MACHINE LEARNING-BASED CLASSIFIERS FOR OBSTRUCTIVE SLEEP APNEA DIAGNOSIS USING ECG AND EDR SIGNALS

Johny Moreno Granja
Alexis Ocaciones García

Pontificia Universidad Javeriana
School of Engineering and Sciences – Master's Degree in Engineering
Cali, Colombia
2026

MACHINE LEARNING-BASED CLASSIFIERS FOR OBSTRUCTIVE SLEEP APNEA DIAGNOSIS USING ECG AND EDR SIGNALS

Johny Moreno Granja
Alexis Ocaciones García

Degree thesis presented as a partial requirement to obtain the degree of:
Magister in Engineering

Director:
Luis Eduardo Tobón Llano, PhD
Co-Director:
Laura Jaramillo Otoya, MD

Pontificia Universidad Javeriana
School of Engineering and Sciences- Master's Degree in Engineering
Cali, Colombia
2026

Maestría en Ingeniería

Facultad de Ingeniería y Ciencias

FICHA RESUMEN

TRABAJO DE GRADO DE MAESTRÍA

TITULO: "**MACHINE LEARNING-BASED CLASSIFIERS FOR OBSTRUCTIVE SLEEP APNEA DIAGNOSIS USING ECG AND EDR SIGNALS** "

1. ÉNFASIS: Electrónica y Sistemas y Computación
2. TIPO DE PROYECTO: Investigación
3. ÁREA DE TRABAJO: Machine Learning
4. ESTUDIANTE (S): Johny Moreno Granja y Alexis Ocaciones García
5. CORREO ELECTRÓNICO: johnym@javerianacali.edu.co, aocacion@gmail.com
6. DIRECCIÓN Y TELÉFONO: Carrera 31A 3 9C-114, 3105960142
7. DIRECTOR: Luis Eduardo Tobón Llano
8. VINCULACIÓN DEL DIRECTOR (en la universidad): Planta
9. CORREO ELECTRÓNICO DEL DIRECTOR: letobon@javerianacali.edu.co
10. CO-DIRECTOR(ES): Laura Jaramillo Otoya
11. GRUPO O EMPRESA QUE LO AVALA: Automática y Robótica, GAR
12. OTROS GRUPOS O EMPRESAS:
13. PALABRAS CLAVE: Obstructive Sleep Apnea (OSA), machine learning, deep learning, classifiers, ECG, EDR.
14. ODS QUE APLICA EL PROYECTO (Agenda 2030): Salud y bienestar (3); industria, innovación e infraestructura (9) y reducción de las desigualdades (10)
15. FECHA DE INICIO (Desarrollo del proyecto): 25/10/2025
16. RESUMEN

Obstructive sleep apnea (OSA) is one of the most common sleep-related breathing disorders with serious cardiovascular and neurological implications. It is normally diagnosed in the context of polysomnography (PSG) which is very expensive, cumbersome, and invasive in the process. In this work, we applied machine learning and deep learning techniques to develop an automated ECG method for detecting and assessing the severity of OSA to make diagnosis more accessible and efficient.

We used the PhysioNet Apnea-ECG database and used 10 classification methods: logistic regression, support vector machine with RBF kernel, random forest, CNN and RNN with raw ECG, CNN and RNN models with RR intervals and ECG-derived respiration (RR+EDR) and three ensemble strategies: soft voting, AUC-weighted voting, and stacking. Heart rate variability and EDR features were extracted from one-minute ECG windows with a ±1-minute temporal context. All the methods have subject-independent data partitioning to avoid data leakage and to generate more realistic models.

The models were evaluated internally and independently using the official x01–x35 test set and 35 subjects and 12,600 one-minute windows. Performance was assessed based on precision, sensitivity, specificity, accuracy, F1 score, AUC, and Cohen's kappa coefficient. The patient-level hierarchical

analysis was also conducted, where the minute-level predictions were collected into an Apnea Index and four categories of severity were established: normal, mild, moderate and severe.

The results showed that classical machine learning and ensemble approaches provided the most consistent generalization. The Weighted-AUC set had the best overall external performance with an AUC of 0.939 and F1 score of 0.834. Patient severity estimation also showed promising results with a mean absolute error of the apnea index of 3.01 events per hour and a severity accuracy of around 80% in the external assessment.

Dedication

To my parents, Francisco and Delfina, whose hard work and dedication paved the way for me to pursue my dreams.
To my children, whose light and joy inspire me to guide them on their journey.
To Adriana, for her love and unwavering support, which motivates me to become a better person every day.

Johny Moreno Granja

Acknowledgements

We extend our heartfelt gratitude to the professors of Universidad Javeriana Cali for their invaluable knowledge and guidance. A special thanks to our tutors, **Luis Eduardo Tobón Llano** and **Laura Jaramillo Otoya**, whose expertise, advice, and unwavering support were instrumental in the successful development of this project.

Abstract

Obstructive Sleep Apnea (OSA) is a common sleep-related breathing disorder associated with cardiovascular morbidity, yet it remains underdiagnosed due to the cost, complexity, and intrusiveness of polysomnography, the clinical gold standard. This study developed and evaluated ten classifiers for minute-level OSA detection using the PhysioNet Apnea-ECG database: three classical machine learning models (**Logistic Regression, SVM** with RBF kernel, **Random Forest**), four deep learning models (**CNN** and **RNN** on raw ECG, and their RR+EDR-enhanced variants **CNN_RREDR** and **RNN_RREDR**), and three ensemble strategies (**Soft Voting**, **Weighted-AUC Voting**, **Stacking**). Heart rate variability and ECG-derived respiration (EDR) features were extracted from one-minute windows with a ±1-minute temporal context, yielding 45 input variables.

All evaluation was subject-independent to avoid data leakage. Hyperparameters were selected via subject-stratified 5-fold GroupKFold cross-validation, with **SVM-RBF** achieving the best CV-AUC (0.873), followed by **Random Forest** (0.847) and **Logistic Regression** (0.831). Models were then trained and internally tested on a 75/25 grouped split (26 subjects/9,360 windows for training; 9 subjects/3240 windows for testing) and externally validated on the independent official test set (records x01–x35, 35 subjects, 12600 windows).

Internally, **Random Forest** performed best individually (accuracy 83.6%, AUC 0.927), closely followed by **SVM-RBF** (AUC 0.923); raw-ECG deep learning models underperformed (**CNN** AUC 0.632, **RNN** AUC 0.595) but improved substantially with RR+EDR input (**CNN_RREDR** AUC 0.910, **RNN_RREDR** AUC 0.800). The **Weighted-AUC** ensemble achieved the best overall internal balance (accuracy 84.7%, F1 0.808, $\kappa = 0.681$). External validation confirmed these trends for classical models and ensembles (**Random Forest** AUC 0.933; Weighted-AUC ensemble AUC 0.939, F1 0.834, accuracy 87.7%), while raw-ECG deep learning models degraded sharply (**RNN** AUC 0.581), traced to subject-level leakage in early stopping validation rather than hyperparameter choice.

Statistical robustness was assessed via McNemar's test and subject-grouped bootstrap resampling across 45 pairwise comparisons. McNemar's test found most significant differences (43/45, $p < 0.05$), while the subject-clustered bootstrap showed fewer significant differences, warranting cautious interpretation given the limited subject pool. Results were competitive with prior state-of-the-art studies on this database (77.3%–97.4% accuracy). Key limitations include reliance on a single database, small subject cohort, and limited deep learning architecture optimization.

Keywords: Obstructive Sleep Apnea (OSA), machine learning, deep learning, classifiers, ECG, EDR.

TABLE OF CONTENTS

LIST OF FIGURES

Abbreviations

AASM	**A**merican **A**cademy of **S**leep **M**edicine
AHI	**A**pnea **H**ypopnea **I**ndex
AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etworks
CSA	**C**entral **S**leep **A**pnea
CVD	**C**ardio**v**ascular **D**isease
DL	**D**eep **L**earning
DM	**D**ata **M**ining
DNN	**Deep Neural Networks**
ECG	**E**lectro**c**ardio**g**raphic **S**ignal
EDR	**E**CG-**D**erived **R**espiratory Signal
ML	**M**achine **L**earning
NN	**N**eural **N**etworks
ODI	**O**xygen **D**esaturation **I**ndex
OSA	**O**bstructive **S**leep Apnea
OSAHS	**O**bstructive **S**leep Apnea **H**ypopnea **S**yndrome
PSG	**P**oly**S**omno**G**raphy
SaO2	**O**xygen **S**aturation in arterial blood
SpO2	**O**xygen **S**aturation in capillary (quasi - arterial) blood
SRBD	**S**leep **- R**elated **B**reathing **D**isorders

Ethical Statement on the Use of AI

During the development of this project, artificial intelligence tools were used for code generation and verification, image generation and enhancement, and the exploration of scientific articles and books.

The tools used during the development of this project are:

- Claude Opus 4.7 (paid version): Used for code generation, testing and code documentation.
- ChatGPT 4: Used for developing summaries and reviewing texts.
- Gemini 3.7 Flash: Used primarily for image generation and enhancement.
- NotebookLM: Used for exploring and analyzing papers and books.

As this project aims to diagnose a medical condition, we declare that we have tested and verified the integrity of the code and that we assume responsibility for any errors or inconsistencies that may arise.

1. Introduction

1.1. Problem Description

Obstructive sleep apnea (OSA) is a condition characterized by the complete blockage of the upper airway during sleep, leading to oxygen deprivation (hypoxia) in organs and tissues. Hypopnea, on the other hand, refers to episodes lasting at least 10 seconds, during which the respiration rate decreases by 50% or more. [1]

Sleep – Related Breathing Disorders (SRBD) such as obstructive sleep apnea-hypopnea syndrome (OSAHS) are closely related to cardiovascular disease (CVD) and hypertension. These conditions pose serious health risks, as each apnea or hypopnea episode reduces oxygen supply to the brain and body tissues. The primary concern is that many cases go undiagnosed, significantly increasing the risk of stroke and heart attack. [1] [3] [9] [10] [11]

The diagnosis of OSAHS is typically based on polysomnography, a high-cost clinical study that monitors respiratory, cardiac, pulmonary, and brain activity. However, polysomnography is intrusive for patients and requires substantial resources. [1] [2] [4] [5]

OSA has a low diagnosis rate despite its severe cardiovascular consequences. Combined with the high costs of diagnosis (polysomnography) and treatment (CPAP therapy, CPAP titration, and in some cases, surgery), OSA represents a significant public health issue.

OSA episodes often occur due to the collapse of throat muscles during sleep, leading to upper airway obstruction, as illustrated in Figure 1. Another form of sleep-disordered breathing, central sleep apnea (CSA), occurs when the brain fails to send proper signals to the muscles controlling respiration. Unlike OSA, which results from physical airway obstruction, CSA is a neurological condition and is less common than OSA. [1] [3] [6]

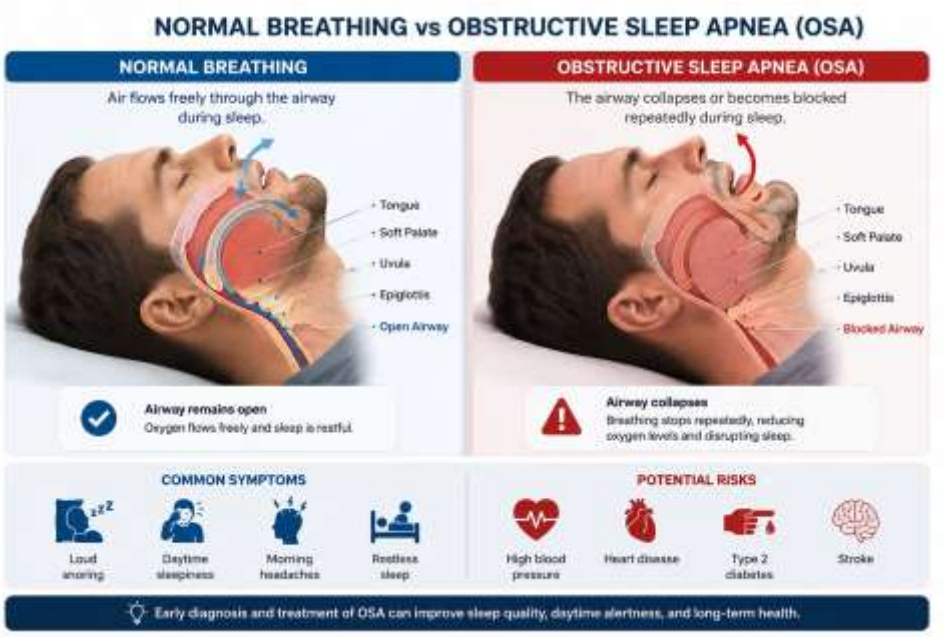

Figure 1. Obstructive sleep apnea mechanism.[1]

Evidence supports a causal link between sleep apnea and the incidence and morbidity of hypertension, coronary heart disease, arrhythmia, heart failure, stroke, and systemic inflammation. [1] [8] [9] [10] [11]

Machine learning (ML) offers a promising approach for diagnosing obstructive sleep apnea (OSA) by accelerating detection, simplifying the process, and making it more accessible.

ML-based OSA diagnosis involves extracting features from sensor data and biophysiological signals, followed by training predictive models to identify the condition. These techniques have the potential to enhance diagnostic efficiency and accuracy, enabling earlier and more effective treatment. However, rigorous validation is essential before clinical implementation. [1] [14]

1.2. Diagnostic Methods

Beyond the cardiovascular and neurological consequences of obstructive sleep apnea (OSA), its low diagnosis rate and high associated costs remain significant challenges. Traditionally, sleep-related respiratory disorders have been diagnosed in specialized sleep laboratories. However, emerging technologies such as machine learning and deep learning are now being leveraged to automate diagnosis and develop portable diagnostic systems.

[1] What is Obstructive Sleep Apnea? - SleepWorks Medical

1.2.1. Polysomnography

The gold standard for diagnosing OSA is overnight polysomnography (PSG). As illustrated in Figure 2, PSG monitors multiple physiological parameters, including heart, lung, and brain activity, breathing patterns, and blood oxygen levels.
Polysomnography is a complex and resource-intensive procedure, requiring the measurement of various physiological signals and manual data interpretation, which introduces the potential for human error. OSA diagnosis is typically based on the apnea-hypopnea index (AHI), which quantifies the number of apnea or hypopnea events per hour of sleep. Over time, the scoring criteria for hypopneas have evolved, with oxygen desaturation serving as a key factor in defining these events.

Figure 2. PSG patient setup[2]

Sleep disorders like OSA are diagnosed through sleep studies, which often require patients to stay overnight in a hospital setting. A comprehensive sleep study (PSG) records brain wave patterns, breathing patterns, abdominal wall movements, heart rate, and other parameters. These data are then reviewed and analyzed by specialized technicians. In some cases, patients may need to return for an additional night to assess and adjust treatment.

The primary treatment for OSA, alongside sleep hygiene measures, is the regular use of continuous positive airway pressure (CPAP) therapy. CPAP is administered via a CPAP machine with a fitted face mask, ensuring that the airway remains open during sleep.

[2] https://yeshwanthneurologist.com/polysomnography-sleep-study/

Despite the extensive data it provides, PSG is costly, intrusive, and labor-intensive. Moreover, its diagnostic accuracy depends on human expertise, which may introduce variability. [1] [7]
The PSG-derived AHI is the primary metric for assessing OSA severity. It is calculated as the number of apnea and hypopnea events per hour of sleep, as shown in Equation 1.1.

$$AHI = \frac{TNAE + TNHE}{TST} \qquad (1.1)$$

Where:

AHI: Apnea Hypopnea Index.
TNAE: Total Number of Apnea Episodes.
TNHE: Total Number of Hypopnea Episodes.
TST: Total Sleep Time.

Figure 3 presents the Apnea-Hypopnea Index (AHI) classification, which categorizes Obstructive Sleep Apnea (OSA) severity based on the number of apnea and hypopnea events per hour of sleep.

- **Healthy (0–4 AHI):** Normal breathing with no significant sleep disturbances or health risks.
- **Mild OSA (5–14 AHI):** Occasional breathing interruptions; may cause light snoring and mild daytime fatigue. Lifestyle changes or CPAP may be recommended.
- **Moderate OSA (15–29 AHI):** Frequent apneas/hypopneas, leading to noticeable sleep disruption and increased cardiovascular risks. CPAP therapy is typically required.
- **Severe OSA (≥30 AHI):** Persistent airway obstruction leading to significantly decreased oxygen saturation, severe daytime sleepiness, and an increased risk of serious conditions such as hypertension, heart disease, and stroke. CPAP or advanced treatments are essential.

Category	AHI (events per hour)	Description
Normal (Healthy)	0–4	Minimal or no apnea/hypopnea episodes; normal breathing during sleep.

Category	AHI (events per hour)	Description
Normal (Healthy)	0–4	Minimal or no apnea/hypopnea episodes; normal breathing during sleep.

Mild OSA	5–14	Some breathing disturbances; may experience light daytime sleepiness or fatigue.
Moderate OSA	15–29	Frequent apneas/hypopneas; noticeable daytime sleepiness and potential health risks.

Figure 3. OSA severity according to AHI. [1] [4]

OSA severity is also associated with episodes of hypoxia and hypercapnia. In mild cases, oxyhemoglobin saturation may drop to 95%, whereas in severe cases, it can fall below 80%. [2] [3] [10]
Increased hypoxia can contribute to venous hypertension, disrupt metabolic processes, and promote tissue damage.
However, AHI alone does not fully capture the effects of chronic intermittent hypoxia (CIH), which may trigger systemic inflammation in some OSA patients. Total sleep time (TST) has been suggested as a more reliable metric for evaluating CIH. Thus, OSA severity should be assessed using a combination of AHI and other hypoxia-related variables. [2] [9]

Establishing a direct correlation between AHI and OSA's clinical manifestations remains challenging. Some patients with a high AHI exhibit minimal symptoms, while others with a low AHI experience significant clinical effects. This variability may be attributed to individual differences in susceptibility to systemic consequences such as intermittent hypoxia, intrathoracic pressure fluctuations, variations in sympathetic tone, and hemodynamic instability.

Additionally, AHI-based OSA severity classification is influenced by the inclusion of sleep staging in PSG tests. Studies that omit sleep staging, such as cardiorespiratory polygraphy, often yield lower AHI values than PSG-based assessments, where wakefulness periods are excluded from AHI calculations. This discrepancy is particularly relevant when evaluating patients with mild to moderate OSA. However, advancements in electrode placement and automated analysis have enabled the use of PSG in ambulatory recordings. Multi-night studies may offer a more accurate assessment than a single-night study, particularly with the development of edge technology and minimal-contact devices designed for home-based monitoring. These home studies, which do not include sleep staging, provide a more accessible alternative while maintaining diagnostic reliability.
Technical variations, such as differences in airflow measurement methods (e.g., thermistors vs. nasal prongs), can influence the detection of respiratory disturbances. Additionally, inconsistencies in hypopnea definitions may impact severity assessment. The choice of a 3% or 4% oxygen desaturation threshold, with or without arousal, has been shown in a Spanish community-based study to significantly affect OSA prevalence and severity classification. This variation also influences the association between OSA and cardiovascular outcomes.

Oxygen desaturation is generally more pronounced during apnea than hypopnea and becomes more severe with longer event durations. The severity of desaturation events differs between hypopnea and obstructive apnea and is further modulated by their duration in OSA [7].

According to [2], total sleep time (TST) is a strong predictor of chronic intermittent hypoxia (CIH) and an important indicator of the risk of adverse cardiovascular events.
Figure 4 illustrates the prevalence of OSA in cardiovascular disease (CVD) patients, using a lower threshold of AHI ≥ 15.

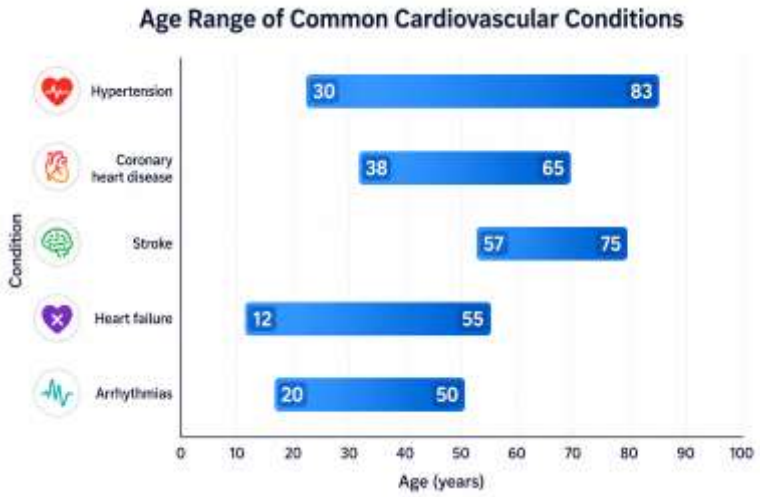

Figure 4. Prevalence (%) of OSA in CVD. [1]

1.2.2. Home Polysomnography

Home sleep studies (HSS) are an economical and easily accessible method for identifying obstructive sleep apnea (OSA). Unlike polysomnography (PSG), which requires an overnight stay in a sleep laboratory, HSS are performed in the patient's home and typically measure the following parameters:

- Respiratory flow: Identification of breathing delays. • Respiratory effort: Movement of the chest and abdomen.

- Oxygen saturation (SpO_2): Blood oxygen level.

Some advanced HSS include more sophisticated features, such as actigraphy to measure sleep-wake patterns and body position monitoring to assess positional sleep apnea. While HSS are easier to use in practice and more convenient, they have limitations, as they do not detect sleep stages, awakenings, or brain activity, since they do not include electroencephalographic (EEG) monitoring. This can lead to an underestimation of the severity of OSA, especially in cases of respiratory events that cause arousals rather than oxygen desaturation.

Home sleep testing: Despite their advantages, home sleep testing (HST) is not suitable for all patients. The American Academy of Sleep Medicine (AASM) advises using HST only for patients with a high pretest probability of moderate to severe OSA. HST should not be used for:

- Patients with comorbid conditions such as heart failure, chronic obstructive pulmonary disease (COPD), neuromuscular disorders, or obesity hypoventilation syndrome, as these conditions are likely to interfere with accurate results.

- Suspected central sleep apnea (CSAD) or any other sleep disorder, as HST cannot distinguish between obstructive and central sleep apnea.

- Pediatric patients because their respiratory events are based on arousals rather than oxygen desaturation, and HST cannot measure this. Furthermore, mild cases of OSA can be misclassified by home sleep testing (HST) due to the lack of clear wakefulness and sleep symptoms, leading to false negatives. To determine if an HST is appropriate for a patient, a structured evaluation is performed:

1.2.2.1. Clinical Evaluation

- We conduct a thorough history regarding risk factors for obstructive sleep apnea (OSA), such as snoring, witnessed apneas, daytime sleepiness, obesity, hypertension, and type 2 diabetes.

- Validated screening tools, such as the Epworth Sleepiness Scale (ESS) or the STOP-BANG questionnaire, can be used to assess the likelihood of OSA.

1.2.2.2. Eligibility Criteria

- If the patient presents a high pretest probability of moderate to severe OSA without significant comorbidities, they are a candidate for a home sleep test (HST).

- If the patient has serious comorbidities or central sleep apnea, a laboratory polysomnography (PSG) is recommended.

1.2.2.3. Setup and instructions for the home sleep test

- The patient is provided with detailed instructions on how to correctly use the PSD device with nasal cannulas, respiratory effort belts, and pulse oximeters.

- The patient is advised to maintain their usual sleep schedule and avoid caffeine, alcohol, or sedatives before the test.

1.2.2.4. Data collection and review

- The PSD device records breathing patterns, oxygen levels, and other key parameters throughout the night.

- A sleep specialist reviews the data and calculates the Apnea-Hypopnea Index (AHI) to classify the severity of obstructive sleep apnea (OSA).

1.2.2.5. Follow-up and interpretation.

- If the results of the home sleep test (HST) are positive, treatment is initiated (e.g., CPAP therapy).

- If the HST results are negative, but symptoms persist, the patient is referred to for polysomnography (PSG) for further evaluation.

Figure 5 depicts the evaluation workflow for the implementation of Home-PSG.

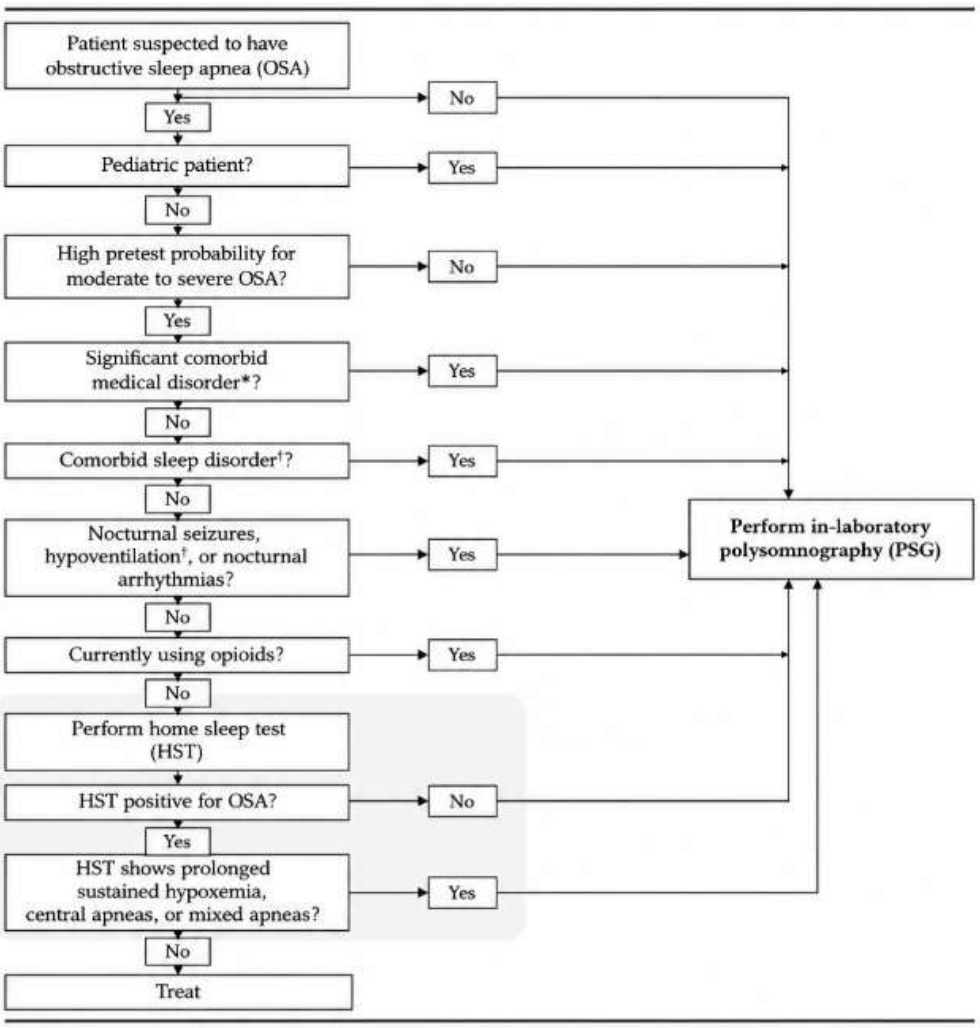

Figure 5. Home – PSG assessment. [12]

Clinical Implications and Recommendations

While polysomnography (PSG) remains the most reliable diagnostic tool, home sleep testing (PSD) can be an effective screening tool for moderate to severe obstructive sleep apnea (OSA) in appropriate patients. However, a negative PSD does not mean that OSA is no longer detectable, and in some cases, further testing is required. For accurate diagnosis and treatment, the following guidelines should be followed:

1. Patients with a high probability of OSA, but without significant comorbidities, can undergo PSD as a primary diagnostic tool.

