

Article

Ensemble of Deep Learning Models for Sleep Apnea Detection: An Experimental Study

Debadyuti Mukherjee ¹, Koustav Dhar ¹, Friedhelm Schwenker ^{2,*} and Ram Sarkar ¹

¹ Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India; debadyuti23@gmail.com (D.M.); koustavdhar.31.05@gmail.com (K.D.); ram.sarkar@jadavpuruniversity.in (R.S.)

² Institute of Neural Information Processing, University of Ulm, 89069 Ulm, Germany

* Correspondence: friedhelm.schwenker@uni-ulm.de

Abstract: Sleep Apnea is a breathing disorder occurring during sleep. Older people suffer most from this disease. In-time diagnosis of apnea is needed which can be observed by the application of a proper health monitoring system. In this work, we focus on Obstructive Sleep Apnea (OSA) detection from the Electrocardiogram (ECG) signals obtained through the body sensors. Our work mainly consists of an experimental study of different ensemble techniques applied on three deep learning models—two Convolutional Neural Network (CNN) based models, and a combination of CNN and Long Short-Term Memory (LSTM) models, which were previously proposed in the OSA detection domain. We have chosen four ensemble techniques—majority voting, sum rule and Choquet integral based fuzzy fusion and trainable ensemble using Multi-Layer Perceptron (MLP) for our case study. All the experiments are conducted on the benchmark PhysioNet Apnea-ECG Database. Finally, we have achieved highest OSA detection accuracy of 85.58% using the MLP based ensemble approach. Our best result is also able to surpass many of state-of-the-art methods.



Citation: Mukherjee, D.; Dhar, K.; Schwenker, F.; Sarkar, R. Ensemble of Deep Learning Models for Sleep Apnea Detection: An Experimental Study. *Sensors* **2021**, *21*, 5425. <https://doi.org/10.3390/s21165425>

Academic Editor: Biswanath Samanta

Received: 16 June 2021

Accepted: 7 August 2021

Published: 11 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: sleep apnea; ECG signal; ensemble; deep learning; health monitoring

1. Introduction

In this modern era, the role of health monitoring systems is increasing in our daily life. Older people are the most benefited ones from the merits of monitoring their health. Smart sensors, under any healthcare system attached to various body parts, can sense and record the required features of the human body. These kinds of sensors can be placed in any smart-watch or smartphone. The introduction of the Internet of Things (IoT) in the healthcare domain has further upgraded the facilities [1]. Health-based alarms, personal smart medical recommendations, etc. have decreased the life-risks caused by sudden health problems.

One such problem under the domain of health monitoring is sleep apnea detection. Sleep Apnea HypoApnea Syndrome (SAHS) or simply apnea is a common sleep disorder related to interruption in breathing during sleep. In most cases, older people are affected by this syndrome [2]. OSA is a category of apnea that causes partial or complete blockage of the airway in our body. OSA may further cause sleepiness, fatigue, morning headache, etc. [3]. Statistics say that almost 9% of the men and 4% of women among the middle-aged people suffer from the mentioned sleep disorder. The sleep loss caused by OSA may lead to some long-term diseases like cardiovascular diseases [4]. Thus, a smart health monitoring system is beneficial for the diagnosis of apnea so that early measures can be taken.

The most common method for apnea diagnosis is ECG [5]. ECG records the electronic signals generated from the human heart. It serves the purpose to detect whether our heart is abnormally working or not. In this work, we have divided the whole time-series ECG data into time-intervals of equal length. Then we have used CNN [6] based

deep learning models along with ensemble learning to detect apnea in the given time-span. We have chosen three previously proposed CNN models as base models: (i) CNN architecture proposed by Wang et al. [7], (ii) CNN model proposed by Sharan et al. [8], and (iii) combination of CNN and LSTM network [9] proposed by Almutairi et al. [10]. To aggregate the base models' predictions and to yield better results, we have applied four ensemble approaches: (i) Majority Voting, (ii) Sum rule, (iii) Choquet Integral based fuzzy fusion and (iv) Trainable ensemble using MLP. Our work involves the experimental study between these four ensemble techniques.

The main advantage of ensemble learning is that it considers and combines all the decisions by different models rather than relying on a single classifier [11]. An ensemble will be successful if its component classifiers have diversity while making the prediction. Also, the ensemble formation will not serve any purpose if all the components generate too many inaccurate predictions [11].

We have chosen the PhysioNet Apnea-ECG Database [12], a standard and publicly available dataset, to conduct all the required experiments. To summarize, first, we form segments from the raw ECG data from the benchmark database, perform necessary pre-processing to derive important features and then train the three base models. Next, all three deep learning models predict the test data and the final prediction is generated by applying the ensemble technique of choice. Figure 1 pictorially represents the above-mentioned process. The rest of the work consists of the four sections, namely, Related Work, Materials and Methods, Results and Discussion, and Conclusions.

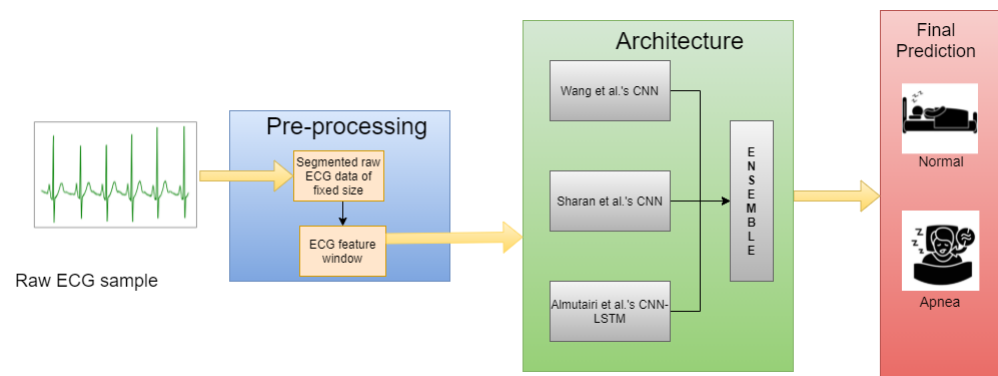


Figure 1. Overall workflow of the ensemble based OSA detection model.

2. Related Work

Since OSA or any other kind of apnea detection is a classification problem of the two classes—normal and apnea, machine learning classifiers like Support Vector Machine (SVM) [13], k-Nearest Neighbours (kNN) [14], Random Forest (RF) [15] etc., and deep learning classifiers like CNN etc., are very much applicable in this domain. Like any other clinical diagnosis, detection of sleep apnea has become an important research topic in the healthcare domain.

