

REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

**ARTIFICIAL INTELLIGENCE-BASED
CLASSIFICATION OF VENTILATION MODES
AND FORECASTING OF RESPIRATORY DATA
FOR INTENSIVE CARE PATIENTS**

Gökçe Nur BEKEN

MASTER OF SCIENCE THESIS

Department of Electronics and Communications Engineering
Program of Electronics

Supervisor

Prof. Dr. Burcu ERKMEN

December, 2024

REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

**ARTIFICIAL INTELLIGENCE-BASED CLASSIFICATION
OF VENTILATION MODES AND FORECASTING OF
RESPIRATORY DATA FOR INTENSIVE CARE PATIENTS**

A thesis submitted by Gökçe Nur BEKEN in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 17.12.2024 in Department of Electronics and Communications Engineering, Program of Electronics.

Prof. Dr. Burcu ERKMEN
Yildiz Technical University
Supervisor

Approved By the Examining Committee

Prof. Dr. Burcu ERKMEN, Supervisor
Yildiz Technical University

Assoc. Prof. Dr. Revna ACAR VURAL, Member
Yildiz Technical University

Asst. Prof. Dr. İpek ŞEN, Member
Istanbul Bilgi University

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Artificial Intelligence-Based Classification of Ventilation Modes and forecasting of Respiratory Data for Intensive Care Patients supervised by my supervisor, Prof. Dr. Burcu ERKMEN. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Gökçe Nur BEKEN

Signature



This work was supported by Yildiz Technical University Scientific Research Projects Coordination Unit under project number FCD-2023-5445.

*Dedicated to my Mother
and
Father*



ACKNOWLEDGEMENTS

I would like to express my gratitude to my esteemed advisor, Prof. Dr. Burcu ERKMEN, whose assistance was crucial to the completion of this project and whose advice, expertise, and experience guided my work throughout my thesis process.

I would like to thank Yildiz Technical University Scientific Research Projects Coordination Unit for supporting my thesis studies under project number FCD-2023-5445. I would like to extend my gratitude to Assoc. Prof. Dr. Revna ACAR VURAL, my instructor, for her guidance and the sharing of her experiences in my thesis study, which was derived from this project. I am also grateful to the doctors of Istanbul Başakşehir Çam and Sakura Urban Hospitals, Prof. Dr. Ayça ÖZKUL, Prof. Dr. Murat ÇABALAR, and Spec. Dr. Mehmet Semih ARI, for their support, knowledge, and guidance in the collection and evaluation of data.

I would like to thank my lovely mother Sultan BEKEN and father Mehmet BEKEN for their unwavering support throughout my studies, as well as my sister Nurbanu CAN for her love and support.

I would like to thank my dear friends Şule Zeynep AYDIN, Seda GÜNAKIN, Sibel ÇİMEN, Hatice Vildan DÜDÜKÇÜ, and Özden NİYAZ for their unwavering support and patience during the course of my thesis. Their kindness and understanding were invaluable.

I would also like to thank Istanbul Başakşehir Çam and Sakura Urban Hospital for providing the data that made this research possible. The resources they provided are of great importance for the feasibility of my thesis study.

I sincerely thank everyone who contributed and stood by me throughout this intense research process, knowing that my thesis is the product of a dedicated effort.

Gökçe Nur BEKEN

TABLE OF CONTENTS

LIST OF SYMBOLS	viii
LIST OF ABBREVIATIONS	ix
LIST OF FIGURES	xi
LIST OF TABLES	xiv
ABSTRACT	xv
ÖZET	xvii
1 INTRODUCTION	1
1.1 Literature Review	1
1.2 Objective of the Thesis	6
1.3 Hypothesis	6
2 MACHINE LEARNING ALGORITHMS	7
2.1 Decision Tree	7
2.2 Bootstrap Aggregating (Bagging)	8
2.3 Random Forest	9
2.4 Gradient Boosting	9
2.5 Adaptive Boosting (AdaBoost)	11
2.6 K-Nearest Neighbors (KNN)	12
2.7 Gaussian Naive Bayes	13
2.8 Extreme Gradient Boosting (XGBoost)	13
3 DEEP LEARNING ALGORITHMS	15
3.1 Long Short Term Memory (LSTM)	15
3.2 Convolutional Neural Network (CNN)	16
3.3 Temporal Convolutional Network (TCN)	17
4 DATASET AND PERFORMANCE METRICS	19
4.1 Dataset	19
4.1.1 Ventilation Modes	21

4.1.2	Data Properties	22
4.1.3	Dataset Preprocessing	23
4.2	Performance Metrics	26
4.2.1	Threshold Based Metrics	26
4.2.2	Rank Based Metrics	27
4.2.3	Regression Metrics	28
5	VENTILATION MODE CLASSIFICATION AND RESPIRATORY DATA ESTIMATION	30
5.1	Ventilation Mode Classification	30
5.1.1	Classification with Machine Learning Algorithms	31
5.1.2	Classification with Deep Learning Algorithms	36
5.1.3	Ventilation Mode Classification Results	47
5.2	Forecasting of Time Series Respiratory Data	49
5.2.1	Forecasting with LSTM Model	50
5.2.2	Forecasting with TCN Model	52
5.2.3	Forecasting with CNN-LSTM Model	52
5.2.4	Forecasting with TCN-LSTM Model	54
5.2.5	Forecasting Results of Time Series Respiratory Data	54
6	POSITION CLASSIFICATION BASED ON RESPIRATORY MONITORING WITH THORAX AND PULSE OXIMETER	63
6.1	Dataset	64
6.2	Classification with Machine Learning Algorithms	66
6.3	Position Classification Results	67
7	CONCLUSION	71
	REFERENCES	74
	PUBLICATIONS FROM THE THESIS	79

LIST OF SYMBOLS

R^2	Coefficient of Determination
$\oplus$	Component-Wise Addition
$\odot$	Component-Wise Multiplication
$\tanh$	Hyperbolic Tangent Function
μ_C	Mean for Class C
σ	Sigma
sig	Sigmoid Function
σ_C	Standard Deviation for Class C
Σ	Summation

LIST OF ABBREVIATIONS

AdaBoost	Adaptive Boosting
AI	Artificial Intelligence
ANN	Artificial Neural Network
AP	Arterial Pressure
ARDS	Acute Respiratory Distress Syndrome
ASV	Adaptive Support Ventilation
AUC	Area Under the Curve
Bi-LSTM	Bidirectional LSTM
CMV	Continuous Mandatory Ventilation
CNN	Convolutional Neural Network
CNN-LSTM	Convolutional Neural Network with Long Short Term Memory
COPD	Chronic Obstructive Pulmonary Disease
ECG	Electrocardiogram
FN	False Negative
FP	False Positive
HLA	hysteresis Loop Analysis
ILC-PID	Iterative Learning PID Controller
KNN	K-Nearest Neighbors
LSTM	Long Short-Term
MAE	Mean Absolute Error
MLP	Multilayer Perceptron
MSE	Mean Squared Error
OPVNN	Orthogonal Polynomial-Based Variant Nearest Neighbor

PCA	Principal Component Analysis
PEEP	Positive End-Expiratory Pressure
PMEAN	Mean Airway Pressure
PPEAK	Peak Pressure
PPG	Photoplethysmographic
PPLAT	Plateau Pressure
PSIMV	Pressure-Supported Intermittent Mandatory Ventilation
PVA	Patient-Ventilator Asynchrony
RBM	Rule-Based Method
ReLU	Rectified Linear Unit
RESP	Respiratory Rate
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
ROC	Receiver-Operating Characteristic
sEMG	Surface Electromyogram
SPONT	Spontaneous Ventilation
SSL	Semi-Supervised Learning
TCN	Temporal Convolutional Network
TCN-LSTM	Temporal Convolutional Network with Long Short Term Memory
TCNN	Three-Input Convolutional Neural Network
TN	True Negative
TP	True Positive
TPOT	Tree-Based Pipeline Optimization Technique
VTE	Expiratory Tidal Volume
VTI	Inspiratory Tidal Volume
XGBoost	Extreme Gradient Boosting

LIST OF FIGURES

Figure 2.1	Decision Tree algorithm.	8
Figure 2.2	Bagging algorithm.	8
Figure 2.3	Random Forest classifier.	10
Figure 2.4	AdaBoost algorithm.	12
Figure 2.5	KNN algorithm.	12
Figure 2.6	XGBoost algorithm.	14
Figure 3.1	LSTM Cell	16
Figure 3.2	TCN architecture.	18
Figure 4.1	Graph of time-dependent data value.	20
Figure 4.2	Basic structure of time series.	21
Figure 4.3	Bar chart showing class distribution.	22
Figure 5.1	General scheme of classification.	31
Figure 5.2	Comparison of metrics obtained as a result of applying machine learning algorithms to normalized data.	31
Figure 5.3	ROC curve plots and confusion matrices obtained as a result of applying Random Forest, Gradient Boosting, AdaBoost and Bagging algorithms to normalized data.	34
Figure 5.4	ROC curve plots and confusion matrices obtained as a result of applying K-Nearest Neighbors, Decision Tree, Gaussian NB and XGBoost algorithms to normalized data.	35
Figure 5.5	Metric values of moving average filtered data according to machine learning models.	36
Figure 5.6	ROC curve plots and confusion matrices obtained as a result of applying Random Forest, Gradient Boosting, AdaBoost and Bagging algorithms to moving average filtered data.	37
Figure 5.7	ROC curve plots and confusion matrices obtained as a result of applying K-Nearest Neighbors, Decision Tree, Gaussian NB and XGBoost algorithms to moving average filtered data.	38
Figure 5.8	Metric values of Gaussian filtered data according to machine learning models.	39

Figure 5.9	ROC curve plots and confusion matrices obtained as a result of applying Random Forest, Gradient Boosting, AdaBoost and Bagging algorithms to Gaussian filtered data.	40
Figure 5.10	ROC curve plots and confusion matrices obtained as a result of applying K-Nearest Neighbors, Decision Tree, Gaussian NB and XGBoost algorithms to Gaussian filtered data.	41
Figure 5.11	Comparative graph of machine learning models' accuracy values for three different datasets.	42
Figure 5.12	Bar graph of accuracy and error value for mean absolute error of LSTM network.	42
Figure 5.13	Confusion matrix for MAE loss function of LSTM network. . .	43
Figure 5.14	Bar graph of accuracy and error value for mean squared error of LSTM network.	44
Figure 5.15	Confusion matrix for MSE loss function of LSTM network. . .	45
Figure 5.16	CNN-LSTM model used.	45
Figure 5.17	Confusion matrix of the CNN-LSTM model.	46
Figure 5.18	LSTM architecture implemented for forecasting.	50
Figure 5.19	Stacked bar chart of the mean absolute error values of the LSTM prediction.	51
Figure 5.20	Plots of PEEP and SPO2 features' actual value and LSTM predicted values.	51
Figure 5.21	Stacked bar chart of the mean absolute error values of the TCN prediction.	52
Figure 5.22	The actual value of the Oxygen feature and the TCN predicted value.	53
Figure 5.23	Stacked bar chart of the mean absolute error values of the CNN-LSTM prediction.	53
Figure 5.24	The actual value of the Oxygen feature and the CNN-LSTM predicted value.	54
Figure 5.25	TCN-LSTM architecture implemented for forecasting.	56
Figure 5.26	Stacked bar chart of the mean absolute error values of the TCN-LSTM prediction.	56
Figure 5.27	The actual value of the Expiration feature and the TCN-LSTM predicted value.	57
Figure 5.28	Bar chart of metric values obtained as a result of estimation of Pulse and RESP features for all models.	58
Figure 5.29	Bar chart of metric values obtained as a result of estimation of Oxygen and PEEP features for all models.	59

Figure 5.30	Bar chart of metric values obtained as a result of estimation of PPEAK and PMEAN features for all models.	60
Figure 5.31	Bar chart of metric values obtained as a result of estimation of SPO2 and Expiration features for all models.	61
Figure 5.32	The actual value and TCN-LSTM predicted value of eight features.	62
Figure 6.1	Sensors used in the study.	64
Figure 6.2	Data collection setup.	66
Figure 6.3	Force over time plot for three places.	66
Figure 6.4	Comparison of metrics obtained as a result of applying machine learning algorithms to thorax and pulse oximeter sensor data. . .	67
Figure 6.5	ROC curve plots and confusion matrices obtained as a result of applying Random Forest, Gradient Boosting, AdaBoost and Bagging algorithms to thorax and pulse oximeter sensor data. . .	68
Figure 6.6	ROC curve plots and confusion matrices obtained as a result of applying K-Nearest Neighbors, Decision Tree, Gaussian NB and XGBoost algorithms to thorax and pulse oximeter sensor data.	69

LIST OF TABLES

Table 4.1	5 hours of data from a patient.	21
Table 4.2	5 hours normalized data of a patient.	24
Table 5.1	Hyperparameter configurations for machine learning models. . .	33
Table 5.2	Comparison of performance metrics across different data preprocessing methods.	36
Table 5.3	Hyperparameter Configurations for LSTM Network.	39
Table 5.4	Classification metrics for the MAE loss function of the LSTM network.	43
Table 5.5	Classification metrics for the MSE loss function of the LSTM network.	44
Table 5.6	CNN-LSTM Model Hyperparameters.	46
Table 5.7	Classification metrics for CNN-LSTM model.	46
Table 5.8	Hyperparameters of deep learning models.	49
Table 5.9	Respiratory data forecasting model performance values (MAE). .	55
Table 6.1	Thorax sensor features.	65
Table 6.2	Pulse oximeter features.	65
Table 6.3	Comparison of performance metrics across different machine learning algorithms.	67

ABSTRACT

Artificial Intelligence-Based Classification of Ventilation Modes and forecasting of Respiratory Data for Intensive Care Patients

Gökçe Nur BEKEN

Department of Electronics and Communications Engineering
Master of Science Thesis

Supervisor: Prof. Dr. Burcu ERKMEN

In hospitals numerous patients in intensive care units necessitating respiratory support due to pulmonary pathologies and diverse ailments. Mechanical ventilation support is one of the most often utilized techniques for individuals experiencing respiratory insufficiency. The necessity for mechanical ventilation may fluctuate throughout the patient's treatment procedure. During this phase, the likelihood of problems and infections for patients escalates. Minimizing the duration of patients' reliance on artificial ventilation is essential for expediting their recovery. The appropriate choice of mechanical ventilation mode is crucial for expediting recovery and minimizing the risk of problems.

This study aims to support healthcare professionals by enabling accurate decision-making for ventilation modes and the prediction of respiratory data through artificial intelligence. The data were collected from 23 patients in the neurology intensive care unit, recorded at 1-hour intervals over a 24-hour period. The dataset, containing a total of 18 features, was labeled with four different ventilation modes (CMV, PSIMV, ASV, SPONT) and underwent preprocessing steps such as Min-Max normalization, moving average filtering, and Gaussian filtering to enhance the performance of the artificial intelligence models. For ventilation mode classification, machine learning algorithms such as decision tree, Bagging, random forest, gradient boosting, adaptive boosting (AdaBoost), K-Nearest Neighbors, gaussian naive bayes, and extreme gradient boosting (XGBoost) were used, along with deep learning models like Long Short-Term

Memory (LSTM) and convolutional neural network with long short term memory (CNN-LSTM). The XGBoost algorithm achieved the highest performance, with an accuracy rate of 98%.

The prediction of respiratory data through time series analysis is essential for overseeing the respiratory condition of intensive care patients and for the early identification of probable complications. Deep learning models utilized for this purpose include LSTM, Temporal convolutional network (TCN), CNN-LSTM, and Temporal convolutional network with long short term memory (TCN-LSTM). The TCN-LSTM model attained optimal performance, exhibiting the minimal error rate and the maximum R^2 value.

Additionally, respiratory data about chest force, respiratory rate, SpO₂, and PRbpm were collected from 10 volunteers in various situations (sitting, walking, sleeping) utilizing thoracic sensors and a pulse oximeter. The XGBoost algorithm achieved a position classification accuracy of 99.74%. These findings illustrate the capacity of artificial intelligence to execute precise classification and prediction on respiratory data. This study's findings underscore the potential of AI-assisted health decision systems to foster novel solutions in patient management.

Keywords: Mechanical ventilator, breathing, thoracic sensor, machine learning, deep learning.

Yoğun Bakım Hastaları İçin Yapay Zeka Tabanlı Ventilasyon Modlarının Sınıflandırılması ve Solunum Verilerinin Tahmini

Gökçe Nur BEKEN

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Prof. Dr. Burcu ERKMEN

Hastanelerde, akciğer patolojileri ve çeşitli hastalıklar nedeniyle yoğun bakım ünitelerinde solunum desteğine ihtiyaç duyan birçok hasta bulunmaktadır. Solunum yetmezliği olan hastalar için en sık uygulanan uygulamalardan biri mekanik ventilasyon desteğidir. Mekanik ventilasyon ihtiyacı, hastanın tedavi süreciyle birlikte zamanla azalabilmekte veya artabilmektedir. Ancak bu süreçte, hastaların komplikasyon ve enfeksiyon riskine maruz kalma olasılığı artmaktadır. Hastaların mekanik ventilasyona daha kısa süre bağımlı kalması, iyileşme süreçlerinin hızlanması açısından önemlidir. Mekanik ventilasyon modunun doğru seçimi, hastaların iyileşme sürecini hızlandırmada ve komplikasyon risklerini azaltmada büyük rol oynar.

Bu çalışmada, yapay zeka desteği ile ventilasyon modlarının doğru karar verilebilmesi ve solunum verilerinin tahmin edilmesini sağlayarak sağlık çalışanlarına destek sağlanması hedeflenmektedir. Veriler, nöroloji yoğun bakım ünitesinde yatan 23 hastadan, 24 saat boyunca 1 saatlik periyotlarla kaydedilerek oluşturulmuştur. Toplamda 18 özellik içeren veri seti, dört farklı ventilasyon modu (CMV, PSIMV, ASV, SPONT) ile etiketlenmiş ve yapay zeka modellerinin daha iyi sonuç vermesi amacıyla Min-Max normalizasyonu, hareketli ortalama filtresi ve Gaussian filtresi ön işleme adımlarından geçirilmiştir. Ventilasyon mod sınıflandırması için Rastgele Orman, Karar Ağacı, Gradyan Artırma, Uyarlamalı Artırma (AdaBoost), Çok Modelli Bootstrap (Bagging), K-En Yakın Komşu, Gaussian Naive Bayes ve Aşırı Gradyan Artırma (XGBoost) gibi makine öğrenmesi

algoritmalarının yanı sıra Uzun kısa süreli bellek (LSTM) ve CNN-LSTM gibi derin öğrenme modelleri kullanılmıştır. Sınıflandırma sonucunda, %98 doğruluk oranı ile en yüksek performansı XGBoost algoritması sağlamıştır.

Solunum verilerinin zaman serisi tahmini ise yoğun bakım hastalarının solunum durumu yönetimi ve olası sorunların erken tespiti açısından büyük önem arz etmektedir. Bu amaçla, derin öğrenme modelleri olan LSTM, Zamansal Evrişimsel Ağlar (TCN), CNN-LSTM ve TCN-LSTM modelleri kullanılmıştır. TCN-LSTM modeli, en düşük hata oranı ve en yüksek R^2 değeri ile en başarılı sonucu elde etmiştir.

Ayrıca, toraks sensörleri ve pulse oksimetre ile 10 gönüllüden farklı pozisyonlarda (oturma, yürüme, yatma) solunuma bağlı göğüs kuvveti, solunum hızı, SpO2 ve PRbpm verileri toplanmış ve XGBoost algoritması ile %99.74 doğruluk oranında pozisyon sınıflandırması yapılmıştır. Bu sonuçlar, yapay zekanın solunum verilerinde yüksek doğruluk oranlarıyla sınıflandırma ve tahmin yapabilme kabiliyetini ortaya koymaktadır. Bu çalışmanın bulguları, yapay zeka destekli sağlık karar sistemlerinin gelişimine katkı sağlayarak hasta yönetiminde yenilikçi çözümler sunma potansiyelini göstermiştir.

Anahtar Kelimeler: Mekanik ventilatör, solunum, toraks sensörü, makine öğrenmesi, derin öğrenme.

1

INTRODUCTION

1.1 Literature Review

Hospitals include patients in intensive care units who necessitate breathing assistance due to pulmonary disorders and diverse ailments. Worldwide and nationally, mechanical respiratory support is the most commonly applied procedure for patients in intensive care units who have respiratory failure [1]. The necessity for mechanical ventilation may diminish as the recovery process advances. During mechanical ventilation, patients exhibit heightened susceptibility to complications and infections. Thus, a reduced duration of mechanical ventilation correlates with an expedited recovery process. The importance of managing mechanical ventilators was particularly highlighted during the COVID-19 pandemic [2]. In Europe and many other countries, a shortage of respiratory devices led to patients losing their lives due to lack of access to appropriate treatment. The efficient utilization of mechanical respiratory devices through the reduction of patients' ventilation durations is essential for minimizing waiting times for other patients, particularly during pneumonia-related pandemics.

As the lungs expand and contract during ventilation, both stress and strain change over time. This procedure entails the exertion of work on the pulmonary system, or, in other terms, the application of energy. This energy is either retained and partially recovered or dissipated in the airways and lung tissue [3]. Therefore, selecting the correct ventilation mode and reducing the patient's ventilation duration can help regulate the energy applied to the lungs appropriately, minimizing lung damage during the ventilation process. Mechanical ventilation is a multifaceted procedure that encompasses several characteristics and variables, and selecting an optimal ventilation mode can profoundly influence patient outcomes [4]. The selection of a ventilation mode is contingent upon multiple criteria, including the patient's respiratory mechanics, underlying disease, and the intended objectives of ventilation [5]. Progress in artificial intelligence has created new opportunities for the classification of ventilation modes and respiratory data, resulting in enhanced

patient outcomes [6, 7]. Zhang et al. developed a machine learning model based on multimodal feature fusion to detect abnormal mechanical ventilation by analyzing ventilator waveforms [8]. They performed classification using 112,420 data samples. They employed three machine learning models; random forest, k-nearest neighbors, and support vector machine to identify five forms of anomalous breathing by integrating multimodal information. This strategy markedly improved identification precision, which is essential for real-time patient surveillance in critical care. Akbulut et al. [9] presented an artificial neural network (ANN) based decision support system to recommend ventilator settings based on patients' physiological data. This approach aims to forecast ideal parameters for frequency, tidal volume, FiO₂, and pressure/volume support settings utilizing data from 158 respiratory patients aged over 60. The research determined that Bayesian regularization backpropagation was the most effective approach for classification. A study introduced a classification approach for sputum accumulation in mechanically ventilated patients, whereas a wireless device was developed to collect ventilator airflow signals [10]. Logistic regression, random forest, Naive Bayes, support vector machine, k-nearest neighbors (KNN), and a two-layer LSTM network were compared using data from 15 patients in an intensive care unit. The suggested method attained the maximum accuracy in classifying non-sputum and sputum accumulation using the LSTM network, with an accuracy of $84.7\% \pm 4.1\%$. In a study by Anitha et al. [11], an Iterative Learning PID Controller (ILC-PID) is proposed to optimize mechanical ventilation in intensive care units by achieving accurate pressure and volume settings. The ILC-PID controller uses flow cycle feedback to ensure consistent inhalation. To evaluate the controller's precision, select the best inhalation mode, and potentially reduce the need for mechanical ventilation, machine learning classifiers including Decision Trees, Bayesian Trees, and ensemble models were employed. Performance of the models was analyzed using evaluation metrics such as confusion matrix, training time, specificity, sensitivity, and F1 score. In a study presenting a classifier-based automatic mode-switching technique for ventilation mode changes [12], a KNN classifier was used. Random data were generated from 7000 samples with 5 labels. This approach accurately classified ventilation modes with an efficacy of 83%. In their paper [13], Bakkes et al. presented a technique for the automated detection and classification of patient-ventilator asynchrony. The study used both clinical and simulated data. The suggested neural network attained an accuracy over 90%. The algorithm achieved superior and dependable performance by integrating simulated data with clinical data during the training of the neural network. A study focused on identifying the ideal moment for weaning intubated patients in intensive care units from mechanical ventilation developed a two-stage prediction model

utilizing artificial intelligence [14]. The study utilized electronic medical records of intubated ICU patients to implement seven algorithms; random forest, support vector machine, LightGBM, logistic regression, KNN, XGBoost, and multilayer perceptron. Initially, 5,873 examples were incorporated, and the AUCs of the best models varied from 0.843 to 0.953. In the second step, 4,172 instances were incorporated, with the AUCs of the best models varying from 0.889 to 0.944. This approach precisely and effectively forecasted the ideal timing for weaning ICU patients from ventilator support. In a study by Ang et al. [15], an automated method for detecting patient-ventilator asynchrony (PVA) was proposed, providing full monitoring capabilities not permitted by standard observation methods. This study used a rule-based method (RBM) known as hysteresis loop analysis (HLA) with a three-input convolutional neural network (TCNN) machine learning model to classify seven different forms of PVA. Both classification methods were validated and compared in performance using a dataset of 11 patients undergoing mechanical ventilation. When comparing the two models in terms of specificity, accuracy and sensitivity metrics, the TCNN model achieved results of 89.5%, 98.3%, and 83.9%, respectively, while the HLA obtained results of 97.5%, 96.6%, and 98.1%, respectively. In conclusion, the TCNN model demonstrated higher sensitivity in detecting patient-ventilator asynchrony, while the HLA, due to its rule-based structure, was better at detecting respiratory cycles without patient-ventilator asynchrony.

Accurate classification of ventilation modes is crucial for providing respiratory support tailored to patient needs. In recent years, the application of artificial intelligence techniques in this field has enabled more accurate decision-making in the selection process of ventilation modes. Furthermore, not only the selection of the ventilation mode but also the accurate classification of respiratory data obtained from the patient plays a critical role in the treatment process. AI-supported analysis of respiratory data allows for a detailed evaluation of the patient's respiratory dynamics, facilitating the optimization of individualized treatment plans. Therefore, AI applications for the classification of respiratory data enhance treatment efficacy by ensuring accurate interpretation of patient data during respiratory support. Artificial intelligence has shown great potential in analyzing and predicting medical data, particularly in respiratory data analysis [16]. Lam et al. [17] developed a machine learning algorithm to predict the onset of acute respiratory distress syndrome (ARDS), a prevalent complication in COVID-19 hospitalized patients. This approach utilizes a semi-supervised learning (SSL) technic that depends on a restricted amount of labeled data. It analyzes time-series data from electronic health records when patients' oxygen saturation decreases below 97%. The

model attained a ROC value of 0.73, indicating its potential efficacy. Kim et al. [18] proposed an effective AI-based technique, FAST-PACE, for forecasting adverse catastrophic events through simple trajectories. This approach seeks to forecast cardiac arrest and acute respiratory failure in patients inside critical care. The model utilizes patient data comprising oxygen saturation, systolic blood pressure, respiration rate, pulse rate, body temperature, diastolic blood pressure, and demographic attributes. They used 3984 random samples for the training dataset. Employing an LSTM-based model, they proved that this approach effectively predicts bad outcomes 1 to 6 hours in advance. A study evaluated the efficacy of deep learning models in forecasting respiratory rate utilizing biosensor data, comprising electrocardiogram (ECG), photoplethysmogram (PPG), and surface electromyogram (sEMG) signals [19]. The investigation encompassed the training of multiple deep learning architectures, including Long Short-Term Memory networks, CNN-LSTM hybrids, Convolutional-LSTM networks, Bidirectional LSTM and attention-augmented LSTM variations. Their performance was evaluated using mean absolute error. The findings indicated that Bi-LSTM attained superior performance across all three datasets derived from bio-signals. Wang et al. [20] proposed an Orthogonal Polynomial-Based Variant Nearest Neighbor (OPVNN) method for predicting respiratory motion. In evaluations of respiratory motion data from 27 patients, OPVNN shown superior accuracy relative to alternative models, such as support vector regression and autoregressive models. Ahmed et al. [21] propose a machine learning methodology for forecasting vital indicators associated with cardiovascular and chronic respiratory ailments. This method utilizes sensor data and multivariate time series analysis. They used 50,000 vital sign samples with ten millisecond intervals for training data. Employing an automated machine learning pipeline, particularly the tree-based pipeline optimization technique (TPOT), in conjunction with the Facebook Prophet model, enables the prediction of vital indicators such as heart rate and blood pressure up to 180 seconds ahead. The new model surpassed conventional methods, attaining superior accuracy in forecasting situations such as hypoxemia and hypercapnia.

Accurate classification and AI-supported analysis methods of respiratory data play a critical role in patient monitoring and optimization of ventilation modes. In this context, it is essential to collect reliable respiratory data from patients and ensure accurate evaluation of this data. The use of biosensors, such as thoracic sensors and pulse oximeters, along with patient monitoring systems and mechanical ventilator data, provides detailed and real-time information about the patient's respiratory status. AI algorithms can examine this data to produce significant

outputs for use in clinical decision support systems. A study utilizing sensors positioned on the thoraco-abdominal regions presented a smart T-shirt design for real-time respiratory monitoring [22]. This system acquires data from alterations in chest and abdomen movements resulting from respiration and transmits it to a base station via Bluetooth, with respiratory patterns assessed using frequency shifts on a spiral fiber antenna. Experiments involving seven participants revealed that respiratory metrics, including inhalation and exhalation lengths and respiratory rate, were precisely measured in comparison to a traditional spirometer. Guy et al. [23] created a respiratory data gathering apparatus designed for bidirectional pressure and flow measurement to assess thoracic and abdominal circumference. This enables facilitates non-invasive assessment of respiratory metrics, including lung capacity and patient respiratory exertion. Tests indicate that the gadget may serve as a useful instrument for real-time respiratory evaluations in diverse clinical environments. Hesse et al. [24] propose a sensor for detecting thoracic motions in their investigation. This sensor detects motions during inhalation and exhalation with a force-sensing resistor embedded in a chest strap. The research additionally devised an adaptive peak detection technique (peakDetAdapt) to enhance the computation of respiratory rate. The findings suggest that the sensor is capable of real-time respiratory monitoring during routine activities. Abdollahi et al. [25] performed a study assessing fall risk in stroke patients employing wearable sensors and machine learning models. The researchers collected comprehensive motion data from 21 volunteers executing clinical balance, gait, and sit-to-stand tests, both with and without a cognitive dual task. The study was conducted by placing a thoracic sensor on the participants' upper thigh. The random forest model achieved an accuracy of 91%. Raiano et al. [26] presented a technique to enhance the positioning and quantity of piezoresistive sensors in wearable respiratory monitoring systems. The study employed Principal Component Analysis (PCA) to assess six sensor configurations across multiple body areas (upper chest, lower chest, and abdomen) for precise monitoring of respiratory rate during activities such as resting, walking, and running. This PCA-based system demonstrated efficacy in respiratory monitoring when tested on 10 healthy people. A study measuring oxygen saturation during home activities in patients with Chronic Obstructive Pulmonary Disease (COPD) utilized wearable finger pulse oximeters [27]. A study by Luks et al. [28] indicated that oximeters could be advantageous for home monitoring of COVID-19 patients experiencing silent hypoxemia. In a separate study employing a wearable reflective pulse oximeter sensor to obtain respiratory rate data [29], researchers illustrated that the photoplethysmographic (PPG) signal from the pulse oximeter could be analyzed to discern respiratory patterns through the examination of low-frequency modulations.

1.2 Objective of the Thesis

In intensive care units, providing mechanical ventilation support to patients experiencing respiratory failure is a crucial treatment method. The selection of ventilation mode depends on various factors, including the patient's respiratory mechanics, existing pathological condition, and the desired goals of ventilation. The ventilation mode is selected based on these factors through the assessment of physicians. This thesis aims to develop an AI-based decision support system for classifying ventilation modes and predicting respiratory data to facilitate the monitoring of respiratory parameters in patients undergoing mechanical ventilation in intensive care units. Additionally, it seeks to demonstrate the effectiveness of thoracic and pulse oximeter sensors in respiratory assessment, as evidenced in the literature. This thesis presents a comparative analysis of the performance of machine learning and deep learning algorithms on respiratory data.

