Project Report: DeepMedico™ Health Sensing

Title: Breathing Irregularity Classification using Physiological Signals and Deep Learning

---

Overview

The goal of this project is to design a pipeline that can detect breathing irregularities (Hypopnea, Obstructive Apnea) from overnight sleep data collected from five participants using nasal airflow, thoracic movement, and SpO_2 signals. The project is divided into 4 main phases:

1. Data Understanding and Visualization
2. Data Cleaning
3. Dataset Creation
4. Modeling and Evaluation

Data Understanding & Visualization

Input:

For each participant (AP01 to AP05), we received:

- nasal_airflow.txt (32 Hz)
- thoracic_movement.txt (32 Hz)
- spo2.txt (4 Hz)
- flow_events.txt (annotated apnea/hypopnea events)
- sleep_profile.txt (optional)

Task:

- Plot full 8-hour time-series for all three signals
- Overlay flow events for visual reference
- Export plots as PDF per participant

Output:

- Implemented in vis.py
- Uses timestamps to align 32 Hz and 4 Hz signals via pandas datetime index
- Each visualization overlays apnea/hypopnea events as colored bands
- Generated 5 PDFs in Visualizations/:
    - AP01.pdf, ..., AP05.pdf

Sample Output:

- Nasal Airflow with hypopnea events shaded in blue
- Thoracic Movement overlaid with apnea events in red
- SpO_2 dips visibly coincide with apnea

Data Cleaning

Problem:

High-frequency artifacts due to movement distort the signals.

Task:

Filter out high-frequency noise beyond expected human breathing frequency (0.17 Hz to 0.4 Hz)

Method:

- Used **4th-order Butterworth bandpass filter** with SciPy
- Applied to nasal_airflow and thoracic_movement signals
- SpO_2 was low-frequency, so only minimal smoothing used
- Cleaned signals saved as:

Cleaned/AP01/nasal_cleaned.csv

Cleaned/AP01/thoracic_cleaned.csv

Cleaned/AP01/spo2_cleaned.csv

Justification:

Butterworth filter provides smooth response and minimal distortion in passband. It is effective for breathing-related frequency isolation.

Dataset Creation

Task:

Convert continuous 8-hour signals into overlapping 30-second windows for model training.

Method:

- Window Size: 30 seconds = 960 samples (32 Hz signals), 120 samples (4 Hz SpO_2)
- Overlap: 50% = 15 seconds
- For each window:
  - If overlap > 50% with flow event → assign corresponding label
  - Else → assign Normal

Implementation:

- Script: create_dataset.py
- Input: Cleaned signals + flow events
- Output: breathing_dataset.csv in Dataset/ folder

Format Justification:

CSV was chosen:

- Easy to debug and inspect
- Compatible with modeling tools like TensorFlow and PyTorch
- Tabular structure fits window-based data

Sample CSV:

participant	nasal	thoracic	spo2	label
AP01	[...]	[...]	[...]	Normal
AP01	[...]	[...]	[...]	Hypopnea

Modeling

Goal:

Train models to classify windows as Normal, Hypopnea, or Obstructive Apnea.

Models Implemented:

- cnn_model.py → 1D CNN

- conv_lstm_model.py → 1D Conv + LSTM

Evaluation Strategy:

Leave-One-Participant-Out Cross-Validation (LOPO-CV)

Fold	Train on	Test on
1	AP02–AP05	AP01
2	AP01, AP03–AP05	AP02
...	...	...

LOPO prevents **data leakage** since sleep signals are highly person-specific.

Results

CNN Model Summary:

Fold-wise Accuracy (AP01–AP05): 93%, 91%, 97%, 90%, 80%
Aggregated Metrics:

Class	Precision	Recall	F1-score
Normal	0.91	0.99	0.95
Hypopnea	0.05	0.01	0.01
Obstructive Apnea	0.00	0.00	0.00

- Confusion matrix showed good detection for Normal, poor for apnea/hypopnea
- Accuracy inflated due to class imbalance

Conv-LSTM Model Summary:

Fold-wise Accuracy (AP01–AP05): 95%, 91%, 99%, 91%, 80%
Aggregated Metrics:

Class	Precision	Recall	F1-score
Normal	0.91	1.00	0.95
Hypopnea	0.00	0.00	0.00
Obstructive Apnea	0.00	0.00	0.00

- Slight improvement over CNN in Normal class
- No detection of minority classes

Metric Breakdown (Avg ± Std):

Label	Accuracy	Precision	Recall	Sensitivity	Specificity
Normal	0.91 ± 0.06	0.91 ± 0.06	0.99 ± 0.01	0.99 ± 0.01	0.02 ± 0.02
Hypopnea	0.93 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
Obstructive Apnea	0.98 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00

Key Insights

Why not use random 80-20 split?

- **Risk of data leakage**: Same participant in both train/test can bias model
- **Physiological signals are highly personal**: Need person-independent validation

Class Imbalance Issue

- Normal class dominates (>90% windows)
- Hypopnea and apnea barely detected
- Need techniques like oversampling, weighted loss, or focal loss

---

Conclusion

This project successfully built a full pipeline for detecting breathing irregularities from raw physiological signals:

- Signals were visualized and cleaned
- Dataset was windowed and labeled using flow annotations
- Deep learning models were trained and evaluated with LOPO-CV

Though Normal detection was strong, both CNN and Conv-LSTM failed to detect minority classes. This highlights the need for advanced imbalance-handling methods in future iterations.