Ng et al. [16] have used thoracic and abdominal signals as input features for sleep apnea indication and have achieved 70.29–86.25% sensitivity. Alvarez et al. [17] have worked on the non-linear analysis of blood oxygen saturation (Sa) obtained from nocturnal oximetry. From the experiments, they have discovered 111 out of 187 subjects as OSA positive. Qin et al. [18] have studied the effect of OSA in Heart Rate Variability (HRV). They have conducted the experiments on 426 normal and 826 OSA affected subjects and have discovered that HRV tends to reduce with the severity of apnea disease.

Although there are many statistical body measures like ECG, acoustic speech signal, Sa, Electroencephalogram (EEG) available for apnea diagnosis [5], we have solely focused on ECG signal for our work. A lot of research works on apnea diagnosis from ECG signals have already been performed. Almazaydeh et al. [5] have extracted the relevant statistical features such as mean, standard deviation, median, inter-quartile range and some of their

derivations for an RR interval (interval between two consecutive R peaks) of the raw ECG signals of the PhysioNet Apnea-ECG database [12]. They have applied SVM on these extracted features and have achieved a maximum of 96.5% accuracy. Cheng et al. [19] also have conducted experiments on RR intervals of the ECG signal of the PhysioNet Apnea-ECG database. By applying the Recurrent Neural Network (RNN) [20], they have achieved 97.80% accurate results.

Nguyen et al. [21] have considered the Recurrence Quantification Analysis (RQA) statistics of the HRV data of PhysioNet Apnea-ECG database as features. Initially, they have performed the classification task by using both SVM and Artificial Neural Network (ANN). They have used soft decision fusion to aggregate both the classifiers' scores and have obtained 85.26% accurate results. Hassan et al. [22] have pre-processed the raw ECG signal of the PhysioNet Apnea-ECG database by applying the Tunable-Q factor Wavelet Transform (TQWT). They have used Adaptive Boosting (AdaBoost) [23], an ensemble method applicable to the decision tree and achieved 87.33% accurate results.

Wang et al. [24] have considered the past time-windows for training the MLP architecture. Such time-windows are restricted to have a time-span of a minute, whereas each sample under the respective time-span has the six time-domain RR Interval (RRI) features—MRR (mean of RRI), MHR (mean of heart rates), RMSSD (root mean square of differences between adjacent RRIs), SDNN (standard deviation of RRIs), NN50 (number of adjacent RRIs exceeding 50 milliseconds) and pNN50 (NN50 divided by the number of RR intervals) and six frequency domain R-peak Amplitude features—Very Low Frequency (VLF), Low Frequency (LF), High Frequency (HF), LF/(LF + HF), and HF/(LF + HF). Finally they have achieved the best result with 87.3% accuracy. Shen et al. [25] have proposed MultiScale Dilation Attention 1-D CNN (MSDA-1DCNN) for extracting features from the RRI and have applied Weighted-Loss Time-Dependent (WLTD) classification model for OSA detection and have achieved 89.4% accuracy on the PhysioNet Apnea-ECG database [12].

Chang et al. [26] have proposed a novel 1-D CNN architecture for the purpose of OSA detection. In their work, each one-minute segment of the raw ECG signal is initially undergone through the band pass filtering followed by Z-score normalization before fitted into the CNN model. Overall, they have achieved 87.9% accuracy on the PhysioNet Apnea-ECG database [12] whereas, the performance has increased up to 97.1% in the case of pre-recorded samples. Thompson et al. [27] have proposed a 1-D CNN architecture including a convolution layer, a max pooling layer, a fully connected MLP and a softmax output layer. In their work, they've applied a windowing strategy, with window sizes of 500, 1000, 1500, 2000 and 2500 for validation of their model, which achieved 93.77% accuracy for window size of 500 on the PhysioNet Apnea-ECG database [12]. Mashrur et al. [28] have proposed a novel Scalogram-based CNN to detect OSA using ECG signals. In their work, they've obtained hybrid scalograms from the ECG signals using continuous wavelet transform (CWT) and empirical mode decomposition (EMD). They train a CNN model on these scalograms to extract deep features to detect OSA, achieving an accuracy of 94.30% on the PhysioNet Apnea-ECG database [12].

The majority of the previous works have considered ECG, and this fact motivates us to choose ECG signal data for conducting our work. PhysioNet Apnea-ECG database is also a popular one for working on OSA detection. We have chosen deep learning models for our work as they are very much applicable to the time-series data [6]. However, only raw samples cannot produce outstanding results when fit into CNN models as discussed in the Results and Discussion section, hence it requires some pre-processing. Since our main concern is about the ensemble approaches in apnea detection domain, some of the established works based on ensemble techniques are also discussed.

Faußer et al. [29] have applied Temporal Difference (TD) and Residual-Gradient (RG) update methods on a given set of agents with their own nonlinear function approximator, for instance, an MLP to adapt the weights to learn from joint decisions, such as Majority Voting and Averaging of the state-values. Also, Glodek et al. [30] have worked on ensemble

approaches for density estimation using Gaussian Mixture Models (GMMs) by combining individual mixture models incorporating a high diversity to create a more stable and accurate model. Chakraborty et al. [31] have performed ensemble of filter methods, such as optimal subsets of features using filter methods Mutual Information (MI), Chi-square, and Anova F-Test, and with the selected features building learning models using MLP based classifier.

Kächele et al. [32] have used an ensemble of RF, Radial Basis Function (RBF) networks to determine the intensity of pain based on the video shown and body features such as ECG, Electromyography (EMG). They have used MLP to train from the classification scores obtained from individual base models for score level fusion. Dey et al. [33] have used a weighted ensemble of three CNN based models: ADNet, IRCNN and DnCNN to remove white Gaussian noise from an image. The aforementioned three models' outputs are aggregated in the ratio 2:3:6 respectively. Bellmann et al. [34] have applied various fusion approaches for the Multi-Classifer System (MCS) to effectively measure pain intensity levels. Their case study includes one of the most popular fusion techniques—bagging and boosting. Kundu et al. [35] have proposed a fuzzy-rank based classifier fusion based approach which uses the Gompertz function for determining the fuzzy-ranks of the base classifiers. They have conducted the experiments on the SARS-COV-2 [36] and Harvard Dataverse [37] datasets for diagnosing COVID-19 from the CT-scans and have achieved the best results with 98.93% and 98.80% accuracies respectively by using the ensemble of the pre-trained models VGG-11, Wide ResNet-50-2 and Inception v3.