1.3 Hypothesis

The objective is to explore the potential of artificial intelligence in selecting appropriate ventilation modes and monitoring and evaluating respiratory values in patients receiving mechanical ventilation support in intensive care units. Accurate classification of ventilation modes using AI-based algorithms and time-series prediction of respiratory data can contribute to improving patient outcomes and reducing the duration of mechanical respiratory support. By applying machine learning and deep learning models, it is possible to provide decision support to healthcare professionals, making ventilator use more efficient and effective in the respiratory support process. Additionally, this study demonstrates that the use of thoracic and pulse oximeter sensors can be effective in patients requiring respiratory support or respiratory monitoring. The proposed approach is supported by machine learning and deep learning models.

2 MACHINE LEARNING ALGORITHMS

The machine learning methods employed for ventilation mode classification comprise decision tree, Bagging, random forest, gradient boosting, adaptive boosting (AdaBoost), K-Nearest Neighbors, gaussian naive bayes, and extreme gradient boosting (XGBoost).

2.1 Decision Tree

A decision tree is a supervised learning technique depicted as a tree structure, where decision nodes relate to dataset attributes, branches signify decision criteria, and leaf nodes denote the ultimate outcomes. It operates by recursively partitioning the data space to create a tree-like decision model, where each node performs a test on an attribute, each branch represents the outcome of that test, and each leaf node corresponds to a specific class label. Consequently, the model categorizes the data into uniform subsets that yield a particular output according to decision-making criteria [30]. The fundamental structure of a decision tree is illustrated in Figure 2.1.

A decision tree begins with a root node representing the entire dataset. Each decision node partitions the sample space according to a splitting criterion, such as information gain, Gini impurity, or variance reduction, by evaluating attribute values or ranges. This division procedure guarantees that each subset grows progressively homogeneous. The algorithm iteratively compares attribute values among sub-nodes for each consecutive node, continuing to partition the data until a leaf node is attained. Decision trees are significant algorithms because of their clarity and adaptability. They are extensively utilized in domains spanning customer behavior analysis to healthcare services. Their popularity arises from their capacity to properly manage both numerical and category characteristics [31].

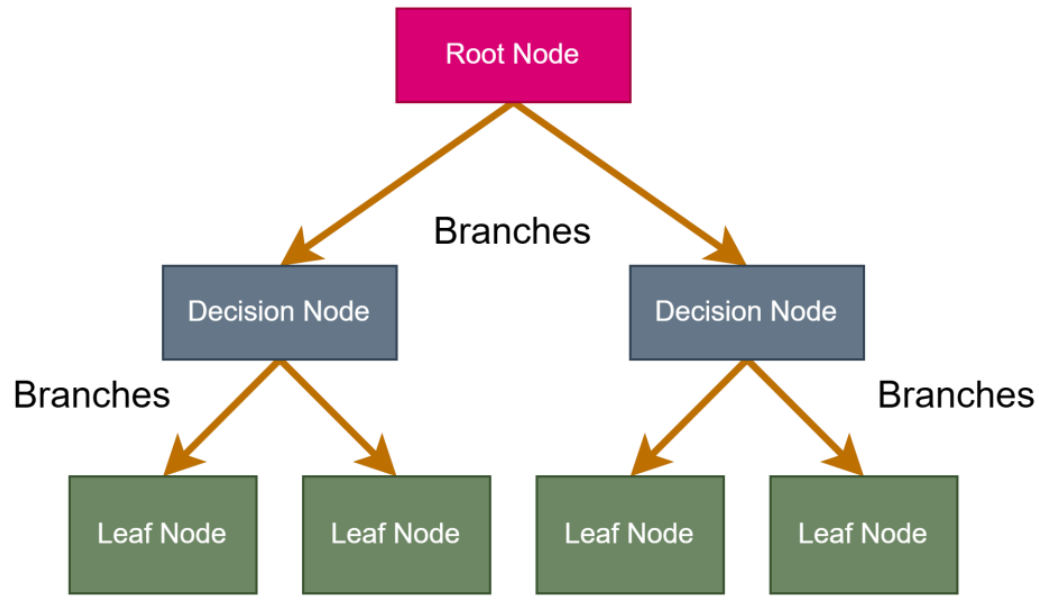


Figure 2.1 Decision Tree algorithm.

2.2 Bootstrap Aggregating (Bagging)

Bagging is an ensemble learning method developed by Leo Breiman to enhance the accuracy and stability of models, especially those exhibiting significant variation, such as decision trees [32]. Bagging generates several models of a singular kind, each trained on distinct random portions of the original dataset. Predictions from each model are then combined to produce the final output, typically by averaging in regression tasks or using majority voting in classification tasks.

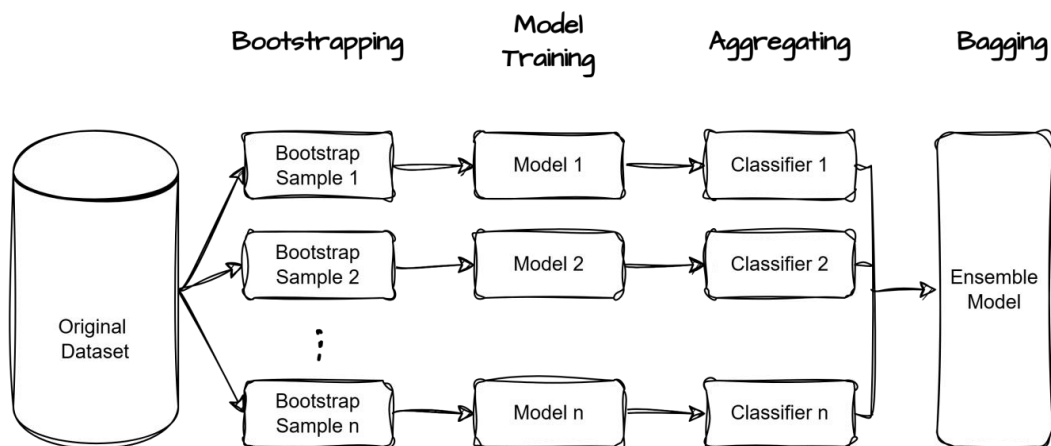


Figure 2.2 Bagging algorithm.

The Bagging algorithm, shown in Figure 2.2, functions as follows:

- The data is partitioned into random subgroups, each subset maintaining the

same size as the original dataset.

- A model is trained on every subset. Each model is derived from a unique data subset, resulting in multiple separate models.
- The final prediction for classification is established using majority voting, wherein the most commonly predicted class among the models is chosen as the definitive output.

2.3 Random Forest

The Random Forest algorithm is a machine learning method that builds multiple decision trees during training to improve prediction accuracy. It extends the bagging technique by incorporating both bagging and feature randomness. The algorithm consists of numerous independent base decision trees. Each tree is constructed from a randomly chosen subset of the dataset and considers a random subset of features at each split. This randomization introduces variability to each tree, mitigating the danger of overfitting and enhancing overall predictive performance [33]. Figure 2.3 gives a schematic representation of the random forest classifier.

The Random Forest algorithm includes the following steps:

- The data is divided into random subsets.
- A new subset of features is selected independently from the other subsets. Each subset is used to create different trees.
- Each tree is trained and makes classifications based on a subset of randomly selected features.
- The test data is applied to all trees, and each tree makes its own classification. The class with the majority vote is selected as the final class.

2.4 Gradient Boosting

Gradient Boosting is an ensemble learning method that trains multiple models, typically decision trees, sequentially [34]. This technique leverages boosting, where each model aims to correct the errors of its predecessors. By iteratively minimizing the loss function, the algorithm refines predictions to enhance accuracy. It estimates a target function by adding weak learners to reduce the overall prediction error. In each iteration, a new model is trained to match the negative gradient of the

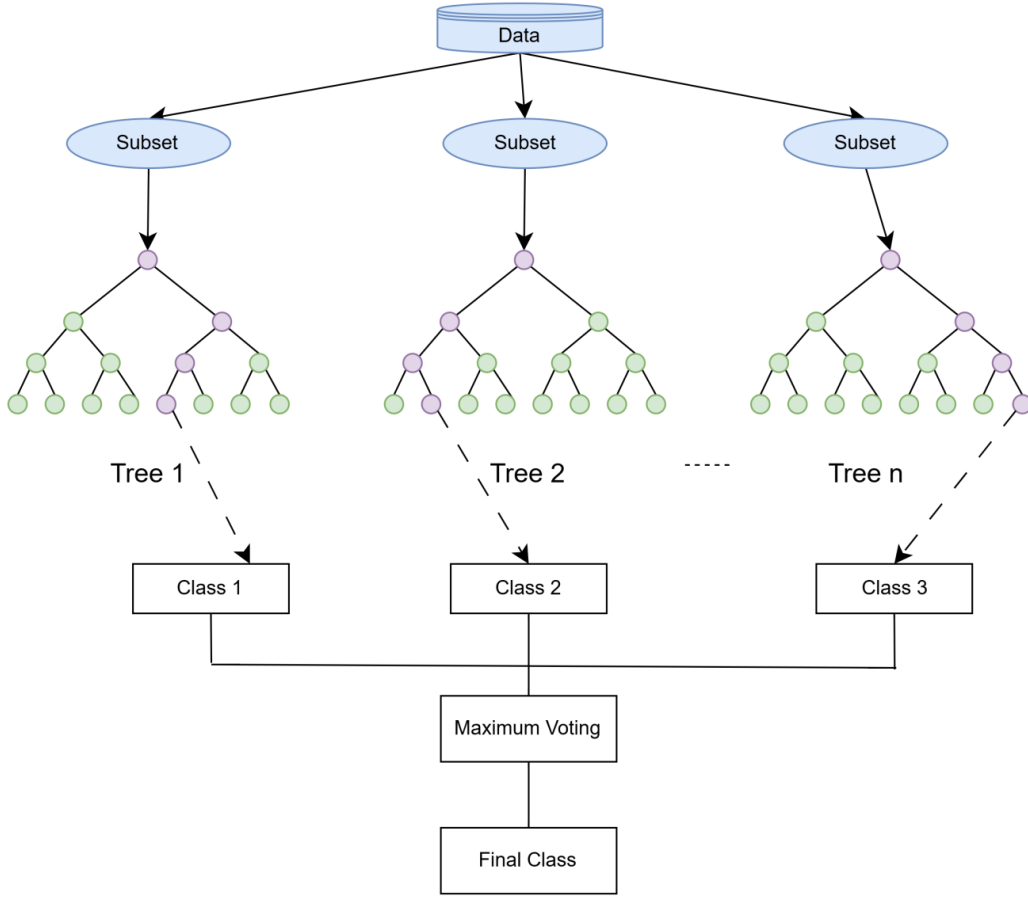


Figure 2.3 Random Forest classifier.

loss function, indicating the optimal direction for reducing errors. The process typically starts with an initial model based on the mean value of the target variable, as expressed in equation 2.1.

$$F_0(x) = \arg \min_{\rho} \left[\sum_{i=1}^N L(y_i, \rho) \right] \quad (2.1)$$

In the equation, L represents the loss function, and ρ is the predicted value. In each iteration, the residuals with negative gradients based on the current model's predictions are calculated. The residual is calculated as shown in Equation 2.2.

$$\tilde{y}_i = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}, \quad i = 1, \dots, N \quad (2.2)$$

Here, m represents the number of iterations. A new weak learner is fitted to these

residuals, allowing it to learn to correct the errors of the previous model. The value of the decision tree generated during the residual computation is depicted as follows:

$$a_m = \arg \min_{a, \beta} \left(\sum_{i=1}^N [\tilde{y}_i - \beta h(x_i; a)]^2 \right) \quad (2.3)$$

The output value of the decision tree is found as shown in equation 2.4. This equation streamlines the computation by averaging all values within a leaf.

$$\rho_m = \arg \min_{\rho} \left[\sum_{i=1}^N L(y_i, F_{m-1}(x_i) + \rho h(x_i; a_m)) \right] \quad (2.4)$$

The model is updated by adding a specific weight to this new learner. The equation that updates the model is given in equation 2.5.

$$F_m(x) = F_{m-1}(x) + \rho_m h(x; a_m) \quad (2.5)$$

The process continues until a stopping requirement is satisfied, such as a predetermined number of iterations or a minimal degree of enhancement.

2.5 Adaptive Boosting (AdaBoost)

AdaBoost is a machine learning algorithm that combines multiple weak learners to construct a stronger and more robust predictive model. The method functions by assigning weights to samples in the training dataset based on the accuracy of prior classifications, enabling the model to enhance its performance progressively via repeated rounds. In each iteration, AdaBoost increases focus on previously misclassified examples by assigning them higher weights, while reducing the weight of correctly classified examples [35]. The procedure initiates by allocating uniform weights to all instances. In every iteration, a weak learner is trained on the data utilizing the current example weights, and its error rate is assessed based on performance on the weighted examples. The weak learner is thereafter allocated a weight according to its accuracy, so augmenting the influence of more proficient learners. Additionally, the weights of misclassified examples are increased, while the weights of correctly classified examples are decreased. The final model is constructed by aggregating all weak learners by a weighted majority vote, with each learner's impact corresponding to its accuracy.

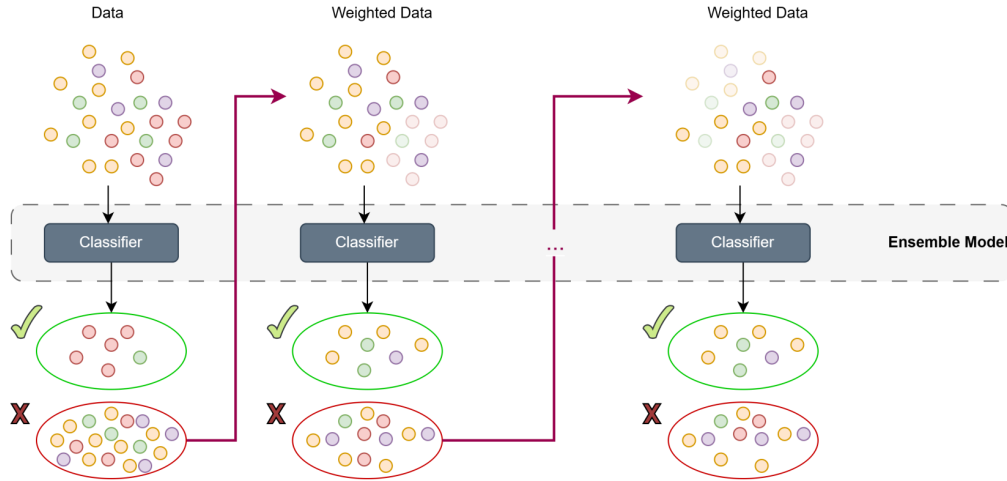


Figure 2.4 AdaBoost algorithm.

2.6 K-Nearest Neighbors (KNN)

K-Nearest Neighbors is a versatile supervised learning algorithm commonly applied to both classification and regression tasks. KNN determines the K closest points in the feature space and classifies the new data based on the predominant class among these neighbors, assessing their similarity. It employs distance metrics, such as Euclidean distance, to ascertain proximity, rendering it especially efficient for datasets where analogous instances are clustered in the feature space [36]. To classify a new data point utilizing the KNN algorithm, the parameter K is initially determined, as its value affects classification precision. For each test point, the Euclidean distance to all other points in the dataset is computed, and subsequently, the K nearest neighbors are identified based on this distance. Within these K neighbors, the quantity of data points in each category is enumerated, and the class with the highest count of nearby points is designated to the test point. Figure 2.5 demonstrates the KNN algorithm.

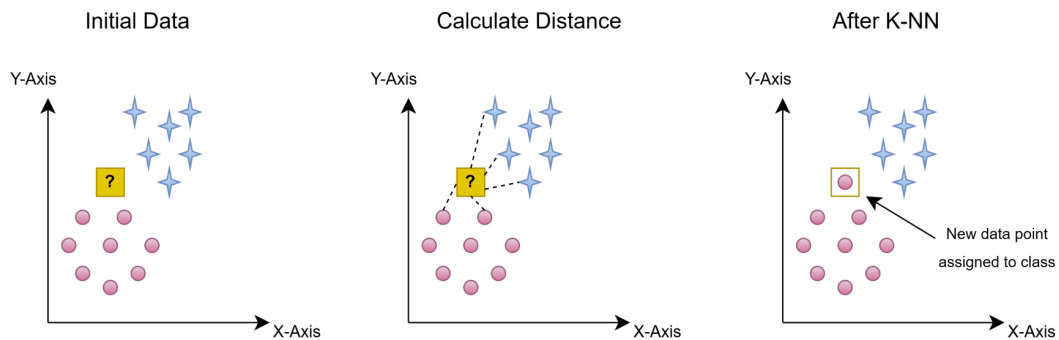


Figure 2.5 KNN algorithm.

2.7 Gaussian Naive Bayes

The Gaussian Naive Bayes algorithm is a probabilistic classification method that models each variable using a Gaussian distribution. This classifier uses Bayes' theorem to assess the likelihood of an event based on existing knowledge of associated conditions, rendering it particularly effective for managing continuous data. Gaussian Naive Bayes assumes that each feature follows a normal distribution within each class [37]. The conditional probability $P(X|C)$ of a feature X given a class C can be represented as illustrated in Equation 2.6.

$$P(X|C) = \frac{1}{\sqrt{2\pi}\sigma_C} \exp\left(-\frac{(X - \mu_C)^2}{2\sigma_C^2}\right) \quad (2.6)$$

Here, μ_C represents the mean and σ_C represents the standard deviation for class C . During training, the mean and variance of each feature for every class are computed. For a new data point, the classifier estimates the probability of the point belonging to each class by combining the feature probabilities under the Gaussian assumption and applying Bayes' theorem. The data point is thereafter allocated to the class exhibiting the highest likelihood.

2.8 Extreme Gradient Boosting (XGBoost)

XGBoost is a powerful machine learning algorithm used in regression, classification problems, and complex tasks such as large-scale data processing [38]. XGBoost is a highly scalable ensemble method based on gradient boosting. Like standard gradient boosting, XGBoost builds an additive model by minimizing a specified loss function. The XGBoost algorithm is illustrated in Figure 2.6.

The algorithm begins with a foundational model, and predictions are continuously updated. Every subsequent weak decision tree addresses the prediction errors of the preceding model. Every new tree is adjusted by a learning rate and included into the existing model's predictions. XGBoost accelerates the tree-building process due to its parallel processing capabilities. XGBoost constructs a robust model by integrating weak decision trees [39]. XGBoost employs decision trees as basis classifiers and utilizes a modified loss function to regulate tree complexity [40]. Minimizing tree complexity facilitates expedited model training and decreases storage requirements. XGBoost employs random subsampling to alleviate overfitting and enhance training efficiency. XGBoost employs a sparsity-aware approach to identify optimal splits, which automatically omits zero or missing values from the gain computation for split candidates. This feature

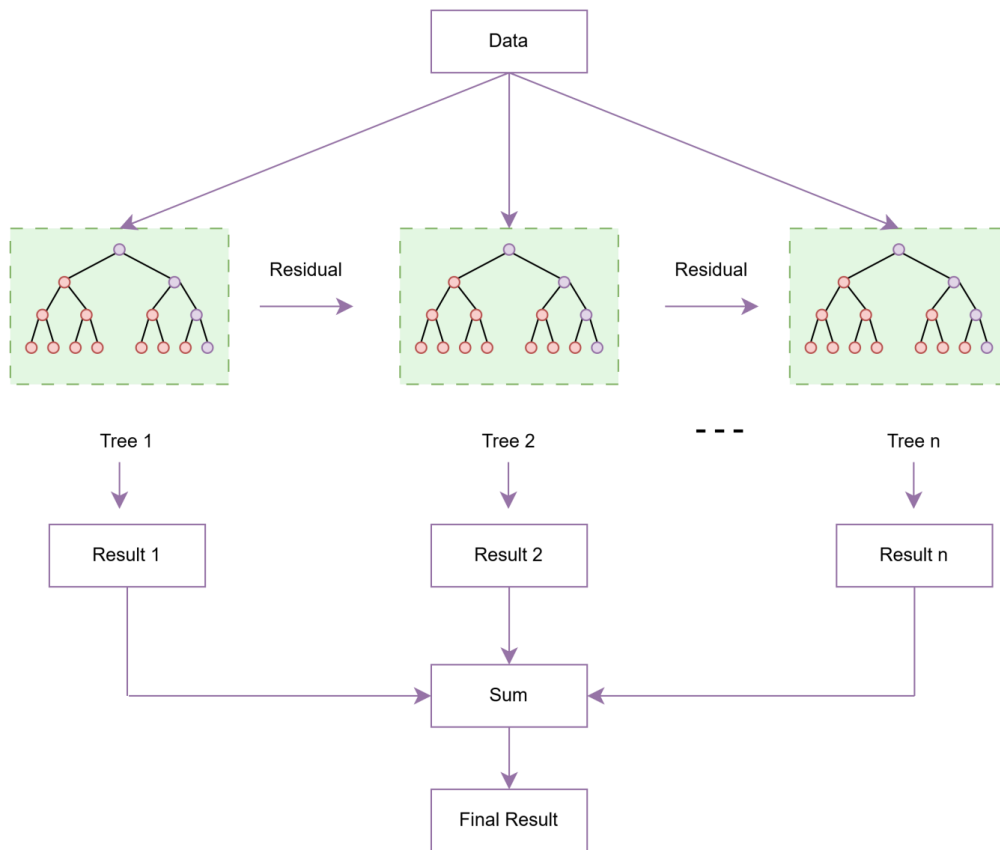


Figure 2.6 XGBoost algorithm.

enhances XGBoost's efficacy in generating very precise forecasts.

3

DEEP LEARNING ALGORITHMS

Deep learning methods used for ventilation mode classification include Long Short Term Memory and Convolutional Neural Network with Long Short Term Memory (CNN-LSTM), while LSTM, Temporal Convolutional Network (TCN), CNN-LSTM, and Temporal Convolutional Network with Long Short Term Memory (TCN-LSTM) models were utilized for respiratory data prediction.

3.1 Long Short Term Memory (LSTM)

LSTM [41] is a variant of recurrent neural networks (RNN) that can preserve long-term dependencies in sequential data. It is a deep learning model frequently employed in time series and sequential data analysis. LSTM was developed to overcome the short-term memory constraints of RNNs in time series forecasting and to mitigate gradient vanishing and gradient explosion problems. An LSTM cell consists of four key components: the input gate, cell gate, forget gate, and output gate. The design of an LSTM cell is illustrated in Figure 3.1.

The gates of an LSTM cell consist of sigmoid or hyperbolic tangent activation functions. The equations for an LSTM cell are provided in Equations (3.1)-(3.7).

$$i_t = \text{sig}(W_{xi} \times x_t \oplus W_{hi} \times h_{t-1} \oplus b_i) \quad (3.1)$$

$$f_t = \text{sig}(W_{xf} \times x_t \oplus W_{hf} \times h_{t-1} \oplus b_f) \quad (3.2)$$

$$\tilde{c}_t = \tanh(W_{xc} \times x_t \oplus W_{hc} \times h_{t-1} \oplus b_c) \quad (3.3)$$

$$o_t = \text{sig}(W_{xo} \times x_t \oplus W_{ho} \times h_{t-1} \oplus b_o) \quad (3.4)$$

$$c_t = c_t \odot f_t \oplus c_{t-1} \odot i_t \quad (3.5)$$

$$h_t = \tanh(c_t) \odot o_t \quad (3.6)$$

$$y_t = W_{hy} \times h_t \oplus b_y \quad (3.7)$$

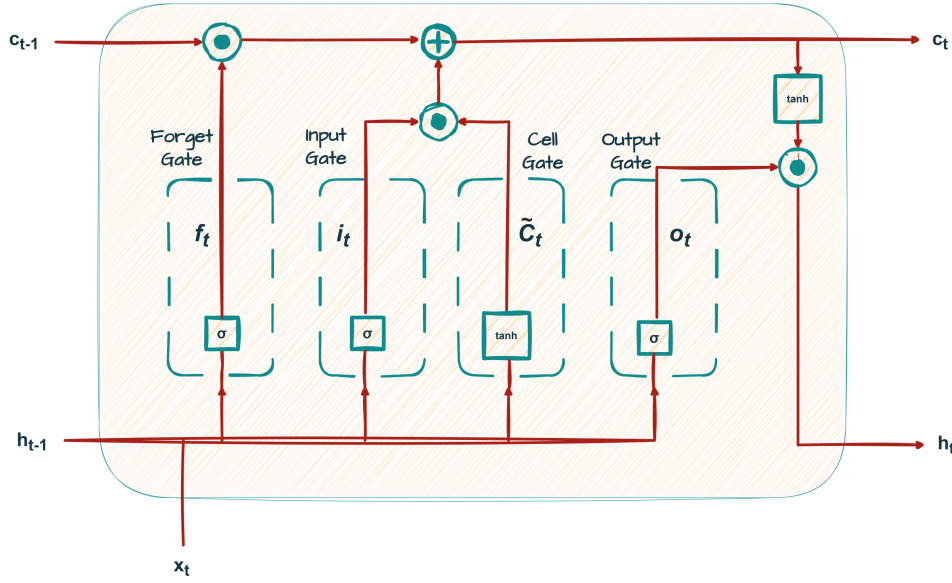


Figure 3.1 LSTM Cell

The gate outputs are $i_t, f_t, \tilde{c}_t, o_t$, and the cell outputs are h_t and c_t . The input weights perform a matrix multiplication with the input matrix x_t , while the recurrent weights perform a matrix multiplication with the hidden state vector h_{t-1} .

3.2 Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are extensively used for classification tasks. Its capacity to extract characteristics from picture data enables effective performance in object recognition and classification applications. The adaptability of CNN architecture to deal with hierarchical data models renders it effective for applications such as forecasting intricate and continuous data, including time series [42]. A CNN architecture comprises several layers, including convolutional layers, pooling layers, and fully connected layers [43].

Convolutional Layer: The convolutional layer is the fundamental unit of a CNN. This layer applies the convolution process to the input data with a collection of filters. This layer produces a collection of feature maps, each depicting distinct characteristics of the supplied data.

Pooling Layer: The pooling layer reduces the dimensions of the feature map generated by the convolutional layer. Pooling reduces the computational complexity of the network.

Dense Layer: The fully connected layer is a traditional neural network layer where each neuron is connected to every neuron in the previous layer. It is usually placed

at the end of the network to generate the final output.

3.3 Temporal Convolutional Network (TCN)

A Temporal Convolutional Network is a type of convolutional neural network architecture designed to model sequential data and effectively capture temporal dependencies. TCNs adhere to two main principles: first, they ensure that the output sequence length matches the input sequence length, similar to LSTM networks. Second, TCNs use causal convolutions to prevent information leakage from future inputs to past inputs, preserving the integrity of temporal relationships [44]. These features enable TCNs to effectively model sequential data, such as time series analysis and segmentation. To adhere to the first principle, Temporal Convolutional Networks (TCNs) employ a 1-dimensional fully convolutional network (FCN) architecture. In this design, each hidden layer matches the length of the input layer, with zero padding applied to maintain the same length across all layers. This convolution increases the receptive field of the network without using pooling operations, thus avoiding resolution loss. To satisfy the second principle, TCNs employ a causal convolution layer, where the output at any time t is connected only to inputs observed at the same time or earlier. This prevents future inputs from influencing past predictions and maintains sequence integrity [45]. TCNs utilize dilated convolutions to exponentially increase the receptive field while keeping the number of parameters constant, allowing the model to effectively capture long-term dependencies. This makes TCNs well-suited for tasks that require knowledge of earlier inputs over extended time periods [46]. The TCN architecture is shown in Figure 3.2.

Dilated Causal Convolution: Dilating convolutions are accomplished by introducing an established period between each filter. A larger dilation is employed to guarantee that a higher-level output covers a broader spectrum of inputs. In this context, d signifies the dilation factor, while k indicates the filter size.

Residual Connections: TCN architectures use residual block connections to facilitate the learning process [47]. Each residual block consists of a dilated convolutional layer, weight normalization, ReLU, and a dropout layer. If needed, a 1×1 convolution layer can be applied to match the shapes of the input and output.

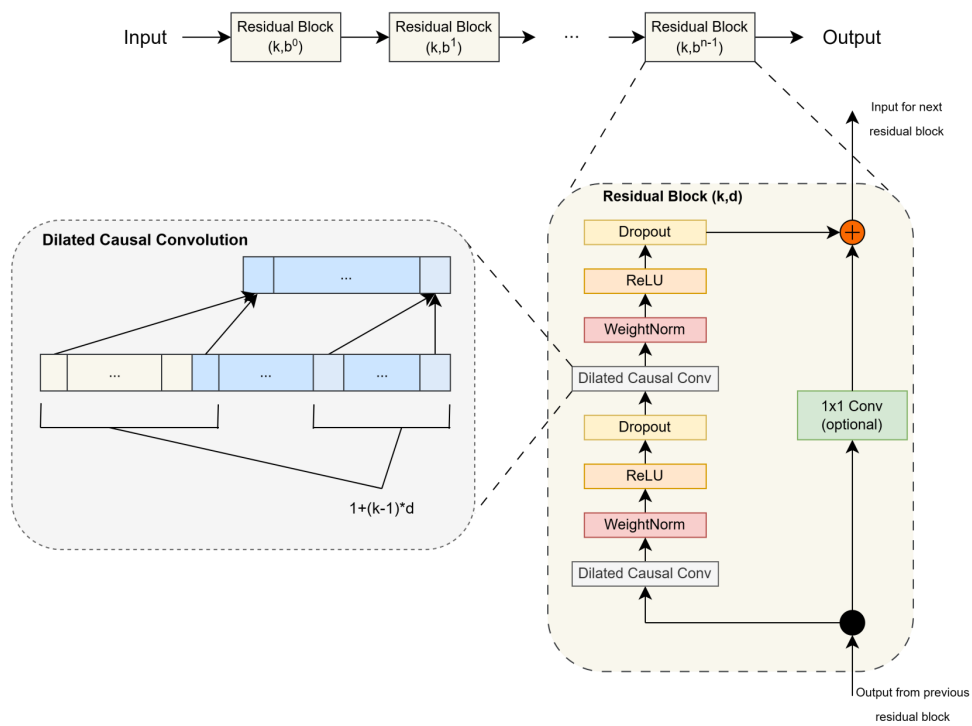


Figure 3.2 TCN architecture.

DATASET AND PERFORMANCE METRICS

This section presents actual patient data obtained from 23 individuals undergoing mechanical breathing in the neurology intensive care unit. The preprocessing steps used before applying machine learning and deep learning algorithms for ventilation mode classification and respiratory data prediction will be discussed. The study delineates performance criteria that enable the assessment of data-model fit and overall model efficacy through several evaluative measures of algorithm performance.

4.1 Dataset

The dataset used in this thesis was assembled from authentic patient data acquired under clinical conditions, consistent with the study's aims. Data from mechanical ventilators was gathered from patients undergoing mechanical respiratory support in the intensive care unit of the neurology department of Istanbul Başakşehir Çam and Sakura City Hospital, as part of a collaborative research initiative funded by Higher Education Institutions. Data were gathered from 23 patients, comprising 15 men and 8 women aged 45 to 90, who had resided in the neurology critical care unit with mechanical ventilation for the preceding year. These patients are persons with neurological or pulmonary disorders who were intubated and necessitated mechanical ventilation for over 5 days.

Data for each patient were collected in 1-hour periods for 24 hours. The data collection period in terms of the number of days varies for each patient. This period is at least 1 month (30 days) and at most 3 months (90 days). 18 features were collected, 7 of which were monitor data; pulse, Mean AP, systolic artery, diastolic artery, temperature, RESP, SPO2 and 11 of which were ventilator data; oxygen, VTE, VTI, PEEP, PEEP Setting, PPEAK, PPLAT, PMEAN, FTOT, inspiration and expiration. The graph of a patient's respiratory data measured over time is given in Figure 4.1. The same patient's 5-hour measurement data is shown in Table 4.1.

