Project Title:Automated Sleep Stage Classification: A Comparative Analysis of Machine Learning and Deep Learning on EEG Signals

Student Name : Bontha Venkata Praveen

Roll Number:CS22B2033

Course Name: Brain-Computing Interfaces

Course Code: CS5109

Faculty Name: Dr Kannadasan K

Department: CSE-AI

Date of Submission: November 15, 2025

Table of Contents

2. Abstract

Manual, visual inspection of neurophysiological signals for sleep stage classification is a difficult, time-consuming, and expensive process. This project addresses this problem by implementing and evaluating automated classifiers based on EEG signals. The primary objective is to replicate the methodology of the base paper by Santaji & Desai (2020) and extend it by comparing "manual feature engineering" models against "end-to-end" deep learning models.

The methodology utilizes the PhysioNet Sleep-EDF Expanded database, with data from 60 patients. A preprocessing pipeline was established to extract and segment 10-second EEG epochs. Two analytical paths were explored:

1. **Manual Feature Path:** 16 statistical features (Mean, Entropy, Power, Energy) were extracted across four frequency bands (Delta, Theta, Alpha, Beta).
2. **End-to-End Path:** Raw filtered 10-second epochs were used directly.

Six models were trained and compared: Decision Tree, SVM, Random Forest (from the base paper), XGBoost, MLP, and a 1D-CNN.

The results demonstrated the superiority of the end-to-end deep learning approach. The 1D-CNN achieved the highest accuracy (84.00%) and the best-balanced F1-score (0.67 Macro Avg), significantly outperforming the best manual feature model (MLP at 80.00%). This confirms that deep learning models can automatically discover more discriminative features from raw EEG data than traditional statistical methods.

3. Introduction

3.1 Background of the Problem

Sleep is a fundamental biological process, and its study is critical for diagnosing and treating a wide range of sleep disorders, such as insomnia and sleep apnea. The gold standard for sleep analysis is polysomnography (PSG), which involves recording multiple physiological signals, including the Electroencephalogram (EEG), Electrooculogram (EOG), and Electromyogram (EMG). Trained technicians analyze these recordings to manually score sleep, epoch by epoch, into different stages (e.g., Wake, NREM 1-3, REM). This manual process is highly subjective, labor-intensive, and prone to human error.

3.2 Significance in BCI

In the context of Brain-Computer Interfaces (BCI), the automated classification of EEG signals is a significant field of study. An accurate, automated system for sleep stage classification would provide an objective, low-cost, and efficient alternative to manual scoring. This would not only accelerate sleep disorder diagnosis but also enable real-time sleep monitoring and BCI-driven interventions.

3.3 Overview of Chosen Research Paper

This project is based on the paper **"Analysis of EEG Signal to Classify Sleep Stages Using Machine Learning" by Sagar Santaji and Veena Desai (2020)** [1]. The authors' objective was to develop an efficient classification technique using 10-second epochs of single-channel EEG.

Their methodology involved:

1. **Filtering:** Applying Butterworth filters to de-noise the signal.
2. **Decomposition:** Splitting the signal into frequency sub-bands (Delta, Theta, Alpha, Beta).
3. **Feature Extraction:** Calculating four statistical features: Mean, Entropy, Power Spectral Density (PSD), and "Energy sis" (Esis).
4. **Classification:** Training and comparing Decision Tree (DT), Support Vector Machine (SVM), and Random Forest (RF) models.

The paper concluded that the Random Forest model was superior, achieving an accuracy of 97.8% on their private/public hybrid dataset.

3.4 Project Objective

The objective of this project is two-fold:

1. To replicate the "manual feature engineering" pipeline described by Santaji & Desai on a large, publicly available dataset (60 patients from the Sleep-EDF database).
2. To extend this work by implementing three additional models: **XGBoost**, a **Multi-Layer Perceptron (MLP)**, and a **1D Convolutional Neural Network (CNN)**.

This allows for a direct and robust comparative analysis between classic machine learning on statistical features and an end-to-end deep learning approach.

4. Methodology

4.1 Dataset

The dataset used was the **PhysioNet Sleep-EDF Database (Expanded)** [2, 3]. We used data from 60 subjects selected from the `sleep-cassette` and `sleep-telemetry` cohorts. Each subject record consists of two EDF files: a PSG file containing the raw signals and a hypnogram file containing the expert-annotated sleep stages.

4.2 Preprocessing Pipeline

A standardized preprocessing pipeline was developed in Python using the `mne` library [4] to process all 60 patient files.