All these previously established works prove that the application of ensemble is spread out through many research fields. The huge success and scope of research in classifier fusion are the main reasons for its popularity. Still based on our knowledge, any ensemble technique based work has not been conducted for apnea diagnosis till now. This has motivated us to conduct experimental studies base on ensemble techniques on the OSA detection domain. Additionally, we have chosen the three deep learning models—(i) Wang et al.'s [7] proposed CNN model, (ii) Sharan et al.'s [8] proposed CNN model, (iii) Almutairi et al.'s [10] proposed CNN-LSTM model as base models. The reason for such choice is these three models are all CNN based which are robust, excellent classifiers in general. The fact that the chosen three models have previously been used for OSA detection further encourages us to work with them. Thus, we have conducted our work by applying an ensemble of CNN based architectures in the popular PhysioNet Apnea-ECG database [12].

3. Materials and Methods

3.1. Datasets Used

We chose the PhysioNet Apnea-ECG database [12] for conducting all the experiments. A total of 70 records sampled at 100 Hz frequency were present in the database out of which 35 records were in the training set and the rest belonged to the testing set. People belonging to the 27–30 year age group volunteered in the data collection process. The data collection procedure lasted approximately 7–10 h per subject.

After the segmentation process, train data and test data had 15,961 and 15,938 pre-processed noiseless 1-min samples respectively. The training set had 9832 samples of apnea class and 6129 samples of normal class respectively, and the test set had 9838 samples of apnea class and 6100 samples of normal class respectively.

3.2. Methodology

3.2.1. Pre-Processing

We performed the steps to convert the raw ECG signal data into 2-D matrices in this stage by following the pre-processing approach according to the GitHub link (<https://github.com/zzklove3344/ApneaECGAnalysis>, accessed on: 6 August 2021, accessed from: Kolkata, India). So, the pre-processing consists of the following steps:

- First, the raw data were transformed into segments having a time-span of 1 min. Thus, each segment had a size of 6000×1 .
- We further divided each 1 min sample by dividing it into 240 parts so that each division contained 25 consecutive samples. Next, we extracted the features from each sub-division.
- We applied the Multilevel Teager Energy Operator (MTEO) algorithm by using the Matlab ToolBox BioSigKit to evaluate the R peaks, ECG Derived Respiration signal (EDR) from the raw segments. MTEO algorithm was used for action potential detection to locate the QRS complexes in Electromyography signals [38].
- From the R peaks, we further derived (i) RRI—the interval between two consecutive R peaks, (ii) R peak Amplitude (RAMP). Thus, three features were derived from each segment. Based on RRI, Each ECG segment is classified as (i) noise, (ii) clear.
- RRI and RAMP features went through the Smoothing and Spline interpolation processes. We also downsampled the EDR signal values.
- After all these processes, the clear segments were present as 2-D matrices having the size of 240×3 .
- We normalized each feature by applying Z-score normalization. The formula of Z-score normalization was: $\hat{X} = \frac{(X - \mu_X)}{\sigma_X}$ where, X is the original feature column, \hat{X} is the normalized feature column and μ_X, σ_X are the mean and standard deviation of the feature values in X .

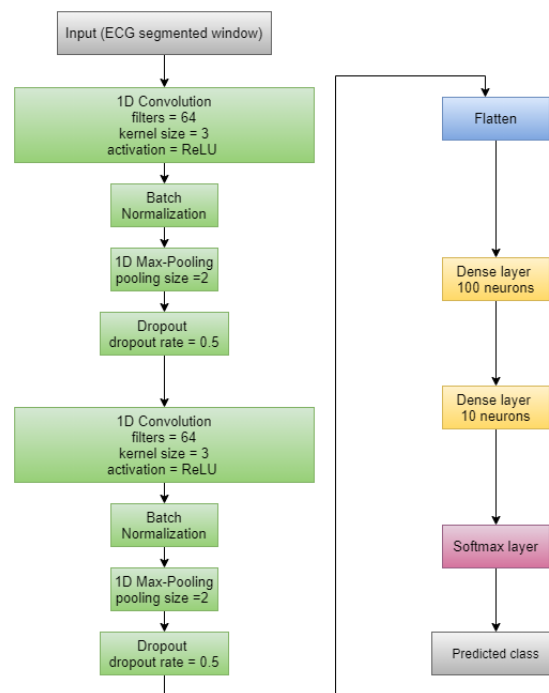
To perform five-fold cross-validation, we combined the pre-processed training and test data, accumulating to a total of 31,899 samples. Then, the combined dataset underwent shuffling before splitting for the cross-validation. For the five-fold cross-validation, the overall dataset was divided into five parts where five parts were used for training and the remaining part is used for testing purposes, over all the combinations.

3.2.2. Models Used

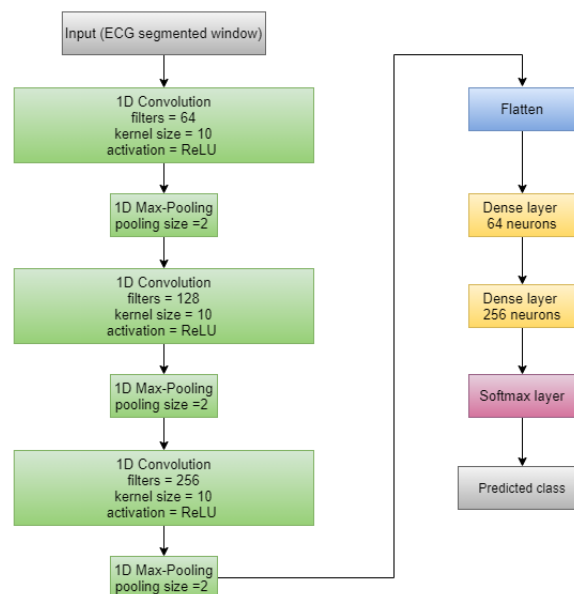
In this work, we used three existing CNN based models—(i) Wang et al.'s [7] proposed CNN architecture, (ii) Sharan et al.'s [8] proposed CNN model, (iii) Almutairi et al.'s [10] proposed CNN-LSTM model for making the initial predictions. These models have previously been proposed in this very domain. However, we made some minor changes to the original architecture.