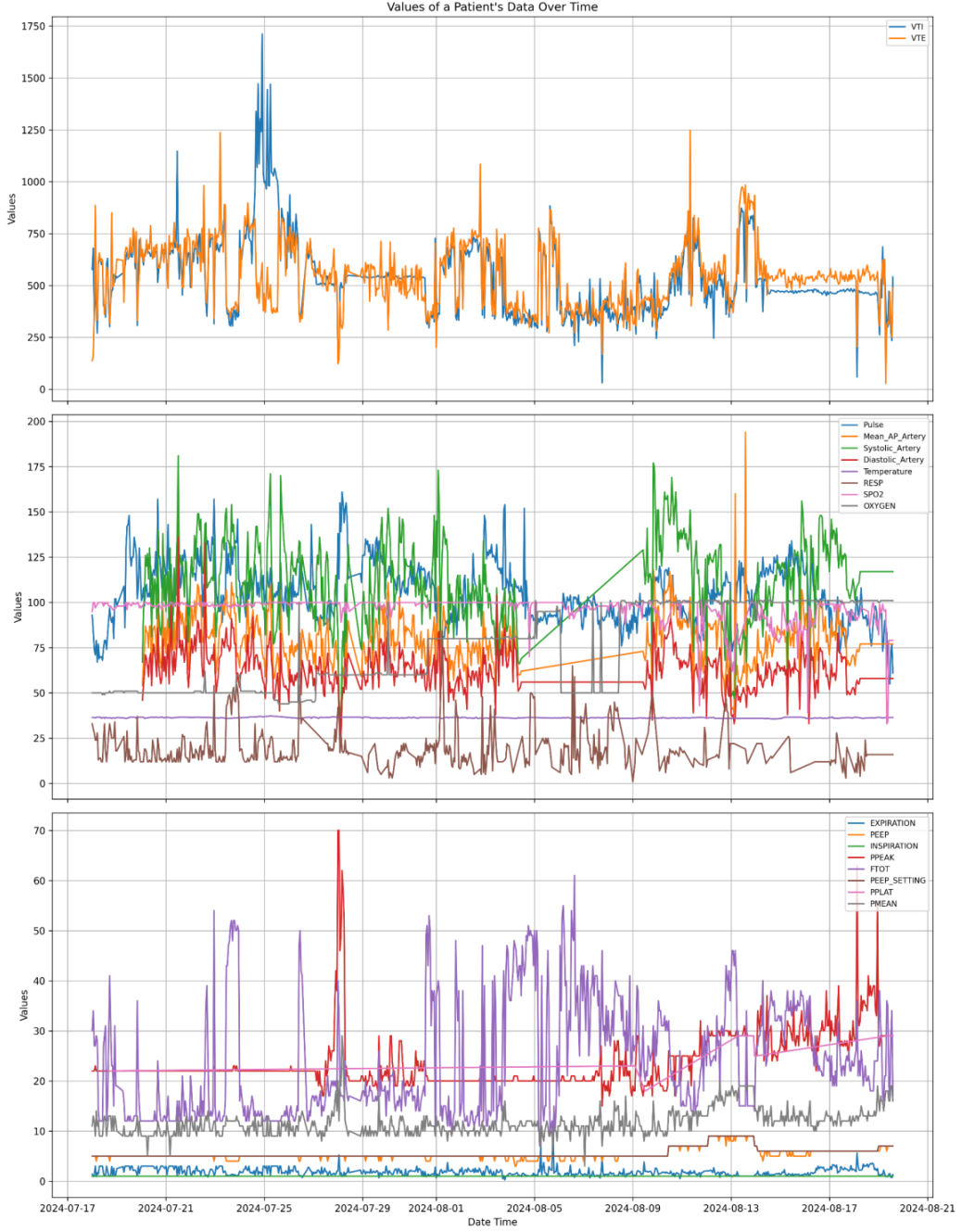


Figure 4.1 Graph of time-dependent data value.

Each observation in time series data $x_{i,t} \in R^F$ (feature dimension F) represents a single data point captured at a specific timestamp t . The basic structure of a collected time series is shown in Figure 4.2. Here i denotes the index. This time series data is a sequence of measurements at regular intervals. The sampled data of a parameter at a specific time interval (T) can be expressed as $x_i = \{x_{i,t} | t = 1, \dots, T\}$.

Patient data is classified into four ventilation classes: Continuous Mandatory Ventilation (CMV), Pressure-Supported Intermittent Mandatory Ventilation

Table 4.1 5 hours of data from a patient.

Date and Time	07/20/2024 1:00 AM	07/20/2024 2:00 AM	07/20/2024 3:00 AM	07/20/2024 4:00 AM	07/20/2024 5:00 AM
Pulse	103.00	108.00	98.00	108.00	110.00
Mean_AP_Artery	54.00	66.00	74.00	87.00	84.00
Systolic_Artery	67.00	94.00	105.00	126.00	120.00
Diastolic_Artery	46.00	54.00	61.00	71.00	69.00
Temperature	36.33	36.60	36.00	36.33	36.33
RESP	12.00	12.00	19.00	19.00	17.00
SPO2	98.00	99.00	100.00	100.00	100.00
OXYGEN	50.00	50.00	50.00	50.00	50.00
VTE	674.00	615.00	652.00	643.00	720.00
VTI	643.00	599.00	642.00	670.00	686.00
PEEP	5.00	5.00	5.00	5.00	5.00
PEEP_SETTING	5.00	5.00	5.00	5.00	5.00
PPEAK	22.00	22.00	22.00	22.00	22.00
PPLAT	9.00	9.00	9.00	9.00	9.00
PMEAN	12.00	12.00	12.00	12.00	12.00
FTOT	1.00	1.00	1.00	1.00	1.00
Inspiration	1.00	1.00	1.00	1.00	1.00
Expiration	3.00	3.00	3.00	3.00	3.00
VENTILATION MODE	PSIMV	PSIMV	PSIMV	PSIMV	PSIMV

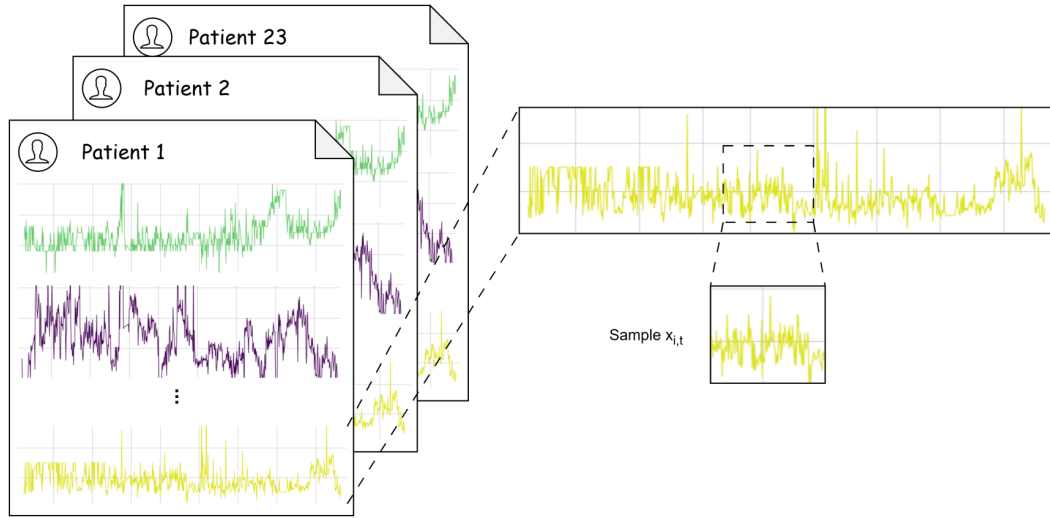


Figure 4.2 Basic structure of time series.

(PSIMV), Adaptive Support Ventilation (ASV), and Spontaneous Ventilation (SPONT). Figure 4.3 presents the bar chart depicting the distribution of these classes.

4.1.1 Ventilation Modes

The data collected from patients includes four different ventilation modes: Continuous Mandatory Ventilation (CMV), Pressure-Supported Intermittent Mandatory Ventilation (PSIMV), Adaptive Support Ventilation (ASV), and Spontaneous Ventilation (SPONT).

Continuous Mandatory Ventilation (CMV)

CMV is a mechanical ventilation method generally utilized for patients with severe respiratory failure who cannot breathe independently [48].

Pressure-Supported Intermittent Mandatory Ventilation (PSIMV)

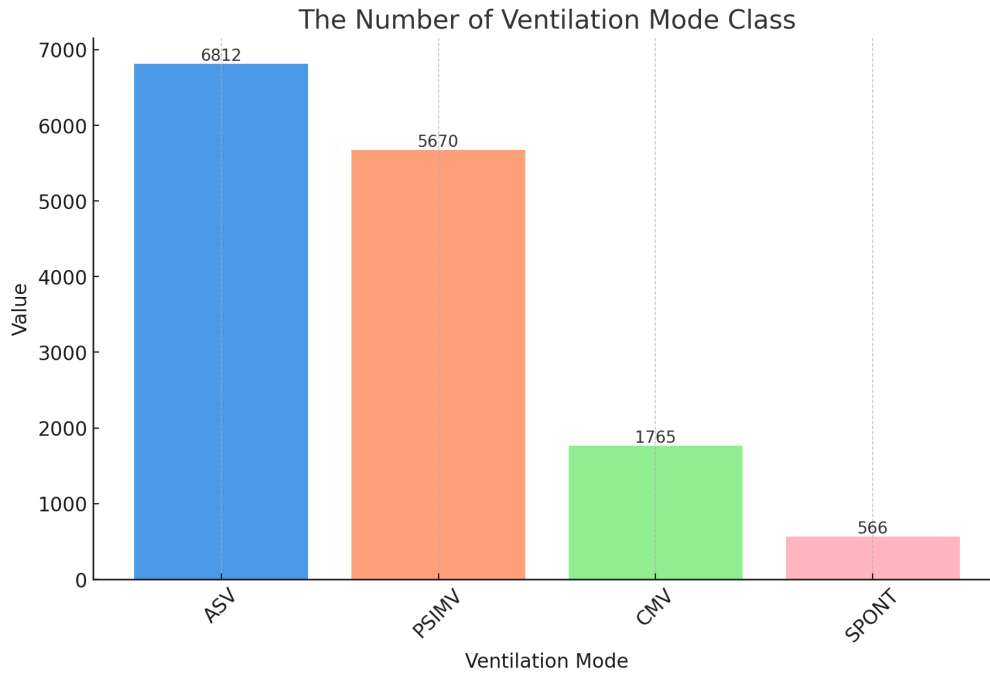


Figure 4.3 Bar chart showing class distribution.

PSIMV is a pressure-controlled mode that facilitates spontaneous respiration [49]. The ventilator delivers a predetermined quantity of necessary breaths at a designated pressure, while permitting the patient to initiate spontaneous breaths with pressure assistance.

Adaptive Support Ventilation (ASV)

The ASV mode is a ventilation mode that autonomously adjusts to the patient's respiratory condition [50]. The ventilator provides mandatory breaths when the patient's efforts are insufficient, ensuring appropriate ventilation while permitting spontaneous breathing whenever feasible.

Spontaneous Ventilation (SPONT)

In spontaneous ventilation modes, the patient control their breathing efforts without mandatory support from the ventilator.

4.1.2 Data Properties

The data consists of 18 features, including monitor and mechanical ventilator data [51], [52]:

- **Pulse:** Indicates the frequency of heartbeats, usually measured in beats per

minute.

- **Mean Arterial Pressure (Mean AP):** It is the average value of blood pressure.
- **Systolic Blood Pressure (Syatolic Artery):** It represents the maximum pressure applied to the arteries during the heart's contraction phase.
- **Diastolic Blood Pressure (Diastolic Artery):** It indicates the minimum pressure that blood exerts in the arteries during cardiac contraction.
- **Temperature:** It refers to body temperature.
- **Respiratory Rate (RESP):** It indicates the number of breaths inhaled in a single minute.
- **Oxygen Saturation (SPO2):** Indicates the rate of oxygen in the blood.
- **Oxygen:** Indicates the amount of oxygen delivered by the ventilator.
- **Expiratory Tidal Volume (VTE):** It is the volume of gas breathed into the patient's lungs with each breath.
- **Inspiratory Tidal Volume (VTI):** Indicates the set tidal volume.
- **Positive End-Expiratory Pressure (PEEP):** It represents the remaining positive pressure in the lungs after exhalation.
- **PEEP Setting:** The PEEP value is determined clinically.
- **PPEAK:** The peak pressure provided by the ventilator during inspiration.
- **Plateau pressure (PPLAT):** The pressure measured in the lungs when the airflow provided by the ventilator is stopped.
- **Mean airway pressure (PMEAN):** The average airway pressure in the lungs.
- **FTOT:** The total airflow provided by the ventilator for each breath.
- **Inspiration:** The duration of airflow into the lungs during inspiration.
- **Expiration:** The duration of airflow out of the lungs during expiration.

4.1.3 Dataset Preprocessing

Various preprocessing steps were applied to the respiratory dataset to obtain more reliable and meaningful results before analysis. Min-Max normalization, a Gaussian filter, and a Moving Average filter were applied to the data.

4.1.3.1 Min-Max Normalization

The Min-Max normalization method is a scaling approach that rescales the original data to a range of $[0, 1]$ [53]. The equation for calculating the normalized value of each feature is given in equation 4.1.

$$x' = \frac{x - X_{min}}{X_{max} - X_{min}} \quad (4.1)$$

Here, x' is the normalized value, x is the original value of the feature, X_{min} is the minimum value of the feature in the dataset, and X_{max} is the maximum value of the feature in the dataset. After the normalization process, the values given in Table 4.1 are presented in Table 4.2.

Table 4.2 5 hours normalized data of a patient.

Date and Time	07/20/2024 1:00 AM	07/20/2024 2:00 AM	07/20/2024 3:00 AM	07/20/2024 4:00 AM	07/20/2024 5:00 AM
Pulse	0.45	0.50	0.41	0.50	0.52
Mean_AP_Artery	0.15	0.22	0.27	0.35	0.33
Systolic_Artery	0.24	0.42	0.49	0.63	0.59
Diastolic_Artery	0.17	0.25	0.31	0.40	0.39
Temperature	0.45	0.59	0.24	0.45	0.45
RESP	0.13	0.13	0.22	0.22	0.19
SPO2	0.97	0.99	1.00	1.00	1.00
OXYGEN	0.11	0.11	0.11	0.11	0.11
VTE	0.53	0.48	0.51	0.50	0.57
VTI	0.36	0.34	0.36	0.38	0.39
PEEP	0.33	0.33	0.33	0.33	0.33
PEEP_SETTING	0.00	0.00	0.00	0.00	0.00
PPEAK	0.13	0.13	0.13	0.13	0.13
PPLAT	0.23	0.23	0.23	0.23	0.23
PMEAN	0.09	0.09	0.09	0.09	0.09
FTOT	0.26	0.26	0.26	0.26	0.26
Inspiration	1.00	1.00	1.00	1.00	1.00
Expiration	0.02	0.02	0.02	0.02	0.02
VENTILATION MODE	PSIMV	PSIMV	PSIMV	PSIMV	PSIMV

4.1.3.2 Moving Average Filter

The moving average filtering algorithm is a method frequently employed for the analysis of time series data. It executes low-pass filtering, distinguished by little computational complexity and rapid processing speed [54]. The procedure of the algorithm is outlined as follows:

- **Step 1** The window size is defined for averaging the sampled data.
- **Step 2** The arithmetic mean of the data points within the window is calculated.
- **Step 3** The window is shifted one position at a time, adhering to the first-in, first-out principle to select data points.
- **Step 4** As the window moves over the data, the output values form a smooth curve.

$$P(n) = \frac{1}{N} \sum_{i=0}^{N-1} x_{n-i} \quad (4.2)$$

The one-sided moving average filter is given in equation 4.2. Here, $P(n)$ is the average value calculated for the n -th movement, N is the number of elements in the iterative average, and x_{n-i} is the sampling value.

When applying the moving average filter to respiratory data, window sizes of 2, 3, and 5 were tested in the deep learning algorithm. Using a window size of 3 yielded higher accuracy in the algorithm.

4.1.3.3 Gaussian Filter

The Gaussian filter is a linear filter that utilizes the Gaussian function to eliminate outliers in images and signals, enhance predictions, and minimize error [55]. This filter is frequently employed in time series data. The Gaussian function is delineated in equation 4.3 [56].

$$G(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{t^2}{2\sigma^2}} \quad (4.3)$$

Here, σ is the standard deviation value that controls the width of the Gaussian curve. In time series data, the filtering process involves the convolution of the time series data with a Gaussian kernel. The convolution is expressed as in equation 4.4.

$$y[n] = \sum_m x[m]G[n - m] \quad (4.4)$$

Here, $y[n]$ represents the filtered time series, $x[m]$ is the input data, and $G[n - m]$ represents the Gaussian kernel centered around each point in the input time series.

When applying the Gaussian filter to respiratory data, values of σ , 2, 4, and 6 were tested in the deep learning algorithm. A higher accuracy was achieved in the algorithm when the data was filtered with $\sigma = 2$.

4.2 Performance Metrics

Evaluation metrics are essential for measuring the performance of classifiers in machine learning and deep learning models. These metrics facilitate the selection and evaluation of suitable metrics to enhance classifiers and guarantee their effective generalization to test data, hence enabling model adjustments based on these findings [57]. Each metric type offers insights into diverse objectives, including the evaluation of generalization capacity, model selection, and optimization of solutions during classification training. Evaluation metrics are classified into three categories: threshold-based, rank-based, and probabilistic metrics [58]. This study employed threshold-based metrics including accuracy, precision, F1 score, and recall; rank-based metrics such as Area Under the ROC Curve (AUC); and regression metrics comprising Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 (coefficient of determination).

4.2.1 Threshold Based Metrics

Accuracy

Accuracy, a basic evaluation metric, quantifies the overall correctness of a model by calculating the ratio of correctly predicted instances to the total number of examples. The equation for accuracy is given in equation 4.5.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.5)$$

Here, TP represents true positives, FP false positives, TN true negatives, and FN false negatives. A high accuracy value indicates that the model performs well and is suitable for the data and classification problem. Low accuracy may suggest that the model struggles with classification and that there may be imbalanced classes in the data.

Precision

Precision measures the proportion of true positive predictions out of all positive predictions made by the model. Precision is calculated according to equation 4.6.

$$Precision = \frac{TP}{TP + FP} \quad (4.6)$$

A high precision value signifies that the model's positive class predictions are typically accurate. A low precision score, on the other hand, suggests that the model often incorrectly predicts positive classes, indicating poor model performance.

Recall

Recall measures the proportion of true positive predictions relative to the total number of actual positive examples. Recall is given in equation 4.7.

$$Recall = \frac{TP}{TP + FN} \quad (4.7)$$

Recall measures the model's capacity to recognize all relevant instances within the dataset. A high recall value indicates that the model successfully captures most of the relevant data and positive instances. A low recall value suggests that the model struggles to correctly identify positive instances.

F1-Score

The F1 score is the harmonic mean of precision and recall, offering a balanced measure that considers both metrics.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (4.8)$$

If the F1 score, given in equation 4.8, has a high value, it indicates a good balance between precision and recall. This means the model has a high rate of identifying true positive examples while minimizing false positives. A low F1 score, on the other hand, indicates that the model is unsuccessful in predicting positive classes accurately.

4.2.2 Rank Based Metrics

Area under the ROC Curve (AUC)

The ROC curve visually depicts the relationship between the true positive rate and the false positive rate across different threshold values. The area under the curve serves as a metric to evaluate the classifier's capability to differentiate between positive and negative classes. An AUC value ranges from 0 to 1, where 0.5 signifies

no discrimination ability, and 1.0 represents perfect classification performance.

4.2.3 Regression Metrics

Mean Squared Error (MSE)

The mean squared error calculates the average of the squared deviations between the predicted values and the actual values. The squaring process eliminates negative signs [59], ensuring that the MSE is always a positive value. MSE is presented in equation 4.9.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_{true,i} - Y_{pred,i})^2 \quad (4.9)$$

Here, n is the number of predictions, Y_i are the observed values, and $\hat{Y}_i$ are the predicted values. The closer the MSE is to 0, the more accurate the model's predictions.

Root Mean Squared Error (RMSE)

RMSE is the square root of MSE and provides the standard deviation of the errors. RMSE is given in equation 4.10. A low RMSE value, like MSE, indicates good model performance.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_{true,i} - Y_{pred,i})^2} \quad (4.10)$$

Mean Absolute Error (MAE)

The MAE calculates the average of the absolute differences between predicted and actual values. Smaller MAE values suggest that the model's predictions are closer to the true values. The formula for MAE is presented in Equation 4.11.

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_{true,i} - Y_{pred,i}| \quad (4.11)$$

R-Squared (R^2)

The coefficient of determination, R^2 , indicates the proportion of variance in the dependent variable that is accounted for by the independent variables. It serves as a measure of the regression model's goodness of fit. R^2 is given in equation 4.12.

$$R^2 = 1 - \frac{SSR}{TSS} \quad (4.12)$$

SSR is the sum of squared residuals, representing the difference between predicted and actual values. SSR , as given in Equation 4.13, measures this difference.

$$SSR = \sum_{i=1}^n (Y_{true,i} - Y_{pred,i})^2 \quad (4.13)$$

TSS represents the total variance in the actual data. It is given in Equation 4.14.

$$TSS = \sum_{i=1}^n (Y_{true,i} - \hat{Y}_{true,i})^2 \quad (4.14)$$

- $R^2 = 1$: It indicates that the model explains all the variance in the data well.
- $R^2 < 0$: It indicates poor model performance and suggests that the model is not a good fit for the data.
- $R^2 = 0$: It indicates that the model explains none of the variance in the data.

VENTILATION MODE CLASSIFICATION AND RESPIRATORY DATA ESTIMATION

The monitoring and assessment of respiratory data are essential for the treatment and follow-up of patients with respiratory diseases. For patients connected to mechanical ventilators, selecting the correct ventilation mode is critically important. Proper evaluation of respiratory values is necessary when choosing the ventilation mode. For this purpose, this study performs ventilation mode classification and respiratory data prediction. This section describes the applications of machine learning and deep learning algorithms used for ventilation mode classification and respiratory data prediction, along with their comparative results.

5.1 Ventilation Mode Classification

Mechanical ventilation techniques are crucial for sustaining the lives of individuals with severe conditions that require prolonged hospitalization. However, the imbalance between the respiratory demands and capacities of hospitalized patients may lead to inconsistencies in their inhalation. As the parameters of ventilation equipment are entirely determined by the physician, their expertise and experience in selecting these settings play a critical role in the accuracy and effectiveness of their decisions. Therefore, this thesis aims to support healthcare professionals in the decision-making process by classifying ventilator modes to facilitate more accurate ventilation mode selection.

Machine learning algorithms used for ventilation mode classification included decision tree, Bagging, random forest, gradient boosting, adaptive boosting, K-Nearest Neighbors, gaussian naive bayes, and extreme gradient boosting, along deep learning algorithms such as LSTM and CNN-LSTM multi-models. The general scheme of classification is given in Figure 5.1.

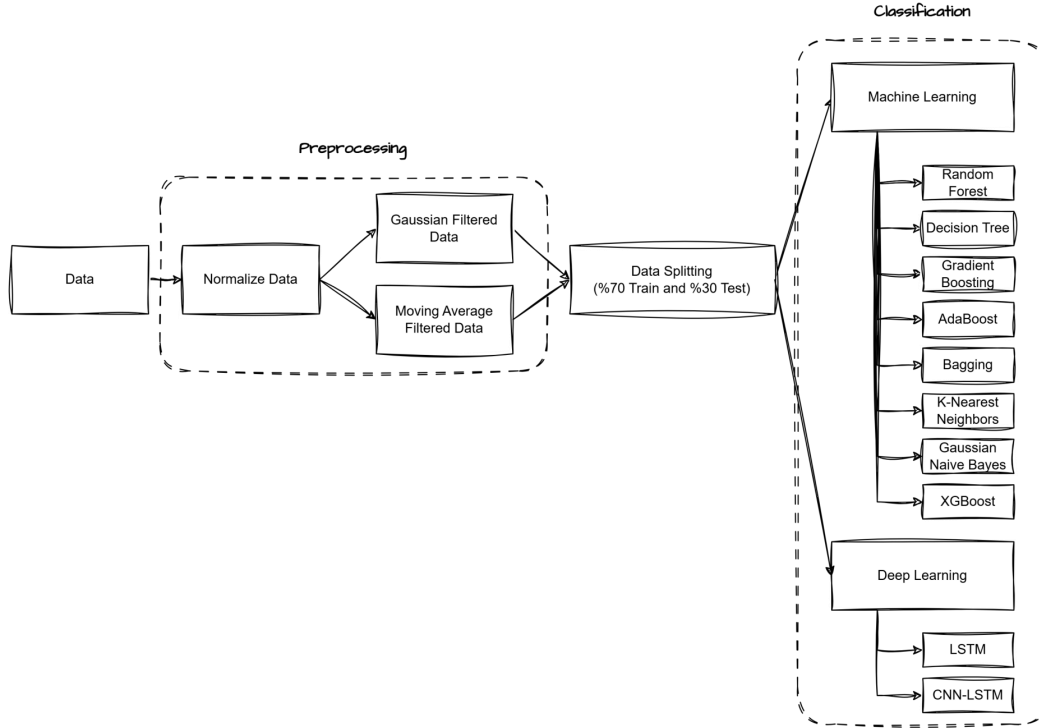


Figure 5.1 General scheme of classification.

5.1.1 Classification with Machine Learning Algorithms

Machine learning algorithms are used for purposes such as classification and prediction in the analysis of time series data. In recent years, the use of machine learning algorithms for time series analysis has increased. In this section, the performance of machine learning algorithms on ventilation mode classification is evaluated.

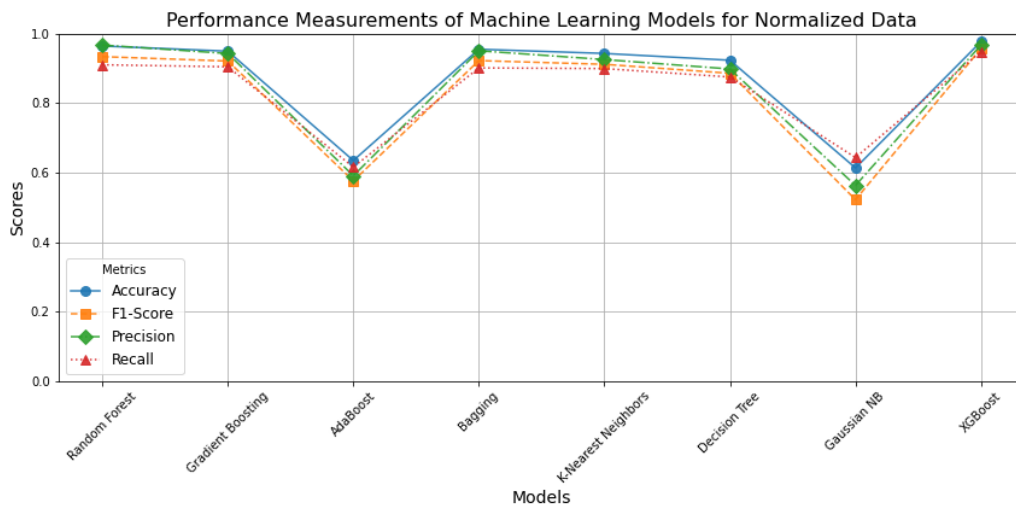


Figure 5.2 Comparison of metrics obtained as a result of applying machine learning algorithms to normalized data.

This study uses Random Forest, Decision Tree, Gradient Boosting, AdaBoost, Bagging, K-Nearest Neighbors, Gaussian Naive Bayes, and XGBoost algorithms. The data is divided into three groups: normalized data, data with a moving average filter applied, and data with a Gaussian filter applied. Each of these three datasets is applied to each model, and evaluation metrics are obtained. The performance of these eight algorithms is then assessed and compared. These algorithms were tuned and applied according to the hyperparameters provided in Table 5.1.

Figure 5.2 presents the graph comparing the evaluation metrics derived from the application of normalized data to machine learning methods.

XGBoost demonstrated the best performance among the models, with an accuracy rate of 97%. The ROC curve and confusion matrix for each model are presented in Figure 5.3, 5.4.

Based on the ROC curves and confusion matrices evaluated for the four classes, XGBoost was found to have a higher classification capability compared to the other models. Random Forest is the second-best algorithm after XGBoost, but its classification capability is not as strong as that of XGBoost.

A moving average filter was applied to the normalized data as described in Section 4.1.3.2. The graph comparing the evaluation metrics obtained from classifying the filtered data using machine learning algorithms is presented in Figure 5.5.

XGBoost demonstrated the highest performance among the models, achieving an accuracy rate of 98%. Figure 5.6, 5.7 presents the ROC curve and confusion matrix for each model.

Analysis of the ROC curves and confusion matrices for the four classes indicated that XGBoost had superior classification performance relative to the other models.

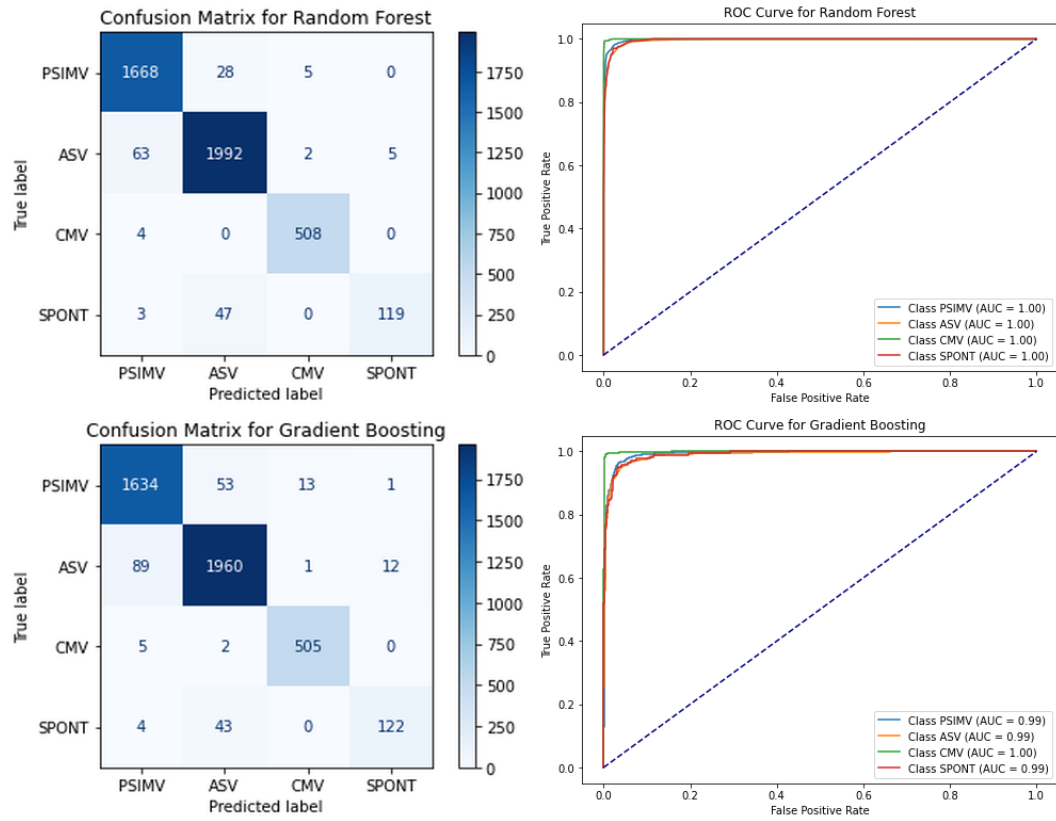
A Gaussian filter was applied to the normalized data as described in Section 4.1.3.3. The graph comparing the evaluation metrics obtained from classifying the filtered data using machine learning algorithms is presented in Figure 5.8.

XGBoost demonstrated the highest performance among the models, achieving an accuracy rate of 98%. Figure 5.9, 5.10 presents the ROC curve and confusion matrix for each model.