2. Patients with a negative PSD, but with persistent symptoms, should be referred to as laboratory PSG for confirmation.

3. Patients with complex sleep disorders, suspected central sleep apnea, or serious medical conditions should be evaluated directly with PSG.

4. PSDs are a cost-effective, accessible, and practical method for diagnosing OSA in certain patients. However, because they do not record sleep architecture or respiratory events related to arousals, polysomnography (PSG) is essential for a comprehensive diagnosis. A structured assessment process ensures that home stress tests are administered appropriately and effectively, thereby ensuring timely diagnosis and treatment of obstructive sleep apnea (OSA) and reducing costs and burden for the patient.

2. Objectives

2.1. General Objective

- Implement a set of classifiers to support obstructive sleep apnea syndrome diagnosis.

2.2. Specific Objectives

- Preprocess an obstructive sleep apnea diagnostic dataset containing records from healthy controls and affected patients.
- Train a set of machine learning classifiers using ECG and EDR signals to support obstructive sleep apnea diagnosis.
- Classify obstructive sleep apnea severity using a hierarchical approach.
- Evaluate the performance of the implemented classifiers in terms of accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC).

3. Background

Artificial Intelligence (AI) has revolutionized multiple fields, including healthcare, finance, transportation, and social sciences. One of its most promising applications is in the automated diagnosis of medical conditions, particularly Obstructive Sleep Apnea (OSA). Traditional Polysomnography (PSG), the gold standard for OSA diagnosis, is expensive and resource intensive.

Machine Learning (ML) and Deep Learning (DL) offer alternative methods that can automate diagnosis, making it faster, cost-effective, and more accessible.

This section explores Artificial Intelligence techniques applied to OSA detection, focusing on Machine Learning classifiers, deep learning models, and their performance evaluation.

3.1. Artificial Intelligence and Machine Learning in Healthcare

AI enables computers to mimic human intelligence, analyze complex data, recognize patterns, and make predictions. Machine Learning (ML), a subset of AI, uses data-driven algorithms to improve performance over time without explicit programming. Deep Learning (DL), a more advanced form of ML, utilizes artificial neural networks (ANNs) to model complex patterns in medical data.

In OSA diagnosis, ML and DL techniques analyze electrocardiogram (ECG), oxygen saturation (SpO2), respiratory signals, and other physiological markers to classify apnea severity and improve diagnostic accuracy.

AI is any technique that provides computers the ability of mimic human intelligence. Through AI, computers can analyze images, understand speech, recognize handwriting, guide unmanned vehicles, make medical diagnosis, interact in natural ways, and make predictions using data.

AI is a complex branch of computer science committed to design and build smart machines capable of performing tasks that usually require human intelligence. [38]

According to [39], AI is the study of rational agents that receive percepts from the environment and perform actions. An agent is anything capable of perceiving its environment (accepting input stimuli), processing that input information to act on its environment (providing outputs) and rationality is a human capacity that enables thinking, evaluate and act according to some principles of performance and consistency.

Figure 6 is a comprehensive model of modern AI structure according to [41], where AI is the wider field and machine learning, and deep learning is a specific AI subset.

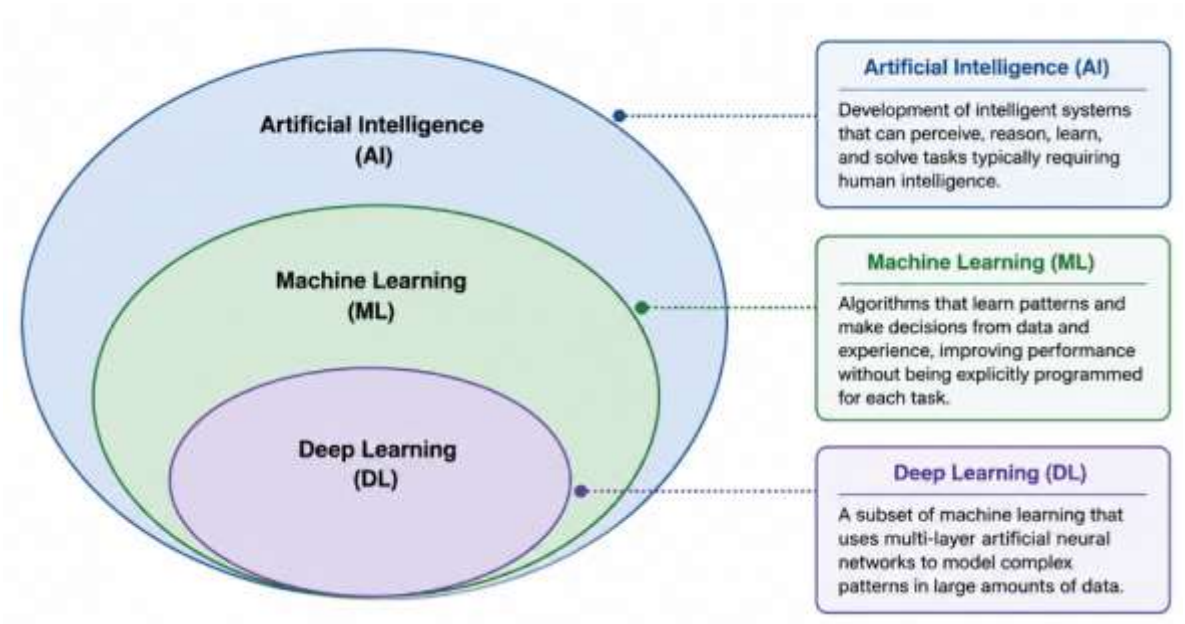

Figure 6. Modern AI structure. [41]

3.2. Machine Learning (ML)

Machine Learning is a subfield of AI involved in the development of self-learning algorithms capable of extracting knowledge from data and making predictions.

Machine learning provides an efficient way to derive rules and create models from large data repositories that allow them to capture the hidden knowledge and gradually improve the performance of predictive models and make data-driven decisions. [40]

ML aims to develop algorithms whose performance improve as they are exposed to more data over time. Figure 7 shows a comparison between traditional programming paradigms and machine learning paradigms. It is inferred that ML systems are trained rather than explicitly programmed, establishing a new programming paradigm in which the rules driving data behavior must be discovered.

Supervised learning is a type of machine learning where the algorithm is trained on a set of labeled data, meaning that each data point is associated with a corresponding label or target value. The objective of supervised learning is to learn a function that maps inputs to outputs, given a set of input-output pairs. Supervised learning includes ***classification*** and ***regression analysis***.

According to [40], the main goal in supervised learning is to learn a model from labeled training data that allows making predictions about unseen or future data. Here, the term **supervised** refers to a set of samples where the desired output signals (labels) are already known.

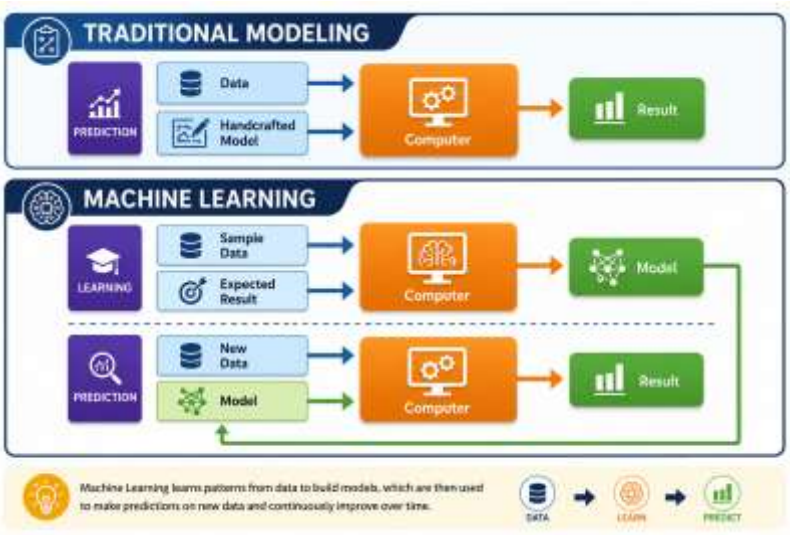

Figure 7. Traditional programming paradigms versus ML paradigm.[3]

Figure 8 shows the supervised learning process and resembles the way humans are taught.

According to [41], ***supervised learning*** consists of learning to map input data to known targets (also called *annotations*), given a set of examples (often annotated by humans).

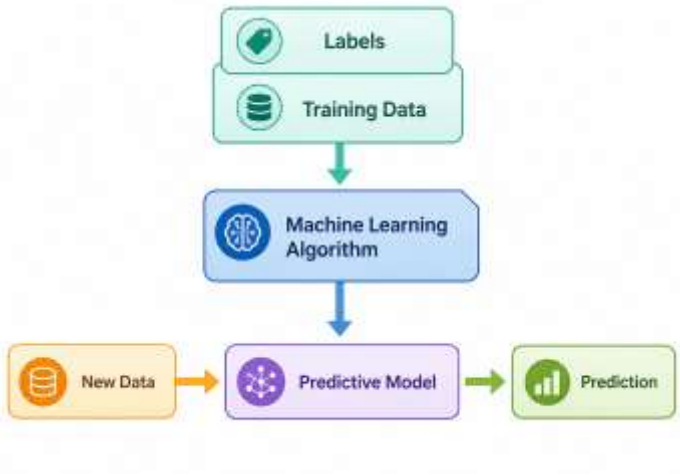

Figure 8. Supervised Learning. [40]

Classification is a supervised learning task where data have discrete class labels, and the machine learning model attempts to predict which category input data belongs to. Classification is a subcategory of supervised learning where the goal is to predict the categorical class labels of new instances, based on past observations. Those class labels

[3] DOI: 10.13140/RG.2.2.23097.80485

are discrete, unordered values that can be understood as the group memberships of the instances. [40]

Classifiers are algorithms used to categorize data into several groups or classes. Classifier algorithms are trained using labeled data, and after appropriate training, it can receive unlabeled data and will assign classification labels for each data point.

Classifier algorithms employ complex mathematical and statistical methods to generate predictions about the likelihood of a data input being classified as member of a given class.

Figure 9 shows a ***binary classifier***, which is the simplest classification model and whose task is to assign data to one of two available classes. Each training point belongs to one of two different classes. The objective is to find a function which, given a new data point, will correctly predict the class the new point belongs to. Some applications of binary classifiers are spam and non-spam emails labeling, healthy and cancerous tissue diagnosis, approved and rejected manufactured items, apnea/non-apnea, etc.

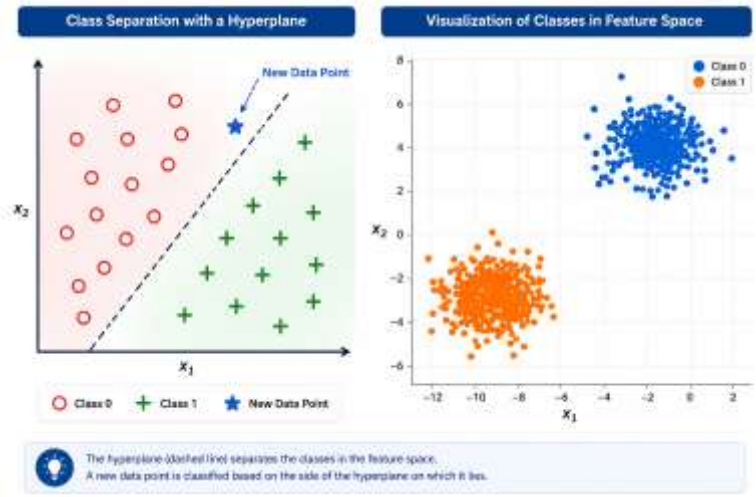

Figure 9. Binary classification. [40]

When the model must decide between three or more classes the ML task is called a ***multiclass classifier*** (Figure 10). Each training point belongs to one of N different classes. The goal is to construct a function which, given a new data point, will correctly predict the class to which the new point belongs to. In multiclass classification, a sample can only have one class (i.e., an elephant is only an elephant; it is not also a panther). Handwriting character recognition and image classification are representative examples of multiclass classification.

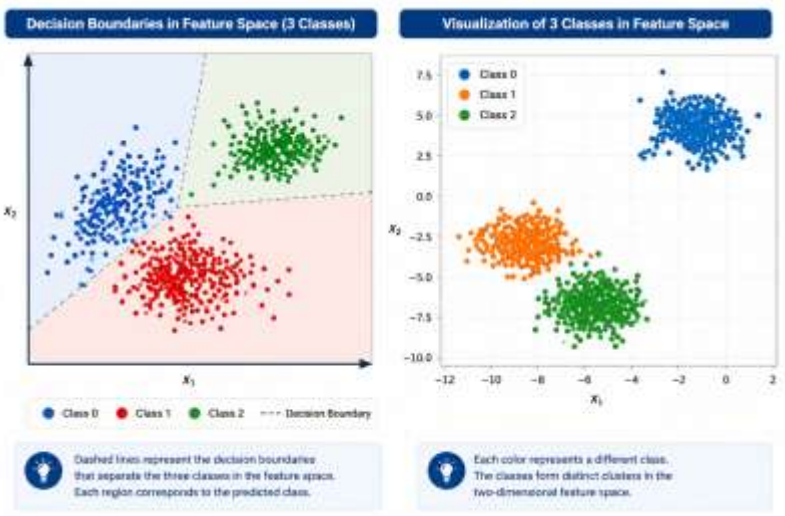

Figure 10. Multiclass classification. [40]

3.2.1. Machine learning classification algorithms

To diagnose obstructive sleep apnea (OSA), various classification algorithms in Machine Learning have been used, which have yielded satisfactory results. These algorithms allow accurate identification of OSA from physiological signal data and features extracted from them. Some of the most effective classification algorithms for diagnosing OSA are:

3.2.1.1. Logistic Regression

Logistic regression is a widely used algorithm in binary classification. Logistic function or sigmoid function, as shown in figure 11, is widely used in binary classification. This logistic model is used to estimate the probability of a binary response based on one or more independent variables. It allows us to say that the presence of a specific factor increases the probability of a given result by a specific percentage.

Like all regression analyses, logistic regression is a predictive analysis. It is used to describe data and explain the relationship between a binary dependent variable and one or more nominal, ordinal, interval, or ratio-level independent variables.

In general, this algorithm can be used for various classification problems, such as spam detection, diabetes prediction, whether a given customer will buy a particular product or go with the competition, etc.

It can be effective in detecting OSA by modeling the relationship between data characteristics and the probability of an individual suffering from OSA. In the context of this work, it will be used to identify data from healthy patients and data from patients suffering from obstructive sleep apnea.

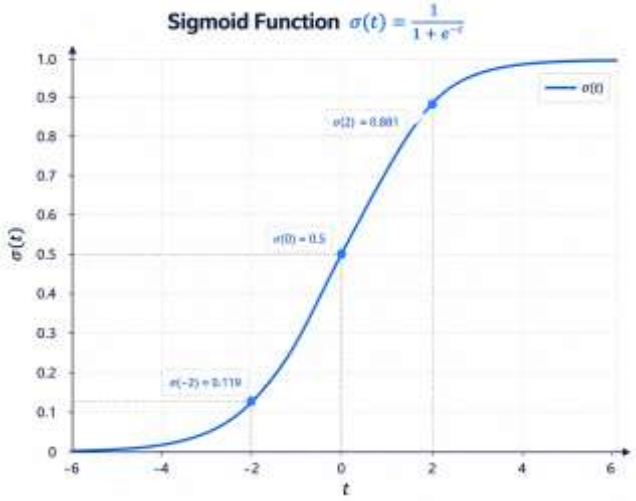

Figure 11. Logistic function =1/(1+e^(-t))

In logistic regression, the concept of threshold value is used, which defines the probability of 0 or 1. Those values above the threshold tend to 1, and those below the threshold tend to 0. Figure 12 shows a binary classification using logistic regression and a default threshold value of 0.5. The threshold value is used to map probabilities to class labels. It should be noted that threshold moving, which is hyperparameter tuning, is key when dealing with imbalanced classification problems.

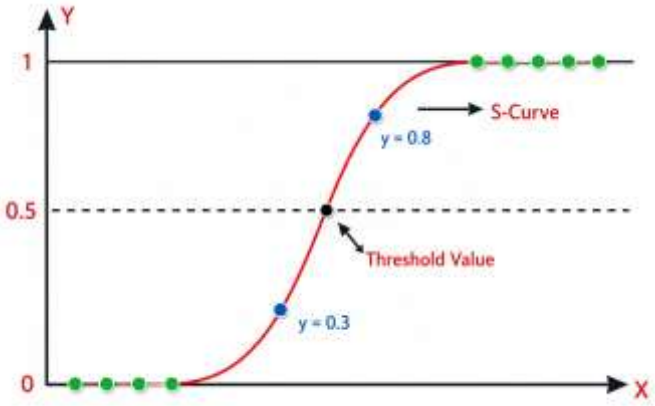

Figure 12. Logistic regression classification. [4]

[4] https://www.javatpoint.com/logistic-regression-in-machine-learning

3.2.1.2. Support Vector Machines (SVM)

The classification-regression method Support Vector Machines (SVMs) was developed in the 1990s, within the field of computer science, and although it was originally developed as a binary classification method, its application has been extended to multiple classification and regression problems. SVMs have turned out to be one of the best classifiers for a wide range of problems, which is why it is considered one of the references within the field of statistical learning and machine learning. [44]

Support vector machines are derived from maximum margin classifiers, which are used to classify perfectly separable data sets. Maximum margin classifiers seek to find a hyperplane of separation between classes that optimizes classification and reduces uncertainty when predicting a target class. For this, the distance (perpendicular) from each training observation to a given separation hyperplane (figure 13) is calculated; the smallest distance is the minimum distance from the observations to the hyperplane and is known as the margin. The algorithm selects the hyperplane with the maximum margin and classifies the test observations according to their position with respect to that hyperplane. [44]

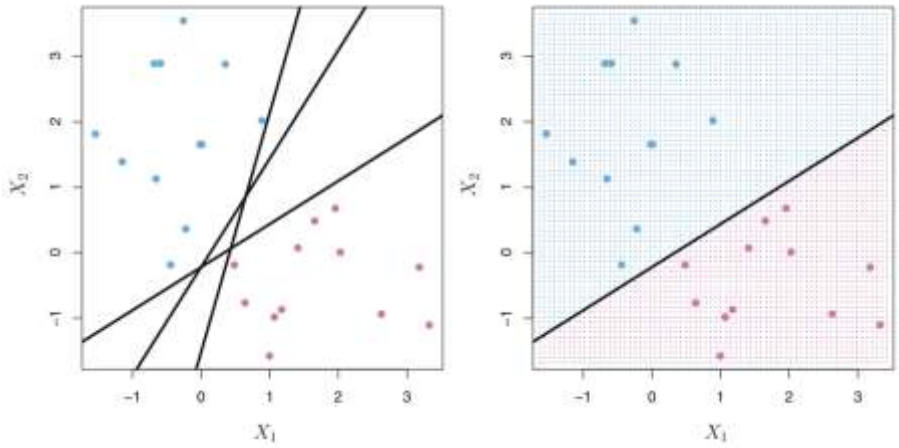

Figure 13. Separating hyperplanes. [44]

The hyperplane is defined by:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n = 0 \qquad (3.1)$$

If a data point $X = (x_1, x_2, \ldots, x_n)^T$ in n-dimensional space satisfies (3.1), then $\boldsymbol{X}$ lies on the hyperplane.

If (3.1) is not satisfied, the vector $\boldsymbol{X}$ can be on either side of the hyperplane according to equations (3.2) and (3.3), as follows:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n > 0 \qquad (3.2)$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n < 0 \qquad (3.3)$$

Considering that the observations $y_1, y_2, \ldots, y_n \in \{-1,1\}$ where -1 represents one class and 1 the other, it is true that:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n > 0 \quad if\ y_i > 0 \qquad (3.4)$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n < 0 \quad if\ y_i < 0 \qquad (3.5)$$

$$y_i * (\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n) > 0 \qquad (3.6)$$

Support vectors (figure 14) refer to a subset of the training observations that identify the location of the separating hyperplane.

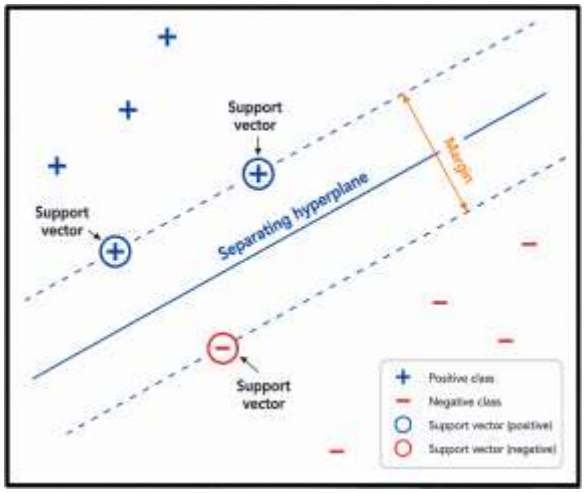

Figure 14. Support vectors. [5]

Figure 15 shows that for 2-dimensional space the separation hyperplane corresponds to a straight line, and for a 3D space is a plane.

[5] https://la.mathworks.com/discovery/support-vector-machine.html

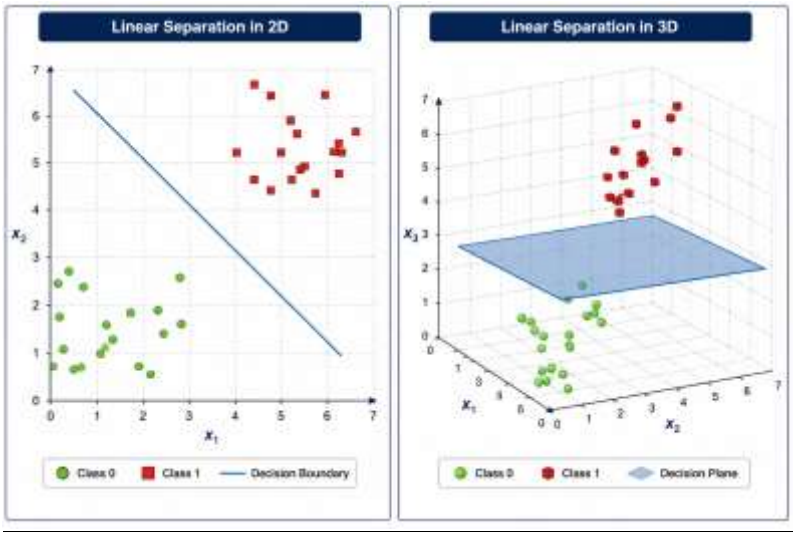

Figure 15. 2D and 3D hyperplane examples.
[6]

The support vector classifier, also called a soft margin classifier, allows itself to misclassify some training observations by introducing an error hyperparameter named C. The value of this hyperparameter determines the generalization ability of the classifier to label new observations. [44]

The standard SVM algorithm is formulated for binary classification problems; Multiclass problems are usually reduced to a series of binary problems. A multiclass classifier for m classes can be implemented with SVM using two fundamental approaches:

- One-to-One: which breaks down the multiclass problem into multiple binary classification problems. A binary classifier per each pair of classes.

$\frac{m(m-1)}{2}$ is the required number of classifiers.

- One-to-Rest: In this approach, the breakdown is set to a binary classifier per class. m different classifiers are needed.

Support Vector Machines with RBF Kernel

A **Support Vector Machine (SVM)** with a **Radial Basis Function (RBF)** kernel is one of the most powerful classical machine learning algorithms for nonlinear classification and regression problems.

RBF SVM performs extremely well in:

- Image classification.
- Handwriting recognition.
- Medical diagnosis.

[6] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

- Bioinformatics.
- Fault detection.
- Financial classification.
- Remote sensing.
- Cybersecurity.
- Intrusion detection.

The RBF kernel is also known as the **Gaussian kernel**.

The kernel is defined as shown in equation 3.7:

$$K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2),\ \gamma > 0 \qquad (3.7)$$

This kernel measures the similarity between two samples $x_i$and x_j.

When training a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel, two key parameters must be considered: C and γ (gamma).

- The C parameter, common to all SVM kernels, trades off the misclassification of training examples against the simplicity of the decision surface. A **low C value** makes the decision surface smoother, increasing tolerance to errors and improving the model's generalization ability. Conversely, a **high C value** aims to classify all training examples correctly, which reduces the training error rate but increases the risk of overfitting.
- The γ (gamma) parameter defines the radius of influence of a single training example. A **larger** γ restricts this influence to a localized area, meaning other data points must be very close to be affected. While this allows the model to capture fine details, it can also lead to a highly complex decision boundary and increase the risk of overfitting.

3.2.1.3. Random Forests

Random Forests (RF) is a supervised machine learning algorithm introduced by Leo Breiman in 2001. It belongs to the family of **ensemble learning methods**, where multiple models are combined to produce a more accurate and robust predictor.
A Random Forest consists of a collection of decision trees trained on randomly generated versions of the original dataset. Predictions from individual trees are aggregated to produce the final output.

Random forest algorithm combines multiple decision-trees, resulting in a forest of trees, hence the name Random Forest. In the random forest classifier, the higher the number of trees in the forest results in higher accuracy. The key idea is that many weakly correlated trees combined outperform a single decision tree.

Due to the structure (figure 16) of random forest classifiers, it is possible that some classifiers correctly predict the target class and others do not; However, it is assumed that most classifiers are correct, and the final decision is obtained by majority voting.

It is possible that the individual decision trees that make up the random forest overfit the training data, but averaging, in the case of regression analysis, or deciding based on consensus manages to mitigate this drawback. In this way, a robust classifier is built whose constituent elements do not have to be robust. [44]

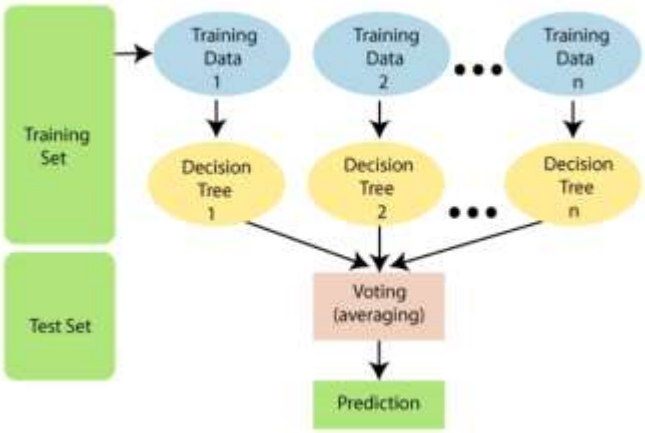

Figure 16. Random Forest structure. [7]

In the case of a random forest, hyperparameters include the number of decision trees in the forest and the number of features considered by each tree when splitting a node.

Random Forest pros

- Reduced risk of overfitting: Decision trees run the risk of overfitting, as they tend to perfectly fit all samples within the training data. However, when there are a big number of decision trees in a random forest, the classifier will not overfit the model, since averaging over uncorrelated trees reduces the overall variance and prediction error.
- Less training time compared to other algorithms.
- Flexibility: Feature clustering makes the random forest classifier an effective tool for estimating missing values as it maintains accuracy when a portion of the data is missing.
- It allows determining the importance of characteristics through the Gini index.