1. Wang et al.'s [7] CNN: We replaced all the 2-D layers of the original architecture with 1-D layers for the convenience of handling the time series ECG signal data. As we considered only three features for each sample, 2-D CNN models were not applicable on the pre-processed data. At first, we used two CNN blocks consisting of a 1-D Convolution layer with a kernel size of 3, 64 filters, Rectified Linear Unit (ReLU) as activation function followed by Batch-Normalization layer and a 1-D Max Pooling layer of size 2. Next, a Flatten layer followed by two Dense layers with 100 and 10 neurons were applied to the output produced by the final CNN block. Finally, the probability for each class was calculated using the Softmax layer. The architecture of this model is shown in Figure 2a.
2. Sharan et al.'s [8] CNN: The model consisted of three CNN blocks having a 1-D Convolution layer with kernel size 10 with ReLU as activation function followed by a 1-D Max Pooling layer with a pooling size of 2. Each of these three blocks differed in the number of filters in the 1-D Convolution layer (64, 128 and 256). Next, two Dense layers with 64 and 256 neurons were applied. Finally, the Softmax layer was used for calculating the probabilities of each class. Figure 2b shows the whole structure of this model.
3. Almutairi et al.'s [10] CNN-LSTM: The proposed model consisted of three 1D-Convolution blocks each having a 1D-Convolution layer with kernel size 3 with ReLU activation function followed by a Batch-Normalization layer, 1D-Max Pooling layer of pooling size = 2 and a Dropout layer having a dropout rate of 0.2. These three blocks differed only in the number of filters applied on the 1-D Convolution layer

(64, 128 and 16). A Flattened layer followed by an LSTM layer having an output size of 64 was applied after the CNN blocks. Next, a Dense layer having 64 neurons was used. To be applicable for the CNN-LSTM architecture, we initially divided each window into four consecutive sub-windows and then have applied the CNN layers on each of the sub-windows. The LSTM layer worked on all four sub-windows collectively. The final output i.e., the probability for each class was given by the Softmax layer. We provided Figure 3a to represent the whole architecture of the proposed CNN-LSTM model whereas Figure 3b was provided to show the CNN-only part separately.



(a)



(b)

Figure 2. Architectures of the CNN models originally proposed by (a) Wang et al. [7] and (b) Sharan et al. [8] respectively.

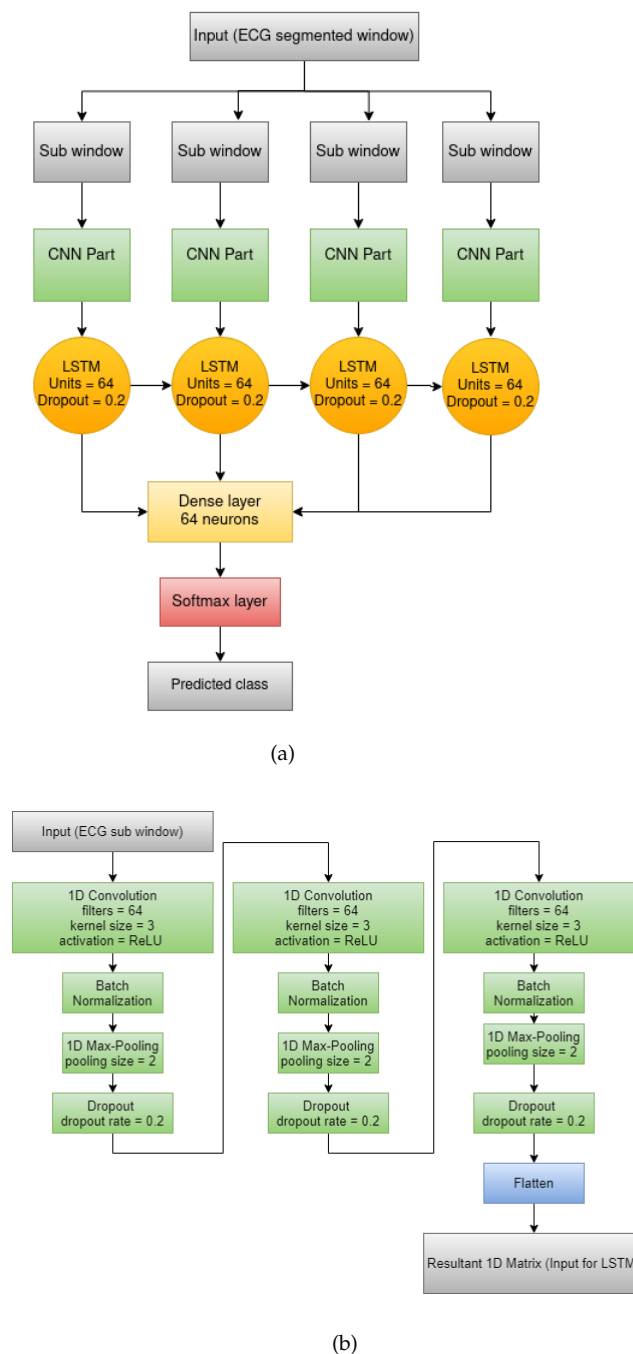


Figure 3. Almutairi et al.'s [10] proposed CNN-LSTM model: (a) overall architecture, (b) architecture of the CNN part present in the overall structure.

3.3. Applied Ensemble Methods

For the later parts of the experiment, we used different ensembles approaches to further increase the overall classification accuracy. Ensemble approach could be classified into two categories,

1. Decision level fusion: Here the predicted class i.e., only decision was considered for each model.
2. Score level fusion: Here the prediction score for each of the class was considered for each model.

We applied one decision level ensemble— Majority Voting and three score level ensemble procedures, namely Sum Rule, Choquet integral based fuzzy fusion and trainable MLP en-

semble. Let us assume there are a total of k classes c_1, c_2, \dots, c_k and l classifiers m_1, m_2, \dots, m_l . Model m_j predicts for a sample x being in class c_i with probability p_{ij} . So, the prediction class of that sample given by model m_j will be $\hat{c}_j = \operatorname{argmax}_i(p_{ij})$.