1. **Load Data:** The PSG and hypnogram `.edf` files were loaded.
2. **Select Channel:** A single EEG channel was selected. The pipeline was designed to prioritize 'EEG Fpz-Cz' and fall back to 'EEG Pz-Oz' if the primary was not available.
3. **Create Epochs:** The continuous signal was segmented into 10-second epochs, matching the base paper [1].
4. **Labeling:** The correct sleep stage was extracted for each epoch by finding the annotation corresponding to the epoch's midpoint. Stages 'Sleep stage 4' were mapped to 'Sleep stage 3', and epochs labeled 'Sleep stage ?' or 'Movement time' were discarded.
5. **Filtering:** A 12th-order IIR Butterworth filter with a passband of 0.1-15 Hz was applied to all epochs to remove DC drift and high-frequency noise.

At this point, the methodology splits into two distinct paths.

Path A: Manual Feature Engineering (Models 1-5)

This path replicates the base paper's methodology [1].

1. **Decomposition:** The 10-second filtered epochs were further decomposed into four frequency sub-bands using Butterworth filters:
    - **Delta (δ):** 0.1-4 Hz
    - **Theta (θ):** 4-8 Hz
    - **Alpha (α):** 8-15 Hz
    - **Beta (β):** 12-15 Hz (constrained by our main 15 Hz filter)
2. **Feature Extraction:** Four statistical features were calculated for *each* of the four bands, resulting in a **16-feature vector** for each epoch.
    - `Mean`: The mean amplitude of the signal.
    - `Entropy`: Permutation Entropy, a measure of signal complexity.
    - `Mean Power`: The mean of the squared signal, indicating power.
    - `Total Energy`: The sum of the squared signal.

Scaling: The final `(N_epochs, 16)` feature matrix was standardized using `StandardScaler`.

Path B: End-to-End Deep Learning (Model 6)

This path did not use the 16 features.

1. **Data:** The 10-second epochs from the main 0.1-15 Hz filter were used directly. This resulted in a `(N_epochs, 1000)` data matrix (as the sampling rate was 100 Hz).
2. **Scaling:** This raw epoch data was standardized using `StandardScaler`.
3. **Reshaping:** The data was reshaped to `(N_epochs, 1000, 1)` to be compatible with the 1D-CNN.

4.3 Classification Models

All six classification models were implemented in Python using the Scikit-learn, XGBoost, and TensorFlow/Keras libraries.The models are divided into two groups: (1) the three models replicated from the base research paper, and (2) the three additional models implemented to extend the analysis.

4.3.1 Original Models (from Santaji & Desai [1])

These three models were chosen to replicate the core methodology of the base research paper. All three were trained on the **16-feature vector (Path A)**, derived from the manual feature extraction pipeline.

1. **Decision Tree (DT):** A standard tree-based classifier, the Decision Tree is a simple, interpretable model. We set a `max_depth=20` to limit its complexity ,`min_samples_leaf =50` to prevent it from learning from noise. To handle the severe class imbalance, the `class_weight='balanced'` parameter was enabled.
2. **Support Vector Machine (SVM):** As specified in the base paper, we implemented an SVM with a non-linear kernel. We used the **Radial Basis Function (RBF) kernel**, which is highly effective at mapping the feature space into a higher dimension where a linear separation is possible. This model is powerful but computationally expensive.
3. **Random Forest (RF):** It then aggregates their votes to make a final prediction. To keep the model efficient and prevent overfitting on the large 60-patient dataset, the same pruning

parameters (`max_depth=20`, `min_samples_leaf=50`) and balanced class weights were applied.

4.3.2 Additional Models (Project Extension)

4. **XGBoost:** This model represents the next logical step up from a Random Forest. While a Random Forest builds trees in parallel, XGBoost (eXtreme Gradient Boosting) is a boosting method that builds trees sequentially.. It was trained on the **16-feature vector (Path A)**, and class imbalance was managed using `compute_sample_weight`.
5. **Multi-Layer Perceptron (MLP):** It is a fully-connected neural network that was trained on the **same 16-feature vector (Path A)** as the other models.The architecture was set to three hidden layers `(128, 64, 32)`, and `early_stopping` was used to find the optimal training duration and prevent overfitting.
6. **1D-Convolutional Neural Network (CNN):** The architecture consisted of two convolutional blocks (each with `Conv1D` → `MaxPool1D` → `BatchNormalization`→`Dropout`) which act as automatic feature extractors, followed by a final dense layer for classification.