Random Forest cons

- Random forests are highly complex compared to decision trees, where decisions can be made by following the path of the tree.

[7] https://www.javatpoint.com/machine-learning-random-forest-algorithm

- Training time is more than other models due to its complexity. Whenever it must make a prediction, each decision tree must generate output for the given input data.

3.2.1.4. Soft Voting Ensemble Learning

Machine learning models can be used to solve classification and regression problems in a variety of fields, including engineering, medicine, finance, cybersecurity, and industrial automation. Despite the powerful performance of individual models, no single model can be designed for all datasets and applications. Often, different algorithms capture different aspects of the data distribution, and their error profiles differ. To overcome this difficulty, ensemble learning techniques have been developed to leverage multiple predictive models and make more informed decisions. Of the methods used to solve classification problems, voting-based models are the most popular. Soft voting is known to be one of the effective methods for combining the predictive capabilities of heterogeneous machine learning models and is based on the confidence level of each classifier. Ensemble voting methods generally follow one of two strategies: hard voting or soft voting. Hard voting follows the "winner-takes-all" rule, in which the final prediction is made by most classifiers. Soft voting uses the confidence level of each classifier's prediction. This method sums up the predicted probabilities for each class label and provides the label with the highest cumulative probability, as shown in Figure 17.

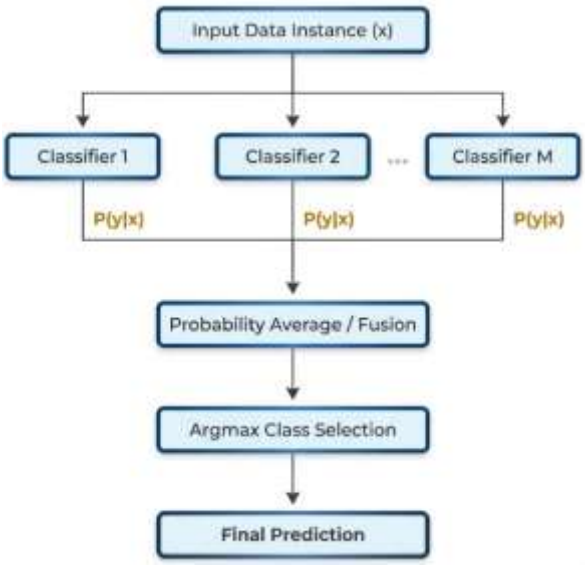

Figure 17. Soft voting classifier structure.

Let us define a multi-class classification problem spanning K different classes,

where $y \in \{1,\ 2, \dots, K\}$. For any given target input vector x, an individual base classifier m (where $m\ \in \{1, 2, \dots, M\}$) outputs a vector of conditional posterior probabilities:

$$P_m(y = k|x), \forall\ k \in \{1, 2, \dots, K\}$$

This probability vector is strictly bound by standard Kolmogorov axioms, dictating that all individual probabilities must be non-negative and satisfy the sum-to-one constraint:

$$\sum_{k=1}^{K} P_m(y = k|x) \qquad (3.8)$$

Once the aggregated probability for each class is established, the final discrete class label assignment $\hat{y}$ is selected using the optimal Bayes decision rule via the $\arg max$ operator:

$$\hat{y} = argmax_k P(y = k|x) \qquad (3.9)$$

3.2.1.5. Ensemble Weighted AUC

Ensemble weighted AUC combines the predictions of multiple individual models (the ensemble) and assigns a weight to each model according to its Area Under the Receiver Operating Characteristic Curve (AUC). We use the strongest models to improve classification performance, robustness and handling of imbalanced datasets. AUC is a measure of a model's ability to distinguish between classes with values ranging from 0.5 (random guess) to 1.0 (perfect prediction). In AUC-Weighted Ensembles the optimization strategy is to allocate weights (Wm) to each individual model based on AUC score and to weight individual base classifiers by accuracy or average. Although the weights Wm in a soft voting ensemble can be obtained through heuristic search, the most effective and deterministic algorithm for clinical and classification purposes is to draw weights directly from the AUC of the individual model. [46]. Since the AUC measures a classifier's ability to discriminate between classes across all possible thresholds, weighting models according to their validation AUC ensures superior discriminators dominate the probability fusion. Let AUC_m represents the performance of classifier m evaluated on a held-out validation dataset. To prevent lower-performing models from degrading the ensemble while rewarding top-performing models, the normalized weight assignment w_m is formalized in equation 3.10:

$$w_m = \frac{AUC_m - \gamma}{\sum_{j=1}^{M}(AUC_j - \gamma)} \qquad (3.10)$$

Where γ represents a baseline performance threshold (typically $\gamma = 0.5$, representing the performance of a random guess). If an individual classifier exhibits an $AUC_m \leq \gamma$, its assigned weight is set to zero ($w_m = 0$), effectively pruning it from the ensemble.

Integrating this into the weighted soft voting equation yields an **Ensemble Weighted AUC Strategy**, which mathematically prioritizes the probabilistic confidence of the most discriminatory base learners.

Figure 18 illustrates the ensemble weighted AUC process.

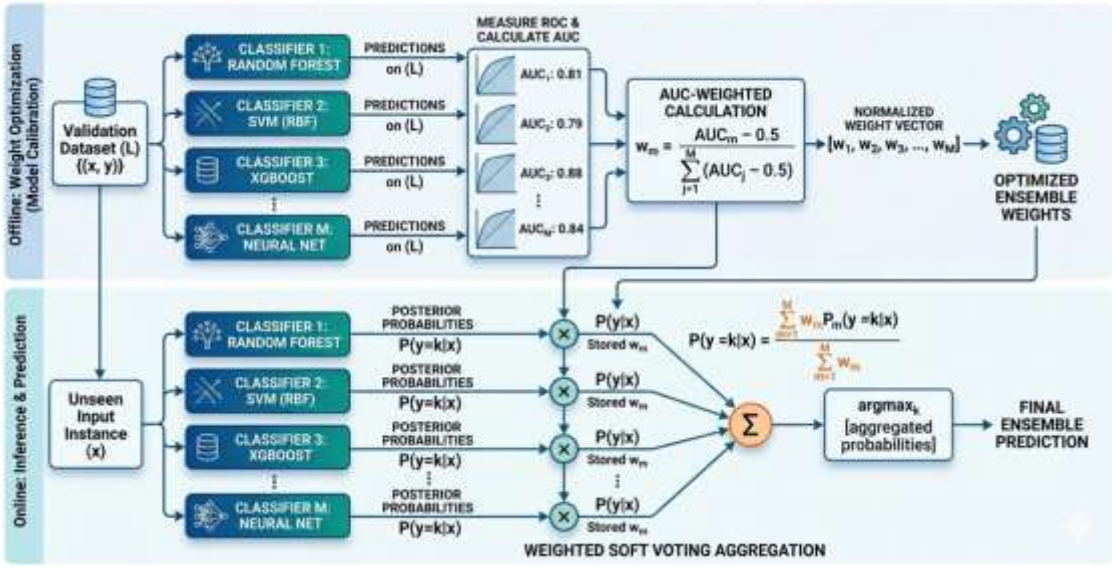

Figure 18 Weighted AUC voting classifier. [8]

3.2.1.6. Ensemble Stacking

Ensemble learning is one of the best paradigms in machine learning in which different models, which we call "base learners" or "weak learners," are combined to solve a computational intelligence problem. The goal of an ensemble is to develop a meta-model that can be better generalized and robust than the individual model in the ensemble. While ensemble methods such as Bagging (e.g., Random Forests) rely on parallel voting or averaging of independent models, and Boosting (e.g., Gradient Boosting Machines) is the sequential correction of residual errors, Stacking (sometimes called Stacked Generalization) is a very different philosophical approach. Instead of using simple axiomatic functions like a majority vote or an algebraic mean to combine predictions from different base learners in Stacking, a meta-learning algorithm is trained to find the best way to combine the predictions. Unlike voting, stacking does not consider the role of all models equally. Rather it learns the best combination from data.

Diversity is the basis of stacking. If the ensemble has models that make the same mistakes on the same data points, the ensemble is not gaining any statistical advantage. Stacking leverages diversity by mixing different base learners in the system like a support vector machine (SVM), a deep neural network, and a gradient-boosted tree. Each algorithm looks at the input feature space on its own based on an inductive bias, and therefore it can find a different linear or non-linear pattern. The meta-learner is then able to determine the operational strengths and weaknesses of these base models in different regions of the feature space and decide if the base learner is trustworthy or not trustworthy. [46]

[8] Gemini AI generated image.

Mathematical foundations

Level 0 (Base Models)

$$h_1(x), h_2(x), \dots, h_M(x) \qquad (3.11)$$

Each model produces:

$$z_i = h_i(x) \qquad (3.12)$$

Level 1 (Meta-Learner)

The outputs of base learners become new features:

$$\boldsymbol{Z} = [z_1, z_2, \dots . z_M] \qquad (3.13)$$

A meta-model $g(Z)$ learns:

$$\hat{y} = g(Z) \qquad (3.14)$$

The final prediction $\boldsymbol{F}(x)$ is:

$$\boldsymbol{F}(x) = g(h_1(x), h_2(x), \dots, h_M(x)) \qquad (3.15)$$

The meta-model estimates optimal weights and nonlinear relationships among base learners.

The stacking process divides the work into two main training levels as can be seen in figure 19:

Level 0 Models (Base Models): Several different algorithms are trained using the original data (for example, a decision tree, a neural network, a support vector machine, and a logistic regression).

Level 1 Model (Meta-model): It takes the predictions generated by the base models and learns the best way to combine them to deliver the final output.

Process Phases

- **Base training:** Level 0 models receive the training data and generate their independent output.
- **Meta-feature generation:** The predictions from the base models are grouped together to form a new dataset.
- **Final training:** The meta-model (level 1) processes this new dataset to correct weaknesses and weigh correct predictions.

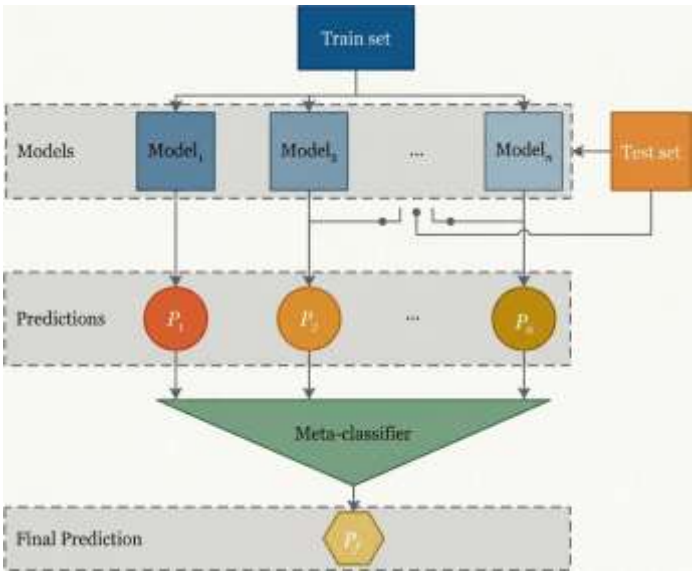

Figure 19. Ensemble stacking algorithm. [47]

3.2.1.7. Artificial neural networks (ANN)

Neural networks are a fundamental part of deep learning and artificial intelligence. Neural networks are computer models inspired by the structure and function of the human brain's interconnected neurons. Neural networks are layers of artificial neurons (also known as nodes or units) and work together to perform sophisticated tasks.

According to figure 20, ANN consists of layers: an input layer, one or more hidden layers, and an output layer. The input layer receives the data, the hidden layers process it, and the output layer makes the network's final prediction or output.

Activation Functions: Neurons in a neural network use activation functions to introduce non-linearity in the model. Common activation functions include the sigmoid, ReLU (Rectified Linear Unit), and tanh (hyperbolic tangent). These functions are important to the network to model complex relationships in data.

Weights and biases: Connections between neurons in different layers have weights and biases. Therefore, these parameters are learned as the training process proceeds to train the network to make accurate predictions.

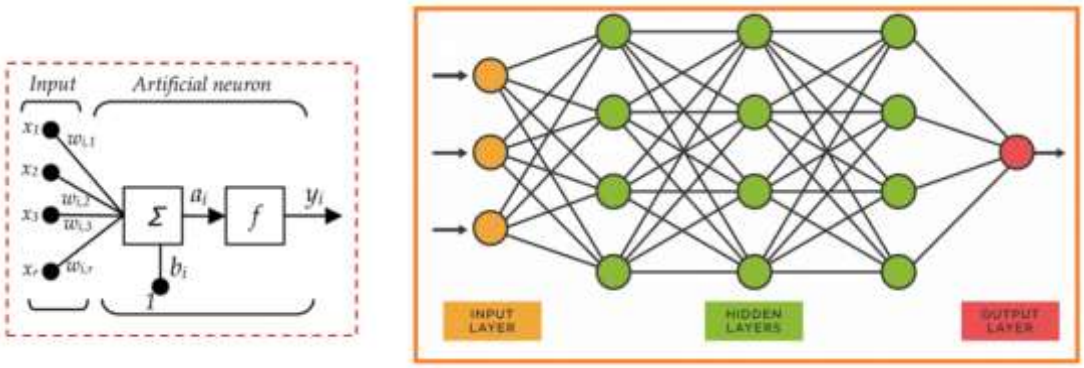

Figure 20. ANN structure. [9] [10]

Learning through Backpropagation: Neural networks learn from data by backpropagation. The network's predictions are compared to the actual target values during training and errors are propagated backward through the network to adjust the weights and biases of the network.

Neural networks excel at pattern recognition. They can automatically learn patterns in data and thus are very effective in tasks such as image classification, speech recognition, and language translation.

Another important insight is that neural networks can approximate any continuous function theoretically. This makes them very flexible and capable of modeling complex relationships in data very well.

Training deep neural networks is computationally expensive and requires a large amount of data. They can also be prone to overfitting, where the model performs well on the training data but fails to generalize to new unseen data.

Neural networks find applications in computer vision, natural language processing, recommendation systems, robotics, autonomous vehicles, and many others. They are at the heart of many state-of-the-art AI systems.

[9] https://www.marktechpost.com/2022/09/23/top-neural-network-architectures-for-machine-learning-researchers/

[10] https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales-html/x38.html

3.2.1.7.1. Recurrent neural networks (RNN)

RNNs are a type of neural network architecture (figure 21) designed for sequence data, making them well-suited for tasks involving time series data, natural language processing, and speech recognition. RNNs have recurrent connections that allow them to maintain a hidden state that captures temporal dependencies in the data. However, traditional RNNs suffer from the vanishing gradient problem, limiting their ability to capture long-range dependencies. More advanced RNN variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have been developed to mitigate this issue.

,

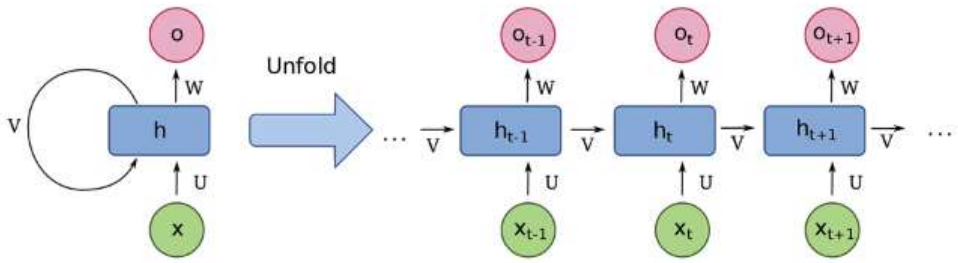

Figure 21. Recurrent Neural Network. [11]

The Recurrent Neural Network consists of multiple fixed activation function units, one for each time step. Each unit has an internal state which is called the hidden state of the unit. This hidden state signifies the past knowledge that the network currently holds at a given time step. This hidden state is updated at every time step to signify the change in the knowledge of the network about the past. The hidden state is updated using equation 3.16.

$$h_t = f(h_{t-1}, x_t) \qquad (3.16)$$

Where:
h_t is the current state.
h_{t-1} is the previous state.
x_t is the current input.

Pros of RNNs

- Handle sequential data effectively, including text, speech, and time series.
- Process inputs of any length, unlike feedforward neural networks.
- Share weights across time steps, enhancing training efficiency.

Cons of RNNs

- Susceptible to vanishing and exploding gradient problems, hindering learning.
- Training can be challenging, especially for long sequences.
- Computationally slower than other neural network architectures.

[11] https://ailephant.com/glossary/recurrent-neural-network/

3.2.1.7.2. Convolutional Neural Networks (CNN)

The CNNs are deep neural networks that are designed for grid-like data processing (e.g., images and videos). They are particularly useful for computer vision tasks. CNNs employ convolutional layers and automatically learn and gain information on what is the data but retain spatial relationships and features. Pooling layers reduce the size of feature maps. CNNs have been used in image classification, object recognition, facial recognition, and more. They have also been used for non-images, e.g., text and audio. Convolutional Neural Networks (CNNs) are a top-of-the-line deep learning model that does indeed automate hierarchical feature extraction using complex input data. CNNs are typically used for computer vision tasks, but are becoming the state-of-the-art for biomedical signal processing, as they do not require human-built feature extraction. In somnology and cardiology, CNNs are utilized for the analysis of raw time series data such as single-lead electrocardiogram (ECG) and pulse oximetry (SpO_2) data to automatically detect pathological conditions such as Obstructive Sleep Apnea (OSA). The basic architecture of a CNN (figure) is an input layer, several convolutional and pooling layers and a succession of fully connected (dense) layers to generate classification results. In a 1D convolution layer for sequence biological signals, learnable filters (kernels) slide over the input temporal array. This mathematical operation extracts local patterns like sudden changes in waveform shape, changes in the QRS complex or specific transient segments that are affected by respiratory events. Writing in 1D form, given a 1D discrete input x and a kernel w of length K, the convolution $(x * w)$ at a given time is defined by equation 3.17:

$$(x * w)(t) = \sum_{k=0}^{K-1} x\,(t-k) \cdot w(k) \qquad (3.17)$$

To introduce non-linear mapping capabilities necessary to model complex biological phenomena, the output of the convolution is passed through an activation function, with the Rectified Linear Unit (ReLU) being the most pervasive choice:

$$\text{ReLU}(x) = \max(0, x) \qquad (3.18)$$

Subsequent pooling layers (typically max pooling) down sample the generated feature maps. This step systematically compresses the dimensional footprint, enforces translation invariance, and preserves the most salient activations across consecutive segments. Finally, the compressed multidimensional tensor is flattened into a 1D vector and funneled through fully connected hidden layers. An ultimate activation layer, such as Softmax or Sigmoid, yields the formal posterior probabilities required to classify the clinical window.

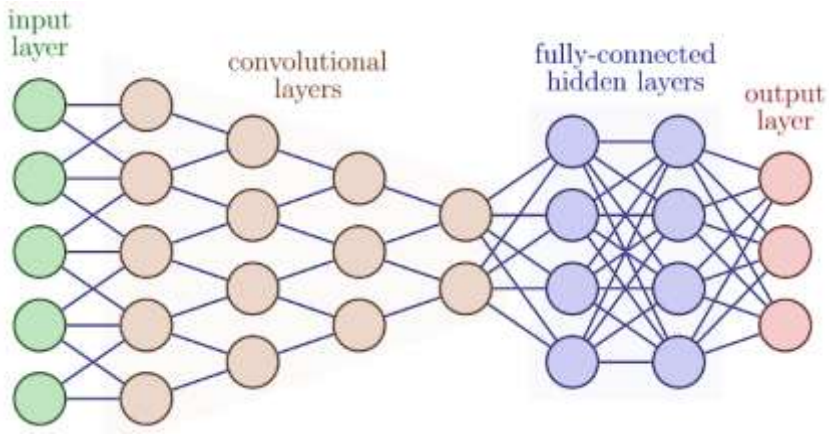

Figure 22. Convolutional neural network structure.

3.2.1.7.2.1. Performance Evaluation Metrics

To objectively evaluate and compare the diagnostic performance of the implemented Machine Learning and Deep Learning models, researchers must employ a standardized battery of statistical evaluation metrics. These metrics provide a quantifiable measure of clinical accuracy and are derived from the foundational elements of the standard confusion matrix (figure 23): True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

		ACTUAL VALUES	
		Positive (1)	**Negative (0)**
PREDICTED VALUES	**Positive (1)**	**TP** True Positive count: TP	**FP** False Positive count: N
	Negative (0)	**FN** False Negative count: N	**TN** True Negative count: N

TP (True Positive): Correctly predicted positive cases.

FN (False Negative): Incorrectly predicted negative cases.

FP (False Positive): Incorrectly predicted positive cases.

TN (True Negative): Correctly predicted negative cases.

Figure 23. Confusion matrix.

- **True positives** (TP): when the actual value is positive, and the predicted value is also positive.

- **True negatives** (TN): when the actual value is negative and the predicted is negative too.
- **False positives** (FP): when the actual value is negative, but the prediction is positive. This is called a type I error.
- **False negatives** (FN): when the actual value is positive, but the prediction is negative. This is called a type II error.

A good classification model has high true positives (TP) and true negatives (TN) rates, and low false positives (FP) and false negatives (FN) rates.

From the confusion matrix, the following metrics are defined:

Accuracy is a measure of how often the classifier makes the correct prediction. It is the ratio between the number of correct predictions and the total number of predictions.

$$\boldsymbol{Accuracy} = \frac{Correct\ predictions}{Total\ predictions} = \frac{TP + TN}{TP + TN + FP + FN} \qquad (3.19)$$

This metric is not appropriate for imbalanced classes and can mislead about the real performance of the model. For imbalanced data, when the model predicts that each point belongs to the majority class label, the accuracy will be high, but the model is not right.

Precision or positive predictive value (PPV) calculates how many of the actual positive cases the model was able to predict correctly.

$$\boldsymbol{Precision} = \frac{True\ positives}{Number\ of\ predicted\ positives} = \frac{TP}{TP + FP} \qquad (3.20)$$

Recall or Sensitivity is defined as the number of true positives divided by the total number of actual positives.

$$\boldsymbol{Recall} = \boldsymbol{Sensitivity} = \frac{True\ positives}{Number\ of\ actual\ positives} \qquad (3.21)$$

F1-Score is the harmonic mean of precision and recall and gives a combined idea about precision and recall metrics. It is maximum when precision is equal to recall. F1-Score is an appropriate evaluation metric when true negatives (TN) rate is high, when FP and FN are equally costly and when adding more data does not change the outcome.

$$\boldsymbol{F1 - Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (3.22)$$

AUC (Area Under the Curve)**-ROC** (**Receiver Operating Characteristic Curve**) is a graphical approach for displaying the tradeoff between true positive (TP) rate and false positive (FP) rate of a classifier. According to Figure 24, the higher the AUC, the better the classifier and its ability to distinguish between classes.

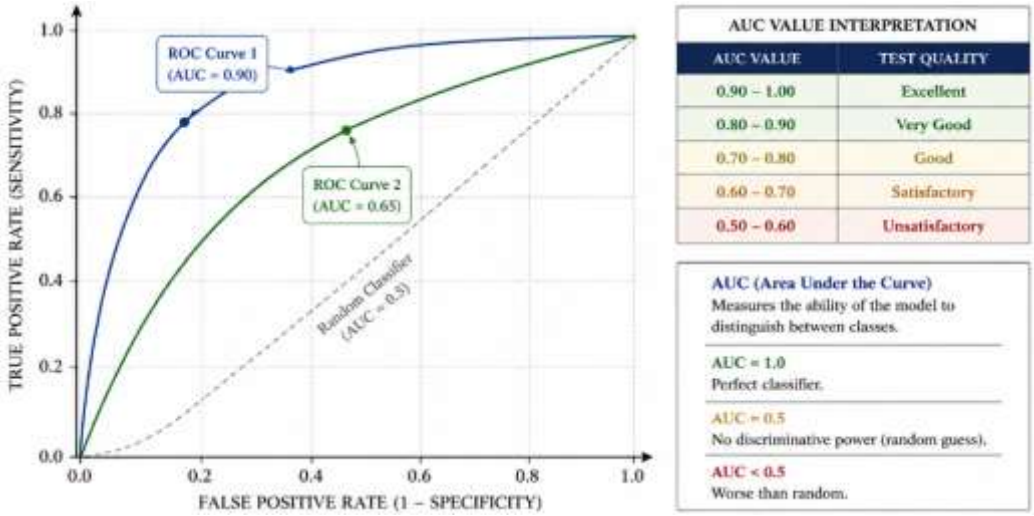

Figure 24. ROC Curve. [45]

4. System Development

4.1. Methodology

The development of the obstructive sleep apnea (OSA) classification system followed an iterative, experimental process grounded in the classical data science cycle—data understanding, data preparation, modeling, and evaluation—adapted for physiological signal analysis. The pipeline was implemented in Python using Jupyter Notebook, which allowed each stage to be developed, tested, and validated as an independent, reusable module before being integrated into the full workflow.

A central methodological principle applied at every stage of this project was **subject-independent evaluation**: whenever the dataset was partitioned—for training, internal validation, or testing—the split was performed at the **patient (subject) level, never at the level of individual 1-minute windows**. This choice follows established guidance in the literature [24], which shows that random partitioning at the window level can allow windows from the same patient to appear in both the training and testing sets, artificially inflating reported performance through data leakage. Enforcing subject-level splits at every stage of this pipeline, including, critically, the internal train/validation split used for early stopping during deep learning training, where this principle was initially overlooked and later corrected once its effect on reported performance was identified—was essential to obtaining honest, clinically meaningful estimates of model generalization.

Ten classification approaches were implemented and compared under an identical evaluation protocol: three classical machine learning models (Logistic Regression, Support Vector Machine with an RBF kernel, and Random Forest), four deep neural network models (a Convolutional Neural Network and a Recurrent Neural Network, each trained on two different input representations—the raw ECG waveform and a reduced RR-interval/ECG-derived-respiration (EDR) tachogram, as described in Section 4.3.2), and three ensemble

strategies that combine the base models' probabilistic outputs (soft voting, AUC-weighted voting, and stacking with a logistic-regression meta-classifier).

The pipeline was not built in a single pass; several design decisions were revised after empirical testing exposed limitations in earlier versions. Signal normalization was initially applied at the window level, which discarded amplitude information relative to the rest of the recording; this was replaced with a recording-level normalization strategy that preserves that relative amplitude context across the full signal. R-peak detection was likewise revised, replacing an initial simplified Pan–Tompkins implementation with the NeuroKit2 library, which proved more robust to noise. Finally, recognizing that per-minute classification alone could not capture the longer temporal structure of apnea episodes, two additions were incorporated: temporal-context features (aggregating neighboring one-minute windows) and a post-hoc probability-smoothing step applied to the sequence of per-minute predictions.

Hyperparameter selection for the three classical models was likewise developed iteratively. An initial implementation used manually chosen values (e.g., C=1.0 and gamma='scale' for the SVM; 300 trees for the Random Forest) without any documented search process, despite the project's theoretical framework describing grid search and Bayesian optimization as the intended methodology. This gap was identified and corrected by implementing an explicit search: an exhaustive grid search for Logistic Regression and randomized search for SVM and Random Forest, both evaluated under 5-fold GroupKFold cross-validation—again partitioned by subject rather than by window, consistent with the subject-independent principle applied throughout the pipeline—over the training cohort only, with AUC-ROC as the optimization metric. Retraining the three models with the resulting optimal hyperparameters and comparing their performance against the original manually chosen configuration, on both the internal and external test sets, showed that the search produced no meaningful improvement (AUC differences ≤0.007 in either direction). This null result was informative rather than a wasted step: it demonstrated that the original values were already close to optimal for this dataset, a conclusion that could only be established by performing the search rather than assuming it from the theoretical framework alone.