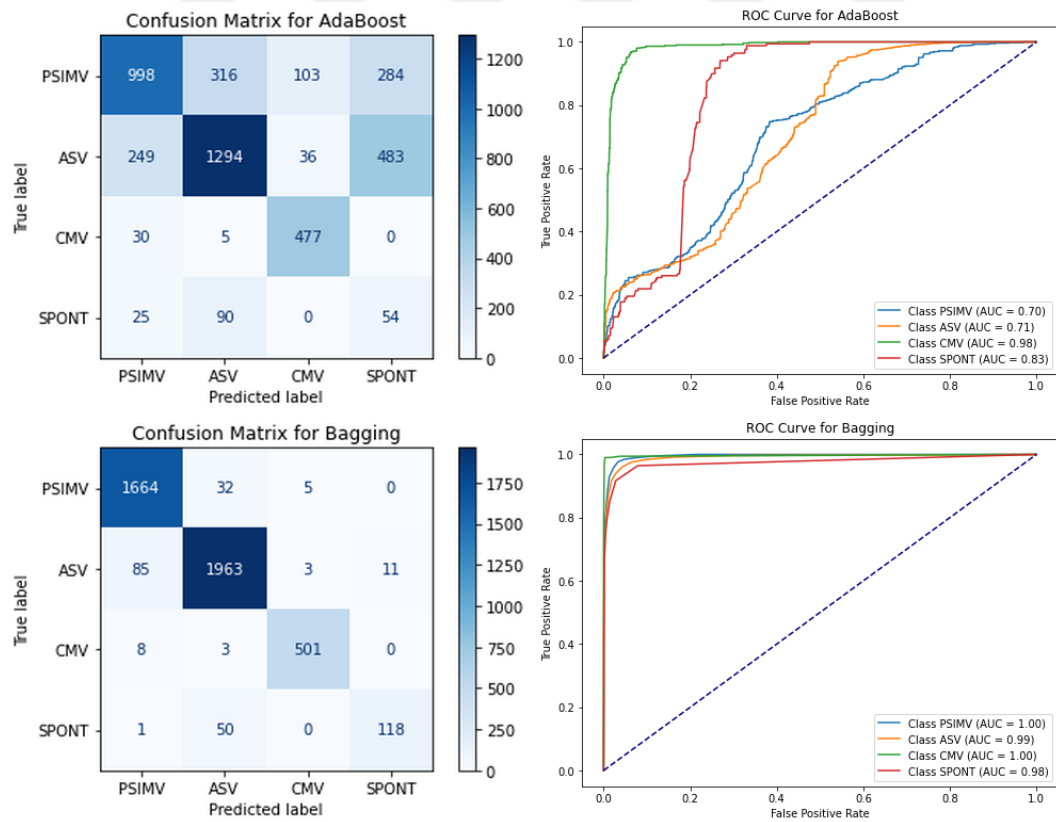
According to the ROC curves and confusion matrices, Random Forest demonstrated higher classification capability for the PSIMV class, KNN for the CMV class, and XGBoost outperformed other models for the remaining two classes.

Table 5.1 Hyperparameter configurations for machine learning models.

Model	Hyperparameter Description	Configured Value
Decision Tree	Minimum samples required to split a node	2
	Minimum samples required at a leaf node	1
Bagging	Total number of base estimators in the ensemble	10
	Proportion of samples drawn from the dataset for training each base estimator	1.0
	Fraction of input features selected for each base estimator	1.0
Random Forest	Total number of trees in the forest	100
	Minimum samples needed to perform a node split	2
	Minimum samples needed to remain at a leaf node	1
Gradient Boosting	Total estimators used in the ensemble	100
	Learning rate to control overfitting	0.1
	Maximum depth of individual trees	3
	Minimum samples needed to split a node	2
	Minimum samples needed at a leaf node	1
	Fraction of data samples used for training each base learner	1.0
AdaBoost	Total number of weak learners in the ensemble	50
	Learning rate applied during weight updates	1.0
K-Nearest Neighbors	Number of neighbors used in predictions	5
	Weighting strategy for neighbors	'uniform'
	Distance metric parameter (p)	2 (Euclidean distance)
Gaussian Naive Bayes	Stability parameter added to variances to prevent numerical instability	1e-9
XGBoost	Objective loss function	'mlogloss'
	Maximum depth of individual trees	6
	Learning rate for boosting	0.3
	Total number of boosting iterations	100
	Proportion of training samples used per boosting step	1.0
	Proportion of input features used to build each tree	1.0

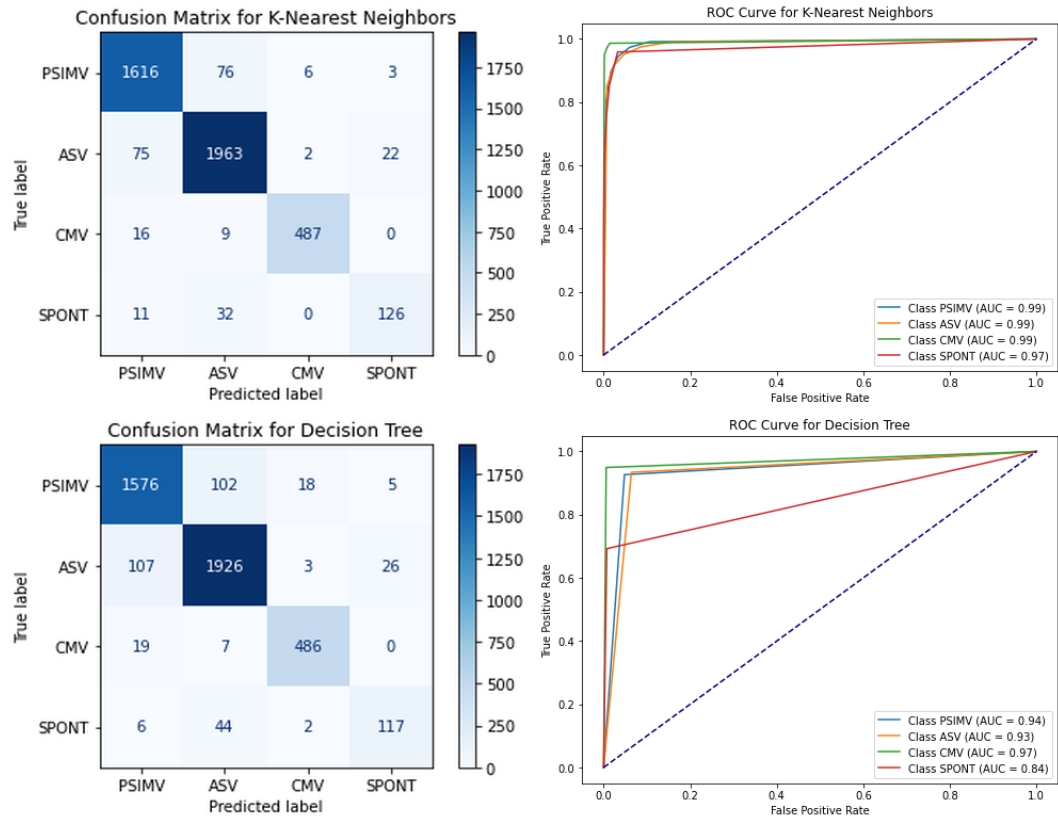


(a) ROC curve plots and confusion matrices of Random Forest and Gradient Boosting algorithms.

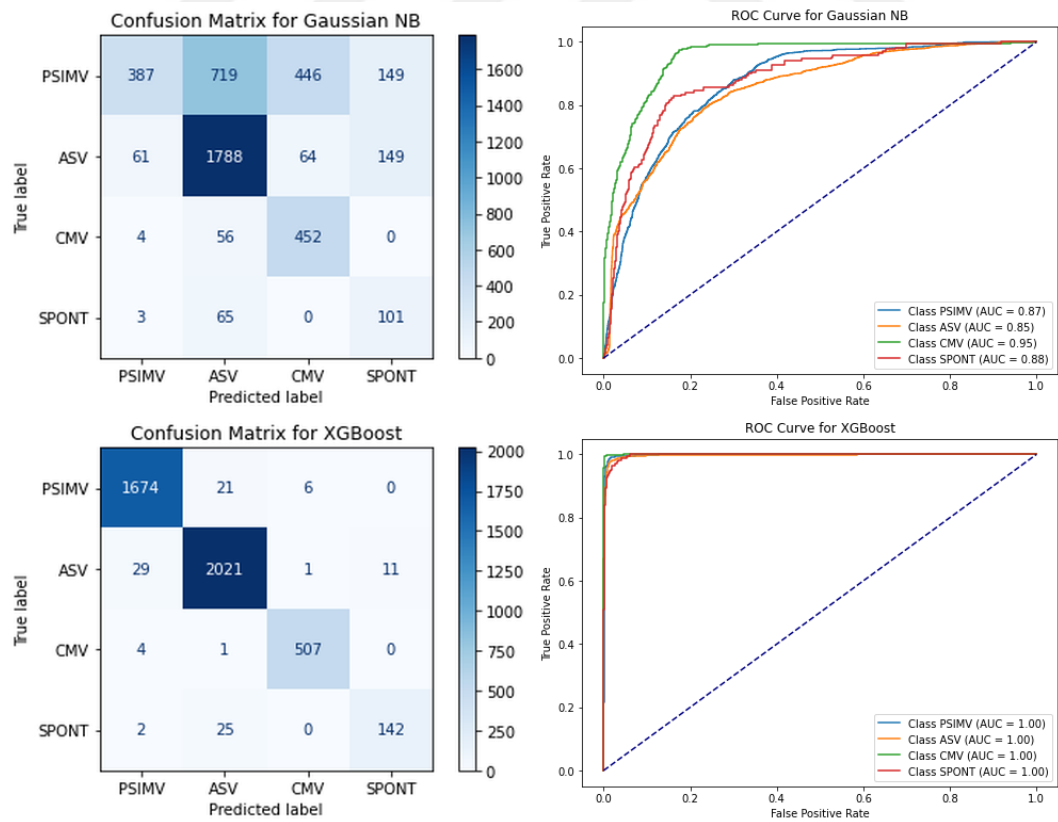


(b) ROC curve plots and confusion matrices of AdaBoost and Bagging algorithms.

Figure 5.3 ROC curve plots and confusion matrices obtained as a result of applying Random Forest, Gradient Boosting, AdaBoost and Bagging algorithms to normalized data.



(a) ROC curve plots and confusion matrices of K-Nearest Neighbors and Decision Tree algorithms.



(b) ROC curve plots and confusion matrices of Gaussian NB and XGBoost algorithms.

Figure 5.4 ROC curve plots and confusion matrices obtained as a result of applying K-Nearest Neighbors, Decision Tree, Gaussian NB and XGBoost algorithms to normalized data.

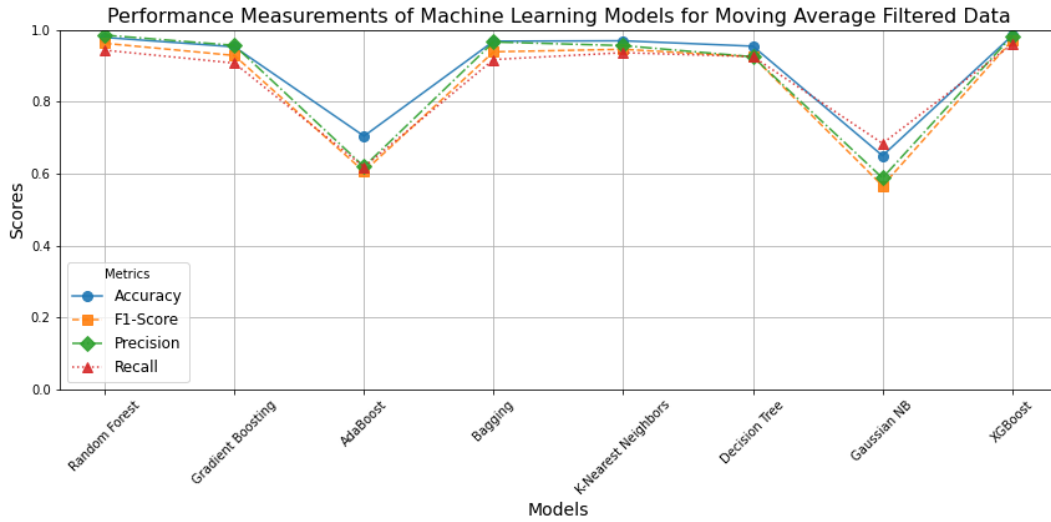


Figure 5.5 Metric values of moving average filtered data according to machine learning models.

In assessing algorithm performance under three different conditions, the XGBoost algorithm demonstrated superior efficacy in classifying ventilation modes. The graph in Figure 5.11 illustrates the accuracy comparison of each model under all situations.

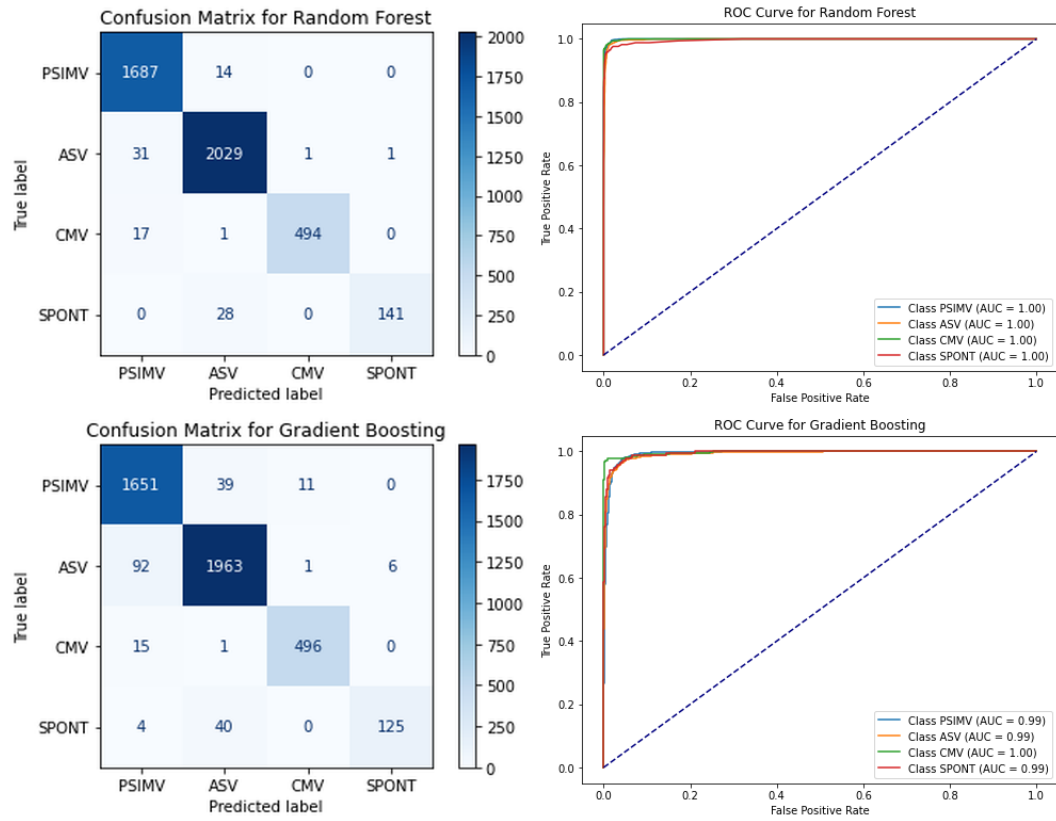
The evaluation metrics are presented in Table 5.2.

Table 5.2 Comparison of performance metrics across different data preprocessing methods.

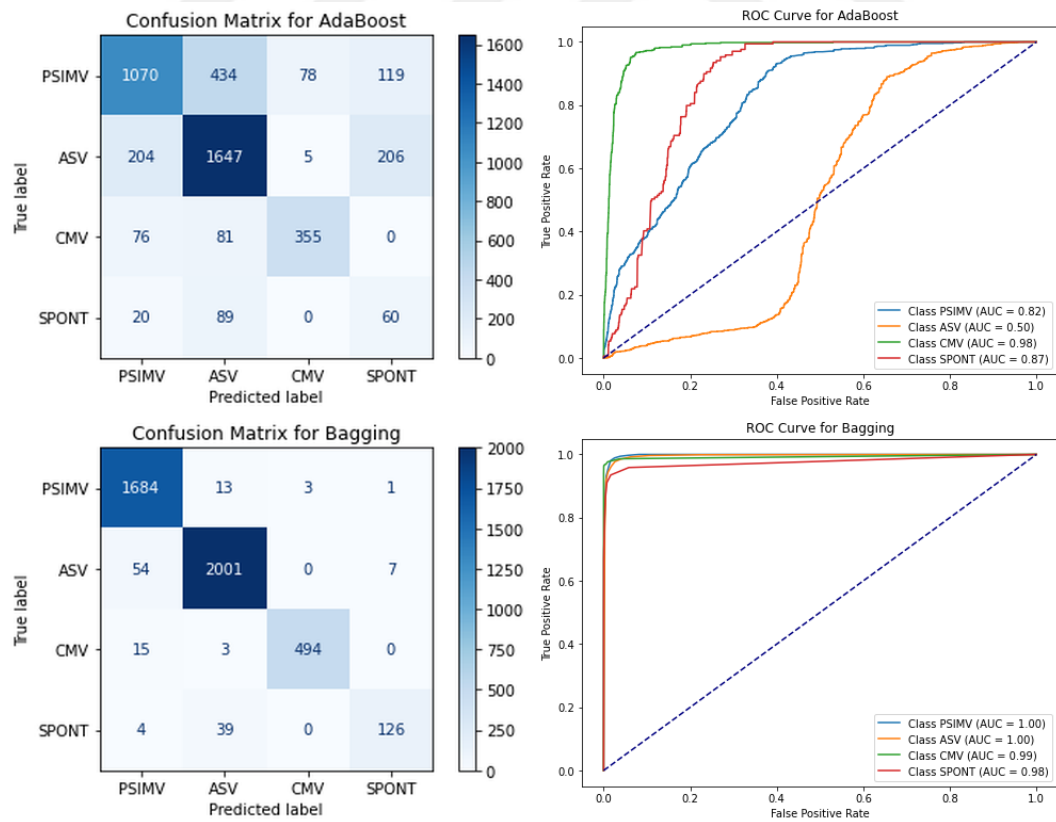
Model	Normalized Data				Moving Average Filtered Data				Gaussian Filtered Data			
	Accuracy	F1-Score	Precision	Recall	Accuracy	F1-Score	Precision	Recall	Accuracy	F1-Score	Precision	Recall
Random Forest	0.9647	0.9341	0.9674	0.9107	0.9791	0.9629	0.9856	0.9437	0.9878	0.9773	0.9913	0.9647
Gradient Boosting	0.9498	0.9214	0.9431	0.9048	0.9530	0.9289	0.9571	0.9077	0.9658	0.9398	0.9756	0.9134
AdaBoost	0.6352	0.5766	0.5914	0.6164	0.7048	0.6061	0.6198	0.6190	0.7459	0.6308	0.6241	0.6439
Bagging	0.9554	0.9227	0.9510	0.9017	0.9687	0.9392	0.9665	0.9177	0.9773	0.9591	0.9817	0.9404
K-Nearest Neighbors	0.9433	0.9120	0.9257	0.8997	0.9698	0.9461	0.9564	0.9368	0.9874	0.9780	0.9853	0.9712
Decision Tree	0.9237	0.8862	0.8986	0.8755	0.9548	0.9256	0.9253	0.9259	0.9624	0.9429	0.9484	0.9377
Gaussian NB	0.6139	0.5226	0.5635	0.6438	0.6503	0.5667	0.5884	0.6844	0.6676	0.5842	0.5959	0.6949
XGBoost	0.9775	0.9577	0.9678	0.9487	0.9842	0.9713	0.9835	0.9603	0.9887	0.9776	0.9886	0.9676

5.1.2 Classification with Deep Learning Algorithms

LSTM networks are commonly used for time series classification due to their ability to process sequential data. Their memory cells, which can retain information over long periods, enable better learning of data. Unlike traditional RNNs, LSTMs reduce the vanishing gradient problem, leading to improved training. They can be integrated with other models, such as CNNs, to capture data features more effectively. For these reasons, LSTM networks were used in ventilation mode classification. As a preliminary preparation for this study, a dataset with similar features was used to perform binary classification with an LSTM network [60]. In this study, high accuracy was achieved, demonstrating the applicability of LSTM

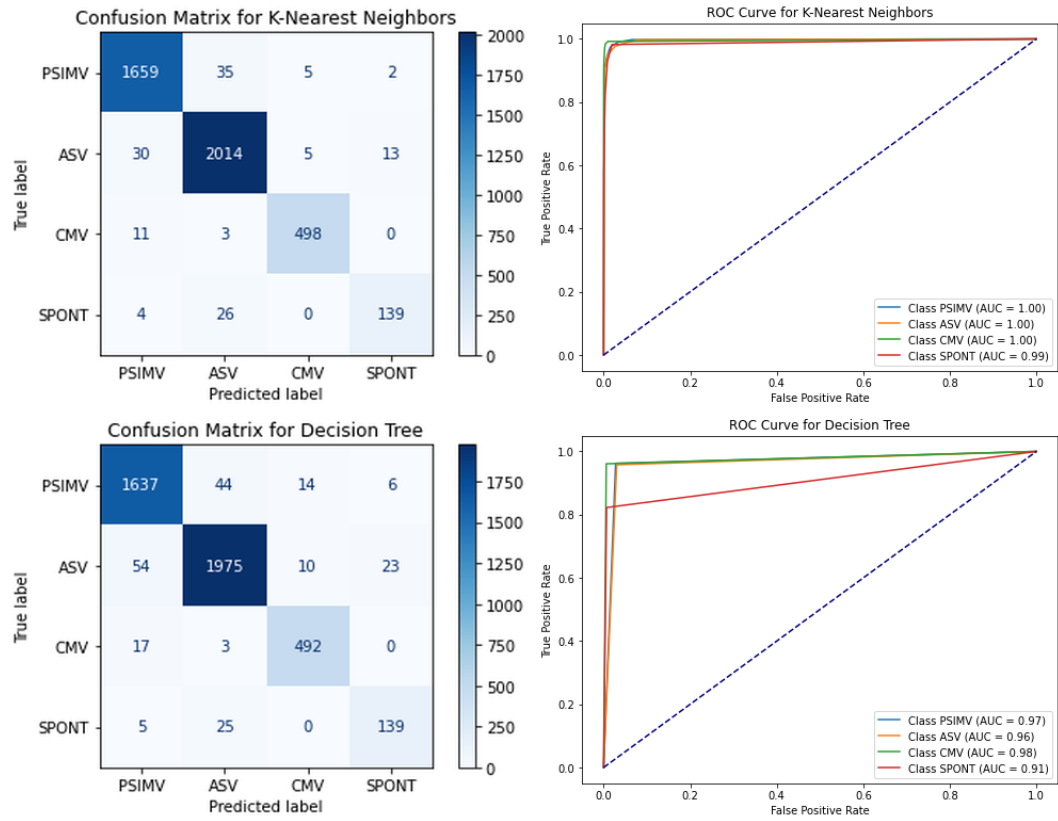


(a) ROC curve plots and confusion matrices of Random Forest and Gradient Boosting algorithms.

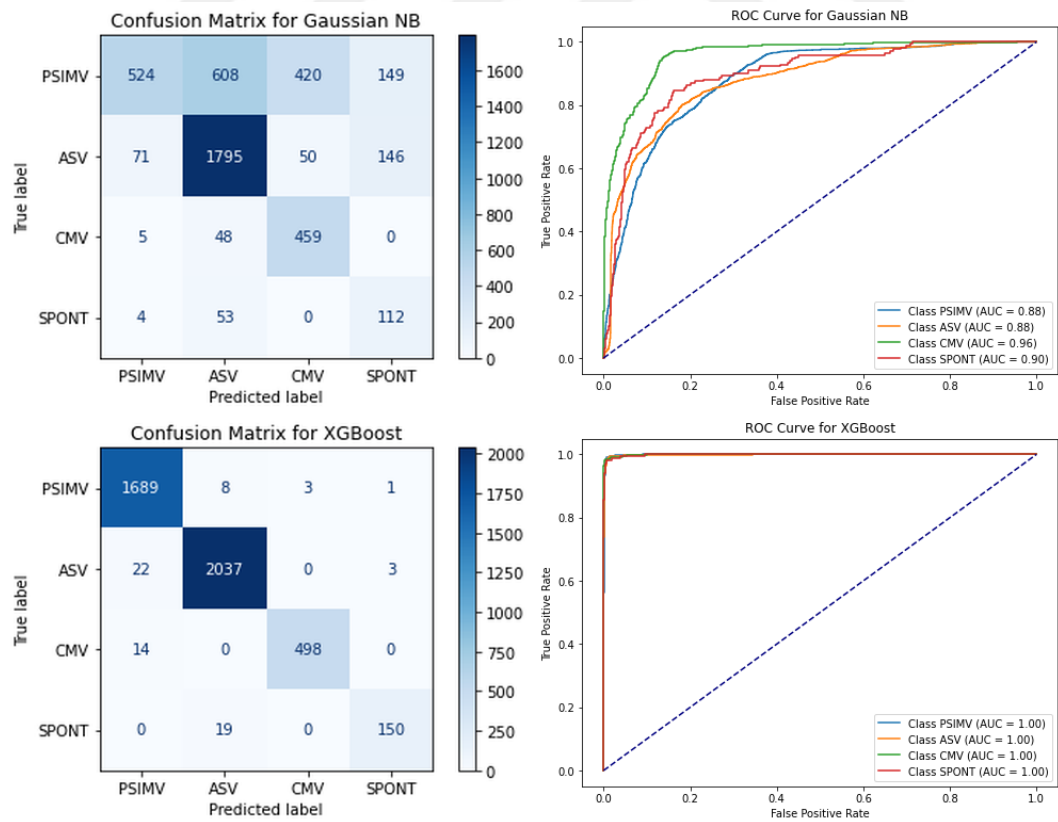


(b) ROC curve plots and confusion matrices of AdaBoost and Bagging algorithms.

Figure 5.6 ROC curve plots and confusion matrices obtained as a result of applying Random Forest, Gradient Boosting, AdaBoost and Bagging algorithms to moving average filtered data.



(a) ROC curve plots and confusion matrices of K-Nearest Neighbors and Decision Tree algorithms.



(b) ROC curve plots and confusion matrices of Gaussian NB and XGBoost algorithms.

Figure 5.7 ROC curve plots and confusion matrices obtained as a result of applying K-Nearest Neighbors, Decision Tree, Gaussian NB and XGBoost algorithms to moving average filtered data.

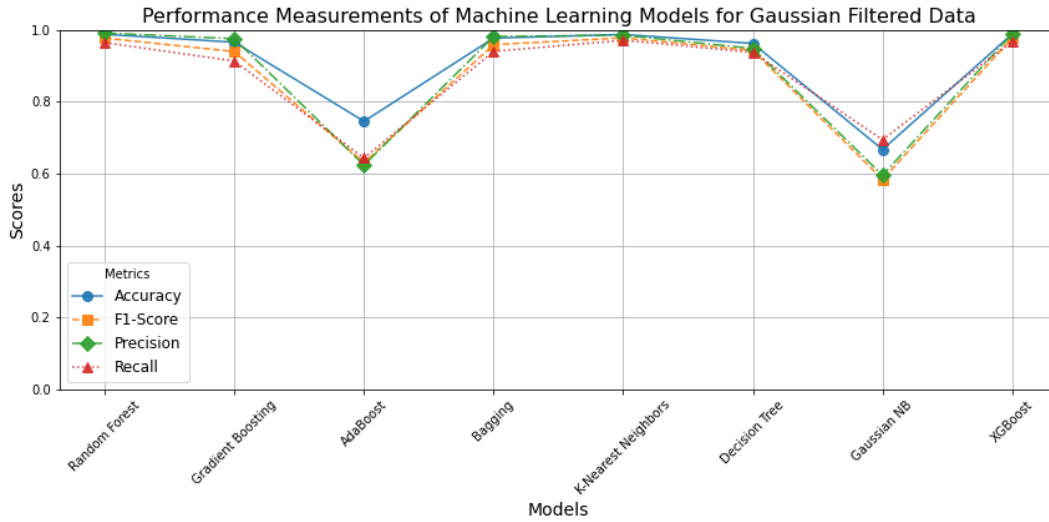


Figure 5.8 Metric values of Gaussian filtered data according to machine learning models.

for this dataset.

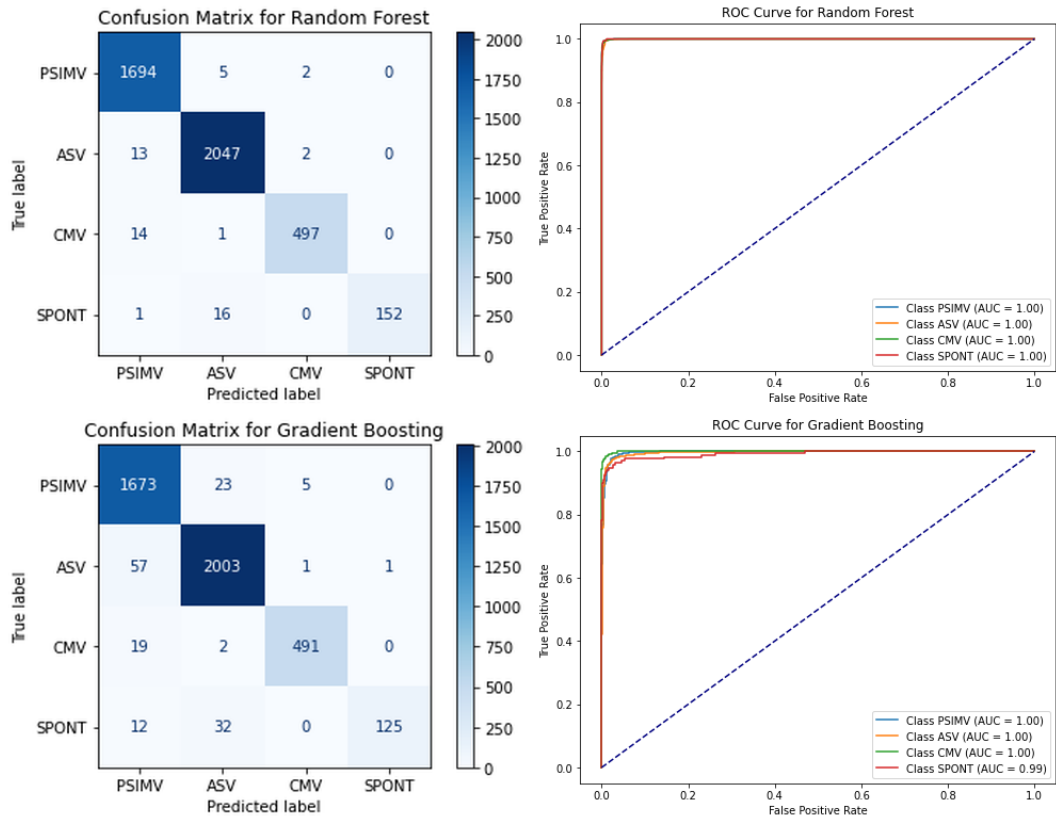
The dataset was created with the ventilation mode labeled for 23 patients. Before applying this data to the LSTM network, normalization was performed as described in Section 4.1.3.1. The normalized data was split into 30% test and 70% training sets. The LSTM network was tested across 144 different configurations, with variations in the number of layers (3, 5, 7, 8), batch sizes (16, 32, 64, 100, 128, 200), and learning rates (0.1, 0.01, 0.001) for two different loss functions: Mean absolute error and mean squared error. The hyperparameter configurations of the LSTM network are given in Table 5.3.

Table 5.3 Hyperparameter Configurations for LSTM Network.