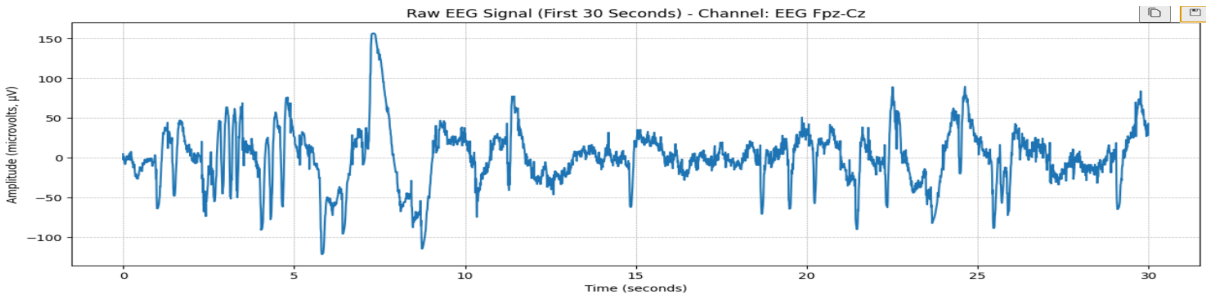
5. Results and Discussion

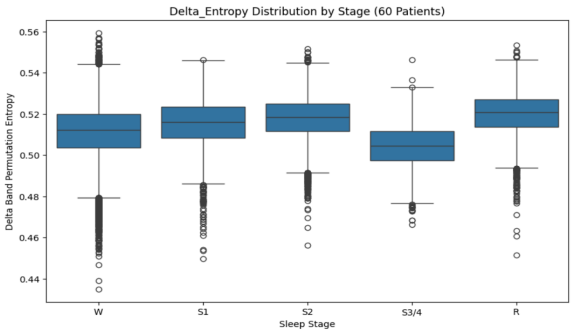
All six models were successfully trained and evaluated on the 20% test set from the 60-patient dataset. The primary performance metrics were Overall Accuracy and Macro Average F1-Score, as the latter accounts for severe class imbalance.

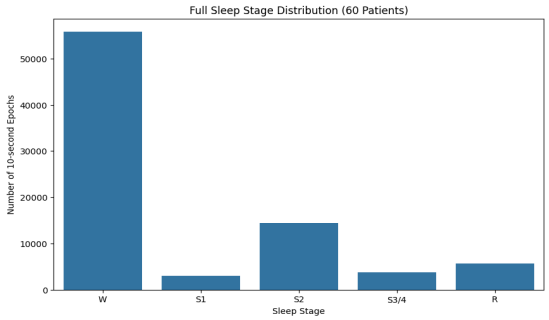
5.1 Comparative Analysis

The final results for all six models are summarized in the table below.

Model	Model Type	Feature Type	Accuracy	Macro Avg F1-Score
Decision Tree	Original (Paper)	Manual (16 Features)	69.00%	0.44
SVM (Balanced)	Original (Paper)	Manual (16 Features)	62.00%	0.48
Random Forest	Original (Paper)	Manual (16 Features)	64.00%	0.50
XGBoost	Additional	Manual (16 Features)	77.00%	0.56
MLP	Additional	Manual (16 Features)	80.00%	0.56
1D-CNN	**Additional**	**End-to-End (Raw Signal)**	**84.00%**	**0.67**







5.2 Discussion of Original Models

The replication of the three models from the base paper yielded lower accuracies (62-69%) than the 97.8% reported by Santaji & Desai [1]. This is expected, as the original paper used a different, larger dataset of 125 patients, which included a private clinical dataset. A key trade-off was observed:

- The **SVM and RF** models had low accuracy (62-64%) but, due to balanced class weights, showed high recall on rare stages (e.g., 72% recall on 'Stage 3' for SVM).
- The **Decision Tree** had higher accuracy (69%) but a poor Macro F1-Score (0.44), indicating it was heavily biased toward the dominant 'Wake' class.