The aforementioned ensemble techniques could also be divided into the following two categories based on whether training was performed for ensemble learning or not:

1. Non-trainable ensemble: In this case, we applied some pre-defined rules to aggregate the decisions/scores made by all the base models. majority voting, Sum rule and Choquet integral based fuzzy fusion were all such ensemble approaches used in this work.
2. Trainable ensemble: In this case, we used another classifier to perform the aggregation of all base models' scores. All class scores given by all the base models were flattened into a single feature vector sample-wise. Next, we chose an appropriate classifier that produced good results when train and test performed against the flattened train and test data scores. In our work, we chose an MLP for the aggregation task. Figure 4 represents the working procedure of a trainable ensemble method.

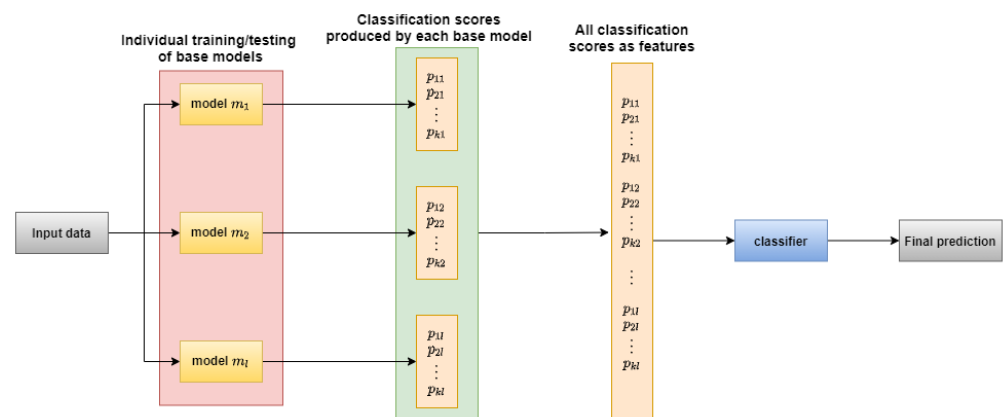


Figure 4. Basic architecture of a trainable ensemble model.

The following paragraphs contain a brief discussion about all of the ensemble approaches used in our work.

1. Majority voting: In this kind of ensemble, the class predicted by maximum number of models is assigned as the final prediction. So, the final class would be $\operatorname{argmax}_i(\operatorname{freq}(c_i))$, where, $\operatorname{freq}(c_i)$ is the number of the number of models predicted class c_i .
2. Sum rule: The class with maximum sum of scores of all classifiers is the final predicted class. So the final prediction class would be $\operatorname{argmax}_i(\sum_j p_{ij})$.
3. Choquet integral based fuzzy fusion: Fuzzy integrals are generally aggregation operators which combine the information in this case, confidence values of all sources and their all possible combinations. Each source i.e., classifier has been given initial weights called fuzzy measures.

Fuzzy measure of set X is a function $g : P(X) \rightarrow [0, 1]$ where $P(X)$ is the power-set of X which holds the following two conditions [39]:

- (a) The boundary of g should follow: $g(\emptyset) = 0$ and $g(X) = 1$.
- (b) Suppose A, B are two subsets of X and $A \subseteq B$ then the $g(A) \leq g(B)$ must be satisfied.

In this case, the set X becomes the set of all classifiers i.e., $X = \{m_1, m_2, \dots, m_l\}$.

Like Pacheco et al.'s [40] work, we used entropy of each classifier's probability vector to evaluate the fuzzy measure. The formula of entropy of probability vector given by classifier m_j , E_j is given as:

$$E_j = \sum_i p_{ij} \cdot \log(p_{ij}) \quad (1)$$

From the function in Equation (1), we understand that entropy becomes less when the probability for one class is much larger than that of other classes. So, it would be inverse to the goodness of a classifier. Hence, we used subtracted entropy value from the maximum possible entropy value (1 in this case) to get the fuzzy measure for each classifier. Finally, each of the fuzzy measure was divided by the sum of all fuzzy measure values. Suppose g_1, g_2, \dots, g_l are the fuzzy-measures provided for the classifiers m_1, m_2, \dots, m_l respectively which means-

$$g_j = \frac{1 - E_j}{\sum_i (1 - E_i)} \quad (2)$$

where, E_j is the entropy of the probability vector provided by classifier m_j evaluated according to Equation (1).

Tahani et al. [41] have introduced the concept of Sugeno λ -measure effective for fuzzy measures. It holds the additional property: if $A \cap B = \phi$ then there exists that $\lambda > -1$ such that:

$$g(A \cup B) = g(A) + g(B) - \lambda \cdot g(A) \cdot g(B) \quad (3)$$

where, A and B are both subsets of X . Considering the additional property, λ can be evaluated as the solution for the equation:

$$\lambda + 1 = \prod_j (\lambda \cdot g(\{m_j\}) + 1) \quad (4)$$

where, $\{m_j\}$ is the singleton set containing m_j . So, after evaluating the fuzzy measures for all singleton sets, we have evaluated the fuzzy measure for the combination of classifiers using Sugeno λ -measure.

The Choquet integral [42] is used for aggregating the scores based on all combination of classifiers. Based on the fuzzy measures, this integral can even combine the empirical strategies like addition, multiplication of scores produced by the classifiers. Choquet integral for class c_i can be evaluated according to the following formula:

$$C_g(X) = \sum_j s_{\pi_j} \cdot [g(A_{\pi_j}) - g(A_{\pi_{j-1}})] \quad (5)$$

where, $s_j = p_{ij} \forall j = 1, 2, \dots, l$ and the set of classifiers X is permuted such that $s_{\pi_1} \geq s_{\pi_2} \geq \dots \geq s_{\pi_l}$. $[g(A_j) - g(A_{j-1})]$ depicts the relative importance of the classifier m_j .

So all the class-scores CS_i are obtained from Equation (5) and class with the maximum score will be the final predicted class which is $\text{argmax}_i(CS_i)$.

4. Trainable ensemble using MLP: The main building blocks of a Neural Network (NN) [43] are nodes. These nodes usually remain collectively as layers. Information in NN passes from layer to layer. In the Feed-forward Neural Network (FNN) [44], the flow of information is fixed in one direction, i.e., from the input layer to the output layer.

Suppose there are n nodes in the previous layer where each node i forwards the value x_i to a particular node j in the current layer. Then, the output y of the node j will be-

$$y = \phi\left(\sum_i (w_i \cdot x_i)\right) \quad (6)$$

where ϕ is the activation function present in the current layer and w_i be the weight assigned to the path from node i to node j . The main objective of FNNs is to optimize these w_i s.