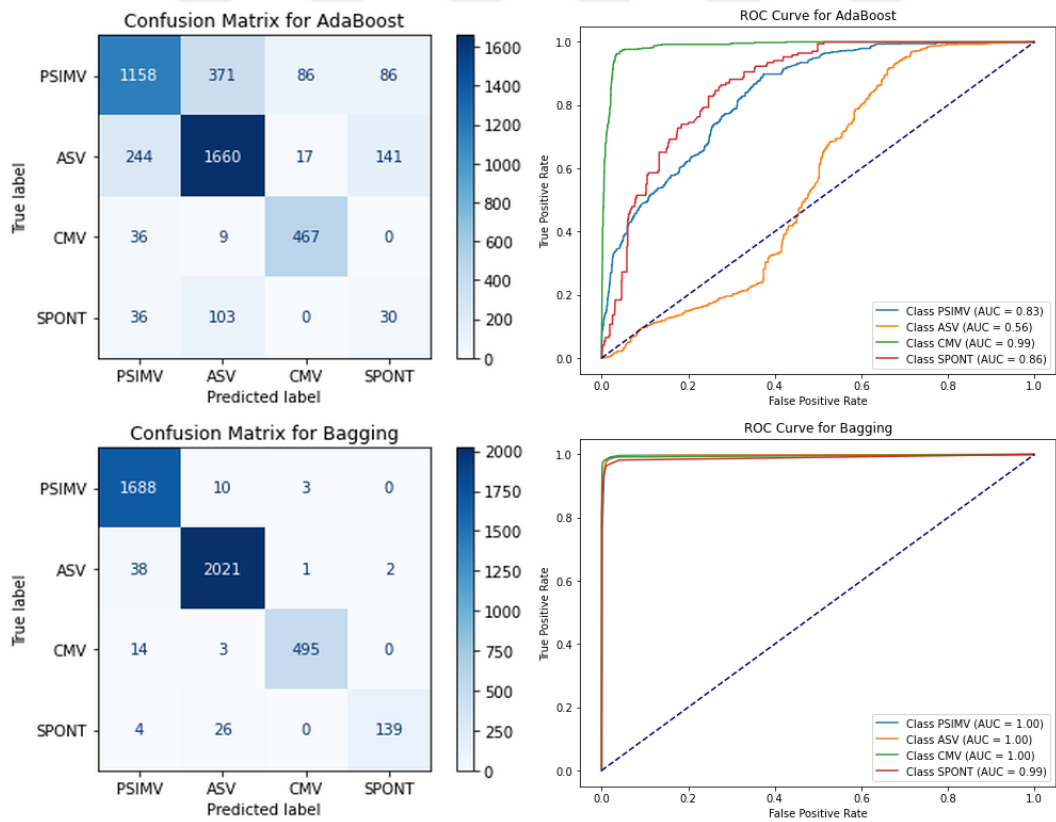
Configuration	Layers	Batch Size	Learning Rate	Loss Function
1	3	16	0.1	MAE
2	3	16	0.1	MSE
3	3	16	0.01	MAE
4	3	16	0.01	MSE
5	3	16	0.001	MAE
6	3	16	0.001	MSE
...	...	...	...	...
143	8	200	0.001	MAE
144	8	200	0.001	MSE

The bar chart showing accuracy and error values obtained when using MAE as the loss function is presented in Figure 5.12.

The evaluation metrics are presented in Table 5.4, and the confusion matrix is shown

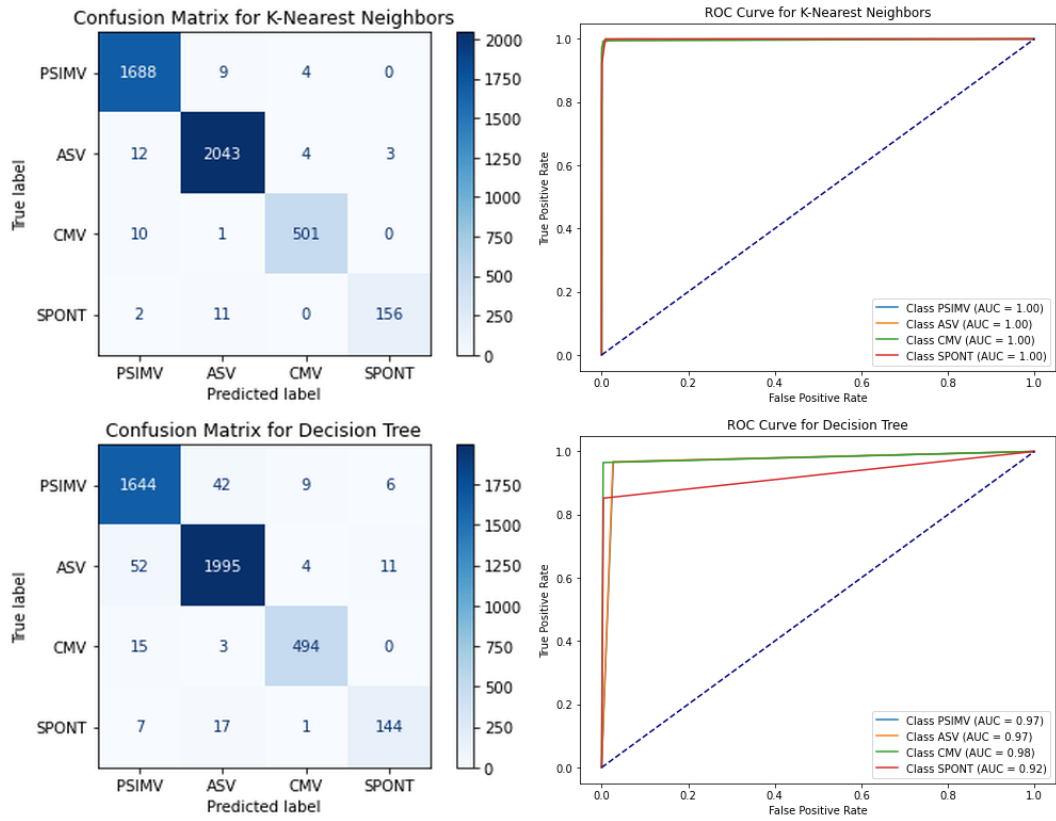


(a) ROC curve plots and confusion matrices of Random Forest and Gradient Boosting algorithms.

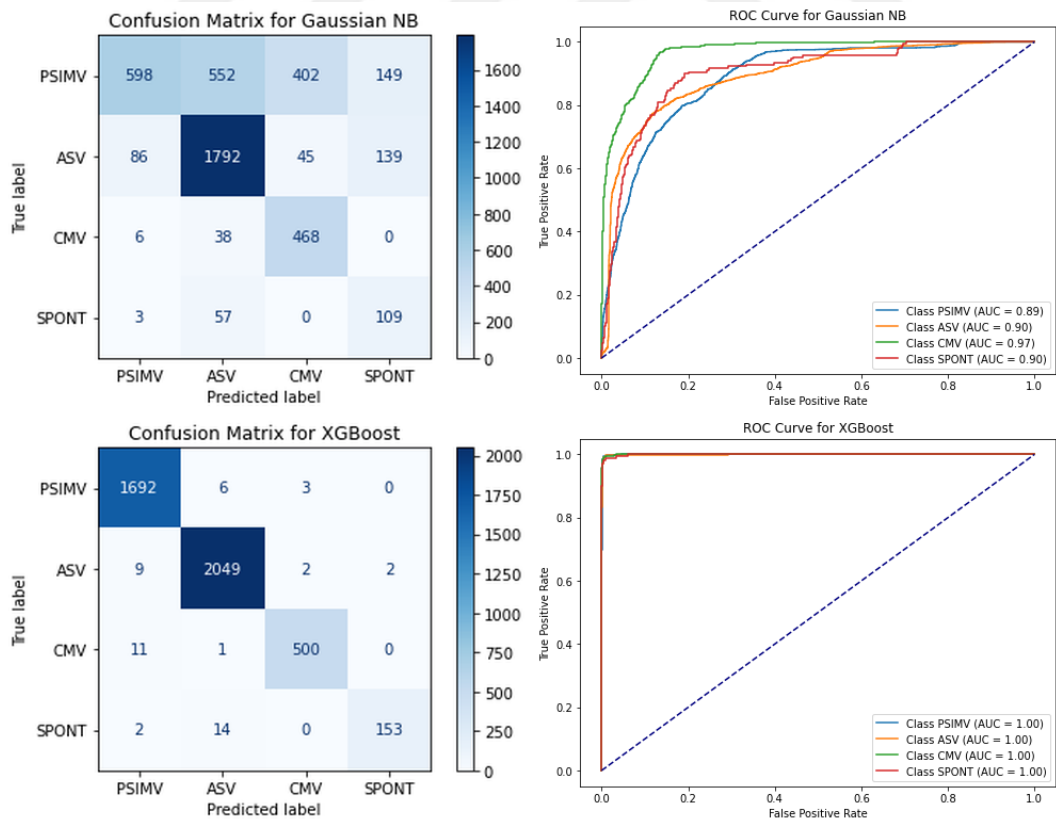


(b) ROC curve plots and confusion matrices of AdaBoost and Bagging algorithms.

Figure 5.9 ROC curve plots and confusion matrices obtained as a result of applying Random Forest, Gradient Boosting, AdaBoost and Bagging algorithms to Gaussian filtered data.



(a) ROC curve plots and confusion matrices of K-Nearest Neighbors and Decision Tree algorithms.



(b) ROC curve plots and confusion matrices of Gaussian NB and XGBoost algorithms.

Figure 5.10 ROC curve plots and confusion matrices obtained as a result of applying K-Nearest Neighbors, Decision Tree, Gaussian NB and XGBoost algorithms to Gaussian filtered data.

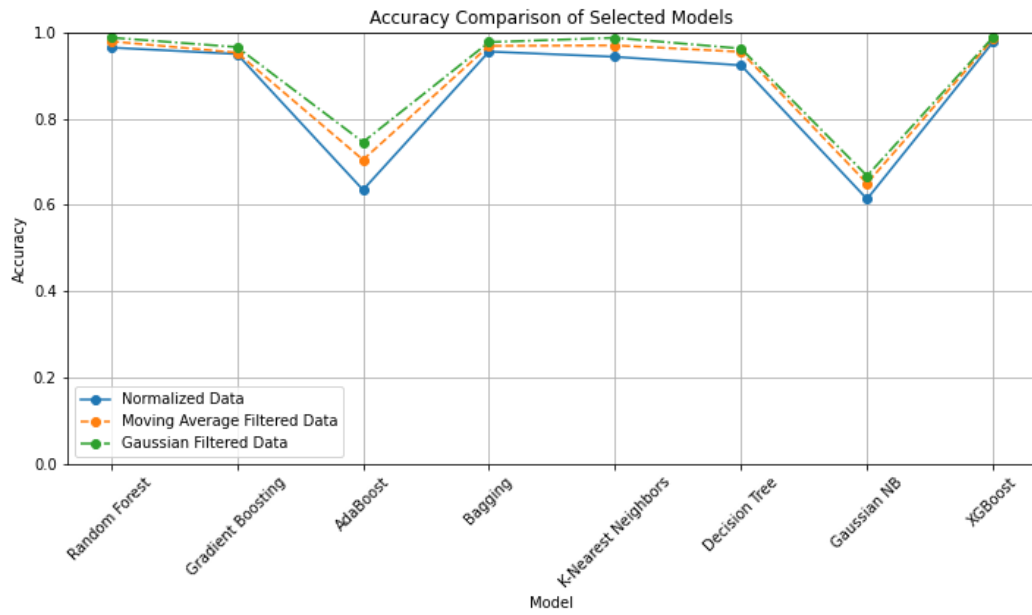


Figure 5.11 Comparative graph of machine learning models' accuracy values for three different datasets.

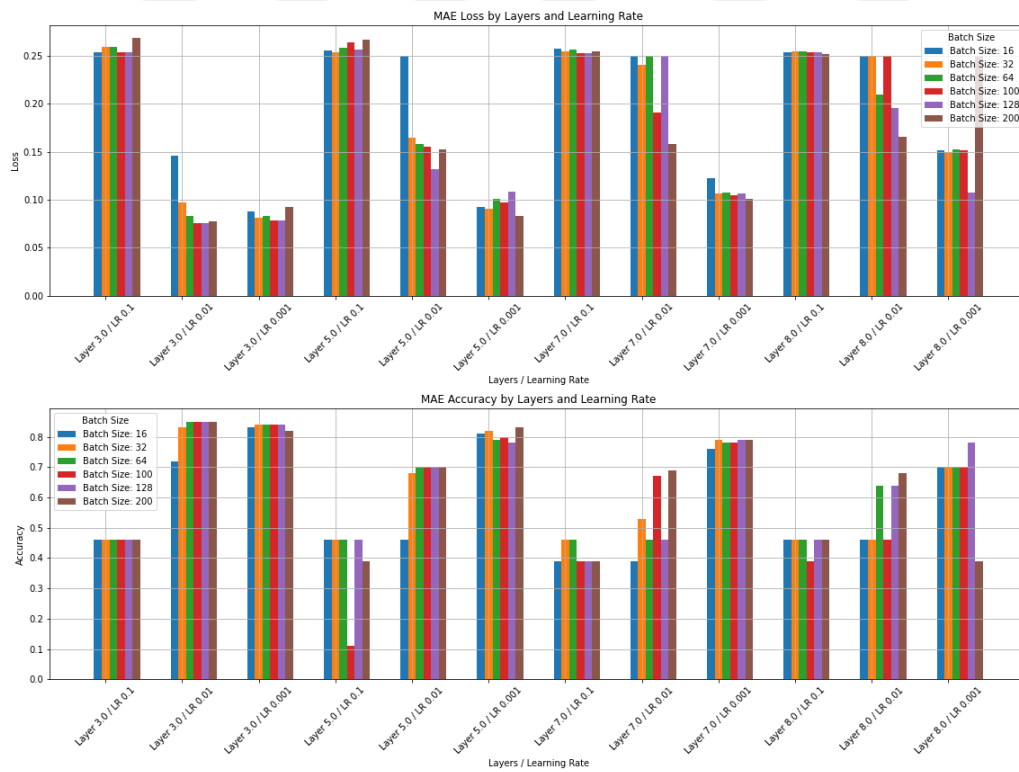


Figure 5.12 Bar graph of accuracy and error value for mean absolute error of LSTM network.

in Figure 5.13.

Table 5.4 Classification metrics for the MAE loss function of the LSTM network.

Class	Precision	Recall	F1-Score
ASV	0.86	0.88	0.87
CMV	0.93	0.93	0.93
PSIMV	0.83	0.89	0.86
SPONT	0.00	0.00	0.00
Accuracy	0.85		

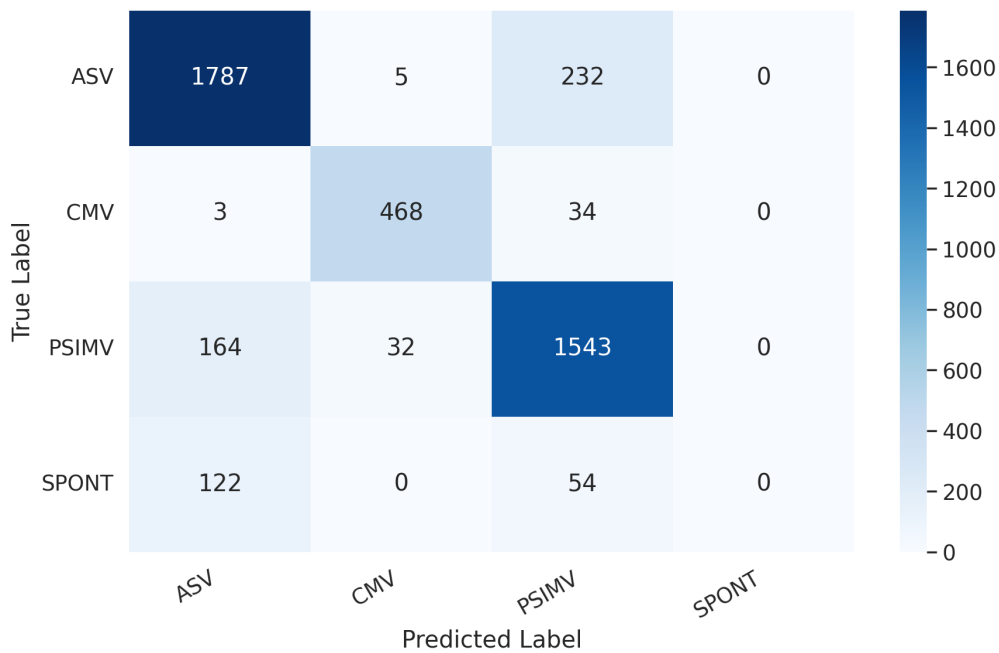


Figure 5.13 Confusion matrix for MAE loss function of LSTM network.

The configuration with the highest accuracy and lowest error is when layer $n = 3$, batch size = 128, and learning rate = 0.01. Here, the accuracy is 0.85 and the error is 0.0759. According to the confusion matrix, the model is generally successful in correctly predicting classes, but it struggles with predicting the spontaneous ventilation class. Based on the metrics, the model shows its best performance in class prediction for the CMV class.

The bar chart showing accuracy and error values obtained when using mean squared error as the loss function is presented in Figure 5.14. The evaluation metrics are shown in Table 5.5, and the confusion matrix is displayed in Figure 5.15.

The configuration with the highest accuracy and lowest error is when $n = 3$, batch size = 100, and learning rate = 0.01. Here, the accuracy is 0.91 and the error is 0.0343. According to the metrics, the CMV class achieved the best prediction performance, while the SPONT class had the worst. However, compared to the

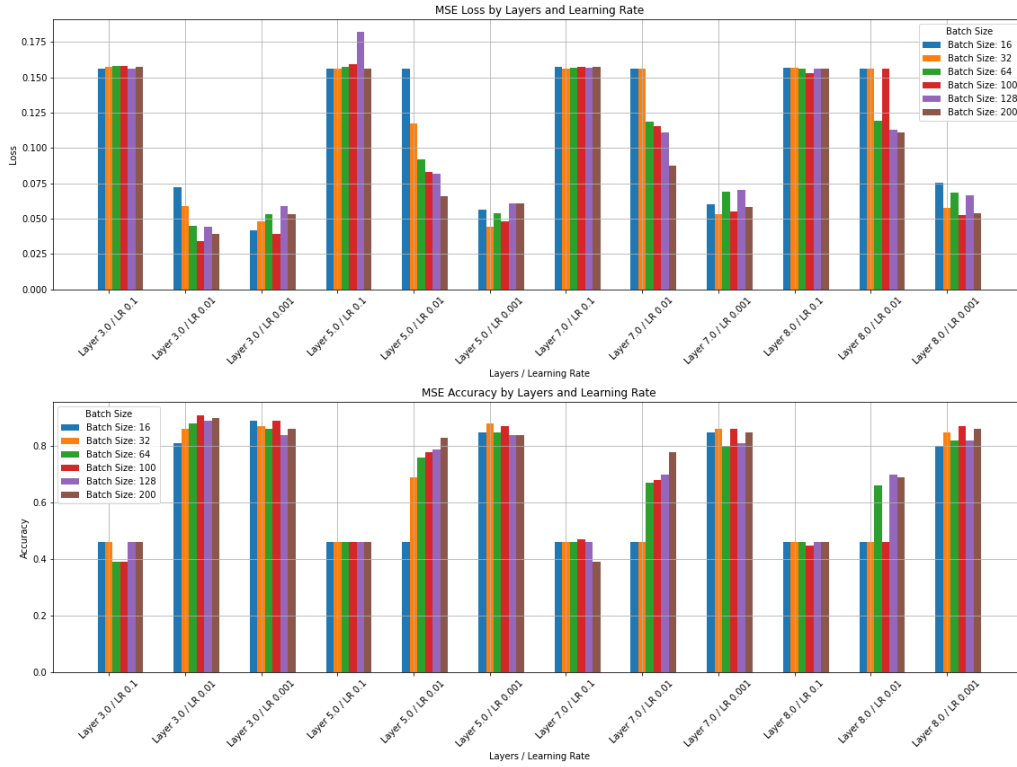


Figure 5.14 Bar graph of accuracy and error value for mean squared error of LSTM network.

Table 5.5 Classification metrics for the MSE loss function of the LSTM network.

Class	Precision	Recall	F1-Score
ASV	0.91	0.94	0.92
CMV	0.95	0.94	0.95
PSIMV	0.91	0.91	0.91
SPONT	0.81	0.58	0.68
Accuracy	0.91		

previous case, the prediction performance for the SPONT class improved. This configuration yielded the highest accuracy among the 144 tested configurations for the LSTM network, indicating that the LSTM network performed optimally under these conditions.

Considering all configurations, the best classification performance for this dataset was achieved with $n = 3$, batch size = 100, learning rate = 0.01, and mean squared error as the loss function, resulting in an accuracy of 91%.

Feature extraction was implemented on the normalized data to increase the accuracy rate. Feature extraction offers advantages including the reduction of data dimensionality, the minimization of noise, the simplification of complex data, and the capture of differences across diverse scales. For inherently non-linear

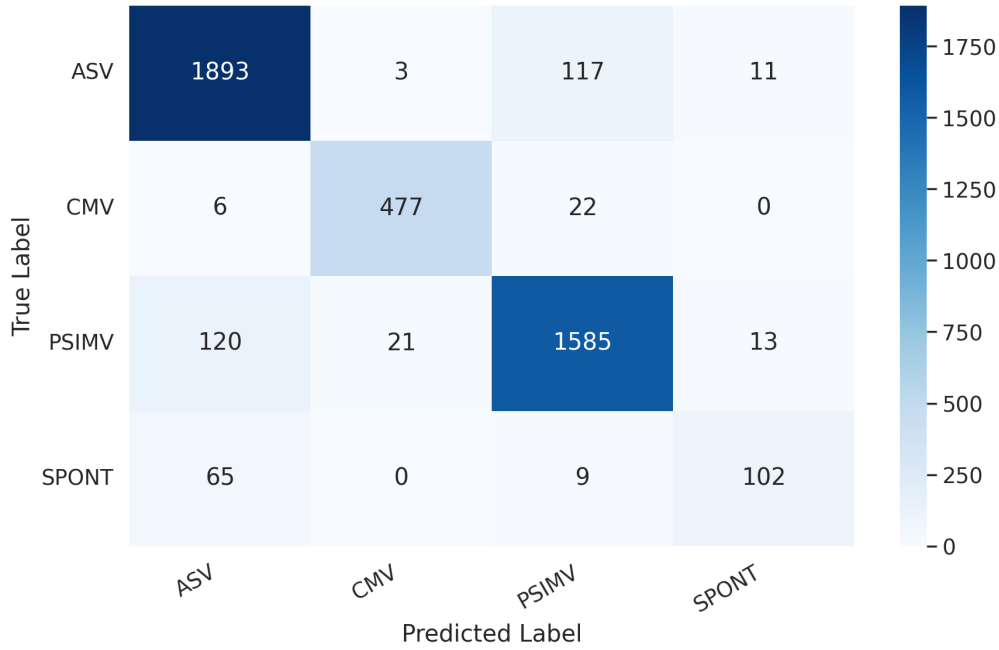


Figure 5.15 Confusion matrix for MSE loss function of LSTM network.

and complex data, such as healthcare data, feature extraction is highly effective. Autoregression, moving average, and convolutional neural networks are often employed techniques for feature extraction. This study employed a CNN network to enhance the accuracy of LSTM. CNN collects spatial information from the time series data, whereas LSTM learns temporal connections. Through feature extraction, CNN reduces network complexity and facilitates the identification of essential features [61]. The CNN-LSTM architecture used is shown in Figure 5.16.

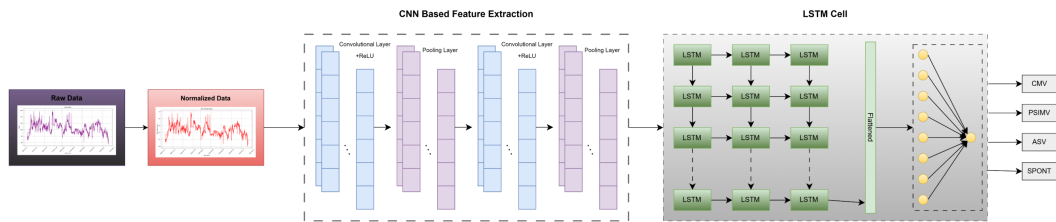


Figure 5.16 CNN-LSTM model used.

In this model, the data was first normalized and split into 30% test and 70% training sets. The training data was initially fed into the CNN block for feature extraction. The CNN block consists of 2 convolution + ReLU layers and 2 pooling layers. After feature extraction, the data was passed to a 3-layer LSTM network. During training, the mean squared error was used as the loss function, with a batch size of 100 and a learning rate of 0.01. These values were chosen because they provided the best performance in the LSTM model classification. The hyperparameters of the CNN-LSTM model are given in Table 5.6.

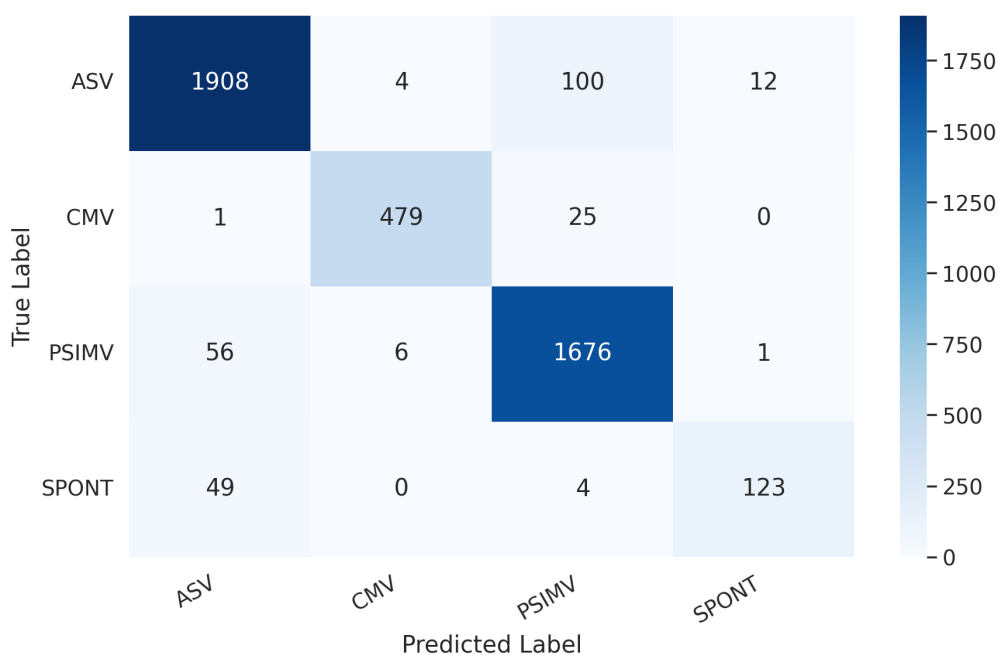
Table 5.6 CNN-LSTM Model Hyperparameters.

Hyperparameter	Value
Layers	3
Batch Size	100
Learning Rate	0.01
Loss Function	Mean Squared Error
Optimizer	Adam
Epochs	1000

The model achieved 94% accuracy, showing better performance than the traditional LSTM network. The CNN-LSTM approach was observed to capture complex patterns and significant features from the respiratory data, resulting in improved classification accuracy. The evaluation metrics obtained from training the CNN-LSTM model are presented in Table 5.7, and the confusion matrix is shown in Figure 5.17.

Table 5.7 Classification metrics for CNN-LSTM model.

Class	Precision	Recall	F1-Score
ASV	0.95	0.94	0.95
CMV	0.98	0.95	0.96
PSIMV	0.93	0.96	0.95
SPONT	0.90	0.70	0.79
Accuracy	0.94		

**Figure 5.17** Confusion matrix of the CNN-LSTM model.

According to the evaluation metrics, the CNN-LSTM model improved the

prediction accuracy for each class compared to the LSTM model. These results indicate that the deep learning-based CNN-LSTM model performs well in ventilation mode classification.

5.1.3 Ventilation Mode Classification Results

In this study, the performance of various machine learning and deep learning algorithms on ventilation mode classification was examined in detail using different data preprocessing techniques. The dataset was divided into three groups: normalized data, data with a moving average filter applied, and data with a Gaussian filter applied. These three data groups were evaluated for performance across each machine learning and deep learning model. The obtained results were analyzed and compared based on accuracy, F1-score, precision, recall, and ROC curve (AUC) metrics.

The analysis of the normalized dataset revealed that the XGBoost algorithm had the highest accuracy rate at 97.75%. The high accuracy of the XGBoost algorithm indicates that this model is highly successful in classification tasks on normalized data. XGBoost achieved an F1 score of 95.77% and a precision of 96.78%, outperforming other algorithms in the classification of ventilation modes. This achievement underscores XGBoost's robust learning capacity, providing an efficient resolution for intricate classification challenges.

The Random Forest algorithm achieved the second-highest performance following XGBoost, with an accuracy rate of 96.47%. It effectively differentiated across categories, obtaining an F1 score of 93.41% and an accuracy of 96.74%. These results indicate that while the Random Forest algorithm performs well with normalized data, it does not reach the same high levels of accuracy and F1 score as XGBoost.

Other algorithms, such as Gradient Boosting and Bagging, also demonstrated effective performance. However, the accuracy rates of AdaBoost and Gaussian Naive Bayes were relatively low. Specifically, AdaBoost showed poor performance on the normalized dataset, with an accuracy rate of 63.52%. This suggests that AdaBoost may lack sufficient classification power for complex time series data.

In the analysis conducted on the dataset with a moving average filter applied, the XGBoost algorithm once again stood out, achieving the highest accuracy rate (98.42%). This success indicates that the moving average filter reduces data noise, enhancing XGBoost's classification performance. With an F1 score of 97.13% and a precision of 98.35%, XGBoost provided more stable and reliable results on filtered

data.

The Random Forest algorithm also achieved a 97.91% accuracy rate with the moving average filter. With an F1 score of 96.29% and a precision of 98.56%, Random Forest demonstrated high performance. This indicates that Random Forest enhances its classification capabilities, particularly when data noise is reduced, making it a competitive model compared to XGBoost.

The Gradient Boosting and Bagging algorithms also showed improved performance with the moving average filter. However, the accuracy rates for AdaBoost and Gaussian Naive Bayes remained low, indicating that even with data filtering, the classification capabilities of AdaBoost and Gaussian Naive Bayes are not as effective as those of XGBoost and Random Forest.

In the analysis conducted on the dataset with a Gaussian filter applied, the XGBoost algorithm once again showed the best performance, with an accuracy rate of 98.87%. With an F1 score of 97.76% and a precision of 98.86%, XGBoost's classification performance improved further due to the additional noise reduction provided by the Gaussian filter. These results indicate that XGBoost offers the most reliable and accurate outcomes in ventilation mode classification.

The Random Forest algorithm ranked second on Gaussian-filtered data with an accuracy rate of 98.78%. With an F1 score of 97.73% and a precision of 99.13%, Random Forest demonstrated the best performance after XGBoost. The Gaussian filter further enhanced Random Forest's classification power, enabling the model to perform effectively on complex time series data.

Among the other algorithms, K-Nearest Neighbors showed high performance with an accuracy rate of 98.74%, following XGBoost and Random Forest. KNN performed particularly well with Gaussian-filtered data. However, AdaBoost and Gaussian Naive Bayes once again stood out with low accuracy rates.

Considering the deep learning models and all data preprocessing methods, the XGBoost algorithm provided the highest classification accuracy across all three cases. XGBoost attained an accuracy rate of 98% or better in classifying ventilation modes due to its proficiency in capturing patterns and features in intricate time series data. This result demonstrates that the XGBoost algorithm is a stronger and more consistent model compared to other machine learning and deep learning algorithms for time series analysis, proving it to be the ideal algorithm for complex tasks such as ventilation mode classification.

5.2 Forecasting of Time Series Respiratory Data

In this thesis, in addition to ventilation mode classification, the goal was to forecast future values of respiratory data collected as time series from patients. Accurately forecasting respiratory data is crucial, especially for managing the treatment processes of patients in intensive care units and for early detection of potential respiratory issues. Being able to anticipate changes in respiratory dynamics over time provides healthcare professionals with more information about patient status, enabling optimization of intervention processes. Therefore, the time series forecasting methods used in this study contribute significantly to the analysis of respiratory data and the prediction of future developments.

Deep learning models, including LSTM, TCN, CNN-LSTM, and TCN-LSTM, were employed for data forecasting. A multi-model approach was preferred in this study to leverage each model's unique advantages in time series data, enabling more effective and comprehensive data analysis. Predictions were made on 8 different features for 6 individuals, including Pulse, RESP, Oxygen, PEEP, PPEAK, PMEAN, SPO2, and Expiration. In time series forecasting, the model requires historical data values to accurately predict future values, based on the assumption that past information has a direct impact on future values in time series data.

In this study, different scenarios were compared with the LSTM network to determine the number of time steps to be used as input for the model and to predict the value at the next time step based on this data. First, the Pulse feature data of an individual was trained with the LSTM network for six data groups: raw data, normalized data, and both of these with a moving average and Gaussian filter applied, for 100, 200, 300, and 400 time steps. The training results were evaluated using regression metrics: mean squared error, root mean squared error, mean absolute error, and the coefficient of determination (R^2). The configuration with the lowest error rate was achieved with 200 time steps using normalized data with a Gaussian filter applied. The hyperparameters used for each model are provided in Table 5.8.