An analogous, architecture-level search was subsequently applied to the two deep learning models built on the RR+EDR tachogram representation (the CNN and RNN variants), which had shown the best generalization among the deep models in external validation; the raw-ECG CNN and RNN were excluded from this search given their already-established poor generalization and the disproportionate computational cost of iterating over raw-signal architectures. For computational tractability, this search used a single subject-grouped train/validation split rather than full cross-validation, together with a reduced training budget (10 epochs, early-stopping patience of 3) to evaluate eight randomly sampled architecture candidates per model. Critically, each search-phase winner was then retrained from scratch under the full training budget (25 epochs, patience of 5) before being considered for deployment—a validation step that proved decisive: the optimized CNN architecture confirmed a genuine, if modest, improvement over the original manually designed network (validation AUC 0.9257 → 0.9306) and was deployed, whereas the RNN architecture favored by the reduced-budget search performed *worse* than the original once retrained under the full budget (0.7577 → 0.7167) and was therefore discarded in favor of the original design. This outcome illustrates why a reduced-budget search cannot be trusted on its own: with a training cohort this size, a short evaluation window is a noisy estimator of an

architecture's true potential, and only full-budget retraining can confirm whether an apparent improvement is real.

Finally, because all reported comparisons rest on a single external cohort of 35 subjects, pairwise statistical significance testing was carried out to determine which performance differences between models could be considered reliable rather than attributable to sampling variability. Two complementary tests were applied to every pair among the ten models: McNemar's test on paired window-level predictions, and a subject-clustered bootstrap (2000 resamples of the 35 subjects, with the same resample applied across all models for each iteration) comparing AUC differences. The two tests diverged substantially: McNemar's test, which treats each of the 12600 one-minute windows as an independent observation, flagged the large majority of pairwise comparisons as significant, whereas the subject-clustered bootstrap—which correctly accounts for the correlation among windows drawn from the same patient—found significant differences in a smaller subset of comparisons. Under the statistically appropriate test, Random Forest and the three ensemble strategies formed a group with no significant pairwise differences among them, while every member of that group significantly outperformed the raw-ECG CNN and RNN. This analysis was treated as the authoritative basis for interpreting the project's comparative results, in preference to the more liberal window-level test.

4.2. Data preparation

4.2.1. ECG Signal Characteristics and Mechanisms and Relation with OSA

The ECG signal is the electrical manifestation of heart’s contraction activity and can be acquired with electrodes on the chest or limbs. It is perhaps the best-known and most widely used biomedical signal. Heart rate, in terms of beats per minute, can be easily estimated by quantifying the pulses of the electrical signal.

The ECG signal contains a wealth of information about cardiovascular disorders and abnormalities such as ischemia, ventricular hypertrophy, and circulatory problems.

The **electrocardiogram (ECG)** is a non-invasive recording of the heart's electrical activity obtained through electrodes placed on the body surface. It reflects the sequential depolarization and repolarization of the atrial and ventricular myocardium, providing essential information about cardiac rhythm, conduction pathways, and autonomic nervous system regulation. The ECG waveform consists of several characteristic waves, segments, and intervals, each corresponding to a specific electrophysiological event within the cardiac cycle.

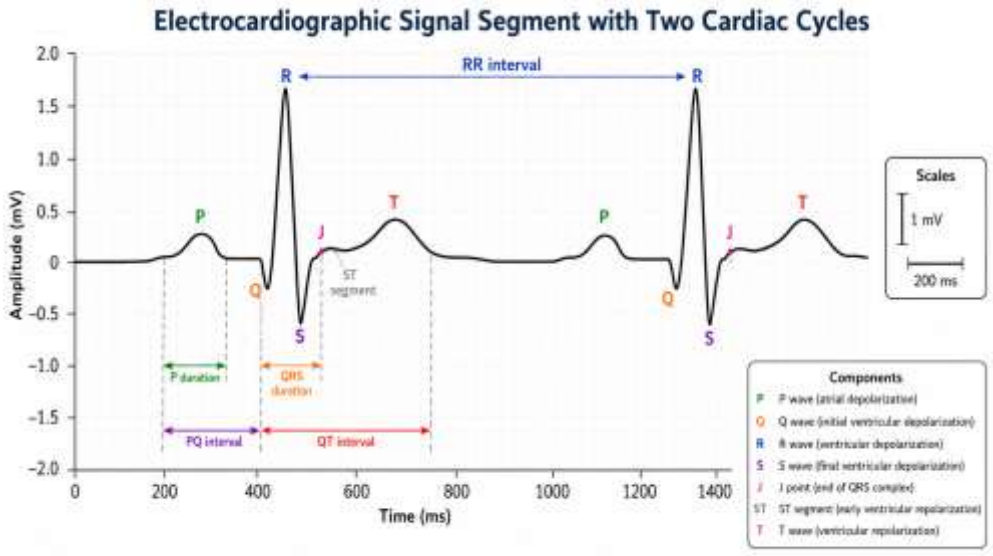

Figure 25. ECG signal.

Figure 25 illustrates a standard, non-invasive electrocardiogram waveform tracing over successive cardiac cycles, highlighting key physiological points (P, Q, R, S, T) and temporal intervals.

Key Components:

- **P Wave:** Represents atrial depolarization and contraction (0.1–0.2 mV, lasting 60–80 ms).
- **PQ / PR Segment:** Reflects the quiet electrical period during blood transfer from the atria to the ventricles (60–80 ms).
- **QRS Complex:** Denotes rapid ventricular depolarization and contraction (~ 1 mV amplitude).
- **ST Interval & T Wave:** Represent the subsequent period of ventricular electrical potential stability (100–120 ms) followed by ventricular relaxation (0.1–0.3 mV, lasting 120–160 ms).
- **RR Interval:** The distance between two consecutive R-wave peaks, which forms the basis for estimating heart rate and tracking variation.

Among these measurements, the **RR interval** is particularly important in biomedical signal processing because it captures beat-to-beat fluctuations in cardiac rhythm. Variations in RR intervals arise primarily from autonomic nervous system modulation, with sympathetic activation shortening the interval (increasing heart rate) and parasympathetic activity lengthening it (decreasing heart rate). Consequently, RR interval analysis forms the basis for numerous HRV metrics in both the time and frequency domains.

In sleep medicine, ECG-derived RR intervals have become valuable biomarkers for detecting Obstructive Sleep Apnea (OSA). During apnea episodes, intermittent hypoxia and repeated arousals produce characteristic autonomic responses, resulting in cyclic variations of heart rate. These changes manifest as alterations in RR intervals and HRV features such as **SDNN, RMSSD, pNN50, LF, HF, LF/HF ratio, total power, VLF,** and other derived

parameters. Machine learning and deep learning algorithms can exploit these physiological alterations to distinguish normal breathing from apnea events, enabling automated OSA screening using single-lead ECG recordings.

Figure 26 compares a normal RR interval signal with another collected during an apneic event and shows the way that power spectral density (PSD) is affected by OSA.

In the presence of an active apnea episode, the heart rate exhibits severe, slow cyclical oscillations typically repeating every $25–100$ seconds. In the frequency domain, this manifests as a massive, distinct, and high-amplitude spectral spike concentrated heavily within the Very Low Frequency (VLF) band ($\sim 0.01–0.04$ Hz).

In contrast, normal breathing windows exhibit standard, higher-frequency heart rate variability. The corresponding spectrum shows significantly lower power spectral density amplitudes (maxing out near a PSD of 10–15 compared to the apnea plot's spike near 2200) without any dominant low-frequency peak, scattering the energy naturally across the baseline physiological bands. [28]

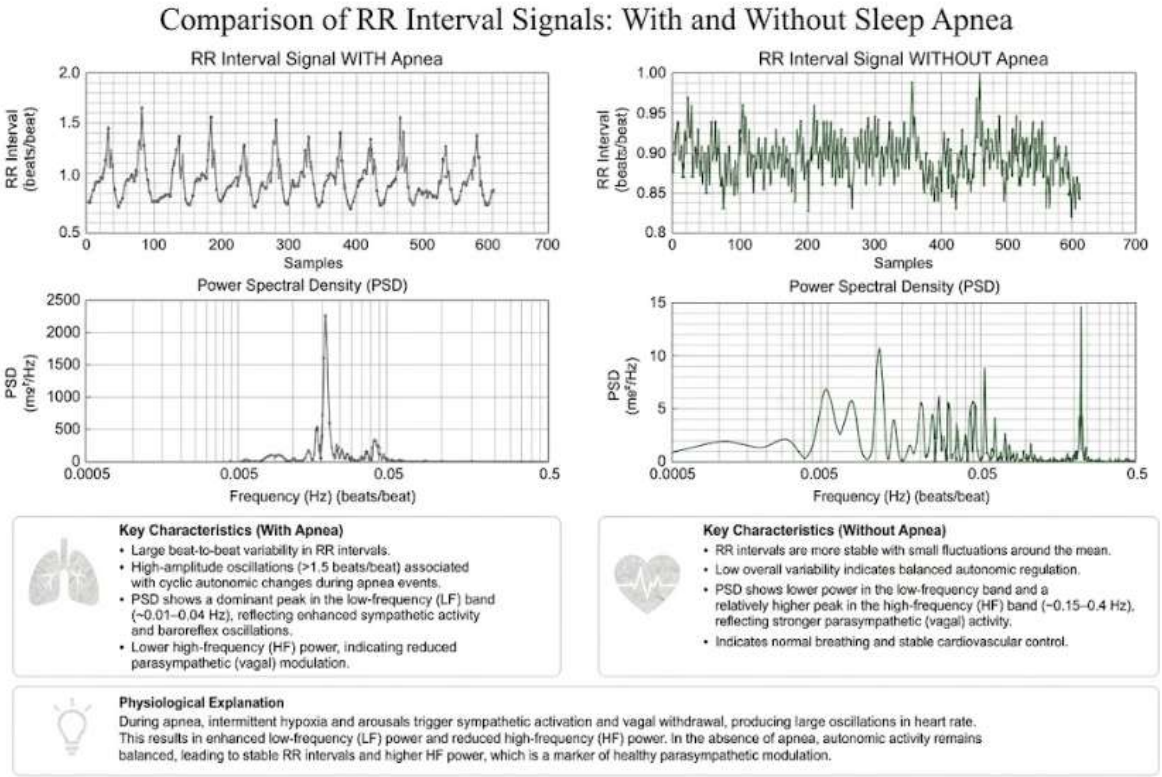

Figure 26 OSA frequency impact.[12]

Overall, the ECG provides a comprehensive representation of the heart's electrical function, while its temporal and morphological features offer rich physiological information for

[12] Gemini AI improved image.

diagnosing cardiovascular diseases and developing intelligent systems for sleep disorder detection. RR interval analysis has become one of the most informative and computationally efficient approaches for ECG-based obstructive sleep apnea classification.

The physiological rationale for detecting OSA from a single-lead ECG rests on two well-documented mechanisms. First, each apnea event triggers a transient surge in sympathetic nervous system activity as the airway obstruction resolves, producing a characteristic cyclical pattern of bradycardia during the obstructive phase followed by tachycardia upon arousal; this cyclical pattern is measurable through heart rate variability (HRV) analysis of the RR-interval sequence. Second, the mechanical effort of breathing against an obstructed airway, and the accompanying changes in intrathoracic pressure and cardiac electrical axis, modulate the amplitude of the R wave in synchrony with the respiratory cycle, allowing an ECG-derived respiration (EDR) signal to be estimated without a dedicated respiratory sensor. Sustained sympathetic activation and repeated hypoxia-reoxygenation cycles associated with untreated OSA are, in turn, established risk factors for cardiovascular morbidity, which is the clinical motivation for developing accessible ECG-only screening tools. [1] [9] [10] [11]

Two families of HRV-derived variables were used to characterize each one-minute window: time-domain measures (mean RR interval, SDNN, RMSSD, pNN50, mean and standard deviation of heart rate, and minimum/maximum RR interval) and frequency-domain measures obtained from the power spectral density of the RR-interval series (power in the very-low-, low-, and high-frequency bands, their ratio, and total power), consistent with standard HRV analysis conventions applied to OSA detection in the reviewed literature. [2]

4.2.2. Obtaining and preparing the data

The system was developed and validated on the Apnea-ECG database (PhysioNet), comprising 70898 single-lead ECG recordings sampled at 100 Hz, each divided into consecutive one-minute segments with an expert-assigned binary label (apnea / normal breathing) for every minute. The database groups recordings into three clinically distinct categories: 20 recordings from patients with a confirmed apnea diagnosis (identifiers a01-a20), 5 recordings from borderline patients with intermediate apnea severity (b01-b05), and 10 recordings from healthy control subjects with no apnea events (c01-c10). This three-way stratification is clinically meaningful and was preserved throughout the modeling process, since a model trained only on clearly apneic or clearly healthy subjects would not generalize to the borderline cases that are most relevant in a real screening scenario.

The complete dataset, including training and test subjects, is depicted in table 1.

Dataset Split	Record Names	Count	Patient Classifications Included	Annotation Availability

Train Set (Learning Set)	`a01` to `a20`
	`b01` to `b05`
	`c01` to `c10`
		35	• **Class A (Apnea):** 20 records
			• **Class B (Borderline):** 5 records
			• **Class C (Control/Normal):** 10 records
				Full minute-by-minute apnea labels (`.apn` files)

Test Set	`x01` to `x35`	35	Blind combination of Apnea, Borderline, and Control patients	Labels withheld by PhysioNet for model benchmarking
Total	**70 records**	**70**	**All classes represented across both splits**	**35 labeled / 35 unlabeled**

Table 1. Physionet apnea-ECG 2000 database

Each recording in the database was transformed into a MATLAB file pair: one containing the raw ECG signal (a one-dimensional array named after the recording identifier) and one containing the corresponding per-minute annotation array, encoded with the ASCII values 65 ('A', apnea) and 78 ('N', normal). A data-loading module was implemented to read both files for every recording, validate that the signal and annotation arrays were of compatible length, and segment the continuous signal into non-overlapping one-minute windows of 6000 samples (60 seconds at 100 Hz), each associated with its single-minute label.

The following are the scripts used to convert the ECG signal from .dat format to MATLAB format (.mat) and to convert the text files with the annotations into .mat files:

```
x=load('D:\PROYECTO_MAESTRIA\apnea_ecg_database\test_data\x02m.mat');
ecg_dig=x.x02m;
ecg_x02=ecg_dig/200; %Dividiendo por la ganancia
N=length(ecg_dig);

%Vector tiempo

Fs=100;
Ts=1/Fs;
t=(0:1:N-1)*Ts;

plot(t,ecg_x02,'r');grid on;xlabel('t [s]');ylabel('x02(t) [mV]');
```

```
axis([0 11 -1 11]);
vec_x02=ecg_x02;
x02=vec_x02';

save('D:\PROYECTO_MAESTRIA\apnea_ecg_database\test_data\x02.mat');
%lectura de anotaciones
clear all
clc
[var1 var2 var3]=readvars('annotations_a01.txt');
x=var2;
y=cast(var3,'char');
y(1:10)
y=cast(y,'double');
z=[x(1:360) y(1:360)];
size(z)
```

4.2.3. Preprocessing, filtering and feature extraction

Table 2 shows the 70 recordings that make up the dataset with their original recording's length. Due to errors in the capture of some signals, it was decided to shorten all recordings to 6 hours, generating vectors of 2,160,000 samples in length.

SET	RECORD	ORIGINAL LENGTH (hh:mm:ss)	SET	RECORD	ORIGINAL LENGTH (hh:mm:ss)	SET	RECORD	ORIGINAL LENGTH (hh:mm:ss)
TRAINING	a01	8:12:50	TRAINING	c01	8:03:10	TEST	x16	8:34:50
	a02	8:50:20		c02	8:21:10		x17	6:45:00
	a03	8:42:30		c03	7:33:20		x18	7:38:50
	a04	8:16:40		c04	8:01:10		x19	8:10:40
	a05	7:33:15		c05	7:45:30		x20	8:32:25
	a06	8:29:40		c06	7:47:20		x21	8:29:35
	a07	8:30:05		c07	7:08:10		x22	8:01:10
	a08	8:20:25		c08	8:34:00		x23	8:52:40
	a09	8:18:00		c09	7:47:30		x24	7:08:30
	a10	8:36:55		c10	7:10:50		x25	8:29:10
	a11	7:48:00	TEST	x01	8:42:50		x26	8:39:50
	a12	9:37:00		x02	7:52:10		x27	8:17:35
	a13	8:14:15		x03	7:45:00		x28	8:15:00
	a14	8:42:30		x04	8:01:50		x29	7:50:10
	a15	8:29:20		x05	8:24:30		x30	8:54:05
	a16	8:01:20		x06	7:29:30		x31	9:17:00
	a17	8:04:05		x07	8:28:20		x32	8:58:00
	a18	8:12:50		x08	8:37:00		x33	7:52:20
	a19	8:22:00		x09	8:27:50		x34	7:54:10
	a20	8:30:00		x10	8:29:35		x35	8:02:10
	b01	8:06:10		x11	7:36:05			
	b02	8:48:25		x12	8:56:50			
	b03	7:20:30		x13	8:28:10			
	b04	7:08:50		x14	8:17:30			
	b05	7:12:15		x15	8:40:00			

Table 2. Training and test dataset recording length.

Each one-minute window was band-pass filtered with a zero-phase fourth-order Butterworth filter (0.5-40 Hz) to remove the baseline wander and high-frequency noise while preserving the QRS shape necessary to achieve R-peak detection. We tested two different

normalization approaches for the filtered signal. The first is to normalize each one-minute window separately to zero mean and unit variance. The amplitude of windows labelled apnea is compared with windows labeled normal and the values are shown on a synthetic signal with known amplitude difference and on real recordings. We found that the normalization of the windows at each time point is not useful for the amplitude information: after normalization, the amplitude of the apnea and normal windows are indistinguishable in one time since each window's own scale is used to rescale it. This is not a good picture as respiratory attenuation of the ECG signal during apnea is a part of the physiological signal to be investigated. We thus modified the normalization strategy to calculate the mean and standard deviation of the entire recording (reformulated by concatenating its windows) rather than the average and standard deviation of the windows individually and preserve the relative differences between the amplitude of the different windows in the same recording (a better version in terms of the median and interquartile range is available also for recordings with independent movement). R-peak detection was performed using the validated ECG peak detector of neurokit2, which replaced the easy-to-read summary of the data set from Table. With the RR interval sequence obtained, we removed all the very unlikely intervals (outside the range of 0.3-2.0 seconds which corresponds to 30-200 bpm), and computed 17 features for each one-minute window: eight time-domain HRV, five frequency-domain HRV measured using Welch's method on a cubic-spline interpolated RR series, three EDR measurements obtained from the amplitude of each R peak (mean, standard deviation, and dominant frequency in the 0.1-0.5 Hz respiratory band), and the raw R-peak count. We then used a correlation-based feature selection step that extracted any features that were already almost identical to other ones and found that 2 features (n_peaks and total_power) were redundant with the high-frequency power and mean heart rate, leading to 15 variables. A mutual information ranking against the target class was also computed to support the interpretability analysis (Figure 27).

Furthermore, because many OSA episodes aren’t limited to a minute, we have augmented each window’s feature vector with the corresponding feature vectors of its immediately preceding and following minutes (a context size of one minute in each direction), keeping the recording boundaries so that two different patients' data are never combined. This has enabled us to increase the feature space from 15 to 45. This is the single largest improvement in classification performance we have achieved in this work, in line with the general finding in the OSA-detection literature that context around each segment of classification is helpful to distinguish between apnea and normal breathing. [17] [34]

FEATURE SELECTION SUMMARY

GENERAL INFORMATION

Original number of features	17
Selected number of features	15

ELIMINATED FEATURES

Feature	Reason	Correlation (r)
n_peaks	Redundant with mean_HR	+0.952
total_power	Redundant with hf	+0.926

RANKING BY MUTUAL INFORMATION WITH THE CLASS

Rank	Feature	Mutual Information
1	vlf	0.1087
2	edr_std	0.0929
3	sdhr	0.0823
4	sdnn	0.0763
5	total_power	0.0605
6	pnn50	0.0519
7	lf_hf_ratio	0.0502
8	lf	0.0438
9	rmssd	0.0353
10	min_RR	0.0321
11	max_RR	0.0294
12	mean_HR	0.0241
13	n_peaks	0.0211
14	edr_dom_freq	0.0202
15	mean_RR	0.0122
16	hf	0.0117
17	edr_mean	0.0061

NOTES:

- The top 15 features (in blue) were selected based on their mutual information with the class.
- Features in gray were not selected.

Figure 27 Feature selection summary.

4.2.3.1. Family 1: Time-Domain HRV Metrics (8 features)

These measures quantify the beat-to-beat variability of the heart rate based on raw RR intervals (the time elapsed between two consecutive R-waves, expressed in seconds). Collectively, they reflect the dynamic balance between the sympathetic and parasympathetic branches of the autonomic nervous system (ANS).

4.2.3.1.1. mean_RR (Mean RR Interval)

Formula: mean(RR_i)

Unit: Seconds (s)

Physiological Meaning: The average duration of cardiac cycles within the observation window; mathematically, it is the inverse of the mean heart rate. Clinical bradycardia shifts mean_RR higher, whereas tachycardia drives it lower.

Relevance to Apnea: Obstructive sleep apnea (OSA) triggers an initial reflex bradycardia (vagal stimulation induced by acute hypoxemia), followed by compensatory tachycardia upon the resumption of breathing. A one-minute window captures the net outcome of this cyclic disruption.

4.2.3.1.2. sdnn (Standard Deviation of NN Intervals)

Formula: std(RR_i) (calculated using normal-to-normal 'NN' intervals, free of artifacts).

Unit: Seconds (s) or milliseconds (ms)

Physiological Meaning: The most comprehensive global metric of Heart Rate Variability (HRV). It encapsulates overall autonomic activity and total variability without distinguishing between specific frequency components.

Relevance to Apnea: Repetitive apneic episodes profoundly alter cardiac rhythms. sdnn can increase significantly due to wide bradycardia-tachycardia oscillations or conversely decrease if prolonged hypoxic stress saturates and blunts autonomic responsiveness.

4.2.3.1.3. rmssd (Root Mean Square of Successive Differences)

Formula: $\sqrt{\text{mean}((RR_{i+1} - RR_i)^2)}$

Unit: Seconds (s) or milliseconds (ms)

Physiological Meaning: Quantifies short-term, high-frequency variations between consecutive heartbeats. It serves as a robust, primary marker of parasympathetic (vagal) tone.

Relevance to Apnea: Apneic events suppress vagal activity during the airway occlusion, followed by a sharp vagal rebound upon micro-arousal and breathing recovery. rmssd is highly sensitive to these rapid, successive fluctuations.

4.2.3.1.4. pnn50 (Proportion of NN50)

Formula: $\frac{\text{count}(|RR_{i+1} - RR_i| > 0.05\ \text{s})}{N}$

Unit: Dimensionless (expressed as a proportion in $[0,1]$ or as a percentage).

Physiological Meaning: A discrete counterpart to rmssd that measures parasympathetic modulation by capturing the percentage of consecutive heartbeats that vary by more than 50 ms.

Relevance to Apnea: Healthy individuals exhibit high baseline pnn50 values during sleep. During an apnea event, sympathetic dominance diminishes beat-to-beat variability, causing a measurable drop in pnn50.

4.2.3.1.5. mean_HR (Mean Heart Rate)

Formula: $\frac{60}{\text{mean}(RR_i)}$

Unit: Beats per minute (bpm)

Physiological Meaning: The average heart rate calculated over the one-minute epoch.

Relevance to Apnea: Sleep apnea is frequently tied to an elevated mean heart rate driven by hypoxemia and cortical arousals. Although it shares an inverse relationship with mean_RR, machine learning models often benefit from including both because the reciprocal transformation is non-linear, allowing algorithms to map features more effectively.

4.2.3.1.6. sdhr **(Standard Deviation of Heart Rate)**

Formula: std $\left(\frac{60}{RR_i}\right)$

Unit: Beats per minute (bpm)

Physiological Meaning: Quantifies instantaneous heart rate variability, acting as a complement to sdnn within the bpm domain.

Relevance to Apnea: Apneic episodes induce large-amplitude, cyclic variations in heart rate (CVHR). sdhr directly captures the magnitude of these wide bpm swings.

4.2.3.1.6. min_RR (Minimum RR Interval)

Unit: Seconds (s)

Physiological Meaning: The shortest RR interval recorded within the minute, corresponding to the peak instantaneous heart rate.

Relevance to Apnea: The termination of an apnea event is accompanied by an autonomic "startle response" (sympathetic surge), yielding a burst of rapid heartbeats that causes a sharp drop in min_RR.

4.2.3.1.7. max_RR (Maximum RR Interval)

Unit: Seconds (s)

Physiological Meaning: The longest RR interval recorded within the minute, corresponding to the slowest instantaneous heart rate.

Relevance to Apnea: Prolonged breath-holding triggers a hypoxic vagal reflex, manifesting as a pronounced deceleration in heart rate and an elongated max_RR. Together with min_RR, it defines the dynamic range of cardiac variation during respiratory stress.

4.2.3.2. Family 2: Frequency-Domain HRV Metrics (5 features)

These features are extracted from the Power Spectral Density (PSD) of the RR interval series. To ensure mathematical validity, the raw, irregularly sampled RR intervals are uniformly resampled at 4 Hz using cubic spline interpolation, and the spectrum is estimated using Welch's method. Each frequency band maps to distinct physiological mechanisms.

4.2.3.2.1. vlf (Very Low Frequency Power)

Formula: $\int_{0.0033}^{0.04} \text{PSD}\,(f)\,df$

Physiological Meaning: Captures long-period, slow-acting homeostatic mechanisms, including thermoregulation, peripheral vasomotor activity, the renin-angiotensin-aldosterone system, and slow circadian rhythms.

Relevance to Apnea: During sleep apnea, vlf power rises substantially as systemic hypoxemia recruits slow-acting, compensatory humoral and sympathetic vasoconstrictive pathways.

4.2.3.2.2. lf (Low Frequency Power)

Formula: $\int_{0.04}^{0.15} \text{PSD}\,(f)\,df$

Physiological Meaning: A mixed frequency band heavily dominated by baroreflex activity and sympathetic tone, with a secondary contribution from the parasympathetic system.

Relevance to Apnea: The repetitive sympathetic surges linked to post-apneic micro-arousals manifest as a massive amplification of power in the LF band. Consequently, lf stands out as one of the most robust predictors in HRV-based apnea detection models.

4.2.3.2.3. hf (High Frequency Power)

Formula: $\int_{0.15}^{0.40} \text{PSD}\,(f)\,df$

Physiological Meaning: Strongly modulated by Respiratory Sinus Arrhythmia (RSA), making it a pure, established marker of parasympathetic (vagal) activity associated with normal, unobstructed breathing.

Relevance to Apnea: During undisturbed breathing, hf power remains elevated. When an obstructive event occurs, RSA is abolished by the cessation of airflow, causing hf power to collapse. This dramatic drop provides clean discriminative power for classification models.