In any MLP [45] architecture, the FNN must have at least one hidden layer between the input and the output layer. In our work, we considered a simple MLP architecture with only one hidden layer having 16 features. The MLP accepted a total of $k.l$ score

features as inputs and returned k score values for all classes so that the deep learning network could be used as an aggregator for the ensemble model. Figure 5 provides the architecture of the MLP used as an ensemble to our base models.

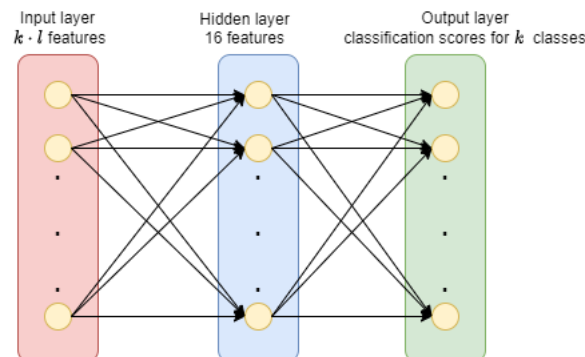


Figure 5. Architecture of MLP used for ensemble where k denotes total number of classes and l denotes total number of base-classifiers.

4. Results and Discussion

In the present work, we used five classification measures—(i) accuracy, (ii) precision, (iii) recall, (iv) F1-score, (v) specificity to evaluate the performance of the base models and their ensemble. Since our only concern was binary classification, we have depicted all the measures as if there were two classes—(i) positive class, (ii) negative class present. Naturally, any classifier would also give the prediction class as either of the two. When the predicted class of a sample matched with its actual class then it was said to be True otherwise False. Thus we defined the five chosen classification metrics based on the terms True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) as follows:

1. Accuracy: It is defined as the ratio of number of correctly classified samples to that of total samples.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (7)$$

2. Precision: Precision of a class is defined as the ratio of correctly classified samples to total number of samples predicted as the given class.

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

3. Recall: Recall of a class is defined as the ratio of correctly classified samples to total number of samples actually belonging to that class.

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

4. F1-score: Sometimes, only Precision and Recall are not enough to measure the performance of a classifier. So, F1-score is presented to combine the both aspects as it is evaluated as the harmonic mean of the two.

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (10)$$

5. Specificity: Specificity is used to measure the proportion of negatives that are correctly identified. It is defined as the ratio of true negatives predicted to total number of samples which belong to negative class.

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (11)$$

Since the current problem consists of only two classes, we used binary cross-entropy as the loss function. We purposefully applied Adam optimizer to optimize the loss value throughout 100 epochs. The training procedure was performed batches with each batch having simultaneously 64 samples. We can observe the change in training accuracy and loss with epochs for all three models in Figure 6a–c.

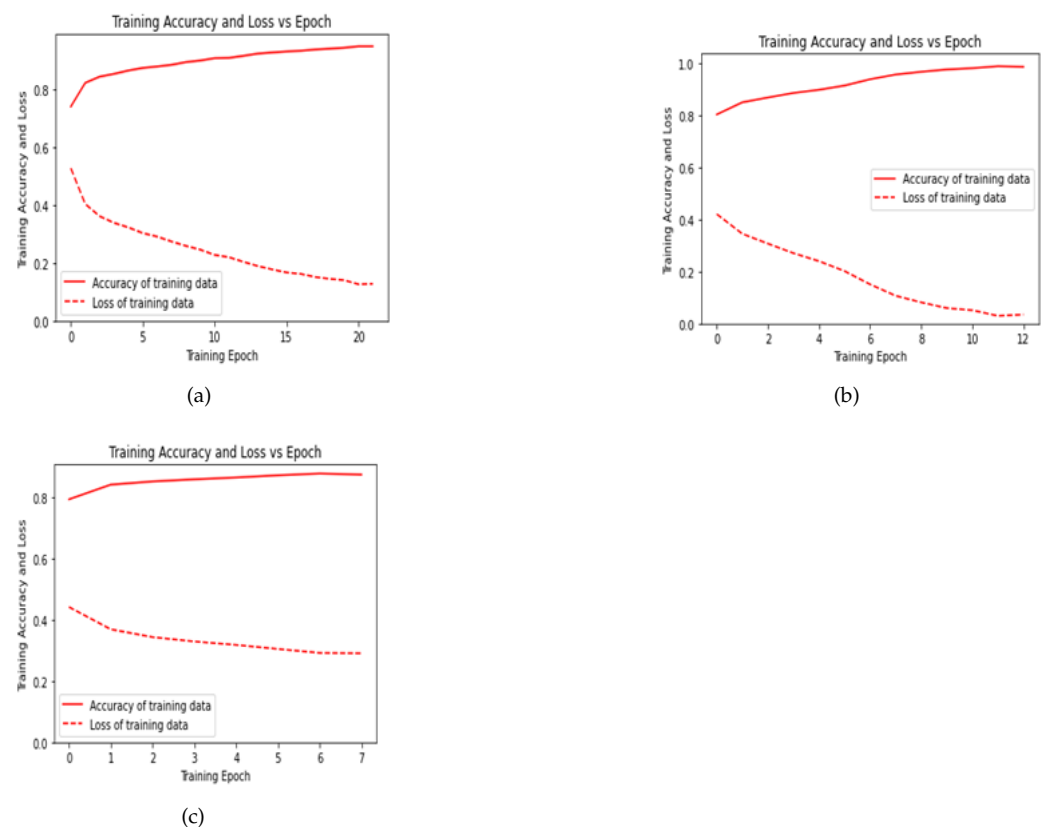


Figure 6. Graphical representation of training loss and accuracy vs epoch for: (a) Wang et al.'s [7] proposed CNN model, (b) Sharan et al.'s [8] proposed CNN model, (c) Almutairi et al.'s [10] proposed CNN-LSTM model.

After training, all the four classification measures were evaluated based on the test data and their prediction for each case in Table 1.

