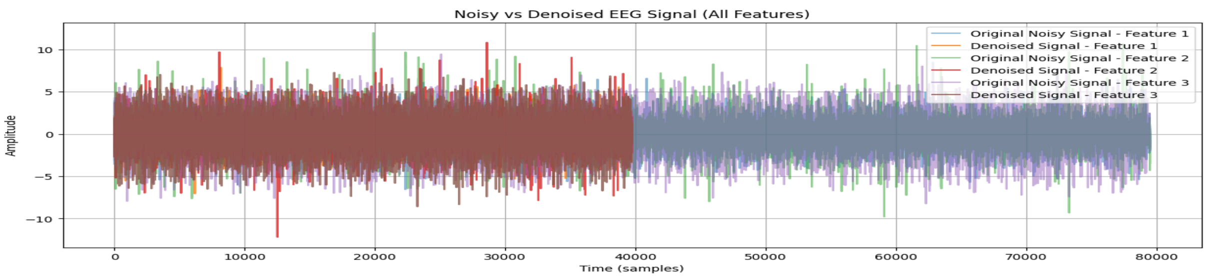
Conclusion and Future Scope

6.1 Conclusion

- The project aimed to develop an efficient system for preprocessing extensive EEG and EOG signal datasets.
- Chunk-based data reading techniques were employed to handle large files without overloading memory resources.
- Accurate timestamps were created using pandas.to_datetime, maintaining proper synchronization across all data entries.
- A robust error-handling mechanism ensured that corrupted or incomplete files were identified and skipped without interrupting the workflow.
- Final outputs, including structured dataframes and graphical visualizations, validated the successful transformation and standardization of raw signals.
- The computed values at the end of processing further confirmed the effectiveness and precision of the system.
- Overall, the project successfully met its objectives by building a scalable and reliable preprocessing pipeline for biomedical signals.



DENOISED SAMPLES OF EEG (Fpz-Cz[uV])



DENOISED SAMPLES OF ENTIRE PSG SIGNAL

```
Mean Squared Error (MSE) on Test Set: 0.000002
Peak Signal-to-Noise Ratio (PSNR) on Test Set: 58.642087 dB
Accuracy on Test Set: 0.999959
Precision on Test Set: 0.999959
Recall on Test Set: 0.999959
```

The above test results are on the basis of the noised input and denoised output.

6.2 Future Scope

6.2 Future Scope

- **Automation and Cloud Integration**: Enhance the pipeline with full automation and migrate processing to cloud platforms for greater scalability.
- **Feature Extraction**: Implement advanced feature extraction methods, such as frequency-domain analysis or statistical summarization, to support machine learning tasks.
- **Real-time Processing**: Modify the system to handle real-time streaming EEG/EOG data for live analysis applications.
- **Interactive Visualization**: Develop dynamic, user-friendly dashboards to facilitate deeper exploration and understanding of the processed data.
- **Integration with AI Models**: Use the cleaned datasets for training AI models aimed at emotion recognition, sleep study analysis, cognitive load assessment, and healthcare diagnostics.
- **Enhanced Error Management**: Introduce detailed error reporting and logging frameworks to further improve system robustness during large-scale deployments.

References

[1]S. Roy, S. Kiral-Kornek, and J. Harrer,
"Deep learning detecting epileptic seizures with interpretable features",
IEEE Transactions on Biomedical Engineering, vol. 66, no. 11, pp. 3104-3113, Nov. 2019..

[2] F. Lopez, A. Vellido,
"Deep Autoencoders for Robust Feature Extraction in Sleep EEG Signal Classification",
Applied Sciences, 10(22), 2020.
[3]A. Faust, Y. Yu, T. Bremer, et al., "Deep Learning for Health Informatics",
IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 1, pp. 4-21, Jan. 2017.

[4]S. K. Bashar, S. M. A. Iqbal, R. Islam, et al.,"Sleep stage classification with bidirectional LSTM and convolutional neural network",
IEEE Access, 7, pp. 155123-155135, 2019.
[5]"Sleep stage classification using attention-based deep learning models",
arXiv preprint arXiv:1908.06387, 2019.
[6]K. Amin, R. Kim,"Noise Robust EEG Signal Denoising with Deep Learning Autoencoders", *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2020.
[7] M. K. Hasan, M. A. Mollah, M. H. Kabir, "BiLSTM Based Deep Learning Model for EEG Signal Denoising"
[8] L. Zhu, J. Cheng, C. Yang, "A Bidirectional LSTM Approach for Sleep Stage Classification Based on EEG Data",*Computational and Mathematical Methods in Medicine*, vol. 2021, Article ID 6631918, 2021.
[9] M. G. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, H. Greenspan,
"Synthetic Data Augmentation Using GAN for Improved Liver Lesion Classification",
IEEE Transactions on Medical Imaging, vol. 38, no. 3, pp. 677-685, 2019.

DENOISING OF POLYSOMNOGRPAHIC SIGNALS USING RNN(Bi-LSTM)

Ritankar Bhattacharya
2205920

Abstract: This project focused on denoising EEG and EOG signals from PSG recordings using the Alice Sleepware system. Key steps included EDF file conversion, preprocessing, and training a Bi-LSTM autoencoder to reconstruct clean signals for reliable sleep analysis.

Individual contribution and findings: I contributed to developing the denoising model using a Bi-LSTM autoencoder, designing the encoder-decoder structure to reconstruct clean EEG and EOG signals. I also worked on applying a Butterworth bandpass filter to remove baseline drift and high-frequency noise during preprocessing. This helped improve signal quality before model training and ensured effective denoising.

Individual contribution to project report preparation: I contributed to writing the sections related to model development, including the architecture details and signal filtering methodology.

Individual contribution for project presentation and demonstration: I presented the model structure and explained how the Bi-LSTM autoencoder helped in signal reconstruction during the demonstration.

Full Signature of Supervisor: ……………………………

Full signature of the student: ……………………………..