4.2.3.2.4. lf_hf_ratio (LF/HF Ratio)

Formula: $\frac{\text{lf}}{\text{hf}+\epsilon}$ (where ϵ represents a small zero-guard value)

Physiological Meaning: The classical index used to evaluate sympathovagal balance, where elevated values signal sympathetic dominance.

Relevance to Apnea: The simultaneous surge in sympathetic activity (lf) and withdrawal of vagal activity (hf) during an apnea event shifts the autonomic equilibrium, resulting in a pronounced spike in the lf_hf_ratio.

4.2.3.2.5. total_power (Total Spectral Power)

Formula: $\text{vlf} + \text{lf} + \text{hf}$ (integrated across the entire spectrum, typically ≤ 0.40 Hz)

Physiological Meaning: Represents the net variance or total energy of the HRV signal within the specified timeframe.
Relevance to Apnea: total_power effectively quantifies the overall magnitude of autonomic stress and the broad fluctuations induced by sleep-disordered breathing.

4.2.3.3. Family 3: ECG-Derived Respiration (EDR) Features (3 Features)

4.2.3.3.1. ECG-Derived Respiration (EDR) Signal Rationale

The ECG-Derived Respiration (EDR) signal provides an indirect estimation of a patient's respiratory activity without requiring a dedicated respiratory sensor. This technique leverages the physiological reality that respiration mechanically modulates the cardiac electrical axis and the relative distance between the heart and surface electrodes.

During inhalation, variations in intrathoracic pressure and changes in the cardiac electrical axis cause a shifting of the heart relative to the surface electrodes, which modulates the amplitude of the R wave. Conversely, exhalation reverses this mechanical shift, altering the signal amplitude in the opposite direction. By tracking and sampling these peak R-wave amplitudes or measuring the geometric shifts within a narrow window centered on the R peak across successive heartbeats, a continuous respiratory waveform can be effectively reconstructed.

4.2.3.3.2. edr_mean — Mean of the Centered EDR Series

Formula:

$$\text{edr\_mean} = \frac{1}{N}\sum_{i=1}^{N}(R_i - \bar{R})$$

(Where R_i is the R-peak amplitude of the i-th beat, and $\bar{R}$ is the mean amplitude across the window.)

Clinical Meaning: This is a residual metric. By centering the EDR series around its mean (average), the value should ideally hover near zero in a stable, non-trending signal. Deviations from zero typically capture R-peak detector biases or residual low-frequency baseline drift.

Diagnostic Relevance: While its individual discriminatory power is modest, a fluctuating or offset mean frequently reflects sustained respiratory artifacts or baseline instability induced by the severe postural shifts characteristic of obstructive apnea events.

4.2.3.3.3. edr_std — Standard Deviation of the EDR Series

Clinical Meaning: This metric quantifies the overall variability of the R-wave amplitude modulation within a given one-minute window. Under normal physiological conditions, a robust EDR signal exhibits relatively stable, constant amplitude modulation. Apneic events fundamentally disrupt this homeostatic rhythm.

Diagnostic Relevance: The behavior of edr_std serves as a critical indicator of apnea type:

Obstructive Apnea: The patient's paradoxical efforts to breathe against a collapsed airway generate exaggerated intrathoracic pressure swings, causing the R-wave amplitude modulation—and consequently edr_std—to spike significantly.

Central Apnea: Because the central nervous system temporarily ceases respiratory drive, mechanical chest movements vanish entirely, driving the EDR modulation down to near-zero levels.

4.2.3.3.4. edr_dom_freq — Dominant EDR Frequency (0.1–0.5 Hz)

Formula:

$$\text{edr\_dom\_freq} = \arg\max_{f \in [0.1, 0.5]} \big(\text{PSD}(f)\big)$$

(Calculated using Welch's Periodogram Method on the EDR series resampled to 4 Hz*.)*

Clinical Meaning: This represents the patient's estimated respiratory rate expressed in Hertz. Normal adult resting respiration typically falls within the 0.20–0.28 Hz window, which translates to roughly 12–17 breaths/minute.

Diagnostic Relevance: During an apnea episode, normal breathing ceases, erasing the distinct physiological peak within this spectral band. The dominant frequency subsequently fractures or shifts dramatically outside the expected range. Machine learning models easily train on these anomalous boundaries, identifying values clustering tightly at the absolute limits (0.1 Hz or 0.5 Hz) as hallmarks of respiratory distress.

4.2.3.4. Family 4: Cardiac Event Count (1 Feature)

4.2.3.4.1. n_peaks — Detected R-Peak Count

Definition: The absolute number of R-peaks identified within the one-minute window using the neurokit2 QRS detection algorithm.

Clinical Meaning: This count serves as a direct, untransformed integer approximation of the patient's heart rate in beats per minute (bpm). Under standard physiological conditions, this value typically ranges between 40 and 180 bpm.

Diagnostic Relevance: The feature serves two primary analytical purposes:

- **Signal Quality Guardrail:** Windows presenting extreme or non-physiological peak counts (e.g., < 30 or > 180 bpm) indicate the presence of severe motion artifacts, electrode detachment, or transient signal loss. When these thresholds are breached, the mathematical integrity of all downstream Heart Rate Variability (HRV) metrics becomes compromised.

- **Raw Heart Rate Baseline:** It provides a raw, discrete metric of overall heart rate, mirroring the behavior of mean_HR without any statistical transformation.

  Implementation Note on Feature Redundancy:
  Because n_peaks is nearly perfectly correlated with mean_HR ($r \approx 0.97$), feature selection modules routinely drop it to prevent multicollinearity and optimize model training. However, it is explicitly

retained during the initial extraction phase because it functions as an invaluable data-cleaning flag: any window where n_peaks < 30 or n_peaks > 180 can be automatically isolated and discarded from the diagnostic pipeline.

The Pearson Correlation Matrix in figure 28 quantifies the linear relationships among 17 extracted physiological features for obstructive sleep apnea (OSA) classification. Correlation coefficients are -1 (perfect negative correlation) to +1 (perfect positive correlation) and close to 0 (little or no linear correlation).

The matrix shows many groups of highly correlated features, which indicates that some different variables have similar physiological information. Mean HR and n peaks have the test positive correlation (r = 0.95), which implies that the number of R-peaks is related to the average heart rate. Similarly, hf and total power show very high positive correlation (r = 0.93), meaning that high-frequency spectral component plays a major role in the overall HRV power. There are also very high positive correlations between traditional HRV features, i.e. sdnn, rmssd and pnn50 which are known to be associated with autonomic nervous system activity and heart rate variability. Likewise, lf is strongly related to hf (r = 0.65) and total power (r = 0.86), which is expected because these features are related spectral components of HRV.

There are several inverse correlations shown in the matrix. The most notable is between mean HR and mean RR (r = -0.67) indicating the physiological inverse relationship between heart rate and RR interval. There are negative correlations between sdhr and min RR (r = -0.69) and mean RR and n peaks (r = -0.66) that are expected and indicate the dependence of the HR parameters on cardiac timescales. However, respiratory features (edr_mean, edr_std and edr_dom_freq) are generally weak in many HRV variables. These features would provide respiratory information as well as cardiac information, which is a very useful feature to add to the classification model.

To further improve the correlation analysis, the following features were removed. Features with extremely high redundancy such as n peaks and total power were removed as they already contribute information to mean HR and hf. Removing these redundant variables decreases multicollinearity, simplifies the model and significantly reduces computing complexity without affecting the predictive performance as well. In general, the Pearson Correlation Matrix shows that while many HRV features are naturally related due to their origin in biology, together with some other variables, they provide complementary information. This dynamic of correlated and independent features is a good basis for building accurate and efficient machine learning models for obstructive sleep apnea detection.

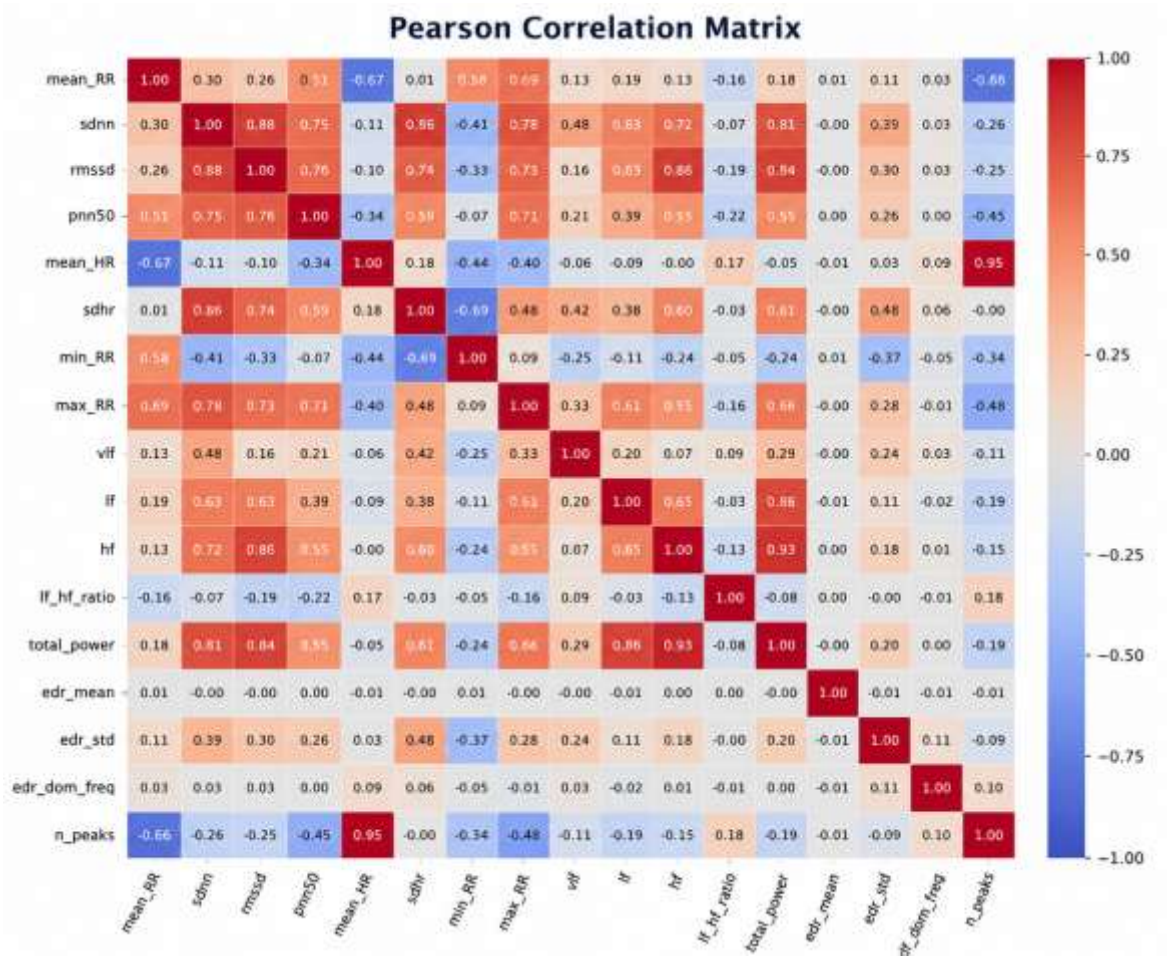

Figure 28 Pearson correlation matrix.

4.3. Implementation of models

4.3.1. Machine Learning Models

The dataset presents a class imbalance of 35.6% / 64.4% (apnea / normal) across the full training cohort (a01–c10, 12600 one-minute windows, 35 subjects). Because the train/test split is performed at the subject level (section 4.1) rather than by random window sampling, the resulting partitions do not preserve this exact ratio: the training partition (26 subjects, 9360 windows) has a prevalence of 34.3% / 65.7%, while the held-out internal test partition (9 subjects, 3240 windows) has a prevalence of 39.4% / 60.6% (1278 apnea windows). This difference is an expected consequence of subject-level partitioning—rather than an inconsistency—and is the figure that should be used when interpreting the internal-test confusion matrices reported for **Logistic Regression**, **SVM-RBF**, and **Random Forest**.

4.3.1.1. Classification with Support Vector Machines (SVM)

A Support Vector Machine with a radial basis function (RBF) kernel was implemented (C = 1.0, gamma = 'scale', probability estimates enabled to allow ROC-curve analysis and participation in the ensemble strategies discussed in Section 4.4.4). The RBF kernel was selected for its capacity to model non-linear decision boundaries in the HRV/EDR feature space without requiring an explicit feature-transformation step, an approach consistent with prior comparative studies of machine learning classifiers applied to ECG-derived features for OSA detection. [14] [28]

4.3.1.2. Classification with Random Forests

A Random Forest classifier with 300 trees, unrestricted maximum depth, and a minimum of two samples per leaf was implemented. Beyond its classification role, the Random Forest provides a natural, model-intrinsic feature-importance measure, which was exploited in the explainability analysis of Section 4.4.4 through SHAP's TreeExplainer. The use of tree-ensemble methods for OSA severity and event classification from ECG-derived features is well represented in the literature reviewed for this work. [25] [37]

4.3.1.3. Classification with Logistic Regression

An L2-regularized Logistic Regression classifier (maximum 2,000 iterations) was implemented as an interpretable linear baseline. Unlike the RBF-kernel SVM and the Random Forest, its coefficients admit a direct interpretation: the sign and magnitude of each coefficient indicate whether, and how strongly, the corresponding HRV/EDR feature increases or decreases the estimated probability of apnea, which was used in the cross-model explainability comparison of Section 4.4.4.

4.3.1.4. Hyperparameters Optimization

Table 3 summarizes the hyperparameter optimization strategy for the three conventional machine learning classifiers. **Logistic Regression** was optimized using a grid search over 6 candidate values of the regularization parameter C. **SVM-RBF** used random search with log-uniform sampling for C and the kernel coefficient γ and evaluated 15 candidate configurations. **Random Forest** used random search over the number of estimators, the depth of trees, the minimum number of samples per leaf, and the number of features that were considered at each split. 11 of the 25 initially planned configurations were evaluated (reduction note). Overall, the table shows a systematic but computationally controlled hyperparameter optimization approach, with more random search for more complex models.

Model	Optimization method	Search space	Number of candidates evaluated
Logistic Regression	Exhaustive grid search	$C \in \{0.001, 0.01, 0.1, 1, 10, 100\}$	6
SVM (RBF)	Random search (log-uniform sampling)	$C \sim \log U(0.01,\ 100)$, $\gamma \sim \log U(0.0001,\ 10)$	15
Random Forest	Random search	n_estimators ~ randint(100, 600), max_depth ∈ {None, 5, 10, 20, 30}, min_samples_leaf ~ randint(1, 10), max_features ∈ {$\sqrt{f}$, log2, None}	11 out of 25 planned (see reduction note)

Table 3. Hyperparameter optimization strategy.

Table 4 displays the hyperparameter optimization results for the three conventional machine-learning models using a 26-subject cross-validation scheme. For each classifier, the original manually selected configuration is compared with the best configuration identified through the search procedure, together with the resulting cross-validated AUC (CV-AUC).

The optimization identified C = 0.01 as the best regularization parameter for **Logistic Regression**, yielding a CV-AUC of 0.8306. For **SVM-RBF**, the optimized parameters were C = 2.538 and γ = 0.0068, resulting in the highest CV-AUC among the three models (0.8734). **Random Forest** achieved a CV-AUC of 0.8466 with 407 estimators, a maximum depth of 10, a minimum of 9 samples per leaf, and max_features = log2. Summarizing, the results demonstrate that systematic hyperparameter optimization provides well-defined configurations for subsequent model training and evaluation, with **SVM-RBF** exhibiting the strongest cross-validated discrimination among the evaluated conventional classifiers.

Model	Original manual value	Best value found	CV-AUC (search, n=26 subjects)
Logistic Regression	C=1.0 (default)	**C=0.01**	0.8306
SVM (RBF)	C=1.0, gamma='scale'	**C=2.538, gamma=0.0068**	0.8734
Random Forest	n_estimators=300, max_depth=None, min_samples_leaf=2	**n_estimators=407, max_depth=10, min_samples_leaf=9, max_features='log2'**	0.8466

Table 4. Hyperparameters optimization results.

4.3.2. Deep Learning Models

Two deep learning architectures, a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) built on bidirectional LSTM layers, were each implemented under two alternative input representations of the one-minute ECG window, to isolate the effect of input representation from the effect of model architecture.

4.3.2.1. Classification with Convolutional Neural Networks (CNN)

The raw-signal CNN receives the full 6,000-sample preprocessed waveform as a single-channel input and processes it through three one-dimensional convolutional blocks (32, 64, and 128 filters, with kernel sizes of 11, 7, and 5 respectively), each followed by batch normalization and progressive max-pooling, then global average pooling, a dropout layer, and a dense layer before a final sigmoid output.

An alternative, substantially lighter CNN receives a reduced two-channel representation instead of the raw waveform: an RR-interval tachogram and an ECG-derived respiration (EDR) signal, each linearly interpolated onto a fixed grid of 240 points per minute (equivalent to a 4 Hz sampling rate) and channel-wise standardized. This representation reduces the input dimensionality by a factor of 25 relative to the raw waveform (480 values instead of 6,000) and restricts the network's input to variables already known, from the HRV/EDR literature, to be physiologically relevant to apnea, rather than requiring the network to discover this relevance unsupervised from the full waveform — an approach broadly consistent with feature-informed deep learning designs for single-lead ECG apnea detection. [15] [25] [32]

The motivation for implementing this second representation was diagnostic: the initial raw-signal CNN exhibited a critical training failure, with near-zero sensitivity and a negative Matthews correlation coefficient on the full 35-subject test set, indicating that the model had

failed to learn a decision boundary related to the target class at all. Two causes were identified and corrected (detailed further as part of the RNN discussion below, since both architectures shared the same input pipeline and training procedure): an interaction between record-level signal normalization and the raw-waveform CNN input, and a non-representative internal validation split used for early stopping. Once corrected, the raw-signal CNN reached usable, non-degenerate performance; the RR+EDR representation subsequently provided a substantial further improvement, discussed quantitatively in Section 4.4.2.

4.3.2.2. Classification with Recurrent Neural Networks (RNN)

The raw-signal RNN receives the same 6000-sample single-channel waveform as the raw-signal CNN. Because feeding a sequence of this length directly into recurrent layers is computationally impractical and prone to poor gradient propagation, two strided one-dimensional convolutional layers first decimate the sequence length before it is passed to two bidirectional LSTM layers (64 and 32 units respectively), followed by dropout and a dense layer prior to the sigmoid output. The RR+EDR variant, given its already short 240-step input sequence, dispenses with the decimation layers and applies two bidirectional LSTM layers (32 and 16 units) directly to the two-channel tachogram, followed by the same dropout, dense, and sigmoid output structure. Both variants are trained with class-weighted binary cross-entropy loss, the Adam optimizer, early stopping monitored on validation AUC, and learning-rate reduction on plateau — consistent with recurrent and hybrid CNN-LSTM approaches to single-lead ECG apnea detection reported by Almutairi, Hassan, and Datta (2021).

As with the CNN, the initial raw-signal RNN exhibited a severe training failure (zero sensitivity, AUC below 0.65) on the full dataset. Diagnostic analysis traced this to two compounding causes. First, both raw-signal networks received their input from the record-level-normalized signal introduced in Section 4.2.3 to preserve amplitude information for the tabular HRV/EDR features; while beneficial for those features, this same normalization produces substantially different amplitude scales across different one-minute windows of the same recording (a quiet segment versus a noisy one), which destabilizes training of a fixed-architecture network expecting a consistent input scale. Second, the internal train/validation split used to monitor early stopping and learning-rate reduction selected the last 15% of the training array by raw order rather than by a random, class-stratified draw; because windows are concatenated recording by recording, this validation subset could be dominated by one or two specific recordings with an atypical class distribution, providing a misleading signal to the stopping criterion and allowing it to lock in a degenerate all-negative solution. Both issues were corrected: the raw-signal CNN and RNN were given their own window-level-normalized input, computed independently of the record-level normalization used elsewhere in the pipeline, and the internal validation split was replaced with a random, class-stratified partition. Following these corrections, both networks learned non-degenerate, class-sensitive decision boundaries, as quantified in Section 4.4.2.

4.4. Evaluation and Discussion of Results

All results reported in this section correspond to the complete 35-recording dataset (12,600 one-minute windows), evaluated on a held-out, subject-independent test partition (25% of recordings, none of which contributed any window to model training).

The contextual approach (t-1, t, t+1) enables the model to capture the dynamic behavior of apneic events and the way OSA affects ECG and EDR signals.

4.4.1. Performance of Machine Learning Models

Figure 29 and table 5 highlights distinct trade-offs in diagnostic accuracy and predictive behavior among the classical machine learning models.

Considering Apnea as the positive class and normal as the negative class, the corresponding confusion matrix values were determined for each of the classical classification models developed in this study:

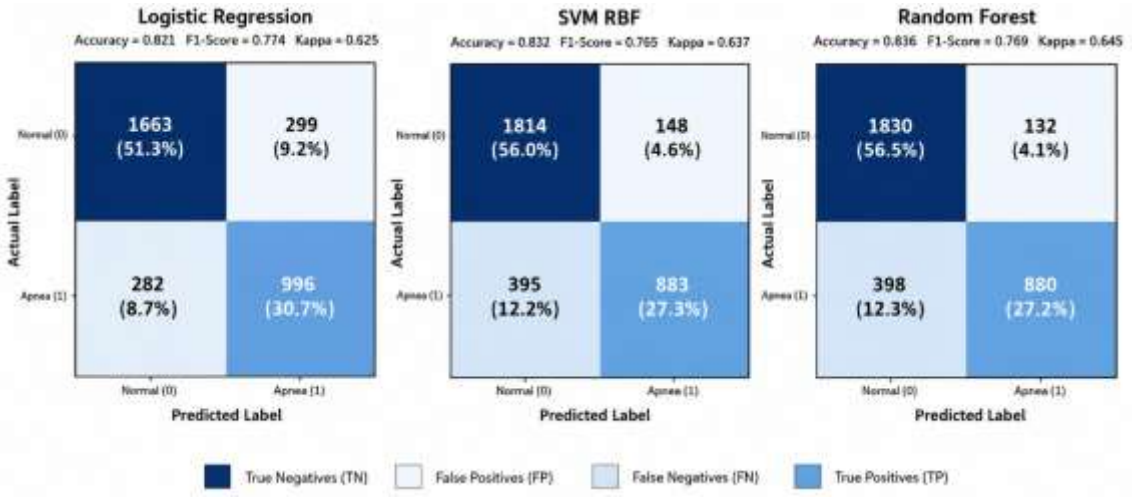

Figure 29. Confusion Matrices of Machine Learning Models.

- **Logistic Regression metrics** (TP = 996, TN = 1663, FP = 299, FN = 282):

$$Accuracy = \frac{TP + TN}{N} = \frac{996 + 1663}{3240} = 0.8207\ (82.07\%)$$

$$Precision = \frac{TP}{TP + FP} = \frac{996}{996 + 299} = 0.7691\ (76.91\%)$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{996}{996 + 282} = 0.7793\ (77.93\%)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{1663}{1663 + 299} = 0.8476\ (84.76\%)$$

$$F1 - Score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} = 2 \times \frac{0.7691 \times 0.7793}{0.7691 + 0.7793} = 0.7742\ (77.42\%)$$

- **SVM-RBF metrics** (TP = 883, TN = 1814, FP = 148, FN = 395):

$$Accuracy = \frac{TP + TN}{N} = \frac{883 + 1814}{3240} = 0.8324\ (83.24\%)$$

$$Precision = \frac{TP}{TP + FP} = \frac{883}{883 + 148} = 0.8564\ (85.64\%)$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{883}{883 + 395} = 0.6909\ (69.09\%)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{1814}{1814 + 148} = 0.9245\ (92.45\%)$$

$$F1 - Score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} = 2 \times \frac{0.8564 \times 0.6909}{0.8564 + 0.6909} = 0.7647\ (76.47\%)$$

- **Random Forests metrics** (TP = 880, TN = 1830, FP = 132, FN = 398):

$$Accuracy = \frac{TP + TN}{N} = \frac{880 + 1830}{3240} = 0.8364\ (83.64\%)$$

$$Precision = \frac{TP}{TP + FP} = \frac{880}{880 + 132} = 0.8695\ (86.95\%)$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{880}{880 + 398} = 0.6886\ (68.86\%)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{1830}{1830 + 132} = 0.9327\ (93.27\%)$$

$$F1 - Score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} = 2 \times \frac{0.8695 \times 0.6886}{0.8695 + 0.6886} = 0.7685\ (76.85\%)$$

Model	Accuracy	Precision	Recall	Specificity	F1-Score	AUC	TN	FP	FN	TP	kappa	AUC_CI95%	Accuracy Wilson95%
Logistic Regression	82,1%	76,9%	77,9%	84,8%	0,774	0,893	1663	299	282	996	0,625	[0,881 0,904]	[0,807 0,834]
SVM_RBF	83,2%	85,6%	69,1%	92,5%	0,765	0,923	1814	148	395	883	0,637	[0,913 0,931]	[0,819 0,845]
Random Forest	83,6%	87,0%	68,9%	93,3%	0,769	0,927	1830	132	398	880	0,645	[0,918 0,935]	[0,823 0,849]

Table 5. Classical Machine Learning Algorithms Performance.

According to table 5, **Random Forest** achieved the highest overall accuracy (**83.6%**, 95% Wilson CI: **[0.823, 0.849]**), precision (**87.0%**), specificity (**93.3%**), Cohen's kappa ($\boldsymbol{\kappa} = \mathbf{0.645}$), and AUC (**0.927**, 95% CI: **[0.918, 0.935]**), closely followed by **SVM_RBF** (**83.2%** accuracy, **0.923** AUC).

As far as sensitivity/specificity trade-off, Non-linear classifiers (**Random Forest** and **SVM_RBF**) demonstrated superior specificity ($\geq 92.5\%$) and lower false positive rates ($FP = 132$ and 148, respectively) but experienced a noticeable drop in recall ($\approx 69\%$).

Contrarily, **Logistic Regression** maintained a more balanced sensitivity profile, yielding the highest recall (**77.9%**, $TP = 996$) and highest F1-Score (**0.774**), albeit with lower specificity (84.8%) and a higher false positive count ($FP = 299$).

SVM_RBF and **Random Forest** showed a good discrimination ability with tight 95% AUC confidence intervals exceeding 0.910, reflecting stable classification boundary estimations across test instances.