Table 5.8 Hyperparameters of deep learning models.

Hyperparameter	Value
Loss Function	Mean Squared Error (MSE)
Optimizer	Adam
Time Step	200
Epochs	1000
Batch Size	32

5.2.1 Forecasting with LSTM Model

LSTM is often used for time series data because of its capacity to learn long-term dependencies in sequential information. LSTM effectively reduces the vanishing gradient problem, allowing for the capture of long-term relationships and enhancing the analysis of patterns in patients' respiratory data.

The dataset was normalized, and a Gaussian filter was applied to the normalized data. The prepared data was then applied to the 2-layer LSTM architecture shown in Figure 5.18.

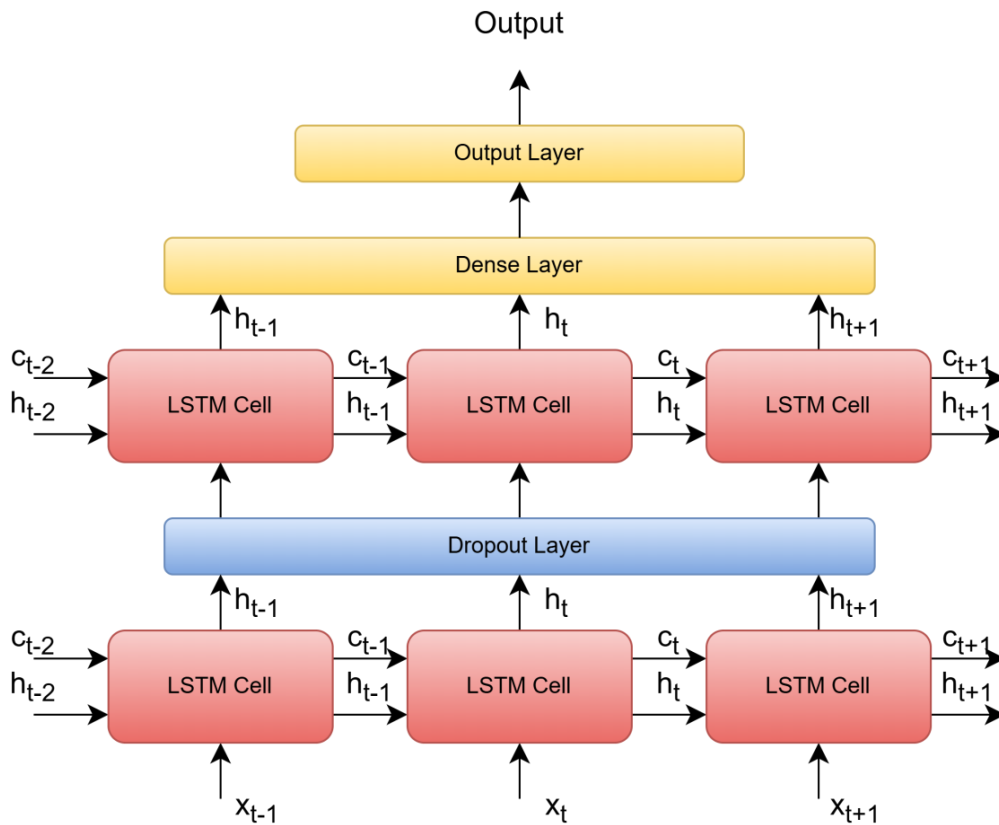


Figure 5.18 LSTM architecture implemented for forecasting.

Each feature for each individual was forecasted with LSTM. Figure 5.19 presents the stacked bar chart showing the mean absolute error values derived from the forecasts.

The PEEP and SPO2 features had the lowest MAE values in the LSTM prediction results, indicating that LSTM performed well in predicting these two features. The graph showing the LSTM prediction of the next 700 time steps for PEEP and SPO2, trained with the first 200 time steps, is presented in Figure 5.20.

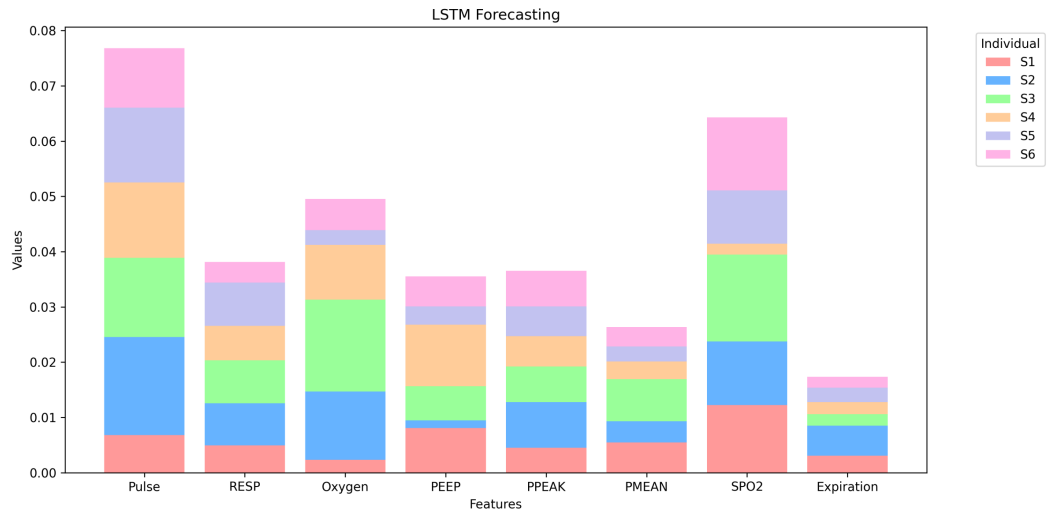
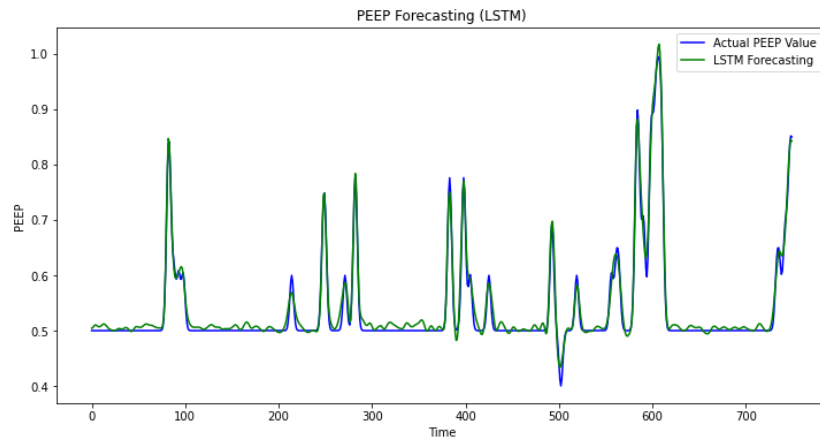
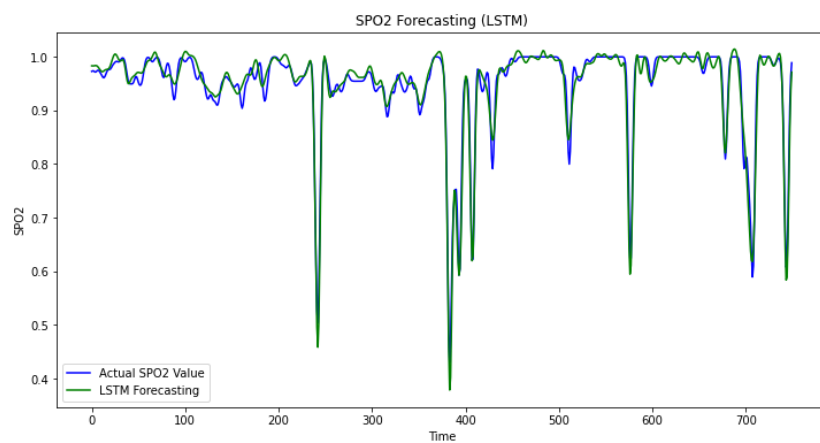


Figure 5.19 Stacked bar chart of the mean absolute error values of the LSTM prediction.



(a) The actual value of the PEEP feature and the LSTM predicted value.



(b) The actual value of the SPO2 feature and the LSTM predicted value.

Figure 5.20 Plots of PEEP and SPO2 features' actual value and LSTM predicted values.

5.2.2 Forecasting with TCN Model

Temporal convolutional network is a model that uses convolutional filters in time series data analysis to reduce data loss while predicting future values and offers faster computation. TCN is particularly strong at capturing short-term relationships, and, due to its more parallel structure compared to LSTM, it provides advantages when working with large datasets.

The dataset was normalized, and a Gaussian filter was applied to the normalized data. The prepared data was then applied to the TCN architecture. Each feature for each individual was predicted using TCN. The stacked bar chart showing the Mean Absolute Error (MAE) values obtained from the predictions is presented in Figure 5.21.

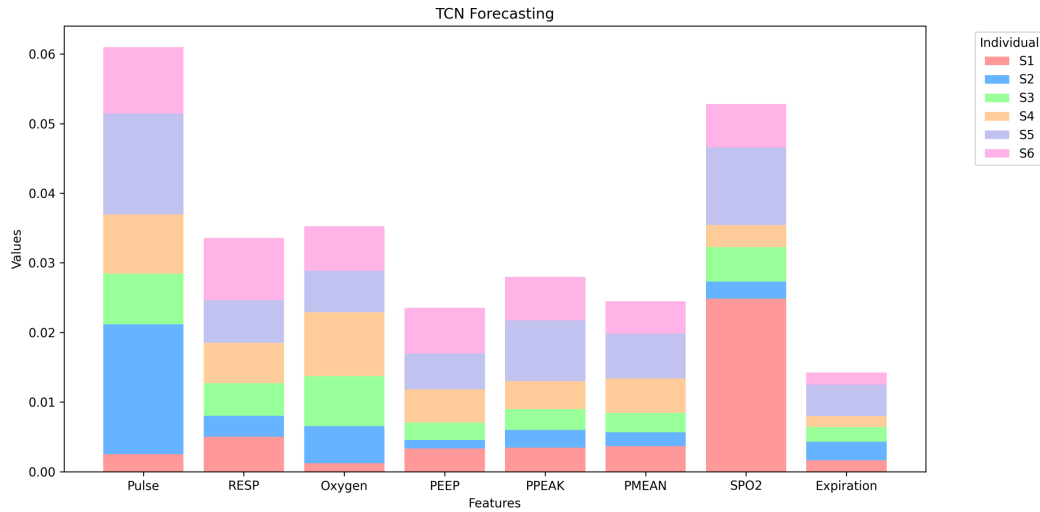


Figure 5.21 Stacked bar chart of the mean absolute error values of the TCN prediction.

The Oxygen feature had the lowest MAE value in the TCN prediction results, indicating strong prediction performance for this feature. The graph showing the TCN prediction of the next 760 time steps for Oxygen, trained with the first 200 time steps, is presented in Figure 5.22.

5.2.3 Forecasting with CNN-LSTM Model

In the CNN-LSTM model, CNN layers extract data features, while LSTM layers learn the temporal dependencies of these features. The CNN layer extracts spatial features from inputs such as pulse and oxygen. These features are then passed to the LSTM layer, which processes them over time, learning long-term dependencies based on past states [61, 62].

The training data is initially processed through the CNN block for feature extraction,

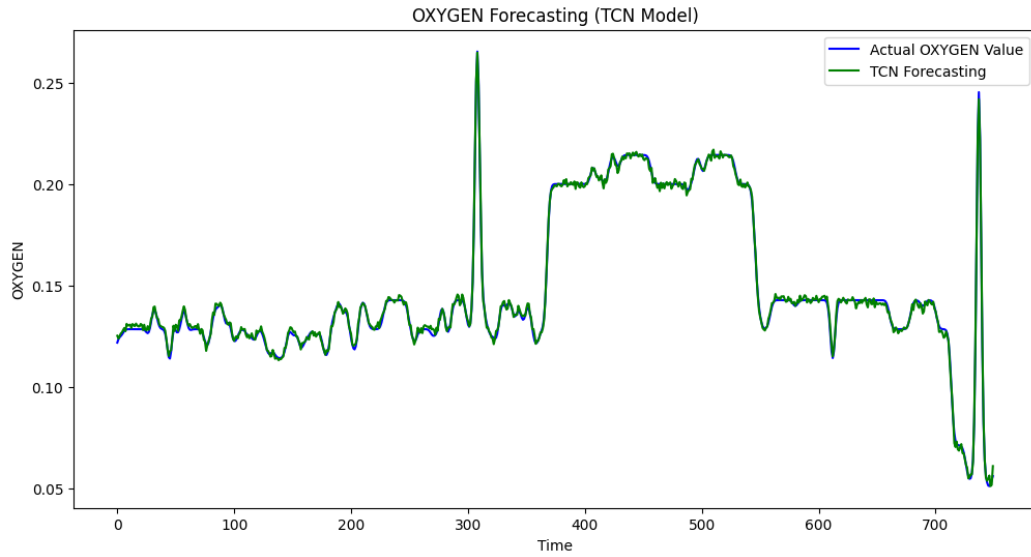


Figure 5.22 The actual value of the Oxygen feature and the TCN predicted value.

comprising 2 convolutional layers and 2 pooling layers. After to feature extraction, the data is transmitted to a two-layer LSTM network. The CNN-LSTM model was employed to predict each feature for every individual. Figure 5.23 displays the stacked bar chart showing the Mean Absolute Error (MAE) values obtained from the predictions.

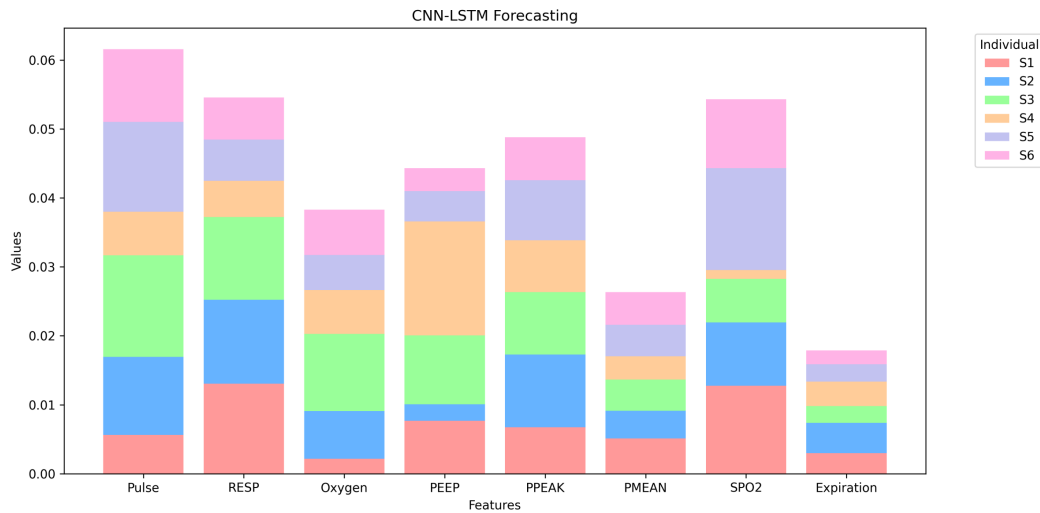


Figure 5.23 Stacked bar chart of the mean absolute error values of the CNN-LSTM prediction.

The Oxygen feature had the lowest MAE value in the CNN-LSTM prediction results, indicating strong predictive performance for this feature. The graph showing the CNN-LSTM prediction of the next 600 time steps for Oxygen, trained with the first 200 time steps, is presented in Figure 5.24.

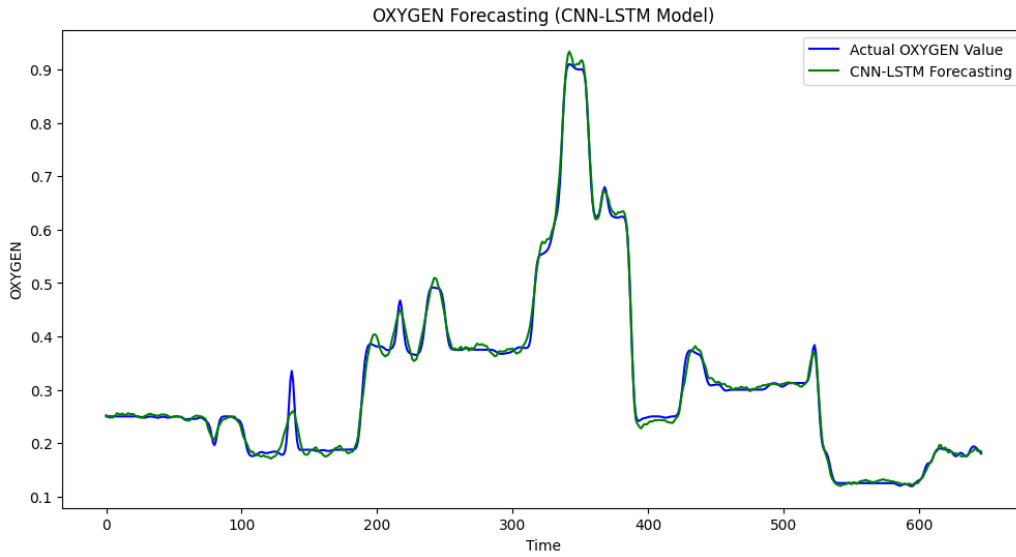


Figure 5.24 The actual value of the Oxygen feature and the CNN-LSTM predicted value.

5.2.4 Forecasting with TCN-LSTM Model

The TCN-LSTM model combines the convolutional structure of TCN, which captures short-term features, with the ability of LSTM to learn long-term dependencies. This structure enables both the detailed extraction of data features and the effective analysis of changing temporal relationships over time [63].

The dataset was normalized, and a Gaussian filter was applied to the normalized data. The prepared data was then applied to the TCN block followed by a 2-layer LSTM architecture, as shown in the diagram in Figure 5.25.

Each feature for each individual was predicted using the TCN-LSTM model. The stacked bar chart showing the Mean Absolute Error values obtained from the predictions is presented in Figure 5.26.

The Expiration feature had the lowest MAE value in the TCN-LSTM prediction results, indicating strong predictive performance for this feature. The graph showing the TCN-LSTM prediction of the next 800 time steps for Expiration, trained with the first 200 time steps, is presented in Figure 5.27.

5.2.5 Forecasting Results of Time Series Respiratory Data

The time series prediction of respiratory data for patients was conducted using four different deep learning models on eight feature datasets for six individuals. A comparison based on the mean absolute error values is presented in Table 5.9.

Table 5.9 Respiratory data forecasting model performance values (MAE).

Subject	Feature	LSTM	TCN	CNN-LSTM	TCN-LSTM	Best Model
S1	Pulse	0.006808	0.002515	0.005608	0.004418	TCN
	RESP	0.004957	0.005034	0.013054	0.008111	LSTM
	Oxygen	0.002322	0.001212	0.002151	0.000931	TCN-LSTM
	PEEP	0.008085	0.003321	0.007662	0.004082	TCN
	PPEAK	0.004536	0.003447	0.006742	0.003439	TCN
	PMEAN	0.005484	0.003674	0.005127	0.002793	TCN-LSTM
	SPO2	0.012266	0.024864	0.012775	0.010073	TCN-LSTM
	Expiration	0.003096	0.001624	0.002962	0.001147	TCN-LSTM
S2	Pulse	0.017725	0.018661	0.011348	0.006160	TCN-LSTM
	RESP	0.007592	0.002976	0.012165	0.005082	TCN
	Oxygen	0.012376	0.005344	0.006922	0.008058	TCN
	PEEP	0.001396	0.001260	0.002415	0.000974	TCN-LSTM
	PPEAK	0.008240	0.002545	0.010546	0.004882	TCN
	PMEAN	0.003807	0.001999	0.004004	0.003504	TCN
	SPO2	0.011480	0.002457	0.009168	0.001709	TCN-LSTM
	Expiration	0.005431	0.002693	0.004416	0.002307	TCN-LSTM
S3	Pulse	0.014360	0.007258	0.014708	0.011824	TCN
	RESP	0.007792	0.004713	0.011994	0.005113	TCN-LSTM
	Oxygen	0.016648	0.007191	0.011209	0.004718	TCN
	PEEP	0.006163	0.002469	0.009980	0.004143	TCN
	PPEAK	0.006444	0.003006	0.009062	0.003426	TCN
	PMEAN	0.007621	0.002789	0.004518	0.003812	TCN
	SPO2	0.015715	0.004955	0.006305	0.001213	TCN-LSTM
	Expiration	0.002050	0.002087	0.002443	0.001220	TCN-LSTM
S4	Pulse	0.013608	0.008537	0.006330	0.004516	TCN-LSTM
	RESP	0.006256	0.005786	0.005255	0.002901	TCN-LSTM
	Oxygen	0.009865	0.009167	0.006355	0.001896	TCN-LSTM
	PEEP	0.011146	0.004795	0.016533	0.003980	TCN-LSTM
	PPEAK	0.005500	0.004014	0.007497	0.002571	TCN-LSTM
	PMEAN	0.003243	0.004941	0.003368	0.001671	TCN-LSTM
	SPO2	0.001963	0.003171	0.001310	0.002417	CNN-LSTM
	Expiration	0.002180	0.001585	0.003529	0.000618	TCN-LSTM
S5	Pulse	0.013551	0.014471	0.013021	0.010826	TCN-LSTM
	RESP	0.007801	0.006126	0.005994	0.006592	TCN
	Oxygen	0.002692	0.005947	0.005077	0.002747	LSTM
	PEEP	0.003288	0.005154	0.004373	0.003677	LSTM
	PPEAK	0.005363	0.008727	0.008720	0.005537	LSTM
	PMEAN	0.002668	0.006421	0.004562	0.001814	TCN-LSTM
	SPO2	0.009682	0.011137	0.014733	0.007594	TCN-LSTM
	Expiration	0.002626	0.004510	0.002534	0.001348	TCN-LSTM
S6	Pulse	0.010782	0.009545	0.010556	0.005330	TCN-LSTM
	RESP	0.003729	0.008940	0.006081	0.002515	TCN-LSTM
	Oxygen	0.005652	0.006402	0.006583	0.004097	TCN-LSTM
	PEEP	0.005460	0.006568	0.003370	0.004515	CNN-LSTM
	PPEAK	0.006445	0.006260	0.006219	0.007775	TCN
	PMEAN	0.003561	0.004646	0.004749	0.002665	TCN-LSTM
	SPO2	0.013181	0.006222	0.010005	0.005294	TCN-LSTM
	Expiration	0.001981	0.001760	0.002003	0.001469	TCN-LSTM

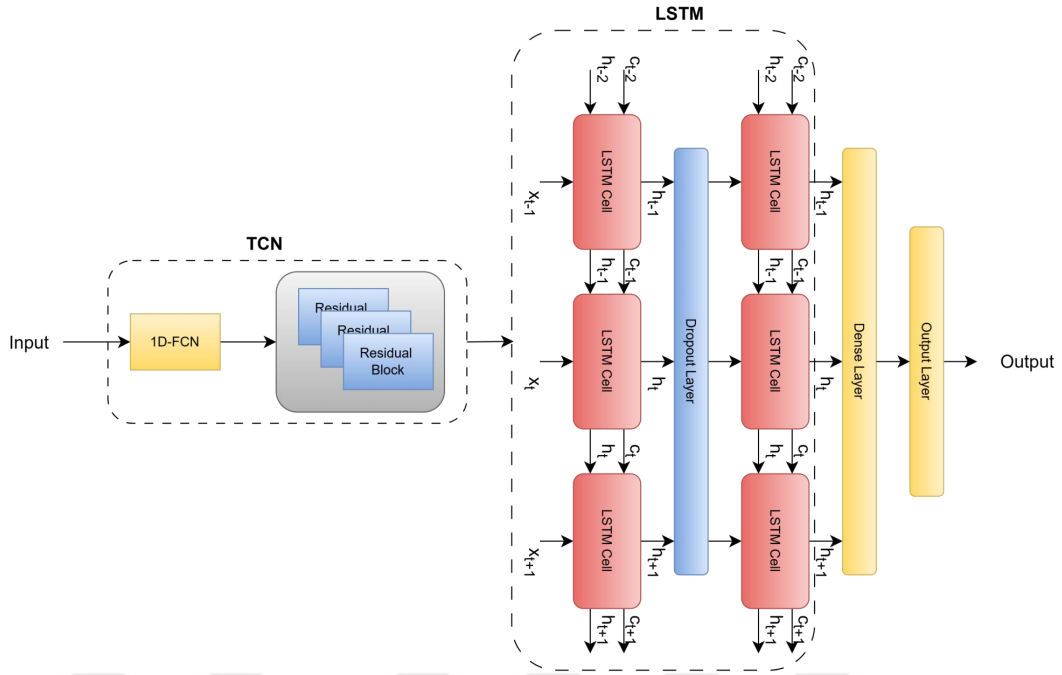


Figure 5.25 TCN-LSTM architecture implemented for forecasting.

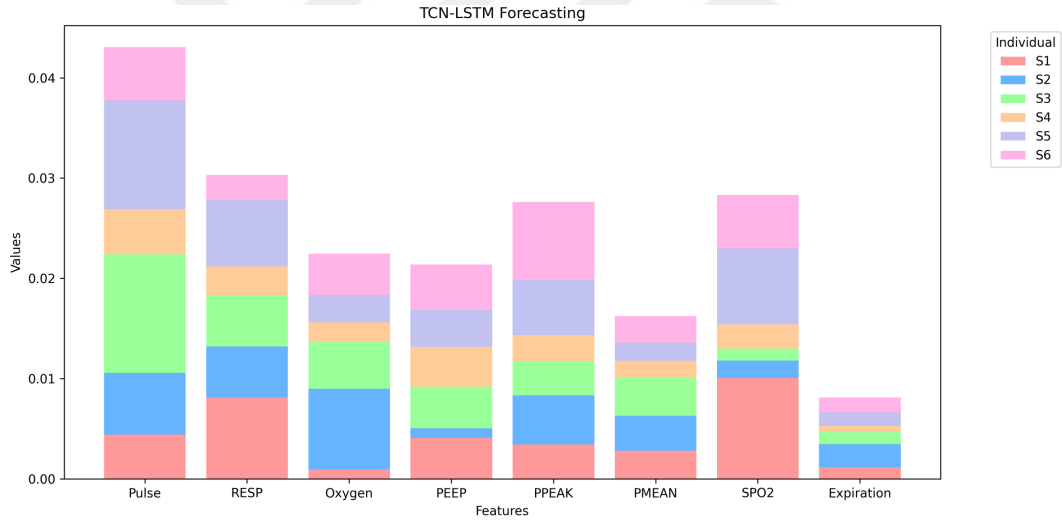


Figure 5.26 Stacked bar chart of the mean absolute error values of the TCN-LSTM prediction.

Figure 5.28,5.29,5.30,5.31 presents a comparison chart of each feature's data trained across four different models for six individuals, based on the mean squared error, root mean squared error, mean absolute error, and the coefficient of determination (R^2) metrics.

According to this table:

In the case of Pulse, the TCN and TCN-LSTM models provided the best results in most cases. For individual S1, the TCN model achieved the lowest MAE value

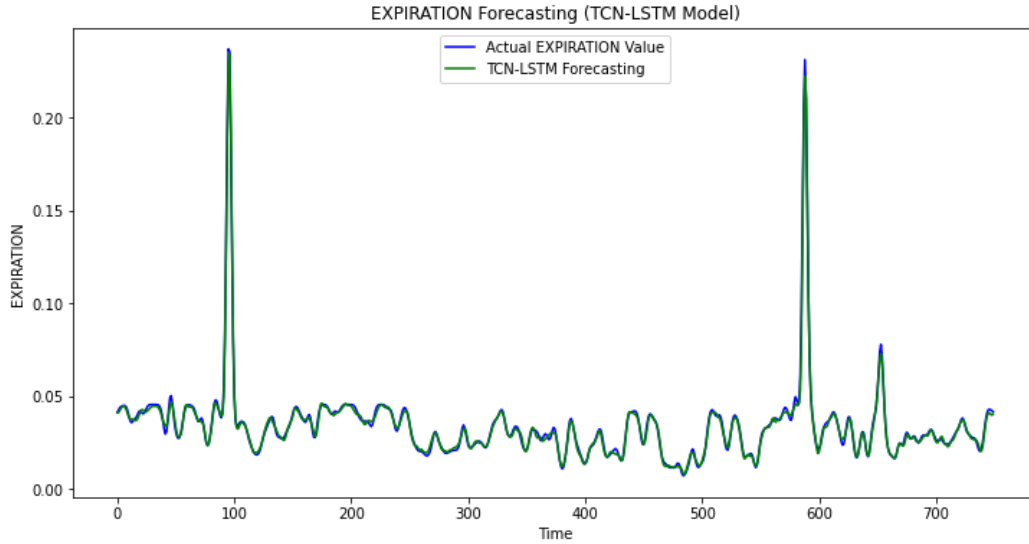


Figure 5.27 The actual value of the Expiration feature and the TCN-LSTM predicted value.

(0.002515), indicating that the TCN model is more successful in modeling the pulse feature.

In the case of RESP, performance varies among the LSTM, TCN, and TCN-LSTM models. The best results for individuals S2 and S6 were obtained with TCN-LSTM, while for other individuals, the TCN or LSTM models achieved the lowest MAE values.

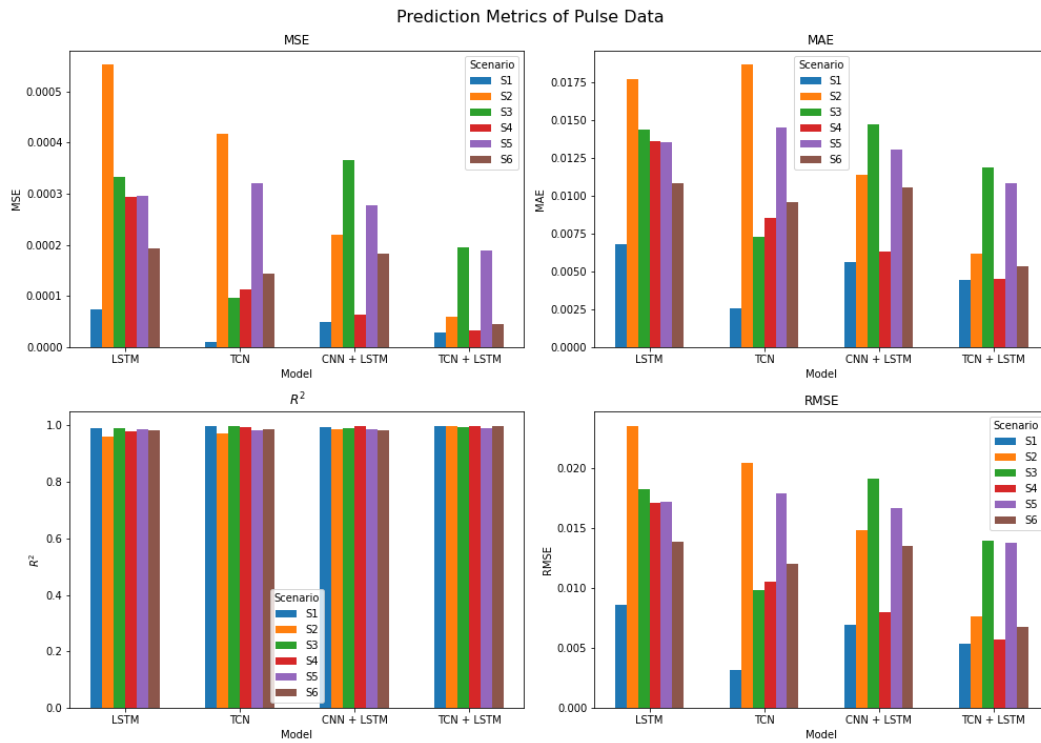
In the case of Oxygen, the TCN and TCN-LSTM models performed better. Specifically, for individual S4, the TCN-LSTM model provided the lowest MAE value (0.001896).

In the case of PEEP, the TCN-LSTM model provided the best results for most individuals. For individual S2, the TCN-LSTM model achieved the lowest error rate with an MAE of 0.000974, indicating that TCN-LSTM yields more accurate results in predicting PEEP.