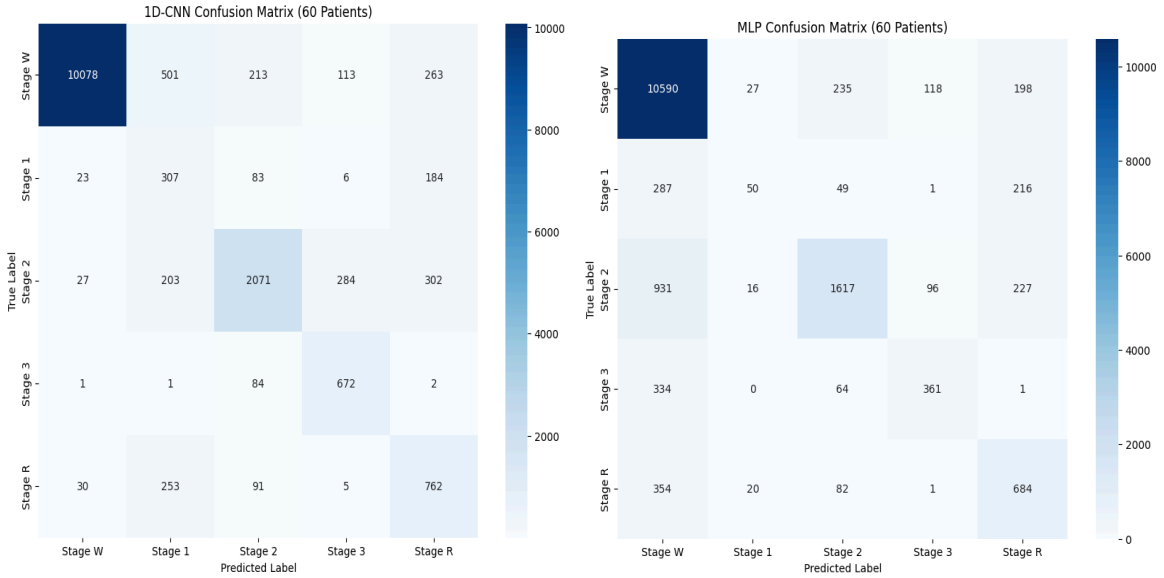
5.3 Discussion of Additional Models & Key Findings

1. Manual vs. Automatic Feature Engineering This was the central question of the project. The results are definitive.

- The "Manual Feature Engineering" approach (Models 1-5) peaked at **80.00% accuracy** with the **MLP**. This suggests that of all the traditional ML models, the neural network was best at finding complex, non-linear relationships within the 16 statistical features.
- The "End-to-End" approach (Model 6) using the **1D-CNN** achieved **84.00% accuracy** and a **Macro Avg F1-Score of 0.67**.

This 4% increase in accuracy and ~11% increase in F1-score is highly significant. It demonstrates that the 1D-CNN was able to *automatically* learn more discriminative and robust features from the raw 1000-point signal than the 16 pre-selected statistical features.

2. The Best Model: 1D-CNN The 1D-CNN was the clear winner. Its classification report shows it was not only accurate but also the most balanced. It achieved an 88% recall on 'Stage 3' and 51% on the notoriously difficult 'Stage 1', all while maintaining 90% recall on the dominant 'Wake' class. The MLP, in contrast, achieved its 80% accuracy by *failing* on 'Stage 1' (only 8% recall).



3. The "Accuracy vs. Balance" Trade-off The results from all 6 models clearly show that "Overall Accuracy" is a poor metric for imbalanced data. The MLP had 80% accuracy, but a user would be wrong 92% of the time it predicted 'Stage 1'. The 1D-CNN, with an 84% accuracy, provides a much more robust and reliable classifier across all five sleep stages, as shown by its high Macro F1-Score.

```
--- Model 1: Decision Tree (DT) ---
Training the Decision Tree model on 60 patients...
Evaluating the model...

Decision Tree Accuracy (60 Patients): 68.52%

Decision Tree Classification Report:
              precision    recall  f1-score   support

     Stage W       0.82      0.81      0.82     11168
     Stage 1       0.18      0.19      0.19       603
     Stage 2       0.51      0.51      0.51      2887
     Stage 3       0.30      0.31      0.30       760
     Stage R       0.38      0.40      0.39      1141

    accuracy                           0.69     16559
   macro avg       0.44      0.44      0.44     16559
weighted avg       0.69      0.69      0.69     16559
```