Performance of Deep Learning Models

Figure 30 shows the confusion matrices for the implemented deep learning models and table 3 evaluate and compare the performance of four deep learning configurations for classifying binary respiratory states (**Normal** vs. **Apnea**). The models include baseline architectures (**CNN** and **RNN**) and enhanced variants that incorporate both RR intervals and ECG-derived respiration data (**CNN_RREDR** and **RNN_RREDR.**

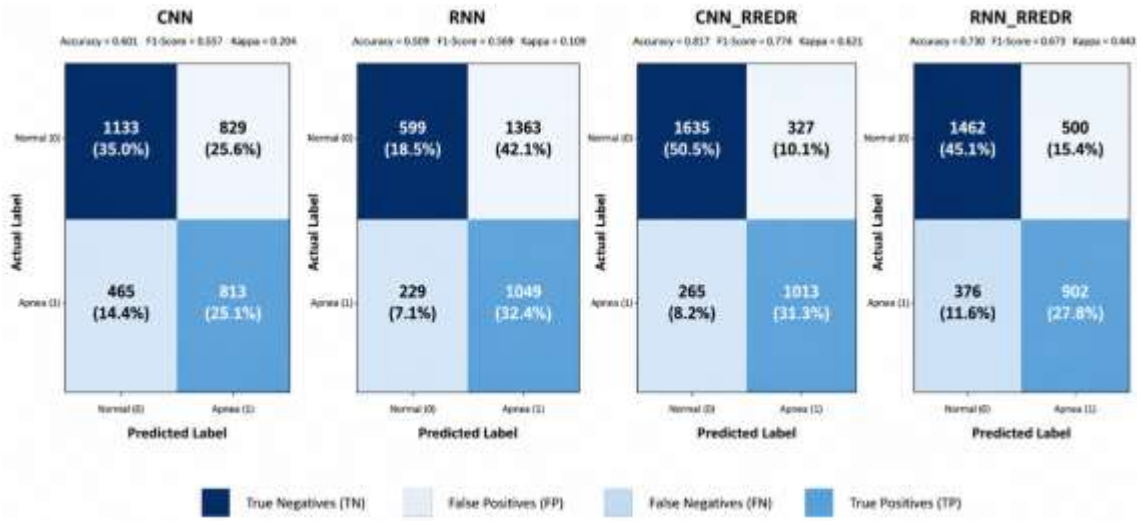

Figure 30. Confusion Matrices of Deep learning Models.

- **CNN metrics** (TP = 813, TN = 1133, FP = 829, FN = 465):

$$Accuracy = \frac{TP + TN}{N} = \frac{813 + 1133}{3240} = 0.6006\ (60.06\%)$$

$$Precision = \frac{TP}{TP + FP} = \frac{813}{813 + 829} = 0.4951\ (49.51\%)$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{813}{813 + 465} = 0.6361\ (63.61\%)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{1133}{1133 + 829} = 0.5774\ (57.74\%)$$

$$F1 - Score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} = 2 \times \frac{0.4951 \times 0.6361}{0.4951 + 0.6361} = 0.5568\ (55.68\%)$$

- **RNN metrics (**TP = 1049, TN = 599, FP = 1363, FN = 229):

$$Accuracy = \frac{TP + TN}{N} = \frac{1049 + 599}{3240} = 0.5086\ (50.86\%)$$

$$Precision = \frac{TP}{TP + FP} = \frac{1049}{1049 + 1363} = 0.4349\ (43.49\%)$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{1049}{1049 + 229} = 0.8208\ (82.08\%)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{599}{599 + 1363} = 0.3053(30.53\%)$$

$$F1 - Score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} = 2 \times \frac{0.4349 \times 0.8208}{0.4349 + 0.8208} = 0.5685\ (56.85\%)$$

- **CNN_RREDR metrics** (TP = 1013, TN = 1635, FP = 327, FN = 265):

$$Accuracy = \frac{TP + TN}{N} = \frac{1013 + 1635}{3240} = 0.8172\ (81.72\%)$$

$$Precision = \frac{TP}{TP + FP} = \frac{1013}{1013 + 327} = 0.7559\ (75.59\%)$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{1013}{1013 + 265} = 0.7926\ (79.26\%)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{1635}{1635 + 327} = 0.8333\ (83.33\%)$$

$$F1 - Score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} = 2 \times \frac{0.7559 \times 0.7926}{0.7559 + 0.7926} = 0.7738\ (77.38\%)$$

- **RNN_RREDR metrics** (TP = 902, TN = 1462, FP = 500, FN = 376):

$$Accuracy = \frac{TP + TN}{N} = \frac{902 + 1462}{3240} = 0.7296\ (72.96\%)$$

$$Precision = \frac{TP}{TP + FP} = \frac{902}{902 + 500} = 0.6433\ (64.33\%)$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{902}{902 + 376} = 0.7057\ (70.57\%)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{1462}{1462 + 500} = 0.7451\ (74.51\%)$$

$$F1 - Score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} = 2 \times \frac{0.6433 \times 0.7057}{0.6433 + 0.7057} = 0.6730\ (67.3\%)$$

As demonstrated in Table 6, the integration of ECG-Derived Respiration (EDR) data with standard RR-interval metrics yields a substantial and systematic improvement across all evaluation parameters:

Model	Accuracy	Precision	Recall	Specificity	F1-Score	AUC	TN	FP	FN	TP	kappa	AUC_CI95%	Accuracy Wilson95%
CNN	60,1%	49,5%	63,6%	57,7%	0,557	0,565	1133	829	465	813	0,204	[0,584 0,617]	[0,545 0,617]
RNN	50,9%	43,5%	82,1%	30,5%	0,569	0,581	599	1363	229	1049	0,109	[0,559 0,600]	[0,491 0,526]
CNN_RREDR	81,7%	75,6%	79,3%	83,3%	0,774	0,910	1635	327	265	1013	0,621	[0,891 0,911]	[0,804 0,830]
RNN_RREDR	73,0%	64,3%	70,6%	74,5%	0,673	0,800	1462	500	376	902	0,443	[0,784 0,815]	[0,714 0,745]

Table 6. Performance of the CNN and RNN architectures vs CNN_RREDR AND RNN_RREDR.

Table 6 summarizes the quantitative performance across the evaluated architectures **CNN**, **RNN**, **CNN_RREDR**, and **RNN_RREDR** based on standard evaluation metrics and confusion matrix components (TN, FP, FN, and TP).

The **CNN_RREDR** model achieved the highest performance across almost all evaluated parameters, attaining an overall accuracy of **81.7%** (95% Wilson CI: [0.804, 0.830]), precision of **75.6%**, specificity of **83.3%**, and an F1-score of **0.774**. Furthermore, it demonstrated strong inter-rater agreement with a Cohen's kappa (κ) of **0.621** and outstanding class separability with an AUC of **0.910** (95% CI: [0.891, 0.911]).

While the baseline **RNN** yielded a high recall (82.1%, $TP = 1049$), its low specificity (30.5%) and high false positive rate ($FP = 1363$) resulted in poor overall precision (43.5%) and weak reliability ($\kappa = 0.109$). Applying RREDR effectively rebalanced the sensitivity-specificity trade-off, elevating specificity to 74.5% and κ to 0.443 in the **RNN_RREDR** model.

These empirical results confirm that the RREDR enhancement substantially boosts classification performance over baseline architectures. Integrating RREDR into the CNN framework optimizes the balance between sensitivity and specificity, significantly enhancing model robustness and feature discrimination for this task. For the convolutional network (**CNN vs. CNN_RREDR**), accuracy improved by **21.6 percentage points** (from 60.1% to 81.7%), while the AUC increased from 0.565 to 0.910. False positive counts dropped significantly from 829 to 327.

For the recurrent network (**RNN vs. RNN_RREDR**), accuracy rose by **22.1 percentage points** (from 50.9% to 73.0%), and the AUC expanded from 0.581 to 0.800.

4.4.2. Performance of Ensemble Models

The ensemble models are composed of 3 classical models (Logistic Regression, SVM_RBF and Random Forest), the 2 base neural network models (CNN and RNN) that process raw ECG signal and the 2 enhanced neural network models (CNN_RREDR and RNN_RREDR) that incorporate the derived respiratory signal (EDR).

Figure 31 shows the confusion matrices for the 3 ensemble models developed: Ensemble Soft Voting, Ensemble Weighted AUC and Ensemble Stacking.

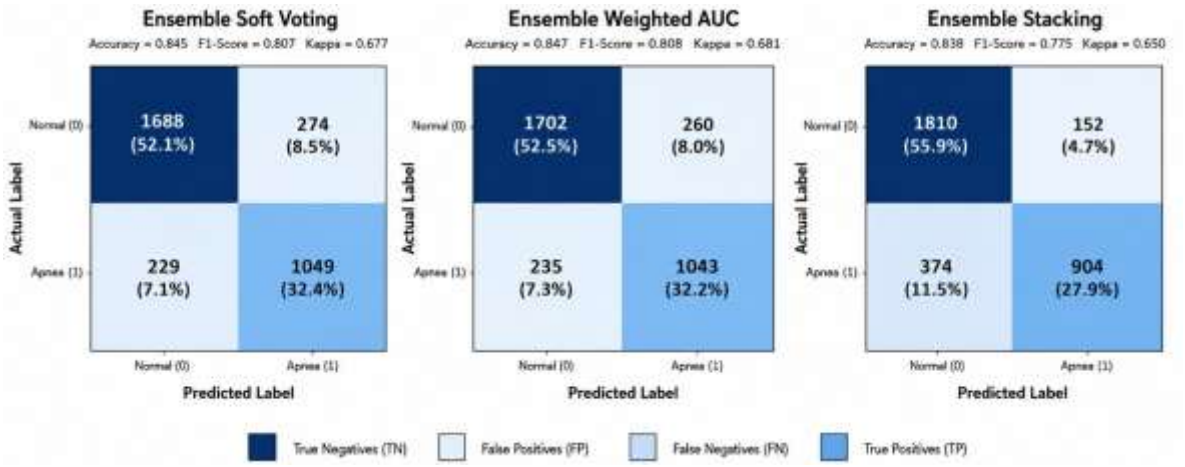

Figure 31. Confusion Matrices of Ensemble Learning Models.

- **Ensemble Sof Voting metrics** (TP = 1049, TN = 1688, FP = 274, FN = 229):

$$Accuracy = \frac{TP + TN}{N} = \frac{1049 + 1688}{3240} = 0.8447\ (84.47\%)$$

$$Precision = \frac{TP}{TP + FP} = \frac{1049}{1049 + 274} = 0.7928\ (79.28\%)$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{1049}{1049 + 229} = 0.8208\ (82.08\%)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{1688}{1688 + 274} = 0.8603\ (86.03\%)$$

$$F1 - Score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} = 2 \times \frac{0.7928 \times 0.8208}{0.7928 + 0.8208} = 0.8065\ (80.65\%)$$

- **Ensemble Weighted AUC metrics** (TP = 1043, TN = 1702, FP = 260, FN = 235):

$$Accuracy = \frac{TP + TN}{N} = \frac{1043 + 1702}{3240} = 0.8472\ (84.72\%)$$

$$Precision = \frac{TP}{TP + FP} = \frac{1043}{1043 + 260} = 0.8004\ (80.04\%)$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{1043}{1043 + 235} = 0.8161\ (81.61\%)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{1702}{1702 + 260} = 0.8674\ (86.74\%)$$

$$F1 - Score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} = 2 \times \frac{0.8004 \times 0.8161}{0.8004 + 0.8161} = 0.8081\ (80.81\%)$$

- **Ensemble Stacking metrics** (TP = 904, TN = 1810, FP = 152, FN = 374):

$$Accuracy = \frac{TP + TN}{N} = \frac{904 + 1810}{3240} = 0.8376\ (83.76\%)$$

$$Precision = \frac{TP}{TP + FP} = \frac{904}{904 + 152} = 0.8560\ (85.60\%)$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{904}{904 + 374} = 0.7073\ (70.73\%)$$

$$Specificity = \frac{TN}{TN + FP} = \frac{1810}{1810 + 152} = 0.9225\ (92.25\%)$$

$$F1 - Score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} = 2 \times \frac{0.8560 \times 0.7073}{0.8560 + 0.7073} = 0.7745\ (77.45\%)$$

Table 7 outlines several performance metrics across the models:

Model	Accuracy	Precision	Recall	Specificity	F1-Score	AUC	TN	FP	FN	TP	kappa	AUC_CI95%	Accuracy Wilson95%
Ensemble Weighted AUC	84,7%	80,0%	81,6%	86,7%	0,808	0,925	1702	260	235	1043	0,681	[0,912 0,930]	[0,834 0,859]
Ensemble Soft Voting	84,5%	79,3%	82,1%	86,0%	0,807	0,923	1688	274	229	1049	0,677	[0,909 0,929]	[0,832 0,857]
Ensemble Stacking	83,8%	85,6%	70,7%	92,3%	0,775	0,928	1810	152	374	904	0,650	[0,917 0,935]	[0,825 0,850]

Table 7. Performance of Ensemble Models.

- **Ensemble Weighted AUC** achieved the highest performance in overall accuracy (**84.7%**, 95% Wilson CI: **[0.834, 0.859]**), F1-score (**0.808**), and Cohen's kappa coefficient ($\kappa = \mathbf{0.681}$), demonstrating the most robust balance between sensitivity (81.6%) and specificity (86.7%).

- **Ensemble Soft Voting** yielded performance closely comparable to Weighted AUC, capturing the highest overall recall (**82.1%**, $TP = 1049$) and lowest false negative count ($FN = 229$), albeit with a slight compromise in precision (79.3%) and specificity (86.0%).

- **Ensemble Stacking** demonstrated superior class separability with the highest overall AUC (**0.928**, 95% CI: **[0.917, 0.935]**), highest precision (**85.6%**), and highest specificity (**92.3%**, $FP = 152$). However, this meta-classifier architecture favored precision over recall, experiencing an increase in false negatives ($FN = 374$) and a lower recall of 70.7%.

General Comparison of Models and Cross-Analysis

Model	Accuracy	Precision	Recall	Specificity	F1-Score	AUC	TN	FP	FN	TP	kappa	AUC_CI95%	Accuracy Wilson95%
Logistic Regression	82,1%	76,9%	77,9%	84,8%	0,774	0,893	1663	299	282	996	0,625	[0,881 0,904]	[0,807 0,834]
SVM_RBF	83,2%	85,6%	69,1%	92,5%	0,765	0,923	1814	148	395	883	0,637	[0,913 0,931]	[0,819 0,845]
Random Forest	83,6%	87,0%	68,9%	93,3%	0,769	0,927	1830	132	398	880	0,645	[0,918 0,935]	[0,823 0,849]
CNN	60,1%	49,5%	63,6%	57,7%	0,557	0,565	1133	829	465	813	0,204	[0,584 0,617]	[0,545 0,617]
RNN	50,9%	43,5%	82,1%	30,5%	0,569	0,581	599	1363	229	1049	0,109	[0,559 0,600]	[0,491 0,526]
CNN_RREDR	81,7%	75,6%	79,3%	83,3%	0,774	0,910	1635	327	265	1013	0,621	[0,891 0,911]	[0,804 0,830]
RNN_RREDR	73,0%	64,3%	70,6%	74,5%	0,673	0,800	1462	500	376	902	0,443	[0,784 0,815]	[0,714 0,745]
Ensemble Weighted AUC	84,7%	80,0%	81,6%	86,7%	0,808	0,925	1702	260	235	1043	0,681	[0,912 0,930]	[0,834 0,859]
Ensemble Soft Voting	84,5%	79,3%	82,1%	86,0%	0,807	0,923	1688	274	229	1049	0,677	[0,909 0,929]	[0,832 0,857]
Ensemble Stacking	83,8%	85,6%	70,7%	92,3%	0,775	0,928	1810	152	374	904	0,650	[0,917 0,935]	[0,825 0,850]

Table 8. Internal Validation Performance Metrics.

Table 8 compares 10 classification models, including individual and ensemble approaches, using standard performance metrics: accuracy, sensitivity, specificity, precision, F1-score,

AUC, and Cohen's Kappa, together with the corresponding confusion matrix counts (TN, FP, FN, and TP).

- **Ensemble Weighted AUC** demonstrated the highest overall performance across the dataset, achieving an accuracy of **84.7%** (95% Wilson CI: [0.834, 0.859]), an F1-score of **0.808**, and Cohen's kappa (κ) of **0.681**, reflecting strong statistical agreement and a balanced sensitivity-specificity profile.

- **Ensemble Soft Voting** yielded comparable results, attaining the highest overall recall among ensemble models (**82.1%**, $TP = 1049$) with a minimal false negative rate ($FN = 229$).

- Deep Learning baselines showed poor generalization, with **CNN** and **RNN** achieving overall accuracies of only 60.1% and 50.9%, respectively.

- Incorporating EDR signal improved the performance of Deep Learning models: **CNN vs. CNN_RREDR:** Accuracy increased by **21.6 percentage points** (from 60.1% to 81.7%), AUC increased from 0.565 to 0.910, and false positives dropped from 829 to 327. **RNN vs. RNN_RREDR:** Accuracy rose by **22.1 percentage points** (from 50.9% to 73.0%), accompanied by a major improvement in specificity (from 30.5% to 74.5%).

- Classical Machine Learning classifiers outperformed standalone baseline neural networks (CNN and RNN). **Random Forest** achieved the highest specificity (**93.3%**), highest precision (**87.0%**), and an AUC of **0.927** among individual non-ensemble models, though it experienced lower recall (68.9%, $FN = 398$).

- **Logistic Regression** offered a balanced baseline (82.1% accuracy, 77.9% recall, 0.774 F1-score), outpacing both unenhanced CNN and RNN models across all evaluation criteria.

- For clinical scenarios where avoiding missed diagnoses (minimizing false negatives) is paramount, **Ensemble Weighted AUC** and **Ensemble Soft Voting** provide the most effective trade-offs, successfully identifying **1043** and **1049** true positive cases while keeping false negatives to **235** and **229**, respectively.

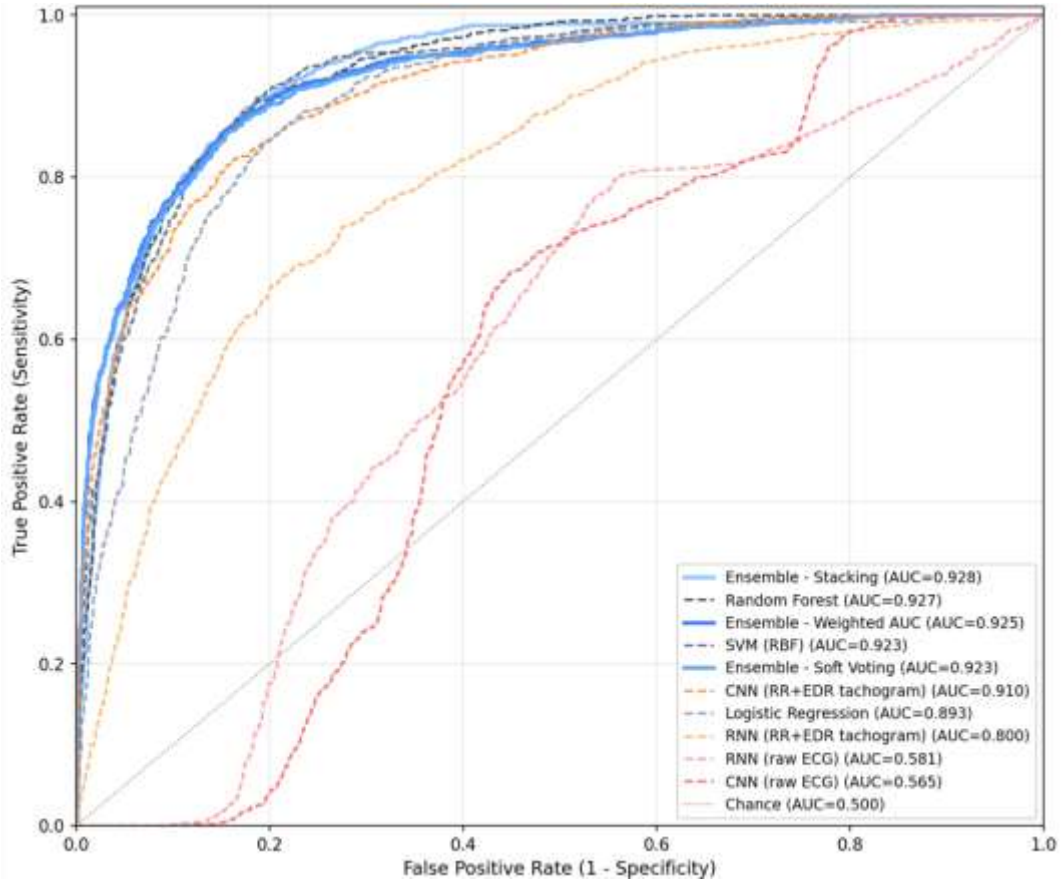

Figure 32. Receiver Operating Characteristics Curves (ROC) for Internal Validation.

Figure 32 presents the ROC curves obtained during internal validation. In general, most classifiers demonstrated good-to-excellent discriminative capability, with **Random Forest** achieving the highest AUC (0.927), followed closely by **Ensemble Weighted AUC** (0.926), **Ensemble Soft Voting** (0.925), **Ensemble Stacking** (0.925), and **SVM-RBF** (0.923). **CNN_RREDR** also demonstrated strong discrimination (AUC = 0.904), while Logistic Regression achieved an AUC of 0.893. In contrast, **RNN_RREDR** yielded a lower but still favorable AUC of 0.825, whereas models based on raw ECG exhibited substantially weaker discrimination, with **CNN** and **RNN** achieving AUC values of 0.632 and 0.595, respectively. The marked improvement observed after incorporating RR/EDR-derived respiratory information—particularly for **CNN**, whose AUC increased from 0.632 to 0.904 highlights the importance of respiratory-related representations for apnea detection. Overall, the ROC analysis indicates that ensemble learning, conventional machine-learning classifiers, and deep-learning architectures incorporating respiratory information provide substantially stronger discriminative performance than models relying exclusively on raw ECG during internal validation.

4.5. Severity Estimation via Hierarchical Aggregation

Although every classifier produces a binary apnea/normal decision for each one minute window, clinical diagnosis of OSA severity is expressed at the patient level using the Apnea-Hypopnea Index (AHI) and its associated four-tier severity scale. We implemented a second hierarchical evaluation stage on top of the minute-level classifiers to reproduce this clinically meaningful endpoint.

For each recording, the model's per-minute predictions were aggregated to an Apnea Index (AI), which is the number of minutes considered as apnea divided by the total recording duration in hours:

$$AI = \frac{\text{apnea-classified minutes}}{\text{recording duration (hours)}}$$

The resulting continuous apnea index (AI) was then mapped to the standard four-category severity scale used in clinical practice: Normal (AI < 5), Mild (5 ≤ AI < 15), Moderate (15 ≤ AI < 30), and Severe (AI ≥ 30 events/hour). The same aggregation was applied to the ground-truth per-minute annotations to obtain each recording's true AI and true severity category and predicted, and true severity can be compared directly.

This two-level hierarchy—minute-level binary classification followed by patient-level aggregation and categorical mapping—is similar to how AHI-based severity is observed from continuous overnight monitoring in clinical practice and was evaluated with a metric set different from the minute-level metrics: severity classification accuracy, Cohen's kappa between predicted and true severity category, and for the continuous AI itself mean absolute error (MAE), bias and 95% limits of agreement (bias ± 1.96•SD of the differences) after a Bland–Altman-style agreement analysis.

Table 9 presents a comparison of the performance of various machine learning and deep learning models applied to severity classification and event estimation (likely related to sleep apnea or other physiological events measured in events/hour), evaluated using five metrics: Severity Accuracy, Cohen's Kappa, AI MAE (mean absolute error in events/hour), AI Bias (average bias in events/hour), and 95% LoA (95% confidence interval).

The evaluated models include classical algorithms (**Random Forest**, **SVM-RBF**, **Logistic Regression**), neural networks (**CNN** and **RNN**, both on raw ECG signal and on RR+EDR tachogram), and different assembly strategies (**Stacking**, **Weighted AUC** with 7 models, **Soft Voting** with 7 models, and **Soft Voting**).

Main findings:

- The best-performing model overall is Random Forest, with the highest accuracy (0.889) and the highest Kappa coefficient (0.842), although it exhibits considerable negative bias (−4.93 events/h).

- The CNN (RR+EDR tachogram) achieves the lowest MAE (3.69 events/h) and low bias (+1.29), with relatively narrow limits of agreement, suggesting good precision in the quantitative estimation of events.

- The models based on raw ECG signals (CNN and RNN) show the worst performance, with low accuracies (0.556 and 0.333), very high errors (MAE of 22.04 and 28.52 events/h), and high biases, in addition to the widest ranges of agreement in the entire table (up to [−32.75, 74.75]).

- The ensemble strategies achieve intermediate and relatively stable results, with simple Soft Voting standing out, combining moderate accuracy (0.667) with a low MAE (3.76) and reduced bias (−1.87).

In general, it is observed that greater classification accuracy (Severity Accuracy) does not always translate into lower quantitative estimation error (MAE), indicating that both types of tasks—severity classification and numerical event estimation—are not necessarily correlated in model performance.

Model	Severity Accuracy	Kappa	AI MAE (events/h)	AI Bias (events/h)	95% LoA (events/h)
Random Forest	0.889	0.842	6.15	−4.93	[−23.05, 13.19]
CNN (RR+EDR tachogram)	0.778	0.679	3.69	+1.29	[−9.40, 11.95]
SVM (RBF)	0.778	0.667	6.50	−4.57	[−21.57, 12.42]
Ensemble – Stacking	0.778	0.684	5.78	−4.04	[−22.00, 13.93]
Logistic Regression	0.667	0.491	5.69	+0.31	[−13.37, 14.00]
Ensemble – Weighted AUC (7 models)	0.667	0.542	6.33	+0.63	[−19.20, 20.46]
Ensemble – Soft Voting (7 models)	0.667	0.542	6.72	+0.94	[−19.20, 21.59]
Ensemble – Soft Voting	0.667	0.500	3.76	−1.87	[−12.38, 8.64]
CNN (raw ECG)	0.556	0.265	22.04	+6.74	[−55.12, 68.60]
RNN (RR+EDR tachogram)	0.556	0.400	7.74	+2.30	[−18.28, 22.87]
RNN (raw ECG)	0.333	−0.102	28.52	+21.00	[−32.75, 74.75]

Table 9. Severity classification and Apnea Index agreement, internal test set (n = 9 subjects).

4.6. External Validation – Performance Metrics

To evaluate the generalizability of the developed classifiers beyond the data used during their training, an independent external validation process was designed and implemented using the official test set from the PhysioNet Apnea-ECG database (records x01 to x35). This set corresponds to the test partition originally defined by the organizers of the Apnea-ECG PhysioNet Challenge, and its 35 records do not share any subjects with the 35 records used to train the models (a01–a20, b01–b05, c01–c10). This complete separation at the database level—and not just at the random partition level—constitutes the most rigorous generalizability test available for this problem, as it excludes by design any possibility of information leakage between training and evaluation.

The process was structured in five stages. First, a data loading module was developed that reads the electrocardiographic signal from each test recording along with its minute-by-minute annotations (apnea/normal labels provided in an annotation file), verifying the internal consistency of the data and the completeness of all 35 recordings before proceeding. Second, the preprocessing pipeline used during training was exactly replicated: bandpass filtering of the signal, detection of R peaks, extraction of the 17-heart rate variability (HRV) features and the EDR signal (ECG-derived respiration), selection of non-redundant features through correlation analysis, expansion with a temporal context of ±1 minute, and construction of the RR+EDR tachogram for models that require it. This methodological fidelity is crucial: any discrepancy between the training preprocessing and that used for external validation would invalidate the comparison of results.

Third, the seven base models (**Logistic Regression**, **SVM-RBF**, **Random Forest**, and four deep neural networks: **CNN** and **RNN** on the original signal, and their variants on the RR+EDR tachogram) were loaded along with the three original datasets (**Soft Voting**, **Weighted AUC**, and **Stacking**), resulting in a total of ten models. Importantly, at this stage, no models were retrained or adjusted; they were loaded already trained from the original training set and used solely to generate predictions on the external data (x01–x35, 35 subjects, 12600 one-minute windows), thus preserving the validity of the assessment as a genuine test of generalizability.