Table 1 suggests that the three chosen CNN based models were compatible for ensemble as each ensemble technique successfully increased the maximum accuracy of all three models by at least 1%. Majority voting gave least accurate results because it only considered the prediction instead of the exact probabilistic values whereas, the other three being score level fusion were able to produce somewhat better results. Trainable ensemble technique performed a little better than the non-trainable ensemble techniques probably due to the fact that weight assigning to the classification scores was performed with the help of a classifier instead of applying a pre-defined weight allocation rule. Besides, MLP itself worked as an excellent classifier because of its utilization of additional hidden features [46]. Thus, it was able to identify the patterns of classification scores as well. Among the non-trainable ensemble techniques, Choquet integral fusion worked better than sum

rule because unlike sum rule, Choquet integral fusion did not assign equal weights to all three. Giving equal importance to all base models' scores may not meet the expectations as the poor performance of an individual performance may affect the overall result. On the other hand, Choquet integral fusion assigned more weight to the model which gave more confident predictions. Among individual models, CNN-LSTM performed better than the rest two base models because LSTM considers the contexts (i.e., previous samples) along with the present sample which was beneficial for any time-series data such as, ECG signals.

Table 1. Classification performances obtained by all the base models and their ensembles.

Model/Ensemble Technique	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)
Wang et al.'s CNN [7]	82.36	81.54	81.78	81.65	83.93
Sharan et al.'s CNN [8]	83.29	82.32	82.80	82.53	87.39
Almutairi et al.'s CNN-LSTM [10]	84.08	83.59	82.94	83.22	86.15
Ensemble—Majority Voting	85.18	84.59	84.40	84.48	87.56
Ensemble—Sum rule	85.34	84.61	84.45	84.52	87.63
Ensemble—Choquet Integral	85.41	84.63	84.48	84.53	87.89
Trainable ensemble using MLP	85.58	84.80	84.43	84.67	88.26

We also performed experiments on the raw 1-min signal windows by taking the Table 1's winner MLP based trainable ensemble for a performance-wise understanding between the raw data and the feature extracted data. Table 2 contains the results for raw ECG segments, which clearly shows that features extracted from the signal greatly outperformed the raw data as the final prediction made by the ensemble with raw data was only 70.77% accurate. The possible explanation for such an outcome was that classifiers may understand the pattern more efficiently in case of certain features which summarized the raw data.

Table 2. Classification performances obtained by considering the raw ECG segments of 1-min time-span.

Model/Ensemble Technique	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)
Wang et al.'s CNN [7]	65.00	63.52	63.99	63.64	66.34
Sharan et al.'s CNN [8]	66.95	65.50	66.01	65.64	68.03
Almutairi et al.'s CNN-LSTM [10]	70.54	68.71	68.38	68.47	71.87
Trainable ensemble using MLP	70.77	69.38	69.96	69.57	72.19

Since, with this amount of data there was a possibility that the distribution of train and test sets may not be uniform, we applied two-fold cross-validation approach by swapping the train set and test set, and taking the average of both the results. The results of the base models and ensembles after this two-fold cross-validation are shown in Table 3.

We also performed five-fold cross-validation over the combined dataset of the train and test sets. The performances of the base models and ensembles after five-fold cross-validation are shown in Table 4.

Table 3. Classification performances obtained by all the base models and their ensemble using two-fold cross-validation.

Model/Ensemble Technique	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)
Wang et al.'s CNN [7]	81.93	78.11	76.40	76.30	81.56
Sharan et al.'s CNN [8]	82.26	79.88	74.41	76.10	84.82
Almutairi et al.'s CNN-LSTM [10]	83.82	80.26	77.79	78.64	84.51
Trainable ensemble using MLP	84.60	82.65	76.88	79.26	86.19

Table 4. Classification performances obtained by all the base models and their ensemble using five-fold cross-validation.

Model/Ensemble Technique	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)
Wang et al.'s CNN [7]	85.07	80.26	82.33	80.74	85.98
Sharan et al.'s CNN [8]	86.31	84.26	81.82	81.99	87.44
Almutairi et al.'s CNN-LSTM [10]	86.88	82.60	83.69	83.00	88.12
Trainable ensemble using MLP	87.91	84.56	84.42	84.25	88.98

From Table 4, we observe that the models and the MLP based ensemble delivered better results during five-fold cross-validation than without doing so, because in five-fold cross validation, every sample from the dataset was there in both training data and test sets at least once, and the amount of training data increased which also included some of the test data used previously. This resulted in better identification of the test data.

Furthermore, we also used standard classifiers such as SVM, ANN with a hidden layer with 100 features and Random Forest to compare how they performed with the trainable ensemble using MLP. We flattened the features for these classifiers and after prediction, reshaped the outputs for the two classes respectively before performing the ensemble on them. The results obtained by the standard classifiers and the ensemble are shown in Table 5.

Table 5. Classification performances obtained by standard classifiers and their ensemble.

Model/Ensemble Technique	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)
SVM	81.64	81.33	91.33	86.00	83.65
ANN	77.48	80.85	83.29	82.01	79.17
Random Forest	78.47	78.08	90.82	83.87	79.59
Trainable ensemble using MLP	81.83	81.70	67.90	74.08	83.09

Table 5 shows that the simple machine learning classifiers performed somewhat worse than the deep learning based models. So, overall ensemble was also affected by the choice of base models. Next, we compared the best performance achieved in our work for the original train and test dataset with some of the previous methods' performance in Table 6.

From Table 6, we observe that the MLP based ensemble delivered better results than some of the previous works. Still Chang et al. [26] and Shen et al. [25] have obtained better results in their respective works. Although the ensemble worked fine for the given combination of models, their individual accuracy could not exceed 84%. So, this worked as a limiting factor for obtaining the better performance by the overall architecture. Additionally, the class-imbalance may be the reason for achieving not so high accuracy. Still, the current work held a good place among all the existing works experimented on the dataset under consideration.

Table 6. Performance comparison of the previous works with our work.

Method	Features Used	Accuracy (%)	Recall (%)	Specificity (%)	Sample Window Size
Tripathi et al. [47]—Kernel Extreme Learning Machine (KELM)	EDR and HRV	76.37	78.02	74.64	-
Hassan et al. [22]—Statistical features with Extreme Learning Machine (ELM)	TQWT extracted features from Raw ECG Signal data	83.77	81.99	90.72	-
Feng et al. [48]—Feature Extraction with Deep Learning Network, SVM with Hidden Markov Model (HMM)	RRI	84.7	68.8	94.5	6000 × 1
Chang et al. [26]—1-D CNN	Raw ECG signal data	87.9	81.1	92.0	6000 × 1
Shen et al. [25]—MSDA-1DCNN	RRI then further features from MSDA-1DCNN	89.4	89.8	89.1	180 × 1
Current study—Trainable Ensemble using MLP	RRI, RAMP and EDR	85.58	84.43	88.26	240 × 3