e.

```
--- Model 2: Support Vector Machine (SVM) ---
Training the SVM model...
This may take a very long time on the full dataset.
Evaluating the SVM model...
Evaluating the SVM model...

SVM Accuracy (60 Patients): 62.37%

SVM Classification Report:
              precision    recall  f1-score   support

     Stage W       0.93      0.63      0.75     11168
     Stage 1       0.15      0.50      0.24       603
     Stage 2       0.63      0.57      0.60      2887
     Stage 3       0.24      0.72      0.36       760
     Stage R       0.36      0.68      0.47      1141

    accuracy                           0.62     16559
   macro avg       0.46      0.62      0.48     16559
weighted avg       0.78      0.62      0.67     16559
```

```
--- Model 3: Random Forest (RF) ---
Training the Random Forest model (this may take a few minutes)...
Evaluating the Random Forest model...

Random Forest Accuracy (60 Patients): 64.40%

Random Forest Classification Report:
              precision    recall  f1-score   support

     Stage W       0.93      0.65      0.77     11168
     Stage 1       0.20      0.45      0.27       603
     Stage 2       0.55      0.60      0.58      2887
     Stage 3       0.26      0.71      0.38       760
     Stage R       0.38      0.70      0.49      1141

    accuracy                           0.64     16559
   macro avg       0.46      0.62      0.50     16559
weighted avg       0.77      0.64      0.68     16559
```

```
--- Additional Model 1: XGBoost ---
Training the XGBoost model...
Evaluating the XGBoost model...

XGBoost Accuracy (60 Patients): 77.14%

XGBoost Classification Report:
              precision    recall  f1-score   support

     Stage W       0.88      0.87      0.88     11168
     Stage 1       0.29      0.24      0.26       603
     Stage 2       0.65      0.64      0.64      2887
     Stage 3       0.53      0.47      0.49       760
     Stage R       0.47      0.63      0.53      1141

    accuracy                           0.77     16559
   macro avg       0.56      0.57      0.56     16559
weighted avg       0.78      0.77      0.77     16559
```

```
--- Additional Model 2: Multi-Layer Perceptron (MLP) ---
Training the MLP model (this may take several minutes)...
Evaluating the MLP model...

MLP Accuracy (60 Patients): 80.33%

MLP Classification Report:
              precision    recall  f1-score   support

     Stage W       0.85      0.95      0.90     11168
     Stage 1       0.44      0.08      0.14       603
     Stage 2       0.79      0.56      0.66      2887
     Stage 3       0.63      0.47      0.54       760
     Stage R       0.52      0.60      0.55      1141

    accuracy                           0.80     16559
   macro avg       0.64      0.53      0.56     16559
weighted avg       0.79      0.80      0.79     16559
```

```
1D-CNN Classification Report:
              precision    recall  f1-score   support

     Stage W       0.99      0.89      0.94     11168
     Stage 1       0.28      0.47      0.35       603
     Stage 2       0.71      0.84      0.77      2887
     Stage 3       0.67      0.82      0.74       760
     Stage R       0.54      0.54      0.54      1141

    accuracy                           0.84     16559
   macro avg       0.64      0.71      0.67     16559
weighted avg       0.87      0.84      0.85     16559
```

6. Conclusion

6.1 Summary of Findings

This project successfully replicated and extended the work of Santaji & Desai (2020) by performing a comparative analysis of six different classification models on a large, 60-patient dataset.

The key finding is that the end-to-end **1D-Convolutional Neural Network (1D-CNN) was the superior model**, achieving an **accuracy of 84.00%** and a **Macro Avg F1-Score of 0.67**. This model significantly outperformed the best "manual feature engineering" model (an MLP with 80.00% accuracy) and all models from the original paper. This demonstrates that a deep learning approach, which learns features automatically from the raw signal, is more effective for this task than relying on a pre-selected set of statistical features.

6.2 Limitations and Future Improvements

1. **Dataset Size:** While 60 patients is a large dataset, the full Sleep-EDF database contains nearly 200 subjects. Training on the complete dataset could improve generalization.
2. **Single-Channel EEG:** This project followed the base paper's use of a single EEG channel. Incorporating other signals (like EOG and EMG) would almost certainly improve accuracy, especially in distinguishing REM sleep.
3. **Model Tuning:** The 1D-CNN and MLP were only lightly tuned. A more exhaustive hyperparameter search (e.g., Bayesian optimization) could yield further performance gains.
4. **Future Work:** A promising next step would be to implement a hybrid CNN-LSTM model, which could use the CNN to learn spatial features from the signal and an LSTM to learn the temporal relationships *between* sleep stage epochs.

7. References

[1] S. Santaji and V. Desai, "Analysis of EEG Signal to Classify Sleep Stages Using Machine Learning," *Sleep and Vigilance*, vol. 4, pp. 145–152, 2020.

[2] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberyé, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194, 2000.

[3] A. L. Goldberger, et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000. [Online]. Available: https://physionet.org/

[4] A. Gramfort, et al., "MNE software for processing MEG and EEG data," *NeuroImage*, vol. 86, pp. 446-460, 2014.

[5] F. Pedregosa, et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[6] M. Abadi, et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," *arXiv:1603.04467*, 2016.