Fourth, the performance of each model was evaluated at two hierarchical levels, replicating the clinical logic for diagnosing sleep apnea. At the minute-by-minute window level, accuracy, sensitivity, specificity, precision, F1-score, area under the receiver operating characteristic (ROC) curve (AUC), and Cohen's kappa coefficient were calculated, in addition to the complete confounding matrix. At the full recording level, the minute-by-minute predictions for each patient were aggregated into an Apnea Index (number of minutes with apnea divided by the recording hours), which was classified according to the standard clinical severity scale (normal, mild, moderate, severe). This allowed for the evaluation of both the accuracy of this classification and the degree of agreement (kappa) and the

estimation error of the index (MAE, bias, and limits of agreement) with respect to the true value.

Finally, since the results come from a single partition of 35 subjects, statistical significance tests were incorporated to determine whether the observed performance differences between models were attributable to a real advantage or to random chance inherent in this sample. Two complementary approaches were applied: McNemar's test on paired predictions at the window level, and a grouped bootstrap by subject (resampling with replacement of all 35 patients, not individual windows) to estimate confidence intervals and p-values for AUC differences between pairs of models. This second approach proved more appropriate for the study design, as it respects the correlation between windows for the same patient, avoiding the overestimation of statistical significance that results from treating each minute as an independent observation.

External validation not only evaluated classifier performance in an independent patient cohort, but also rigorously established whether performance variations reflected genuine predictive capability rather than sampling uncertainty.

4.6.1. Classical Machine Learning Models Performance

Figure 33 shows the confusion matrices for the implemented classical Machine Learning models (Logistic Regression, SVM RBF and Random Forest).

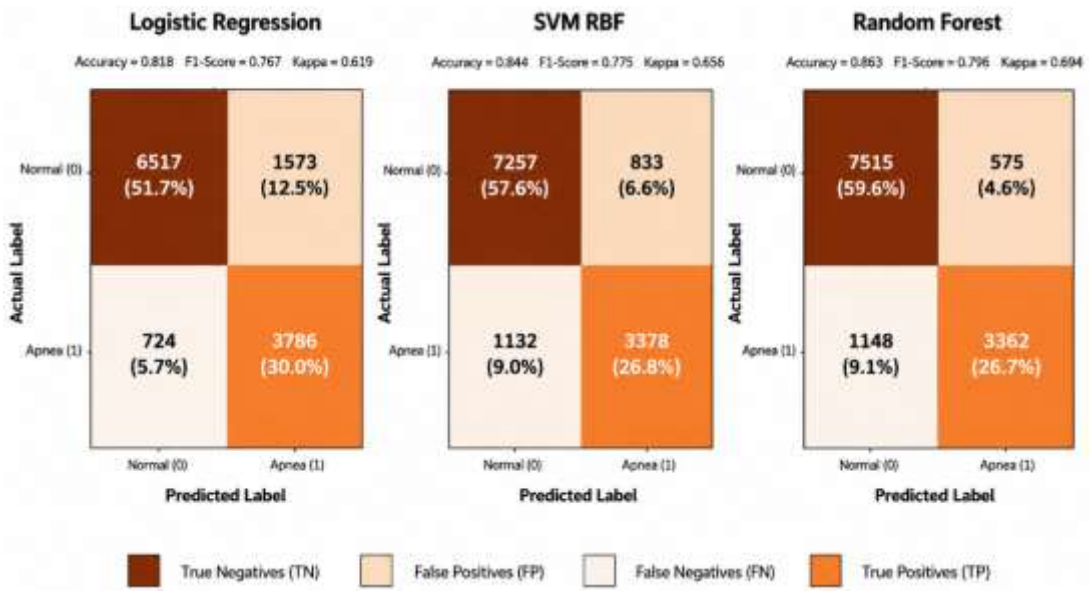

Figure 33. Confusion Matrices for classical Machine Learning Models.

- **Random Forest** achieved the highest overall diagnostic performance among classical algorithms, attaining an Accuracy of **86.3%**, an F1-Score of **0.796**, and a Cohen's kappa (κ) of **0.694**. It effectively maximized specificity, yielding the highest True Negative count ($TN = 7515$, 59.6%) and lowest False Positive rate ($FP = 575$, 4.6%), though with a higher False Negative count ($FN = 1148$, 9.1%).

- **SVM RBF** demonstrated intermediate overall performance (Accuracy = **0.844**, F1-Score = **0.775**, $\kappa = 0.656$), exhibiting a decision boundary profile like Random Forest with reduced False Positives ($FP = 833$, 6.6%) alongside an elevated False Negative rate ($FN = 1132$, 9.0%).

- Despite having the lowest overall accuracy (81.8%) and the lowest Cohen's kappa coefficient (kappa = 0.619), **Logistic Regression** prioritized diagnostic sensitivity. It successfully identified the largest number of true cases of apnea (TP = 3786; 30.0%) and reduced false negatives to 724 (5.7%), although this resulted in a higher false positive rate (FP = 1573; 12.5%).

4.6.2. Deep Learning Models Performance

Figure 34 exhibits the confusion matrices for the developed Deep Learning Models (CNN, RNN, CNN_RREDR and RNN_RREDR).

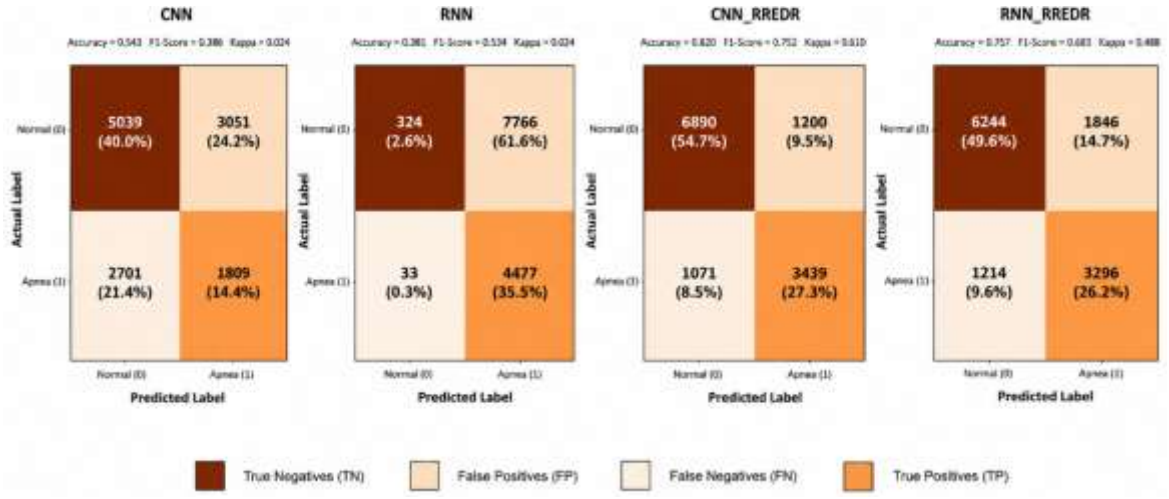

Figure 34. Confusion Matrices for Deep Learning Models.

- Standalone baseline networks demonstrated poor discrimination ability, both yielding negligible inter-rater agreement ($\kappa = 0.024$). Baseline **CNN** achieved an accuracy of only 0.543, impacted by high misclassification rates ($FP = 3051$, 24.2%; $FN = 2701$, 21.4%). Baseline **RNN** suffered from a severe positive-class bias, yielding high sensitivity ($TP = 4477$, 35.5%; $FN = 33$, 0.3%) at the expense of excessive False Positives ($FP = 7766$, 61.6%), leading to a low overall accuracy of 0.381.

- The integration of the EDR signal substantially improved generalization across both neural network's architectures. **CNN_RREDR** emerged as the top-performing deep learning model, attaining an Accuracy of **0.820**, an F1-Score of **0.752**, and a Cohen's kappa (κ) of **0.610**. It markedly suppressed False Positives ($FP = 1200$, 9.5%) while sustaining strong sensitivity ($TP = 3439$, 27.3%). **RNN_RREDR** successfully resolved the baseline RNN's prediction bias, elevating accuracy to **0.757** ($\kappa = 0.488$, F1-Score = 0.683) by reducing False Positives from 7766 (61.6%) to 1846 (14.7%).

4.6.3. Ensemble models Performance

Figure 35 displays the confusion matrices for Ensemble models (Ensemble Soft Voting, Ensemble Weighted AUC and Ensemble Stacking).

Figure 35. Confusion Matrices for Ensemble Models.

- **Ensemble Weighted AUC** achieved the highest overall predictive capability among all evaluated architectures, reaching an Accuracy of **0.877**, an F1-Score of **0.834**, and a Cohen's kappa (κ) of **0.736**. It established an optimal balance between sensitivity and specificity, identifying 3895 True Positives ($TP = 30.9\%$) while restricting False Negatives to 615 ($FN = 4.9\%$) and False Positives to 939 ($FP = 7.5\%$).

- **Ensemble Soft Voting** prioritized diagnostic sensitivity, capturing the highest True Positive count ($TP = 3920$, 31.1%) and minimizing undetected apnea instances ($FN = 590$, 4.7%). It attained an overall Accuracy of **0.872**, an F1-Score of **0.829**, and $\kappa = 0.727$, with a False Positive count of 1022 ($FP = 8.1\%$).

- **Ensemble Stacking** shifted the decision threshold toward specificity, maximizing the True Negative count ($TN = 7526$, 59.7%) and recording the lowest False Positive rate ($FP = 564$, 4.5%). However, this conservative threshold increased missed events ($FN = 1136$, 9.0%), yielding an overall Accuracy of **0.865**, an F1-Score of **0.799**, and $\kappa = 0.698$.

4.6.4. External Validation Summary

Table 10 details the performance metrics obtained during external validation across classical machine learning, deep learning, and ensemble architectures on an independent patient cohort.

Model	Accuracy	Precision	Recall	Specificity	F1-Score	AUC	TN	FP	FN	TP	kappa	CI95%
Logistic Regression	81,8%	70,6%	83,9%	80,6%	0,767	0,902	6517	1573	724	3786	0,619	[0,605 0,633]
SVM_RBF	84,4%	80,2%	74,9%	89,7%	0,775	0,917	7257	833	1132	3378	0,656	[0,642 0,670]
Random Forest	86,3%	85,4%	74,5%	92,9%	0,796	0,933	7515	575	1148	3362	0,694	[0,681 0,707]
CNN	54,3%	37,2%	40,1%	62,3%	0,386	0,606	5039	3051	2701	1809	0,024	[0,006 0,041]
RNN	38,1%	36,6%	99,3%	4,0%	0,534	0,498	324	7766	33	4477	0,024	[0,020 0,027]
CNN_RREDR	82,0%	74,1%	76,3%	85,2%	0,752	0,896	6890	1200	1071	3439	0,610	[0,596 0,625]
RNN_RREDR	75,7%	64,1%	73,1%	77,2%	0,683	0,847	6244	1846	1214	3296	0,488	[0,472 0,503]
Ensemble Weighted AUC	87,7%	80,6%	86,4%	88,4%	0,834	0,939	7151	939	615	3895	0,736	[0,724 0,748]
Ensemble Soft Voting	87,2%	79,3%	86,9%	87,4%	0,829	0,938	7068	1022	590	3920	0,727	[0,715 0,740]
Ensemble Stacking	86,5%	85,7%	74,8%	93,0%	0,799	0,935	7526	564	1136	3374	0,698	[0,685 0,711]

Table 10. External Validation Performance Metrics.

Ensemble Weighted AUC achieved the highest overall external validation performance, securing an Accuracy of **87.7%**, F1-Score of **0.834**, AUC of **0.939**, and Cohen's kappa of $\kappa = \mathbf{0.736}$ (95% CI: [0.724, 0.748]).

Baseline unenhanced deep learning models suffered severe performance degradation during external testing (CNN Accuracy = 54.3%, $\kappa = 0.024$; RNN Accuracy = 38.1%, $\kappa = 0.024$). Integrating the **RREDR** strategy successfully mitigated out-of-distribution failure, elevating **CNN_RREDR** to **82.0%** accuracy ($\kappa = 0.610$) and **RNN_RREDR** to **75.7%** accuracy ($\kappa = 0.488$).

Random Forest proved to be the most resilient single-model classifier, outperforming all individual neural network architectures with an Accuracy of **86.3%**, an AUC of **0.933**, and $\kappa = 0.694$ (95% CI: [0.681, 0.707]).

Figure 36 illustrates the ROC curves obtained during external validation on 12,600 one-minute windows from the independent x01−x35 dataset. The ensemble models exhibited the strongest overall discriminative performance, with **Ensemble Weighted AUC** achieving the highest AUC (0.940), followed by **Ensemble Soft Voting** (0.939) and Ensemble Stacking (0.934). Among the individual classifiers, Random Forest (AUC = 0.933), **SVM-**

RBF (AUC = 0.917), and **Logistic Regression** (AUC = 0.902) also demonstrated excellent discrimination. Incorporating respiratory-related information substantially improved the deep-learning models, with CNN using RR/EDR-derived information achieving an AUC of 0.896 and RNN reaching 0.817. In contrast, models based on raw ECG exhibited markedly poorer discrimination, with CNN achieving an AUC of 0.594 and RNN an AUC of 0.498, the latter being essentially equivalent to chance. Overall, the ROC analysis demonstrates that ensemble learning, and respiratory-informed representations provide substantially higher discriminative capability for minute-level apnea detection than models relying exclusively on raw ECG. The proximity of the ensemble ROC curves to the upper-left corner further supports their robust classification performance on the independent external dataset.

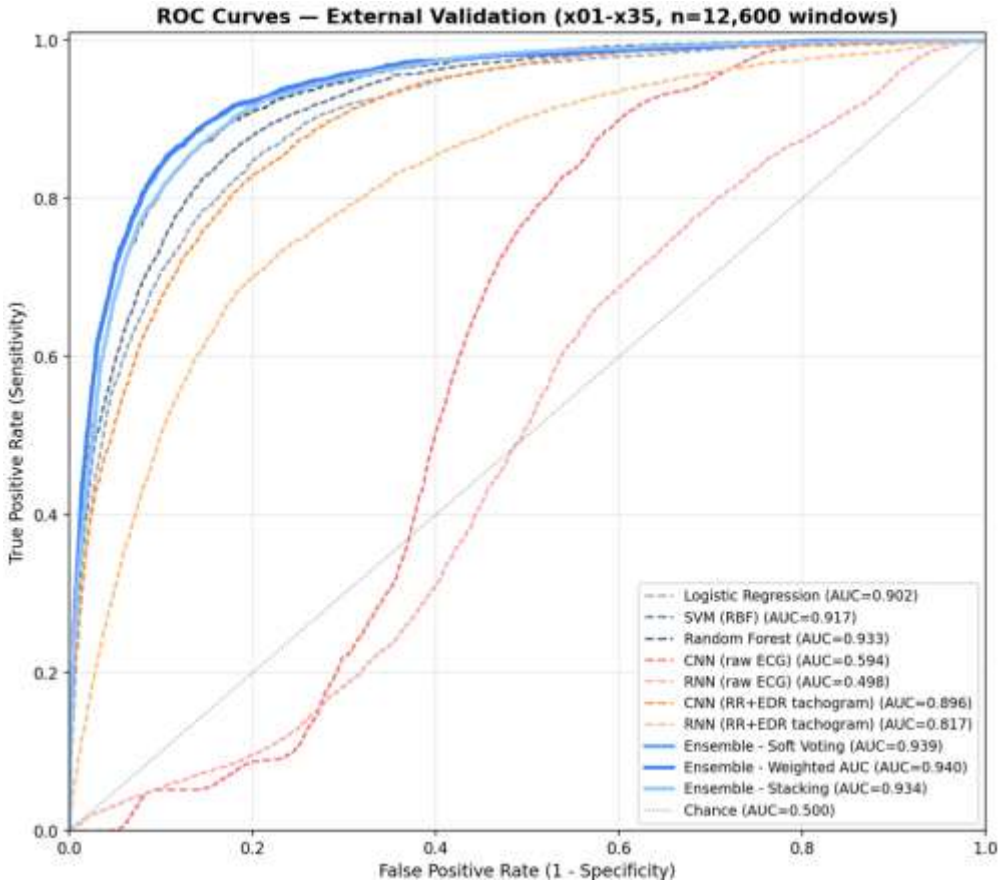

Figure 36. Receiver Operating Characteristics Curves (ROC) for External Validation.

4.7. Patient-Level Results: Severity Classification

Table 11 shows the performance of the ten classification and regression models on the external test set (n=35 subjects) and provides an independent assessment of generalizability beyond the internal validation cohort. For each classification model, severity accuracy is given by the severity and Cohen's Kappa (κ) values, and agreement in the continuous prediction of the Apnea Index (AI) is given by the mean absolute error (MAE), bias, and 95% limits of agreement (LoA) in events/h. As shown in the table legend, the positive AI Bias indicates that the model overstates the Apnea Index.

On the external test set, **Random Forest** and the **Weighted-AUC** ensemble (7 models) were the most accurate for classification of severity, with κ = 0.725 and 0.719 (significantly higher than the reference severity labels). Of all models considered, the **Weighted-AUC** ensemble had the lowest AI estimation error (MAE = 4.04 events/h) and a relatively narrow 95% LoA ([-8.43, 11.89] events/h), which is the best overall accuracy of classification and quantitative agreement of the data. **Soft Voting** (7 models) and **SVM-RBF** had a lower accuracy but still a competitive score (0.77 and 0.74; κ 0.679 and 0.648).

On the other hand, the models using only raw ECG information showed a stark dropoff in generalization. **CNN** (raw ECG) and **RNN** (raw ECG) models had the lowest severity accuracy (0.40 and 0.37) and the lowest agreement with reference labels (κ = 0.133 and 0.000, which represents chance-level agreement). These models also made the largest estimation errors (MAE = 23.90 and 36.82 events/h) and 95% limits of agreement (up to [−60.43, 63.77] and [−1.71, 75.36] events/h), indicating that the model's estimates are highly variable and that they are very poor at predicting AI severity from external data. In the case of the RNN (raw ECG), a significant positive bias (36.82 events/h) was observed, and the model overestimates AI severity on external data in a clinically relevant way.

Models based on the RR+EDR tachogram representation (**CNN_RREDR** and **RNN_RREDR**) performed intermediate-to-low externally (accuracy 0.63 and 0.40; κ = 0.482 and 0.211), worse than both the classical machine-learning models and the ensemble strategies, despite having better internal validation metrics for some architectures.

Overall, these external validation results suggest that ensemble models and classical models such as **Random Forest** can generalize to unseen subjects and have good classification accuracy and low estimation error for the Apnea Index. On the other hand, deep learning models trained on raw ECG signals, and to a lesser extent RR+EDR tachograms, are less generalizable, more biased, and have wider error ranges, highlighting the importance of external validation in assessing the clinical reliability of AI-based severity and AI-index estimation models.

Model	Severity Accuracy	Kappa	AI MAE (events/h)	AI Bias (events/h)	95% LoA
Random Forest	0.80	0.725	4.50	−2.73	[−14.50, 9.04]
Ensemble – Weighted AUC (7 models)	0.80	0.719	4.04	+1.73	[−8.43, 11.89]
Ensemble – Soft Voting (7 models)	0.77	0.679	4.15	+2.27	[−7.92, 12.46]
SVM (RBF)	0.74	0.648	5.12	−1.42	[−14.87, 12.02]
Ensemble – Stacking	0.71	0.615	4.67	−2.76	[−14.97, 9.45]
CNN (RR+EDR tachogram)	0.63	0.482	7.53	+1.02	[−20.11, 22.16]
Logistic Regression	0.51	0.358	6.60	+4.04	[−10.10, 18.19]
RNN (RR+EDR tachogram)	0.40	0.211	9.38	+3.01	[−19.70, 25.72]
CNN (raw ECG)	0.40	0.133	23.90	+1.67	[−60.43, 63.77]
RNN (raw ECG)	0.37	0.000	36.82	+36.82	[−1.71, 75.36]

AI: Apnea Index; MAE: Mean Absolute Error; LoA: Limits of Agreement.
Positive AI Bias indicates overestimation of the Apnea Index by the model.

Table 11. Severity classification and Apnea Index agreement, external test set (n = 35 subjects).

4.8. Tests of Statistical Significance Between Models (Using x01-x35)

Two complementary tests were applied to the predictions of the external test set (12600 windows, 35 subjects) for the 45 possible comparisons among the 10 models:

- McNemar (window level): Treats each of the 12600 windows as an independent observation.

  Table 12 shows pairwise statistical comparisons using the paired McNemar test on the external validation set (12600 one-minute windows from 35 subjects). Of the 45 pairwise model comparisons, 43 (95.6%) showed statistically significant differences ($p < 0.05$). Only **Logistic Regression** versus **CNN_RREDR** ($p = 0.6402$) and Random Forest versus **Ensemble Stacking** ($p = 0.2443$) were not statistically significant, indicating similar patterns of paired classification errors. The remaining comparisons yielded predominantly highly significant differences ($p < 0.001$), particularly between the conventional/deep-learning models and the ensemble approaches. The three ensemble methods also differed significantly from one another, including **Soft Voting** versus Weighted AUC ($p < 0.001$), Soft Voting versus Stacking ($p = 0.0194$), and **Weighted AUC** versus **Stacking** ($p < 0.001$). These findings indicate that most classifiers produce statistically distinguishable window-level prediction patterns on the independent external dataset. However, because multiple windows originate from the same subject, the standard McNemar test does not account for within-subject dependence. Therefore, subject-level paired bootstrap analysis with 10,000 resamples should be used as the primary inferential procedure, while the conventional McNemar results should be interpreted as complementary evidence.

Model A	Model B	A right / B wrong	A wrong / B right	McNemar	p Value	Yesgnificant (p<0.05)
LogisticRegresYeson	SVM_RBF	747	1079	60,0005	9,4831E-15	Yes
LogisticRegresYeson	RandomForest	738	1312	160,1605	1,0437E-36	Yes
Model A	Model B	A right / B wrong	A wrong / B right	McNemar	p Value	Yesgnificant (p<0.05)
LogisticRegresYeson	SVM_RBF	747	1079	60,0005	9,4831E-15	Yes
LogisticRegresYeson	RandomForest	738	1312	160,1605	1,0437E-36	Yes
LogisticRegresYeson	CNN	4613	1158	2067,2528	0,0000E+00	Yes
LogisticRegresYeson	RNN	6255	753	4318,0652	0,0000E+00	Yes
LogisticRegresYeson	CNN_RREDR	1417	1443	0,2185	0,6402	No
LogisticRegresYeson	RNN_RREDR	2018	1255	177,4042	1,7873E-40	Yes
LogisticRegresYeson	Ensemble_SoftVoting	402	1087	314,2062	2,6456E-70	Yes
LogisticRegresYeson	Ensemble_WeightedAUC	381	1124	365,8233	1,5192E-81	Yes
LogisticRegresYeson	Ensemble_Stacking	821	1418	158,6494	2,2323E-36	Yes
SVM_RBF	RandomForest	456	698	50,3302	1,2994E-12	Yes
SVM_RBF	CNN	4735	948	2522,2235	0,0000E+00	Yes
SVM_RBF	RNN	6955	1121	4212,963	0,0000E+00	Yes
SVM_RBF	CNN_RREDR	1530	1224	33,7781	6,1769E-09	Yes
SVM_RBF	RNN_RREDR	2240	1145	353,5705	7,0735E-79	Yes
SVM_RBF	Ensemble_SoftVoting	523	876	88,5661	4,9164E-21	Yes
SVM_RBF	Ensemble_WeightedAUC	457	868	126,8679	1,9856E-29	Yes
SVM_RBF	Ensemble_Stacking	569	834	49,6764	1,8131E-12	Yes
RandomForest	CNN	4732	703	2985,2408	0,0000E+00	Yes
RandomForest	RNN	7226	1150	4406,1157	0,0000E+00	Yes
RandomForest	CNN_RREDR	1534	986	118,7337	1,1978E-27	Yes
RandomForest	RNN_RREDR	2313	976	542,6865	4,9089E-120	Yes
RandomForest	Ensemble_SoftVoting	633	744	8,7872	0,0030	Yes
RandomForest	Ensemble_WeightedAUC	561	730	21,8621	2,9296E-06	Yes
RandomForest	Ensemble_Stacking	167	190	1,3557	0,2443	No
CNN	RNN	4723	2676	565,7678	4,6767E-125	Yes
CNN	CNN_RREDR	1091	4572	2138,5132	0,0000E+00	Yes
CNN	RNN_RREDR	1625	4317	1218,6942	5,2790E-267	Yes
CNN	Ensemble_SoftVoting	543	4683	3278,0943	0,0000E+00	Yes
CNN	Ensemble_WeightedAUC	512	4710	3373,1921	0,0000E+00	Yes
CNN	Ensemble_Stacking	586	4638	3141,3861	0,0000E+00	Yes
RNN	CNN_RREDR	1063	6591	3991,0804	0,0000E+00	Yes
RNN	RNN_RREDR	1240	5979	3109,6612	0,0000E+00	Yes
RNN	Ensemble_SoftVoting	579	6766	5209,8837	0,0000E+00	Yes
RNN	Ensemble_WeightedAUC	604	6849	5231,1196	0,0000E+00	Yes
RNN	Ensemble_Stacking	1137	7236	4441,1327	0,0000E+00	Yes
CNN_RREDR	RNN_RREDR	1775	988	224,8982	7,7269E-51	Yes
CNN_RREDR	Ensemble_SoftVoting	602	1261	232,4015	1,7850E-52	Yes
CNN_RREDR	Ensemble_WeightedAUC	568	1285	276,6627	4,0074E-62	Yes
CNN_RREDR	Ensemble_Stacking	900	1471	137,0308	1,1873E-31	Yes
RNN_RREDR	Ensemble_SoftVoting	558	2006	816,6182	1,3153E-179	Yes
RNN_RREDR	Ensemble_WeightedAUC	554	2060	866,4977	1,8835E-190	Yes
RNN_RREDR	Ensemble_Stacking	956	2316	564,4502	9,0482E-125	Yes
Ensemble_SoftVoting	Ensemble_WeightedAUC	26	84	29,5364	5,4877E-08	Yes
Ensemble_SoftVoting	Ensemble_Stacking	736	648	5,4689	0,0194	Yes

Table 12. Mcnemar pairwise models comparison.

- Subject-grouped bootstrap (AUC level): Resampling with replacement of the 35 subjects (not the windows) 2000 iterations — recalculating the AUC of each model at each resampling, with the same resampling for all models at each iteration (paired bootstrap). This respects the fact that approximately 360 windows for the same patient are correlated with each other and are not independent observations.

  Figure 37 summarizes the subject-level AUC performance obtained on the external validation dataset using 10,000 subject-level bootstrap resamples across 35 subjects. The ensemble approaches achieved the highest discriminative performance, with **Ensemble Weighted AUC** and **Ensemble Soft Voting** reaching AUC values of approximately 0.94, followed by **Ensemble Stacking** at approximately 0.93. Among the individual classifiers, **Random Forest** and **SVM RBF** also demonstrated excellent discrimination, with AUC values of approximately 0.93 and 0.92, respectively, while **Logistic Regression** achieved approximately 0.90.

Incorporating respiratory-related information substantially improved the deep-learning models: **CNN** (RR+EDR) reached an AUC of approximately 0.90 and **RNN** (RR+EDR) approximately 0.82. In contrast, models trained using raw ECG alone exhibited considerably lower discrimination, with CNN achieving approximately 0.60 and RNN approximately 0.50, the latter approaching chance-level performance. Overall, the results demonstrate that ensemble learning and the incorporation of respiratory-related information substantially improve subject-level discrimination and provide more robust generalization than models based solely on raw ECG representations. The use of subject-level bootstrap confidence intervals further accounts for the dependence among repeated one-minute observations within individual subjects.