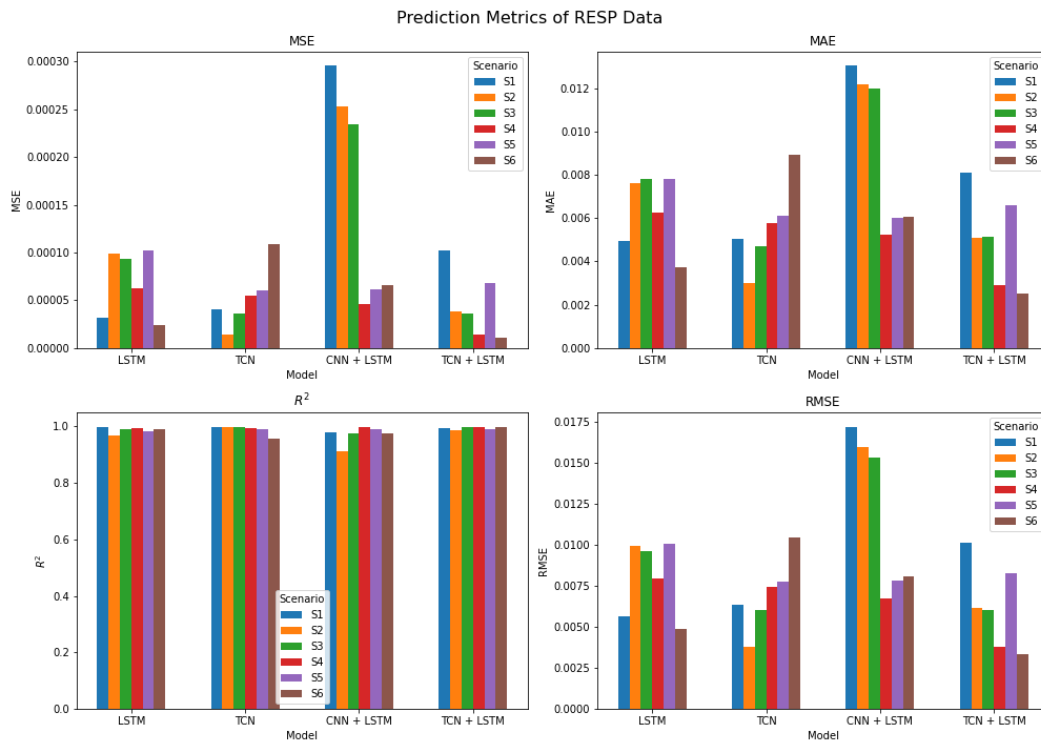
In the case of PPEAK, TCN performed better than the other models. The TCN model achieved the lowest MAE value for individual S5.

In the case of PMEAN, the TCN-LSTM model reached the lowest MAE values in most cases.

In the case of SPO2, the TCN-LSTM model provided the best results in most cases, with the lowest error rate for individual S6 achieved by TCN-LSTM.



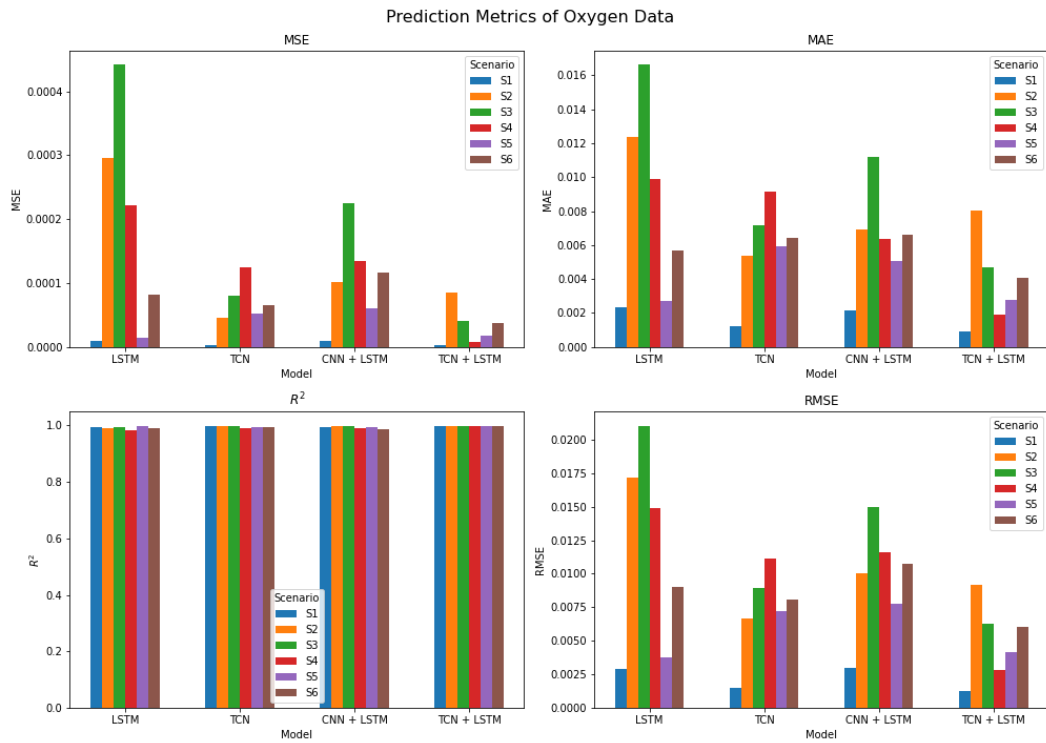
(a) Prediction Metrics of Pulse Data.



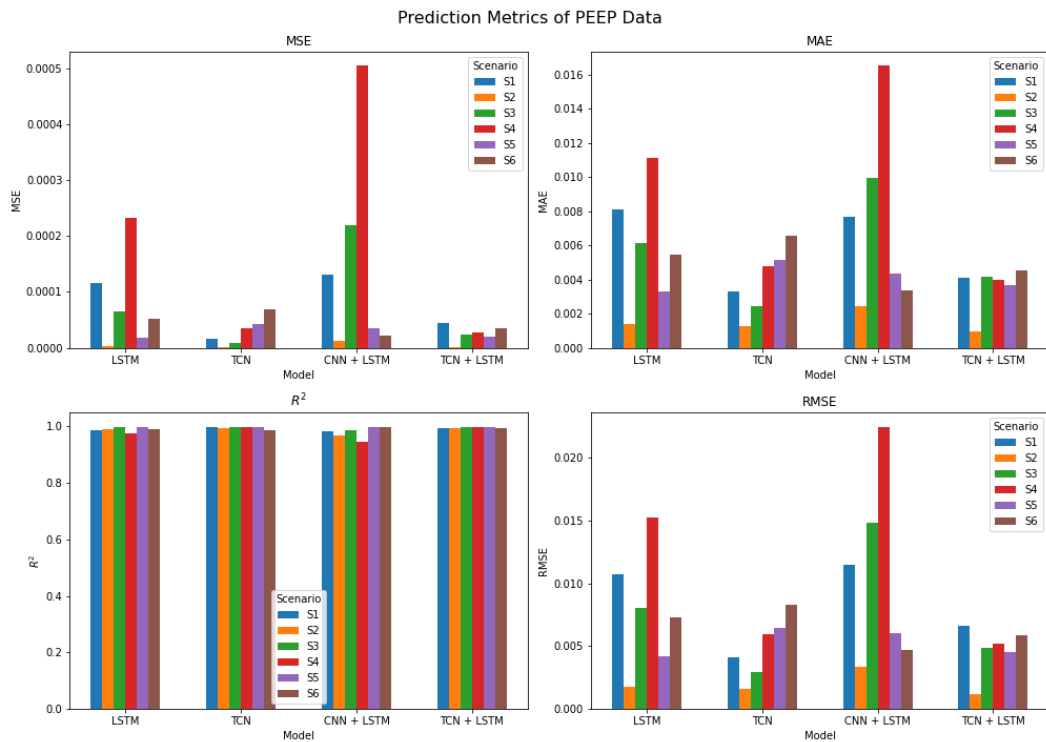
(b) Prediction Metrics of RESP Data.

Figure 5.28 Bar chart of metric values obtained as a result of estimation of Pulse and RESP features for all models.

In the case of Expiration, the TCN-LSTM model gave the best results for most individuals, with the lowest error rate for individual S4 achieved by TCN-LSTM,



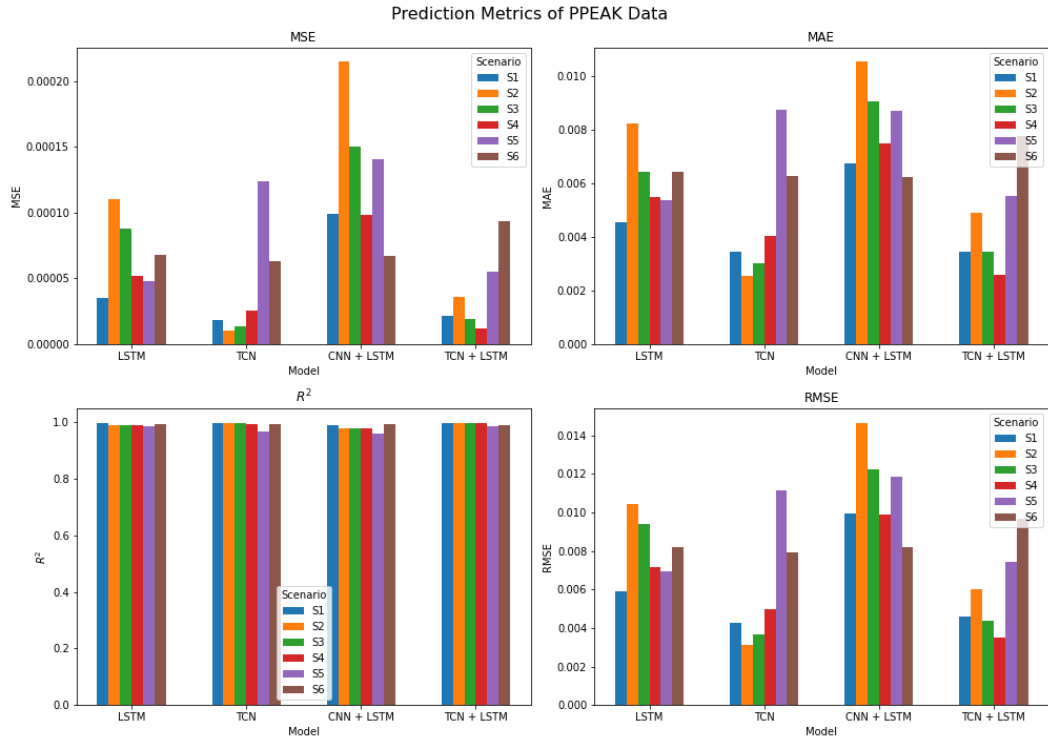
(a) Prediction Metrics of Oxygen Data.



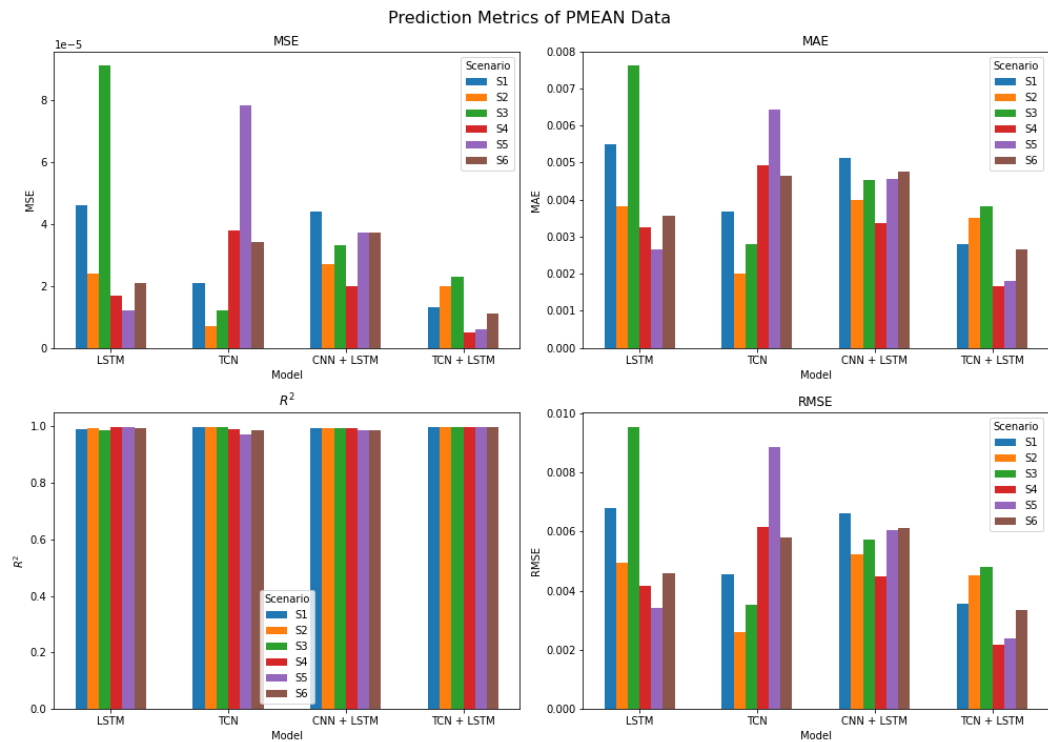
(b) Prediction Metrics of PEEP Data.

Figure 5.29 Bar chart of metric values obtained as a result of estimation of Oxygen and PEEP features for all models.

yielding an MAE of 0.000618.



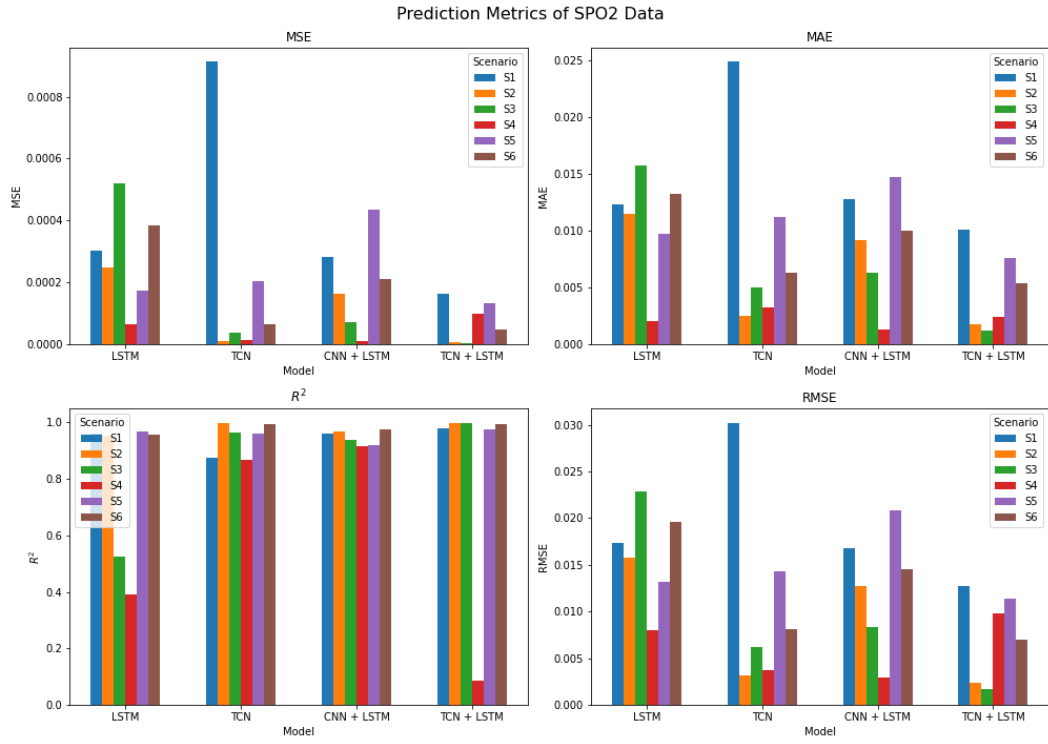
(a) Prediction Metrics of PPEAK Data.



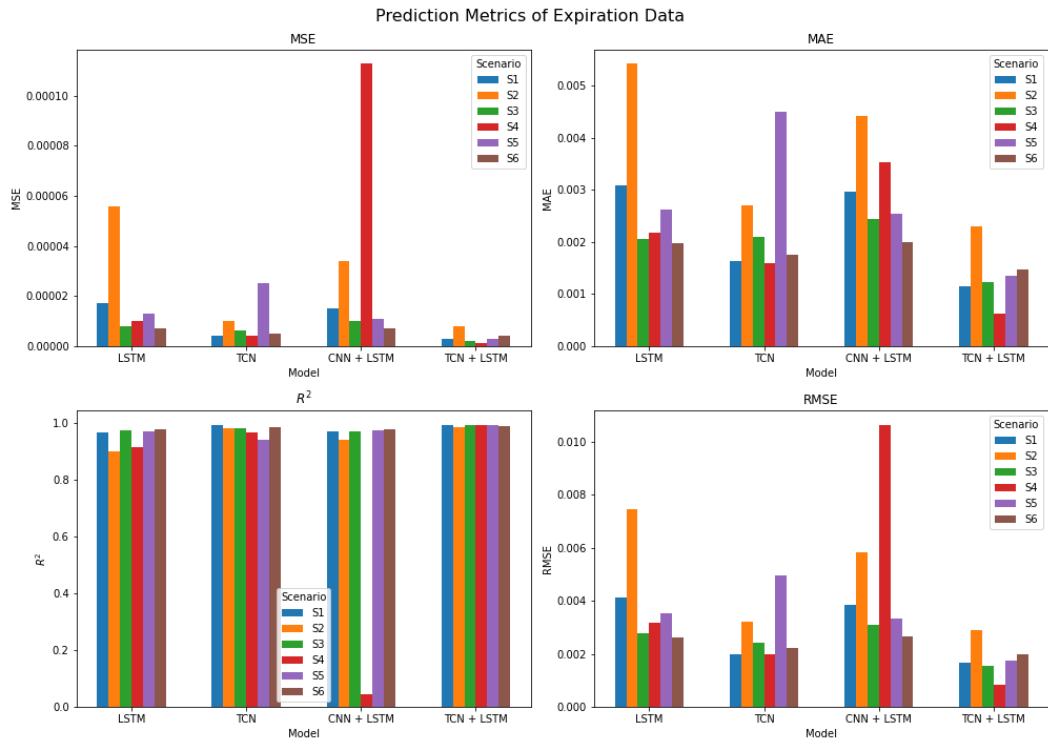
(b) Prediction Metrics of PMEAN Data.

Figure 5.30 Bar chart of metric values obtained as a result of estimation of PPEAK and PMEAN features for all models.

According to these results, the TCN-LSTM model demonstrated the best performance in predicting respiratory data. The forecast results for an individual



(a) Prediction Metrics of SPO2 Data.



(b) Prediction Metrics of Expiration Data.

Figure 5.31 Bar chart of metric values obtained as a result of estimation of SPO2 and Expiration features for all models.

using the TCN-LSTM model are shown in Figure 5.32.

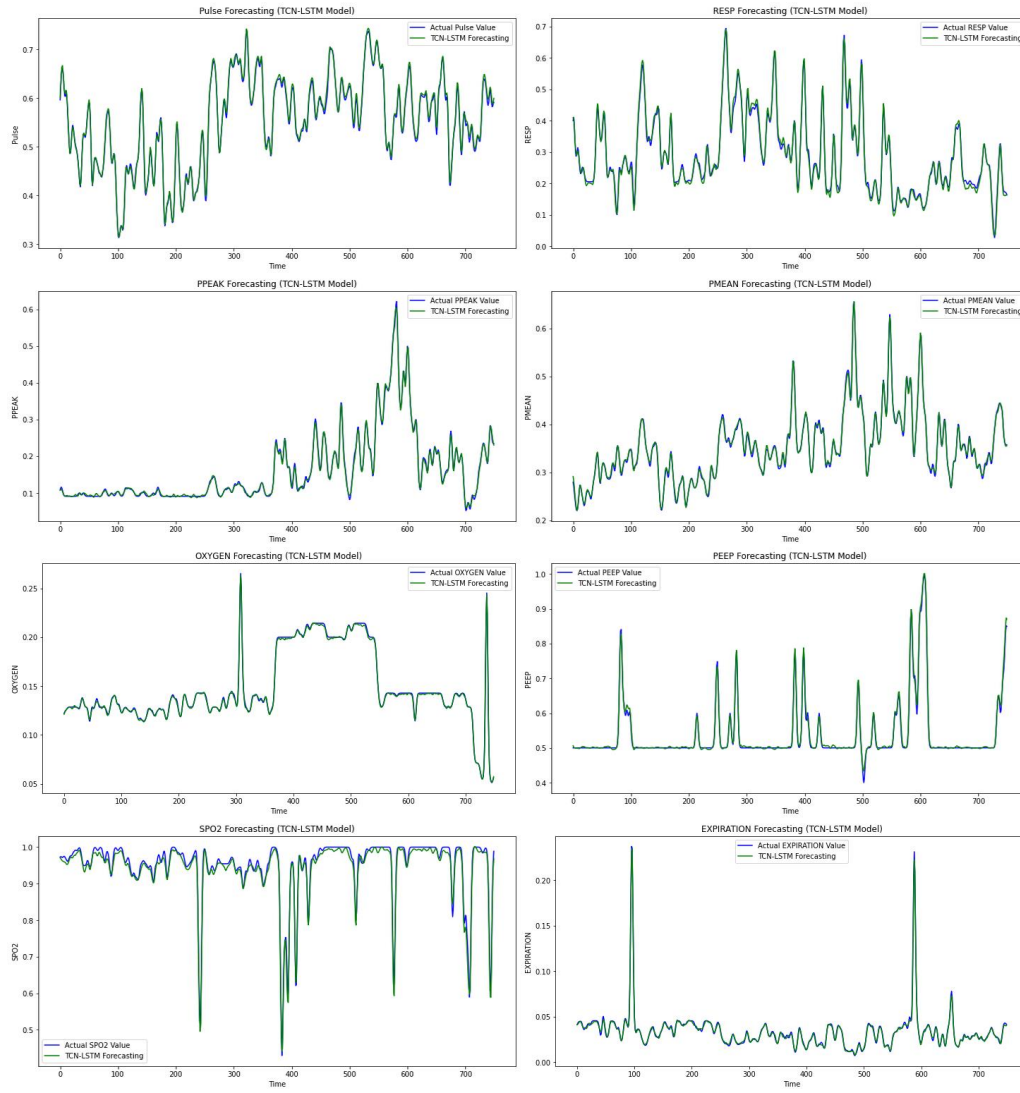


Figure 5.32 The actual value and TCN-LSTM predicted value of eight features.

6

POSITION CLASSIFICATION BASED ON RESPIRATORY MONITORING WITH THORAX AND PULSE OXIMETER

Thorax sensors and pulse oximeter devices are commonly used biomedical tools for respiratory monitoring and assessment. These devices continuously measure vital parameters such as chest movements, oxygen saturation (SpO₂), and heart rate (PRbpm), providing healthcare professionals with detailed information on patients' respiratory health. Specifically, thoracic sensors that measure chest force monitor the effort and depth of respiratory muscles, while pulse oximeters track blood oxygen levels and pulse rate, offering critical insights into the respiratory and circulatory systems. Healthcare professionals can analyze the data from these devices to detect any irregularities or abnormalities in an individual's respiratory system at an early stage. Patients can also monitor their own respiratory status using these sensors. Especially with Bluetooth communication capabilities, these devices provide easily accessible and analyzable data via mobile applications or computer programs.

In this study, four key respiratory and health parameters (chest force, respiratory rate [bpm], SpO₂, and PRbpm) were collected from 10 volunteers using a thoracic sensor and pulse oximeter in three different positions (sitting, walking, and lying). The goal was to classify individuals' positions based on this collected data. Thus, in the first phase of the thesis, an alternative approach for collecting and processing respiratory data was provided, complementing the AI studies conducted with ventilator data. One of the main objectives of this study is to understand the effectiveness of thoracic sensors and pulse oximeters in assessing respiratory data. The sensors used are shown in Figure 6.1.

Thorax sensors are used to monitor chest movements, such as breathing, and other related biological data in the chest area. They are primarily used to measure respiratory rate and the depth of breathing. Their working principle is based on



(a) Thorax sensor.



(b) Pulse oximeter.

Figure 6.1 Sensors used in the study.

sensors located on an elastic band wrapped around the chest, detecting changes as the ribcage expands and contracts [64]. The features of the thoracic sensor used are presented in Table 6.1.

The pulse oximeter is a monitoring device that non-invasively measures arterial oxygen saturation by recording the light absorption of different types of hemoglobin in the arteries [65]. The specifications of the pulse oximeter used are presented in Table 6.2.

6.1 Dataset

Respiratory data was collected from 10 healthy volunteers, aged 20-35, consisting of 6 males and 4 females. The data collection setup is shown in Figure 6.2.

The thorax sensor was attached to the chest area, and the pulse oximeter was placed on the finger of each participant. Each individual was asked to sit, walk, and sleep sequentially, with data collected for 5 minutes in each position. The thoracic sensor recorded chest force (N) and respiratory rate (bpm), while the pulse oximeter

Table 6.1 Thorax sensor features.

Feature	Details
Range	0–50 N
Resolution	0.01 N
Response time	50 ms
Respiratory rate	Sampling window: 30 s Progress interval: 10 s
Step rate	Sampling window: 10 s Progress interval: 10 s
Maximum chest circumference	140 cm
Wireless feature	Bluetooth 4.2
Maximum wireless range	30 m
Battery	300 mA Li-Poly
Battery life (single full charge)	24 hours
Battery life (long-term)	500 full charge cycles (several years depending on usage)

Table 6.2 Pulse oximeter features.

Feature	Details
Measurement range	Oxygen saturation: 0–100% Pulse: 30–250 bpm
Accuracy	Oxygen saturation: 70–100%, $\pm 2\%$ Pulse rate: 30–250, ± 2 bpm
Sensor	Red light (660 nm); infrared light (905 nm); silicon receiver diode
Operating conditions	Temperature: 10°–40°C Humidity: $\leq 75\%$ Pressure: 700–1060 hPa
Power supply	2 x 1.5V AAA battery
Data transmission via Bluetooth	Bluetooth 4.0

measured oxygen saturation (SpO₂) and pulse rate (PRbpm). The recorded thorax data was transferred via Bluetooth to the Vernier Graphical Analysis interface [66]. After attaching the thoracic sensor to the patient, it was powered on using the button, and then connected to the interface via Bluetooth. Once connected, the interface generated a GAMBL file. GAMBL is a project file created by Vernier Graphical Analysis that allows real-time monitoring of force and respiratory rate data, either as a graph or in tabular format. During project setup, the time interval and sampling frequency for data recording can be configured. After data collection from the patient, the project file was used to save the data's graphical representations as images and its values in CSV (comma-separated values) format. Pulse oximeter data was recorded at 30-second intervals. The graphical variation of force and

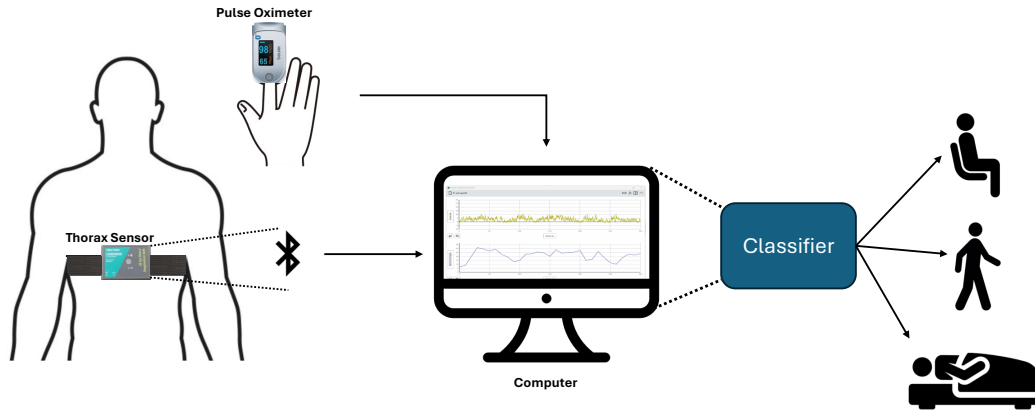


Figure 6.2 Data collection setup.

respiratory rate for a participant in sitting, walking, and lying positions is shown in Figure 6.3.

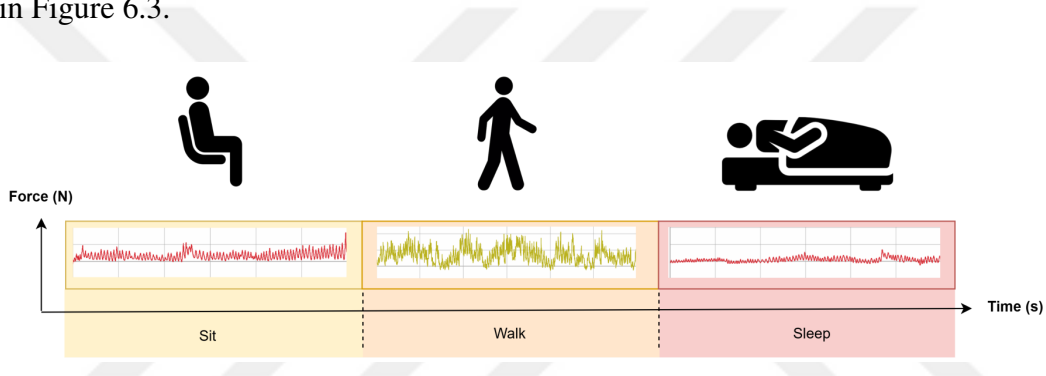


Figure 6.3 Force over time plot for three places.

6.2 Classification with Machine Learning Algorithms

In this study, Random Forest, Decision Tree, Gradient Boosting, AdaBoost, Bagging, K-Nearest Neighbors, Gaussian Naive Bayes, and XGBoost algorithms were used. During the preprocessing step, the data was normalized. This processed data was then applied to each model, and evaluation metrics were obtained. The performances of these eight algorithms were assessed and compared.

The plot comparing the evaluation metrics obtained from applying the machine learning algorithms to the normalized data is shown in Figure 6.4.

Among the models, XGBoost demonstrated the best performance with an accuracy rate of 99%. The ROC curve and confusion matrix for each model are shown in Figure 6.5,6.6.

Based on the ROC curves and confusion matrices, XGBoost demonstrated higher classification capability across the four classes compared to other models. Random

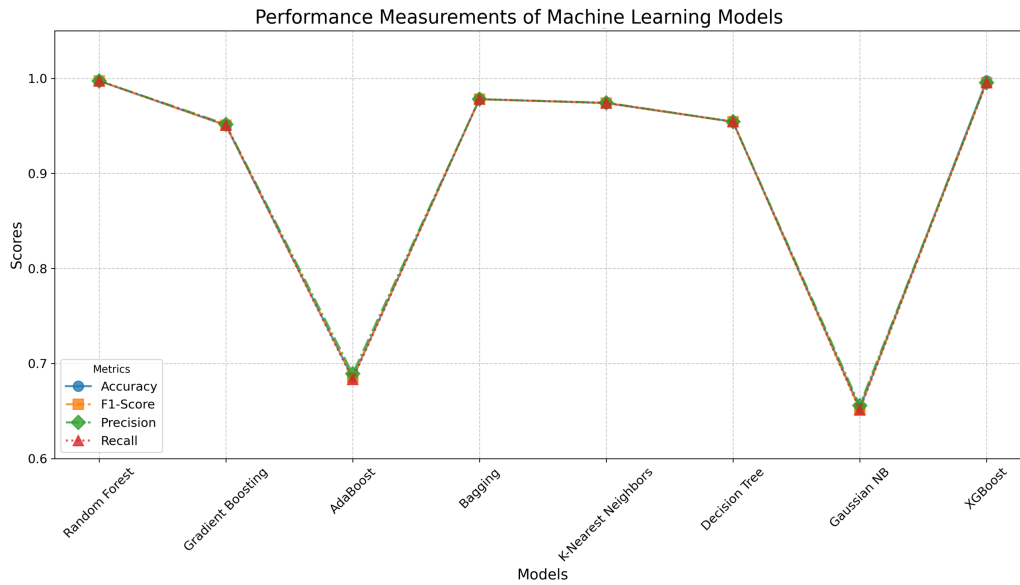


Figure 6.4 Comparison of metrics obtained as a result of applying machine learning algorithms to thorax and pulse oximeter sensor data.