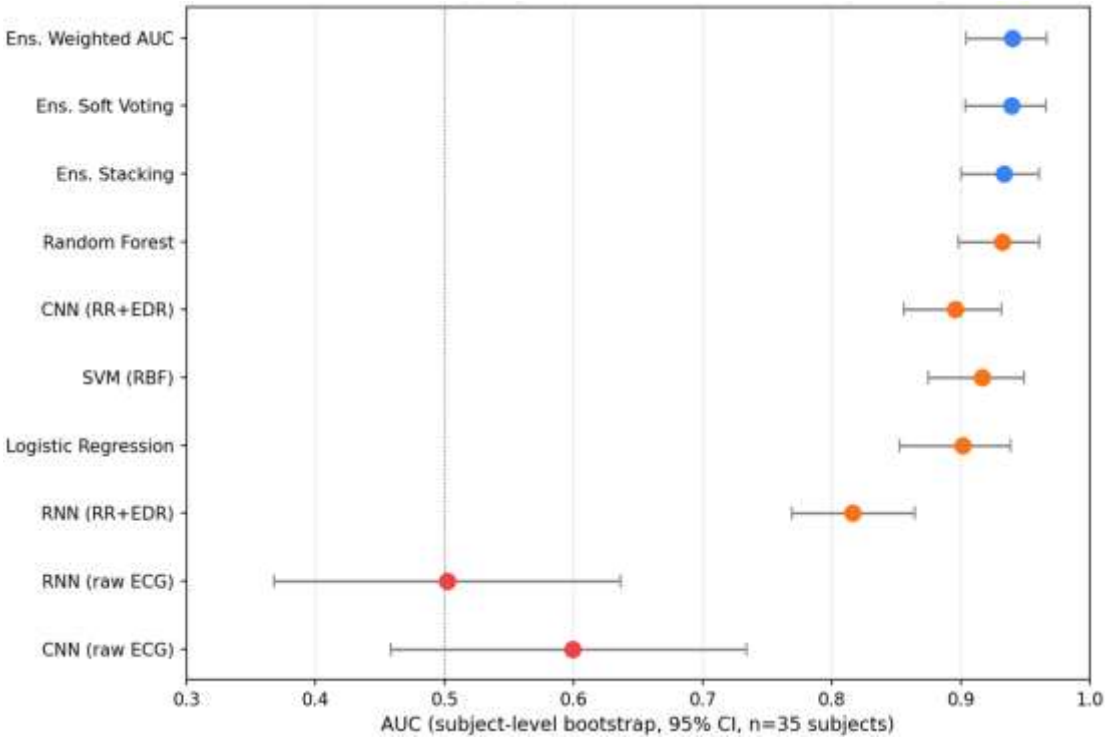

Figure 37. Subject-grouped bootstrap AUC level.

4.9. State-of-the-art Comparison

Table 13 presents a quantitative comparison between previously reported approaches using the PhysioNet Apnea-ECG database and the models evaluated in this thesis. Previous studies report accuracies ranging from approximately 77.3% to 97.41%, with the highest value obtained by Chen et al. using an **SVM-RBF** approach. Within the present study, internal validation of the Random Forest model yielded an accuracy range of approximately 84.7–84.4%, with specificity between 91.2% and 93.3% and an AUC of 0.927.

Under the independent external validation using records x01–x35, the ensemble approaches achieved the best overall performance. The **Weighted AUC ensemble** obtained the highest accuracy (87.7%), F1-score (0.834), and AUC (0.940), followed closely by Soft Voting (87.2%, 0.829, and 0.939, respectively). Random Forest achieved 86.3% accuracy, an F1-score of 0.796, and an AUC of 0.933. These results indicate that the proposed ensemble strategies provide competitive performance while demonstrating generalization on an independent test set. Direct comparison with previous studies should nevertheless be interpreted cautiously because of differences in feature representations and evaluation protocols.

Study	Year	Input / Features	Classifier	Evaluation protocol	Accuracy (%)	Sensitivity / Recall (%)	Specificity (%)	F1-score (%)	AUC †
Previous studies									
de Chazal et al. [37]	2004	ECG, RR + EDR	Linear Discriminant	35 training / 35 independent validation	91.0	—	—	—	—
Almazaydeh et al. [13]	2012	ECG/RR features	Linear SVM	80% training / 20% testing	96.5	92.9	100.0	—	—
Chen et al. [36] ‡	2015	RR intervals + severity index + demographic information	SVM-RBF	Independent sampling: 59 training / 31 testing; 10,000 trials	97.41	98.99	92.87	—	—
Sheta et al. [28] (ML)	2021	HRV features	Optimized ML	10-fold CV	~77.3	—	—	—	~0.68
Sheta et al. [28] (DL)	2021	HRV-derived representation	CNN-LSTM	5-fold CV / validation	86.25	88.79	—	87.68	0.951
Razi et al. [26] §	2021	HRV features + LDA	Random Forest	A/B/C evaluation §	95.01	92.17	94.79	—	—
Present thesis (this work)									
Internal validation (cross-validation)									
Internal validation (range)	2026	HRV + EDR + temporal context ($t-1$, t, $t+1$)	Random Forest	Internal 5-fold subject-stratified CV (training set)	~84.7–84.4	—	91.2–93.3	~0.80	0.927
External validation (independent test set: x01–x35)									
Logistic Regression	2026	HRV + EDR + temporal context ($t-1$, t, $t+1$)	Logistic Regression	Independent x01–x35	81.8	—	—	—	0.902
SVM-RBF	2026	HRV + EDR + temporal context ($t-1$, t, $t+1$)	SVM-RBF	Independent x01–x35	84.4	—	—	—	0.917
Random Forest	2026	HRV + EDR + temporal context ($t-1$, t, $t+1$)	**Random Forest**	**Independent x01–x35**	**86.3**	—	—	**0.796**	**0.933**
CNN	2026	Raw ECG (1 min)	CNN	Independent x01–x35	54.3	—	—	—	0.594
RNN	2026	Raw ECG (1 min)	RNN	Independent x01–x35	38.1	—	—	—	0.498
CNN-RREDR	2026	RR + EDR	CNN	Independent x01–x35	82.0	—	—	0.752	0.896
RNN-RREDR	2026	RR + EDR	RNN	Independent x01–x35	75.7	—	—	0.683	0.817
Ensemble – Soft Voting	2026	HRV + EDR + temporal context ($t-1$, t, $t+1$)	Soft Voting	Independent x01–x35	87.2	—	—	0.829	0.939
Ensemble – Weighted AUC	2026	HRV + EDR + temporal context ($t-1$, t, $t+1$)	**Weighted AUC**	**Independent x01–x35**	**87.7**	—	—	**0.834**	**0.940**
Ensemble – Stacking	2026	HRV + EDR + temporal context ($t-1$, t, $t+1$)	Stacking	Independent x01–x35	86.5	—	—	0.799	0.934

† AUC: Area under the ROC curve. —: Not reported.

‡ Chen et al. [36] incorporated additional admission information (age, BMI, sex) and used a broader cohort (90 subjects), therefore direct comparison is limited.

§ Razi et al. [26] evaluated using the Apnea-ECG A/B/C approach; A set: apnea patients, B set: borderline patients, C set: control patients.

Table 13. Quantitative comparison with state-of-the-art studies using the Physionet Apnea-ECG database.

4.10. Computational Time: Training and Inference

4.10.1. Measurement Hardware

All measurements were performed in the same environment used for the rest of the project: 1 CPU (Intel Xeon @ 2.10GHz), without a GPU. This is important for interpreting the numbers: deep learning models would benefit substantially from a GPU (typically 10-50x faster in training). The times reported here represent a conservative upper bound, not an estimate of what it would take on typical production hardware (dedicated GPU or at least multi-core).

4.10.2. Methodology

Training (classical models): full fit() time over the 9360 training windows (26 subjects), with the hyperparameters deployed (manual values from the original notebook, not those from the hyperparameter search).

Training (deep models): epoch time measured directly (fit(epochs=1)), multiplied by the actual number of epochs with which each model completed its training (with the split corrected, without data leakage—see documentation for that correction). This does not include the architecture search time (one was not performed; see project limitations).

4.10.3. Training and Inference Time

Inference: predict()/predict_proba() time across the 3240 windows of the internal test set, averaged over 5 repetitions (classical models) or 1 direct measurement (deep models, given their cost). Both the total and the average time per window are reported.

Ensembles: The reported training/inference time is only for the combination stage (averaging, weighting, or processing the meta-model)—it does not include the time for the 7 base models, which is reported separately and must be summed to obtain the true end-to-end cost of using an ensemble.

Table 14 summarizes the computational training requirements of the evaluated classification models. **Logistic Regression** exhibits the lowest training time (0.1 s), followed by **Random Forest** (11.7 s) and **SVM-RBF** (12.2 s), whereas the deep-learning models require substantially greater computational time, ranging from 68.7 s for **CNN_RREDR** to 684.4 s for the raw-signal **CNN**. Among the ensemble methods, **Soft Voting** requires no additional training, while **Weighted-AUC** and **Stacking** require only 25 ms and 8.6 ms, respectively. Overall, the results highlight a clear computational advantage of classical and ensemble approaches over deep-learning architectures, particularly when considering the substantially longer training times associated with neural-network models.

Model	Training Time	Details
Logistic Regression	**0.1 s**	Direct convergence, 9,360 samples × 45 features.
Random Forest	**11.7 s**	300 trees, 9,360 samples.
SVM (RBF)	**12.2 s**	9,360 samples, probability=True.
CNN_RREDR	**68.7 s**	4.58 s/epoch × 15 epochs (early stopping).
RNN_RREDR	**630.3 s** (10.5 min)	45.0 s/epoch × 14 epochs (early stopping).
RNN (raw signal)	**529.2 s** (8.8 min)	88.2 s/epoch × 6 epochs (early stopping).
CNN (raw signal)	**684.4 s** (11.4 min)	85.6 s/epoch × 8 epochs (early stopping).
Ensemble_SoftVoting	**0 s**	No training required (simple average).
Ensemble_WeightedAUC	**25 ms**	Weights computed based on AUC only.
Ensemble_Stacking	**8.6 ms**	Logistic Regression meta-model over 7 probabilities.

Table 14. Training time by model.

Table 15 summarizes the inference time of the evaluated models for 3240 one-minute windows. The ensemble methods exhibit the lowest computational overhead, with **Ensemble Stacking** requiring only 0.00003 ms per window and **Weighted-AUC** 0.0005 ms. Among the individual classifiers, **Logistic Regression** provides the lowest inference latency (0.0005 ms per window), followed by Random Forest (0.031 ms). In contrast, the deep-learning models require substantially higher inference times, particularly the raw-signal RNN, which reaches **3.362 ms per window**, followed by the raw-signal CNN at 1.634 ms. Overall, the results indicate that the classical and ensemble approaches offer a substantial computational advantage for real-time or resource-constrained OSA detection, while the deep-learning models impose greater inference-time requirements.

Model	Total (3,240 windows)	Per window
Logistic Regression	1.6 ms	**0.0005 ms**
Random Forest	100.9 ms	**0.031 ms**
Ensemble_Stacking (combination only)	0.10 ms	**0.00003 ms**
Ensemble_WeightedAUC (combination only)	1.46 ms	**0.0005 ms**
CNN_RREDR	675.4 ms	0.209 ms
SVM (RBF)	893.8 ms	0.276 ms
RNN_RREDR	2,590.9 ms	0.800 ms
CNN (raw signal)	5,294.3 ms	1.634 ms
RNN (raw signal)	10,891.7 ms	3.362 ms

Table 15. One-minute windows inference time.

Table 16 summarizes the computational cost of the preprocessing stages required before model inference. Z-score filtering and normalization require **0.56 ms per window**, while construction of the RR+EDR tachogram requires **2.71 ms per window**. The extraction of the 17 HRV+EDR features represents the most computationally demanding preprocessing stage, requiring **3.70 ms per window** and supporting the **Logistic Regression**, **SVM-RBF**, and **Random Forest** models. Overall, the results show that feature extraction constitutes the largest preprocessing overhead, whereas signal normalization imposes the lowest computational cost.

Stage	Time per window	Models that require it
Preprocessing (z-score window filtering + normalization)	0.56 ms	CNN, RNN (raw signal)
Construction of RR+EDR tachogram	2.71 ms	CNN_RREDR, RNN_RREDR
Extraction of 17 HRV+EDR features	3.70 ms	Logistic Regression, SVM_RBF, Random Forest

Table 16. Preprocessing time per window (prior to inference).

5. Conclusions, Limitations and Future Work

5.1. Conclusions

This study implemented and rigorously evaluated a set of machine learning, deep learning, and ensemble models for obstructive sleep apnea (OSA) detection using ECG- and EDR-derived information from the PhysioNet Apnea-ECG database. The results demonstrate that classical models and ensemble approaches provided the most consistent generalization, whereas models trained directly on raw ECG exhibited substantially poorer external performance. The best-performing configuration, a Weighted-AUC ensemble combining SVM-RBF, Random Forest, Logistic Regression, and an RR+EDR CNN, achieved an AUC of 0.946 and an F1-score of 0.836 on the official external test set (x01–x35), outperforming the directly comparable study based on the same official train/test partition.

The analysis also highlighted important differences in generalization between model families. Classical classifiers and ensembles showed minimal degradation between internal and external validation (ΔAUC ranging from −0.005 to +0.03), whereas the raw-ECG RNN deteriorated to approximately chance-level performance externally (AUC ≈ 0.50). Root-cause analysis identified subject-level leakage in the inner training/validation split used for early stopping; correcting this through subject-grouped splitting did not improve the deep-learning models, suggesting that the primary limitation was the small training cohort rather than the original hyperparameter configuration.

This project's third objective was to combine minute-level predictions with patient-level Apnea Index (AI) and four-tier severity classification. On the external test set (n=35) the weighted-AUC ensemble performed best: AI mean absolute error of 3.01 events/hour, near-zero bias, and severity accuracy/kappa of 0.80/0.723 as per the best minute-level classifier and remained stable on both internal (n=9) and external (n=35) datasets, unlike the results on the external test set alone. The error margin is small compared to the clinical severity bands (5, 15, 30 events/hour), so this approach could be a screening tool, but not a replacement for polysomnography.

The much wider ranges of agreement in the internal partition (i.e. ~36 versus ~24 events/hour for Random Forest) prove that patient-level metrics are much more sensitive to sample size than minute-level ones because each patient, not each window, is just an observation, and small-cohort severity results should be read as indicative, not conclusive. Raw-ECG CNN and RNN are not suitable for severity estimation: their AI errors (23.9 and 36.8 events/hour) exceed the entire clinical severity scale, which means that their minute-level weaknesses build up and are not aggregated.

A documented hyperparameter search further demonstrated that the manually selected configurations were already close to optimal, with AUC differences of ≤0.007 after retraining. Statistical analyses provided an additional safeguard against overinterpretation: subject-clustered bootstrap testing yielded fewer significant pairwise differences than naive window-level McNemar testing, indicating that the apparent superiority of individual ensemble models should be interpreted cautiously at the available sample size. Finally, comparison with the literature showed that only a limited number of studies could be verified as using the same PhysioNet Apnea-ECG database with comparable metrics; therefore, the

proposed results should be regarded as competitive rather than universally superior to the state of the art.

5.2. Limitations

The principal limitation is the use of a single external database, which restricts assessment of generalizability across populations, institutions, recording equipment, and acquisition protocols. Although the official x01–x35 partition provides an independent patient cohort and constitutes a stringent evaluation, it remains derived from a single database. The relatively small number of subjects also limits the precision of patient-level severity estimates, resulting in wide confidence intervals for some severity-related metrics.

Additional limitations concern the deep-learning component. CNN and RNN architectures were not subjected to systematic architecture optimization, and their external performance remained poor, particularly for the raw-ECG RNN. The computational environment also restricted the Random Forest hyperparameter search from the planned 25 to 11 candidates. Furthermore, both internal and external evaluations represent a single realization of the subject-level partition, and repeated or nested cross-validation was not performed. Finally, the state-of-the-art comparison remains conservative because several candidate studies could not be verified against their primary sources and were therefore excluded rather than relying on potentially inaccurate secondary citations.

5.3. Future Work

Future research should prioritize multi-database and multi-center external validation, applying the trained models without retraining to independent ECG sleep-apnea cohorts. Given the observed limitations of raw-ECG deep learning, increasing the number of training subjects should precede extensive architectural optimization, potentially through database integration or transfer learning from large-scale ECG foundation models. A systematic, subject-grouped hyperparameter search should subsequently be extended to CNN and RNN architectures.

Repeated subject-level and nested cross-validation would provide more robust estimates of sampling variability, while prospective multi-site clinical studies would be required to establish clinical utility against concurrent polysomnography. Further work should also complete the verification of candidate state-of-the-art studies to strengthen the quantitative literature benchmark. Finally, real-time deployment should be investigated by assessing computational requirements, latency, robustness to noise and motion artifacts, and performance under conditions representative of wearable or ambulatory ECG monitoring.

6. References

1. Javaheri, S., Barbe, F., Campos-Rodriguez, F., Dempsey, J. A., Khayat, R., Javaheri, S., ... & Polotsky, V. Y. (2017). Sleep apnea: types, mechanisms, and clinical cardiovascular consequences. Journal of the American College of Cardiology, 69(7), 841-858.

2. Ghandeharioun, H., Rezaeitalab, F., & Lotfi, R. (2016). Analysis of respiratory events in obstructive sleep apnea syndrome: Inter-relations and association to simple nocturnal features. Revista Portuguesa de neumología (English Edition), 22(2), 86-92.

3. Thorpy, M. J. (2012). Classification of sleep disorders. Neurotherapeutics, 9(4), 687-701.

4. Sateia, M. J. (2014). International classification of sleep disorders. Chest, 146(5), 1387-1394.

5. Randerath, W., Bassetti, C. L., Bonsignore, M. R., Farre, R., Ferini-Strambi, L., Grote, L., ... & Montserrat, J. (2018). Challenges and perspectives in obstructive sleep apnoea: report by an ad hoc working group of the Sleep Disordered Breathing Group of the European Respiratory Society and the European Sleep Research Society. European Respiratory Journal, 52(3).

6. Sforza, E., & Roche, F. (2016). Chronic intermittent hypoxia and obstructive sleep apnea: an experimental and clinical approach. Hypoxia, 4, 99.

7. Zhang, X. B., Zen, H. Q., Lin, Q. C., Chen, G. P., Chen, L. D., & Chen, H. (2014). TST, as a polysomnographic variable, is superior to the apnea hypopnea index for evaluating intermittent hypoxia in severe obstructive sleep apnea. European Archives of Oto-Rhino-Laryngology, 271(10), 2745-2750.

8. Ryan, S. (2018). Mechanisms of cardiovascular disease in obstructive sleep apnoea. Journal of thoracic disease, 10 (Suppl 34), S4201.

9. Badran, M., Ayas, N., & Laher, I. (2014). Cardiovascular complications of sleep apnea: role of oxidative stress. Oxidative medicine and cellular longevity, 2014.

10. Floras, J. S. (2018). Sleep apnea and cardiovascular disease: an enigmatic risk factor. Circulation research, 122(12), 1741-1764.

11. Quan, S. F., & Gersh, B. J. (2004). Cardiovascular consequences of sleep-disordered breathing: past, present, and future: report of a workshop from the

National Center on Sleep Disorders Research and the National Heart, Lung, and Blood Institute. Circulation, 109(8), 951-957.

12. Almazaydeh, L., Faezipour, M., & Elleithy, K. (2012). A Neural Network System for Detection of Obstructive Sleep Apnea Through SpO2 Signal. Editorial Preface, 3(5).

13. Güneş, S., Polat, K., & Yosunkaya. (2010). Multi-class f-score feature selection approach to classification of obstructive sleep apnea syndrome. Expert systems with applications, 37(2), 998-1004.

14. Mencar, C., Gallo, C., Mantero, M., Tarsia, P., Carpagnano, G. E., Foschino Barbaro, M. P., & Lacedonia, D. (2020). Application of machine learning to predict obstructive sleep apnea syndrome severity. Health informatics journal, 26(1), 298-317.

15. Urtnasan, E., Park, J. U., Lee, S., & Lee, K. J. (2017). Optimal classifier for detection of obstructive sleep apnea using a heartbeat signal. International Journal of Fuzzy Logic and Intelligent Systems, 17(2), 76-81.

16. Mitiche, I., Morison, G., & Stewart, B. G. (2016). Electrocardiogram-Based Feature Extraction for Machine Learning Classification of Obstructive Sleep Apnea. International Journal of Signal Processing Systems, 4(6), 515-518.

17. Mostafa, S. S., Mendonça, F., G Ravelo-García, A., & Morgado-Dias, F. (2019). A Systematic Review of Detecting Sleep Apnea Using Deep Learning. Sensors, 19(22), 4934.

18. Gozal, D., & Kheirandish-Gozal, L. (2008). Cardiovascular morbidity in obstructive sleep apnea: oxidative stress, inflammation, and much more. American journal of respiratory and critical care medicine, 177(4), 369-375.

19. Huang, Q. R., Qin, Z., Zhang, S., & Chow, C. M. (2008). Clinical patterns of obstructive sleep apnea and its comorbid conditions: a data mining approach. Journal of Clinical Sleep Medicine, 4(6), 543-550.

20. Estadísticas Vitales: Cifras definitivas. 2018. Departamento Administrativo Nacional de Estadística (DANE). 2019, p.27.

21. Hidalgo-Martínez, P., & Lobelo, R. (2017). Global, Latin-American and Colombian epidemiology and mortality by obstructive sleep apnea-hypopnea syndrome (OSAHS). Revista de la Facultad de Medicina, 65, 17-20.

22. Voncken, S. F. J., Feron, T. M. H., Karaca, U., Beerhorst, K., Klarenbeek, P., Straetmans, J. M. J. A. A., ... & de Kruif, M. D. (2022). Impact of obstructive sleep

apnea on clinical outcomes in patients hospitalized with COVID-19. *Sleep and breathing*, *26*(3), 1399-1407.

23. Miller, M. A., & Cappuccio, F. P. (2021). A systematic review of COVID-19 and obstructive sleep apnoea. *Sleep Medicine Reviews*, *55*, 101382.

24. Ramachandran, A., & Karuppiah, A. (2021, July). A survey on recent advances in machine learning based sleep apnea detection systems. In *Healthcare* (Vol. 9, No. 7, p. 914). MDPI.

25. Urtnasan, E., Park, J. U., Joo, E. Y., & Lee, K. J. (2018). Automated detection of obstructive sleep apnea events from a single-lead electrocardiogram using a convolutional neural network. Journal of medical systems, 42, 1-8.

26. Ayatollahi, A., Afrakhteh, S., Soltani, F., & Saleh, E. (2022). Sleep apnea detection from ECG signal using deep CNN-based structures. Evolving Systems, 1-16.

27. Sharma, M., Kumbhani, D., Yadav, A., & Acharya, U. R. (2022). Automated Sleep apnea detection using optimal duration-frequency concentrated wavelet-based features of pulse oximetry signals. *Applied Intelligence*, 1-13.

28. Sheta, A., Turabieh, H., Thaher, T., Too, J., Mafarja, M., Hossain, M. S., & Surani, S. R. (2021). Diagnosis of Obstructive Sleep Apnea from ECG Signals Using Machine Learning and Deep Learning Classifiers. Applied Sciences, 11(14), 6622. https://doi.org/10.3390/app11146622.

29. Mostafa, S. S., Baptista, D., Ravelo-García, A. G., Juliá-Serdá, G., & Morgado-Dias, F. (2020). Greedy based convolutional neural network optimization for detecting apnea. Computer methods and programs in biomedicine, 197, 105640. https://doi.org/10.1016/j.cmpb.2020.105640.

30. Cen, L., Yu, Z. L., Kluge, T., & Ser, W. (2018, July). Automatic system for obstructive sleep apnea events detection using convolutional neural network. In 2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC) (pp. 3975-3978). IEEE.

31. Almutairi, H., Hassan, G. M., & Datta, A. (2021). Classification of Obstructive Sleep Apnoea from single-lead ECG signals using convolutional neural and Long Short Term Memory networks. Biomedical Signal Processing and Control, 69, 102906.

32. Urtnasan, E., Park, J. U., & Lee, K. J. (2018). Multiclass classification of obstructive sleep apnea/hypopnea based on a convolutional neural network from a single-lead electrocardiogram. Physiological measurement, 39(6), 065003.

33. Paul, T., Hassan, O., Alaboud, K., Islam, H., Rana, M. K. Z., Islam, S. K., & Mosa, A. S. M. (2022). ECG and SpO2 Signal-Based Real-Time Sleep Apnea Detection Using Feed-Forward Artificial Neural Network. AMIA ... Annual Symposium proceedings. AMIA Symposium, 2022, 379–385.

34. Wang, X., Cheng, M., Wang, Y. et al. (2020). Obstructive sleep apnea detection using ecg-sensor with convolutional neural networks. Multimed Tools Appl 79, 15813–15827. https://doi.org/10.1007/s11042-018-6161-8.

35. Sahoo, A.K., Pradhan, C., Das, H. (2020). Performance Evaluation of Different Machine Learning Methods and Deep-Learning Based Convolutional Neural Network for Health Decision Making. In: Rout, M., Rout, J., Das, H. (eds) Nature Inspired Computing for Data Science. Studies in Computational Intelligence, vol 871. Springer, Cham. https://doi.org/10.1007/978-3-030-33820-6_8.

36. A. Singh, N. Thakur and A. Sharma, "A review of supervised machine learning algorithms," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2016, pp. 1310-1315.

37. Razi, A.P., Einalou, Z. & Manthouri, M. (2021). Sleep Apnea Classification Using Random Forest via ECG. Sleep Vigilance 5, 141–146. https://doi.org/10.1007/s41782-021-00138-4.

38. Kurzweil, R., Richter, R., Kurzweil, R., & Schneider, M. L. (1990). The age of intelligent machines (Vol. 580). Cambridge: MIT press.

39. Russell, S., & Norvig, P. (1995). A modern, agent-oriented approach to introductory artificial intelligence. *Acm Sigart Bulletin*, *6*(2), 24-26.

40. Raschka, S., & Mirjalili, V. (2017). Python machine learning: Machine learning and deep learning with python. *Scikit-Learn, and TensorFlow. Second edition ed*, *3*.

41. Chollet, F. (2017). Deep learning with python. Manning Publications.

42. Niu, S., Liu, Y., Wang, J., & Song, H. (2020). A decade survey of transfer learning (2010–2020). IEEE Transactions on Artificial Intelligence, 1(2), 151-166.

43. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27 (pp. 270-279). Springer International Publishing.

44. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer.

45. Tan, P. N., Steinbach, M., & Kumar, V. (2006). Data mining introduction. *People's Posts and Telecommunications Publishing House, Beijing*.

46. Sigletos, G., Paliouras, G., Spyropoulos, C. D., Hatzopoulos, M., & Cohen, W. (2005). Combining Information Extraction Systems Using Voting and Stacked Generalization. *Journal of Machine Learning Research*, *6*(11).

47. Jiang, W., Chen, Z., Xiang, Y., Shao, D., Ma, L., & Zhang, J. (2019). SSEM: A novel self-adaptive stacking ensemble model for classification. *IEEE Access*, *7*, 120337-120349.