Forest achieved the second-best results after XGBoost, though its classification capability was not as high as XGBoost's. The evaluation metrics are presented in Table 6.3.

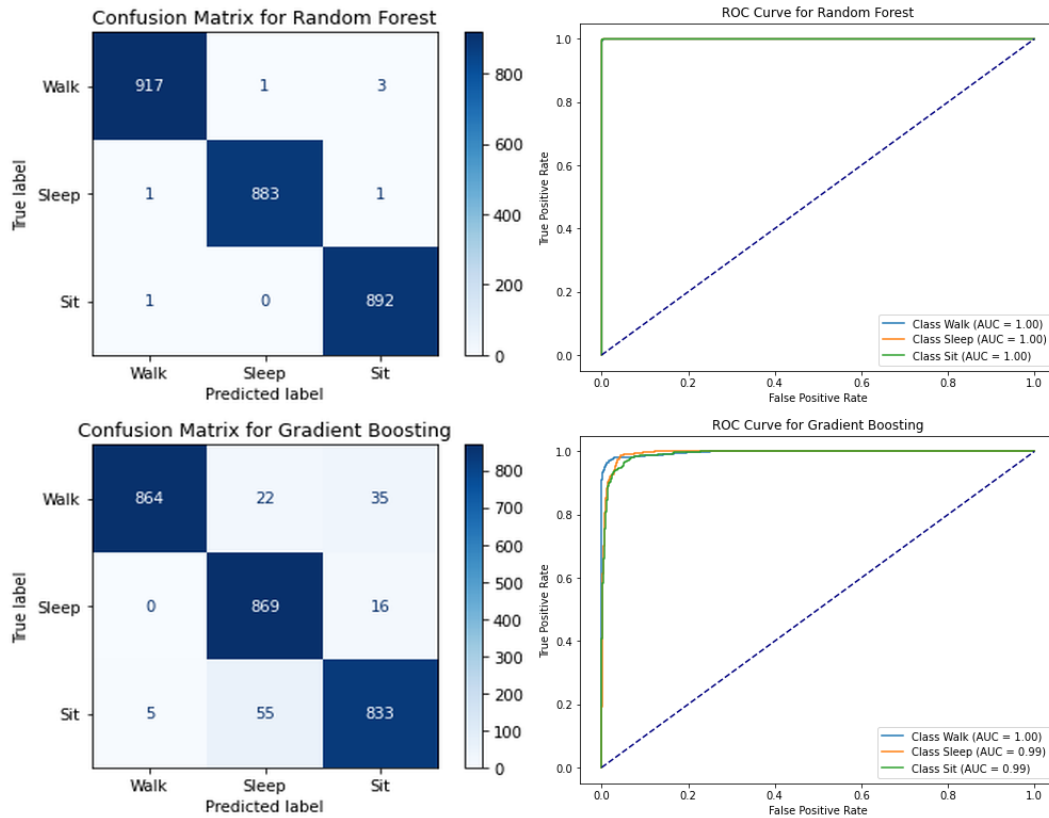
Table 6.3 Comparison of performance metrics across different machine learning algorithms.

Model	Accuracy	F1-Score	Precision	Recall
Random Forest	0.9974	0.9974	0.9974	0.9974
Gradient Boosting	0.9507	0.9507	0.9517	0.9509
AdaBoost	0.6847	0.6853	0.6892	0.6833
Bagging	0.9781	0.9781	0.9781	0.9782
K-Nearest Neighbors	0.9741	0.9740	0.9744	0.9741
Decision Tree	0.9544	0.9544	0.9545	0.9545
Gaussian NB	0.6528	0.6517	0.6558	0.6512
XGBoost	0.9974	0.9956	0.9956	0.9955

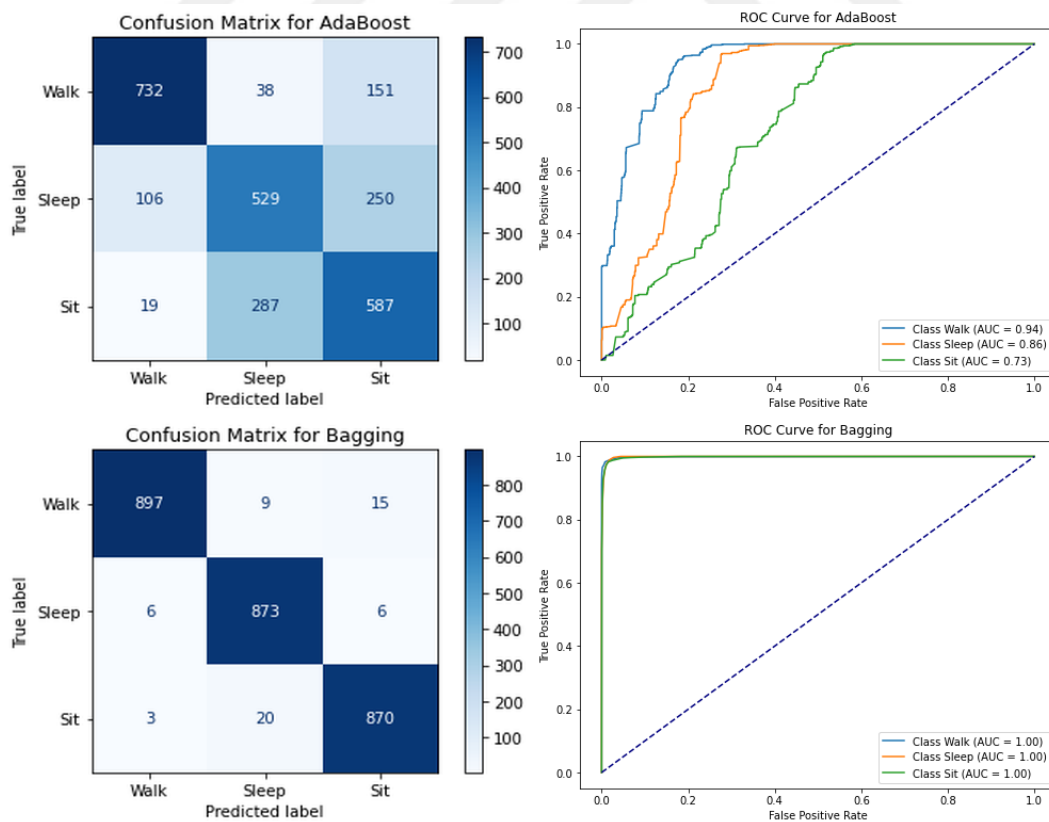
6.3 Position Classification Results

In the process of preparation, the data was normalized, and the resultant dataset was utilized for each method to compute evaluation metrics. The metrics—Accuracy, F1 Score, Precision, Recall, and ROC (AUC) curve—were applied to evaluate the classification performance of each model.

The results of the study showed that the XGBoost algorithm achieved the best performance with an accuracy rate of 99.74%. The high classification accuracy of

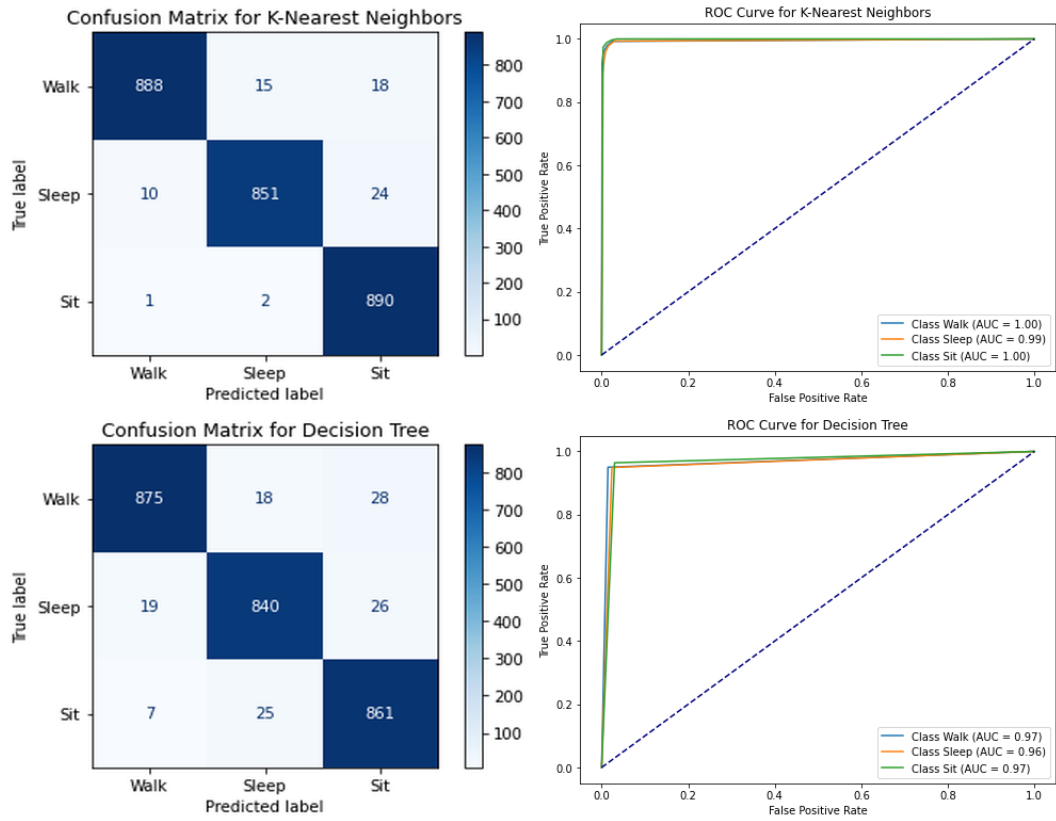


(a) ROC curve plots and confusion matrices of Random Forest and Gradient Boosting algorithms.

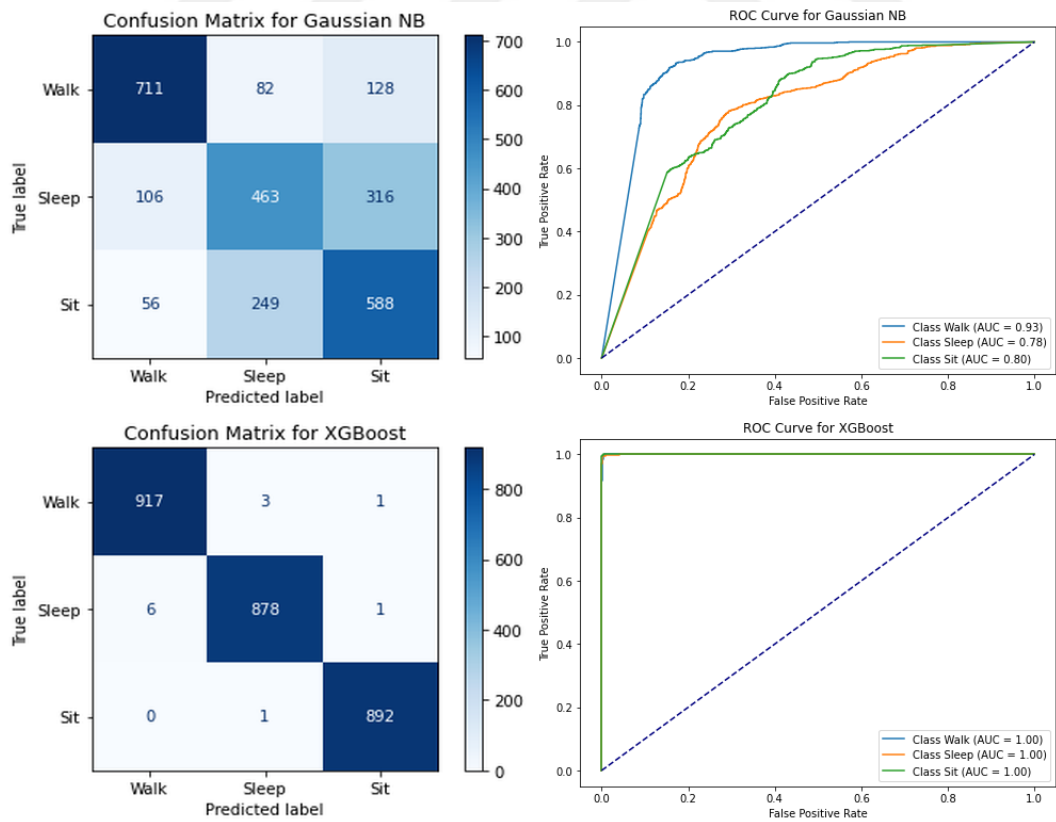


(b) ROC curve plots and confusion matrices of AdaBoost and Bagging algorithms.

Figure 6.5 ROC curve plots and confusion matrices obtained as a result of applying Random Forest, Gradient Boosting, AdaBoost and Bagging algorithms to thorax and pulse oximeter sensor data.



(a) ROC curve plots and confusion matrices of K-Nearest Neighbors and Decision Tree algorithms.



(b) ROC curve plots and confusion matrices of Gaussian NB and XGBoost algorithms.

Figure 6.6 ROC curve plots and confusion matrices obtained as a result of applying K-Nearest Neighbors, Decision Tree, Gaussian NB and XGBoost algorithms to thorax and pulse oximeter sensor data.

XGBoost, along with its superior performance across other metrics, indicates that this model is the most suitable for the classification problem within this dataset. Random Forest followed XGBoost with an accuracy rate of 99.74%. Although Random Forest produced results very close to XGBoost, it fell slightly behind XGBoost in terms of classification capability.

Bagging and K-Nearest Neighbors demonstrated high performance, achieving accuracy rates of 97.81% and 97.41%, respectively. Gradient Boosting and Decision Tree achieved accuracy rates of 95.07% and 95.44%, respectively. AdaBoost and Gaussian Naive Bayes showed worse accuracy rates of 68.47% and 65.28%, respectively, and displayed lower classification ability on the dataset relative to the other models.

Machine learning algorithms were used to analyze the dataset generated from each individual's respiratory data in different positions (sitting, walking, and sleeping) to create a model capable of distinguishing between these positions, achieving high classification accuracy. This classification process demonstrates that sensor data can be accurately classified according to different conditions. In conclusion, this study highlights the effectiveness of devices such as thoracic sensors and pulse oximeters in assessing respiratory status. The use of a sensor with wireless data transmission capabilities, like the thoracic sensor, can significantly facilitate patient monitoring.

7

CONCLUSION

In this study, an AI-based approach was used to classify ventilation modes and perform time series predictions of respiratory data for patients receiving mechanical ventilation support in intensive care units, with the aim of improving patient outcomes. As experienced during the COVID-19 pandemic, the effective use of ventilator devices is critically important for patient health and resource management in intensive care units. Proper management of the mechanical ventilation process accelerates patients' recovery and reduces the risk of complications. Selecting the appropriate ventilation mode is crucial to providing support best suited to the patient's pathological condition, a process that depends on the expertise and knowledge of the physician. Therefore, leveraging AI to accurately determine ventilation modes based on intensive care patient data is of great importance in reducing patients' intubation duration and supporting healthcare professionals.

The data was collected from 23 patients in the neurology intensive care unit, with recordings taken at 1-hour intervals over 24-hour periods. The data collection period varied by patient, ranging from 30 to 90 days. The dataset includes 18 features, with 7 from monitoring devices and 11 from ventilators. The data is labeled into four different classes (CMV, PSIMV, ASV, SPONT). To improve the performance of AI applications, the data was preprocessed with normalization and filtering. Two types of filters, moving average and Gaussian filter, were applied during the filtering step for comparison. When tested with AI algorithms, the Gaussian-filtered data showed higher accuracy.

For ventilation mode classification, the machine learning algorithms used included Random Forest, Decision Tree, Gradient Boosting, AdaBoost, Bagging, K-Nearest Neighbors, Gaussian Naive Bayes, and XGBoost, along with the deep learning models LSTM and CNN-LSTM. For the LSTM model, 144 different configurations were evaluated with variations in the number of layers, batch size, learning rate, and loss functions to select the optimal setup. When assessing the performance of the models, the CNN-LSTM model achieved the best result among deep

learning models with 94% accuracy. Among the machine learning algorithms, XGBoost provided the highest accuracy at 98%. Thus, for ventilation mode classification, XGBoost proved to be the best-performing algorithm, delivering higher classification accuracy than the CNN-LSTM model.

Accurate prediction of respiratory data is crucial for managing the treatment processes of patients in intensive care units and for the early detection of potential respiratory issues. In this study, along with ventilation mode classification, time-series predictions were made for respiratory data collected from patients. The respiratory predictions were made for eight features from six different individuals. Time-series prediction relies on past data to estimate future values. In this approach, the Pulse feature data of one individual was trained using the LSTM network across six data groups: raw data, normalized data, and both datasets with applied moving average and Gaussian filters, at time steps of 100, 200, 300, and 400. The training results were evaluated using regression metrics such as mean squared error, root mean squared error, mean absolute error, and the Coefficient of Determination (R^2). The lowest error rate was achieved when using normalized Gaussian-filtered data with a time step of 200. The data was preprocessed using normalization and Gaussian filtering and then tested using LSTM, TCN, CNN-LSTM, and TCN-LSTM models. The models were trained with the first 200 time steps of the time series data, and predictions were made for the next 600–900 time steps, depending on each individual's data collection interval. Training results were evaluated using MSE, RMSE, MAE, and R^2 metrics. The TCN-LSTM model demonstrated the best performance with the lowest error rate and the highest coefficient of determination.

In this study, chest force, respiratory rate, SpO₂, and PRbpm data were collected from 10 volunteers using thoracic sensors and a pulse oximeter in three different positions, and a position classification approach based on individuals' respiratory values was proposed. For position classification, machine learning algorithms such as Random Forest, Decision Tree, Gradient Boosting, AdaBoost, Bagging, K-Nearest Neighbors, Gaussian Naive Bayes, and XGBoost were used. The XGBoost algorithm achieved the highest classification accuracy with a score of 99.74%.

The ventilator data collected from patients are the types of data assessed by healthcare professionals both in our country and worldwide. In this study, the high accuracy rates achieved in the classification and prediction of this data demonstrate that artificial intelligence provides an effective solution to this problem. Additionally, the proposed approach using thoracic and pulse sensors proved to be

more practical compared to the literature and highly effective based on artificial intelligence outcomes. The results of this study indicate that it could contribute positively to the development of AI-based health decision support systems.



REFERENCES

- [1] M. C. Guilhermino, K. J. Inder, D. Sundin, “Education on invasive mechanical ventilation involving intensive care nurses: A systematic review,” *Nursing in critical care*, vol. 23, no. 5, pp. 245–255, 2018.
- [2] J. S. Alqahtani *et al.*, “Global current practices of ventilatory support management in covid-19 patients: An international survey,” *Journal of multidisciplinary healthcare*, pp. 1635–1648, 2020.
- [3] T. Barnes, D. Enk, “Ventilation for low dissipated energy achieved using flow control during both inspiration and expiration,” *Trends in Anaesthesia and Critical Care*, vol. 24, pp. 5–12, 2019.
- [4] D. S. Cooper *et al.*, “Current challenges in cardiac intensive care: Optimal strategies for mechanical ventilation and timing of extubation,” *Cardiology in the Young*, vol. 18, no. S3, pp. 72–83, 2008.
- [5] S. Gupta, S. K. Sinha, “Newer modalities of mechanical ventilation in the extremely premature infant,” *Paediatrics and child health*, vol. 17, no. 2, pp. 37–42, 2007.
- [6] G. Zhang, L. Luo, L. Zhang, Z. Liu, “Research progress of respiratory disease and idiopathic pulmonary fibrosis based on artificial intelligence,” *Diagnostics*, vol. 13, no. 3, p. 357, 2023.
- [7] S. Rosati, A. Scotto, V. Fanelli, G. Balestra, “Feature selection based on a genetic algorithm for optimizing weaning success,” in *Caring is Sharing—Exploiting the Value in Data for Health and Innovation*, IOS Press, 2023, pp. 566–570.
- [8] H. Zhang, L. Wang, J. Xu, Y. Xiang, Z. Zhang, “Multi-modal feature fusion-based machine learning to detect abnormal mechanical ventilation,” *IEEE Access*, 2023.
- [9] F. P. Akbulut, E. Akkur, A. Akan, B. S. Yarman, “A decision support system to determine optimal ventilator settings,” *BMC Medical Informatics and Decision Making*, vol. 14, pp. 1–11, 2014.
- [10] S. Ren, J. Niu, M. Cai, Y. Shi, T. Wang, Z. Luo, “Sputum deposition classification for mechanically ventilated patients using lstm method based on airflow signals,” *Heliyon*, vol. 8, no. 12, 2022.
- [11] T. Anitha, G. Gopu, M. Assaad, *et al.*, “Machine learning algorithm for ventilator mode selection, pressure and volume control,” *Plos one*, vol. 19, no. 3, e0299653, 2024.

- [12] R. K. Thakur, R. Kumar, R. Kumar, A. Vardhan, "Auto-mode shifting: A machine learning-based automatic ventilation mode selection technique personalized to physiological characteristics of intubated patients," *2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET)*, IEEE, 2023, pp. 1–6.
- [13] T. Bakkes *et al.*, "Automated detection and classification of patient–ventilator asynchrony by means of machine learning and simulated data," *Computer Methods and Programs in Biomedicine*, vol. 230, p. 107 333, 2023.
- [14] C.-F. Liu *et al.*, "An artificial intelligence system to predict the optimal timing for mechanical ventilation weaning for intensive care unit patients: A two-stage prediction approach," *Frontiers in Medicine*, vol. 9, p. 935 366, 2022.
- [15] C. Y. S. Ang *et al.*, "Patient-ventilator asynchrony classification in mechanically ventilated patients: Model-based or machine learning method?" *Computer Methods and Programs in Biomedicine*, vol. 255, p. 108 323, 2024.
- [16] A. Koul, R. K. Bawa, Y. Kumar, "Artificial intelligence techniques to predict the airway disorders illness: A systematic review," *Archives of Computational Methods in Engineering*, vol. 30, no. 2, pp. 831–864, 2023.
- [17] C. Lam *et al.*, "Semisupervised deep learning techniques for predicting acute respiratory distress syndrome from time-series clinical data: Model development and validation study," *JMIR formative research*, vol. 5, no. 9, e28028, 2021.
- [18] J. Kim, M. Chae, H.-J. Chang, Y.-A. Kim, E. Park, "Predicting cardiac arrest and respiratory failure using feasible artificial intelligence with simple trajectories of patient data," *Journal of clinical medicine*, vol. 8, no. 9, p. 1336, 2019.
- [19] A. K. Kumar, M. Ritam, L. Han, S. Guo, R. Chandra, "Deep learning for predicting respiratory rate from biosignals," *Computers in biology and medicine*, vol. 144, p. 105 338, 2022.
- [20] S. Wang, K. Kam, C. Xiao, S. Bowen, W. Chaovalitwongse, "An efficient time series subsequence pattern mining and prediction framework with an application to respiratory motion prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.
- [21] U. Ahmed, J. C.-W. Lin, G. Srivastava, "Multivariate time-series sensor vital sign forecasting of cardiovascular and chronic respiratory diseases," *Sustainable Computing: Informatics and Systems*, vol. 38, p. 100 868, 2023.
- [22] M. Roudjane *et al.*, "Smart t-shirt based on wireless communication spiral fiber sensor array for real-time breath monitoring: Validation of the technology," *IEEE Sensors Journal*, vol. 20, no. 18, pp. 10 841–10 850, 2020.
- [23] E. F. Guy, J. G. Chase, L. R. Holder-Pearson, "Respiratory bi-directional pressure and flow data collection device with thoracic and abdominal circumferential monitoring," *HardwareX*, vol. 12, e00354, 2022.

- [24] M. Hesse, P. Christ, T. Hörmann, U. Rückert, "A respiration sensor for a chest-strap based wireless body sensor," *SENSORS, 2014 IEEE*, IEEE, 2014, pp. 490–493.
- [25] M. Abdollahi, E. Rashedi, S. Jahangiri, P. M. Kuber, N. Azadeh-Fard, M. Dombovy, "Fall risk assessment in stroke survivors: A machine learning model using detailed motion data from common clinical tests and motor-cognitive dual-tasking," *Sensors*, vol. 24, no. 3, p. 812, 2024.
- [26] L. Raiano, J. Di Tocco, C. Massaroni, G. Di Pino, E. Schena, D. Formica, "A pca-based method to select the number and the body location of piezoresistive sensors in a wearable system for respiratory monitoring," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6847–6855, 2020.
- [27] J. Buekers *et al.*, "Wearable finger pulse oximetry for continuous oxygen saturation measurements during daily home routines of patients with chronic obstructive pulmonary disease (copd) over one week: Observational study," *JMIR mHealth and uHealth*, vol. 7, no. 6, e12866, 2019.
- [28] A. M. Luks, E. R. Swenson, "Pulse oximetry for monitoring patients with covid-19 at home. potential pitfalls and practical guidance," *Annals of the American Thoracic Society*, vol. 17, no. 9, pp. 1040–1046, 2020.
- [29] W. Johnston, Y. Mendelson, "Extracting breathing rate information from a wearable reflectance pulse oximeter sensor," *The 26th annual international conference of the IEEE engineering in medicine and biology society*, IEEE, vol. 2, 2004, pp. 5388–5391.
- [30] I. D. Mienye, N. Jere, "A survey of decision trees: Concepts, algorithms, and applications," *IEEE Access*, 2024.
- [31] L. Rokach, O. Maimon, "Top-down induction of decision trees classifiers-a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 4, pp. 476–487, 2005.
- [32] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123–140, 1996.
- [33] A. Parmar, R. Katariya, V. Patel, "A review on random forest: An ensemble classifier," *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*, Springer, 2019, pp. 758–763.
- [34] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.
- [35] Y. Freund, R. E. Schapire, *et al.*, "Experiments with a new boosting algorithm," *icml*, Citeseer, vol. 96, 1996, pp. 148–156.
- [36] M. Suyal, P. Goyal, "A review on analysis of k-nearest neighbor classification machine learning algorithms based on supervised learning," *International Journal of Engineering Trends and Technology*, vol. 70, no. 7, pp. 43–48, 2022.
- [37] H. Kamel, D. Abdulah, J. M. Al-Tuwaijari, "Cancer classification using gaussian naive bayes algorithm," *2019 international engineering conference (IEC)*, IEEE, 2019, pp. 165–170.

- [38] T. Chen, “Xgboost: Extreme gradient boosting,” *R package version 0.4-2*, vol. 1, no. 4, 2015.
- [39] R. Guo, Z. Zhao, T. Wang, G. Liu, J. Zhao, D. Gao, “Degradation state recognition of piston pump based on iceemdan and xgboost,” *Applied Sciences*, vol. 10, no. 18, p. 6593, 2020.
- [40] C. Bentéjac, A. Csörgő, G. Martínez-Muñoz, “A comparative analysis of gradient boosting algorithms,” *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2021.
- [41] S. Hochreiter, “Long short-term memory,” *Neural Computation MIT-Press*, 1997.
- [42] Z. Zeng, H. Xiao, X. Zhang, “Self cnn-based time series stream forecasting,” *Electronics Letters*, vol. 52, no. 22, pp. 1857–1858, 2016.
- [43] M. Krichen, “Convolutional neural networks: A survey,” *Computers*, vol. 12, no. 8, p. 151, 2023.
- [44] P. Lara-Benítez, M. Carranza-García, J. M. Luna-Romera, J. C. Riquelme, “Temporal convolutional networks applied to energy-related time series forecasting,” *applied sciences*, vol. 10, no. 7, p. 2322, 2020.
- [45] S. Bai, J. Z. Kolter, V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.
- [46] F. Yu, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.
- [47] K. Lee, J. Ray, C. Safta, “The predictive skill of convolutional neural networks models for disease forecasting,” *Plos one*, vol. 16, no. 7, e0254319, 2021.
- [48] A. D. Betensley, I. Khalid, J. Crawford, R. A. Pensler, B. Digiovine, “Patient comfort during pressure support and volume controlled-continuous mandatory ventilation,” *Respiratory care*, vol. 53, no. 7, pp. 897–902, 2008.
- [49] M. Keszler, “Mechanical ventilation strategies,” *Seminars in Fetal and Neonatal Medicine*, Elsevier, vol. 22, 2017, pp. 267–274.
- [50] J. Brunner, G. A. Iotti, “Adaptive support ventilation (asv),” *Minerva anesthesiologica*, vol. 68, no. 5, pp. 365–368, 2002.
- [51] R. L. Chatburn, “Classification of ventilator modes: Update and proposal for implementation,” *Respiratory care*, vol. 52, no. 3, pp. 301–323, 2007.
- [52] R. L. Chatburn, E. Mireles-Cabodevila, “Closed-loop control of mechanical ventilation: Description and classification of targeting schemes,” *Respiratory care*, vol. 56, no. 1, pp. 85–102, 2011.
- [53] E. K. Ampomah, G. Nyame, Z. Qin, P. C. Addo, E. O. Gyamfi, M. Gyan, “Stock market prediction with gaussian naive bayes machine learning algorithm,” *Informatica*, vol. 45, no. 2, 2021.
- [54] X. Zhu, Y. Li, “Lstm neural network based on moving average filtering optimization assisted navigation methods,” *2023 International Conference on Computer Applications Technology (CCAT)*, IEEE, 2023, pp. 110–114.

- [55] E. O. Rybakova, E. E. Limonova, D. P. Nikolaev, "Fast gaussian filter approximations comparison on simd computing platforms," *Applied Sciences*, vol. 14, no. 11, p. 4664, 2024.
- [56] K. Ito, K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE transactions on automatic control*, vol. 45, no. 5, pp. 910–927, 2000.
- [57] M. Hossin, M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.
- [58] R. Yacoub, D. Axman, "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models," *Proceedings of the first workshop on evaluation and comparison of NLP systems*, 2020, pp. 79–91.
- [59] A. V. Tatachar, "Comparative assessment of regression models based on model evaluation metrics," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9, pp. 853–860, 2021.
- [60] G. N. Beken, S. Z. Aydin, B. Erkmen, "Deep long short-term memory networks for asthma disease prediction from respiratory dataset obtained from the peep expiratory occlusion," *2024 47th International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, 2024, pp. 239–242.
- [61] X. Jin, X. Yu, X. Wang, Y. Bai, T. Su, J. Kong, "Prediction for time series with cnn and lstm," *Proceedings of the 11th international conference on modelling, identification and control (ICMIC2019)*, Springer, 2020, pp. 631–641.
- [62] V. M. P. Cortes, A. Chatterjee, D. Khovalyg, "Dynamic personalized human body energy expenditure: Prediction using time series forecasting lstm models," *Biomedical Signal Processing and Control*, vol. 87, p. 105381, 2024.
- [63] J. Du, Z. Liu, W. Dong, W. Zhang, Z. Miao, "A novel tcn-lstm hybrid model for semg-based continuous estimation of wrist joint angles," *Sensors*, vol. 24, no. 17, p. 5631, 2024.
- [64] Vernier, *Go direct respiration belt (order code gdx-rb) user manual*, Accessed on: November 3, 2024, Vernier, 2024. [Online]. Available: <https://www.vernier.com/files/manuals/gdx-rb/gdx-rb.pdf>.
- [65] Beurer, *Po 60 pulse oximeter user manual*, Accessed on: November 3, 2024, Beurer, 2024. [Online]. Available: <https://pim.beurer.com/images/attribut/po60.PDF>.
- [66] Vernier, *Vernier graphical analysis*, <https://www.vernier.com/product/graphical-analysis/>, Accessed on: November 3, 2024, 2024.

PUBLICATIONS FROM THE THESIS

Conference Papers

1. G. N. Beken, S. Z. Aydin and B. Erkmen, "Deep Long Short-Term Memory Networks for Asthma Disease Prediction from Respiratory Dataset Obtained from the PEEP Expiratory Occlusion," 2024 47th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 2024, pp. 239-242, doi: 10.1109/TSP63128.2024.10605923.

Projects

1. "AI-Based Decision Support and Monitoring System for VR-Based Respiratory Rehabilitation", supported by Yildiz Technical University Scientific Research Projects Coordination Unit under project number FCD-2023-5445.