5. Conclusions

Proper diagnosis of any severe disease such as OSA is very crucial which further emphasises the application of AI based classification techniques on such kind of data. Outcome of the competent classification models may be helpful even to the medical professionals. For example, in our case, it can help differentiating an Apnetic patient from a non-Apnetic one. In this paper, we have applied four ensemble techniques on deep learning models to detect sleep apnea based on ECG signal data. Although, each ensemble model can predict with at least 85% accurate results on PhysioNet Apnea-ECG database, still some improvements can be applied further. Since our primary focus remains on the ensemble approaches in the OSA detection domain, our experiments show that the deep learning models are compatible to be aggregated as the final ensemble method is able to surpass the best among three models. However, some modifications like considering varied LSTM based structures, attention mechanism can be thought of to the increase the base model's performance. In future, we would consider more ensemble methods such that we can logically assign weights to fuse the classifiers' decisions. We can also use different features from the raw data based on proper feature relevance techniques as well. From the statistics of the database used here, it is evident that the number of samples is lower than the required for any deep learning network. To overcome this problem, data augmentation techniques can be applied to produce artificial data from the original dataset.

Author Contributions: Conceptualization and methodology, R.S. and D.M.; writing code, D.M. and K.D.; validation, R.S. and F.S.; writing original manuscript, D.M., K.D. and R.S.; editing manuscript, F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No datasets are generated during the current study. The datasets analyzed during this work are publicly available.

Acknowledgments: This work was supported by Centre for Microprocessor Application for Training, Education and Research (CMATER) under the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

OSA	Obstructive Sleep Apnea
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
IoT	Internet of Things
SAHS	Sleep Apnea HypoApnea Syndrome
ECG	Electrocardiogram
SVM	Support Vector Machine
kNN	k-Nearest Neighbours
RF	Random Forest
RNN	Recurrent Neural Network
RQA	Recurrence Quantification Analysis
HRV	Heart Rate Variability
ANN	Artificial Neural Network
TQWT	Turnable-Q factor Wavelet Transform
AdaBoost	Adaptive Boosting
MTEO	Multilevel Teager Energy Operator
MLP	Multi-Layer Perceptron
EDR	ECG Derived Respiration
RRI	RR Interval
RAMP	R peak Amplitude
ReLU	Rectified Linear Unit
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
KELM	Kernel Extreme Learning Machine
ELM	Extreme Learning Machine
HMM	Hidden Markov Model
NN	Neural Network
FNN	Feedforward Neural Network
TD	Temporal Difference
RG	Residual-Gradient
MI	Mutual Information
GMM	Gaussian Mixture Model
RBF	Radial Basis Function
EMG	Electromyography
MCS	Multi-Classifer System

References

1. Yin Y.; Zeng, Y.; Chen, X.; Fan, Y. The internet of things in healthcare: An overview. *J. Ind. Inf. Integr.* **2016**, *1*, 3–13. [[CrossRef](#)]
2. Xie, B.; Minn, H. Real-time sleep apnea detection by classifier combination. *IEEE Trans. Inf. Technol. Biomed.* **2012**, *16*, 469–477. [[CrossRef](#)]
3. Punjabi, N.M. The epidemiology of adult obstructive sleep apnea. *Proc. Am. Thorac. Soc.* **2008**, *5*, 136–143. [[CrossRef](#)]
4. Al Lawati, N.M.; Patel, S.R.; Ayas, N.T. Epidemiology, risk factors, and consequences of obstructive sleep apnea and short sleep duration. *Prog. Cardiovasc. Dis.* **2009**, *51*, 285–293. [[CrossRef](#)]
5. Almazaydeh, L.; Elleithy, K.; Faezipour, M. Detection of obstructive sleep apnea through ECG signal features. In Proceedings of the 2012 IEEE International Conference on Electro/Information Technology, Indianapolis, IN, USA, 6–8 May 2012; pp. 1–6.
6. Zhao, B.; Lu, H.; Chen, S.; Liu, J.; Wu, D. Convolutional neural networks for time series classification. *J. Syst. Eng. Electron.* **2017**, *28*, 162–169. [[CrossRef](#)]
7. Wang, X.; Cheng, M.; Wang, Y.; Liu, S.; Tian, Z.; Jiang, F.; Zhang, H. Obstructive sleep apnea detection using ecg-sensor with convolutional neural networks. *Multimed. Tools Appl.* **2020**, *79*, 15813–15827. [[CrossRef](#)]
8. Sharan, R.V.; Berkovsky, S.; Xiong, H.; Coiera, E. ECG-Derived Heart Rate Variability Interpolation and 1-D Convolutional Neural Networks for Detecting Sleep Apnea. In Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 637–640.
9. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]

10. Almutairi, H.; Hassan, G.M.; Datta, A. Detection of Obstructive Sleep Apnoea by ECG signals using Deep Learning Architectures. In Proceedings of the 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, The Netherlands, 18–21 January 2021; pp. 1382–1386.
11. Dietterich, T.G. Ensemble methods in machine learning. In *Lecture Notes in Computer Science, Proceedings of the International Workshop on Multiple Classifier Systems, Cagliari, Italy, 21–23 June 2000*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 1–15.
12. Penzel, T.; Moody, G.B.; Mark, R.G.; Goldberger, A.L.; Peter, J.H. The apnea-ECG database. In Proceedings of the Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163), Cambridge, MA, USA, 24–27 September 2000; pp. 255–258.
13. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
14. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. KNN model-based approach in classification. In *Lecture Notes in Computer Science, Proceedings of the OTM Confederated International Conferences “On the Move to Meaningful Internet Systems”, Catania, Italy, 3–7 November 2003*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 986–996.
15. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
16. Ng, A.S.; Chung, J.W.; Gohel, M.D.; Yu, W.W.; Fan, K.L.; Wong, T.K. Evaluation of the performance of using mean absolute amplitude analysis of thoracic and abdominal signals for immediate indication of sleep apnoea events. *J. Clin. Nurs.* **2008**, *17*, 2360–2366. [[CrossRef](#)] [[PubMed](#)]
17. Alvarez, D.; Hornero, R.; Abásolo, D.; Del Campo, F.; Zamarrón, C. Nonlinear characteristics of blood oxygen saturation from nocturnal oximetry for obstructive sleep apnoea detection. *Physiol. Meas.* **2006**, *27*, 399. [[CrossRef](#)]
18. Qin, H.; Keenan, B.T.; Mazzotti, D.R.; Vaquerizo-Villar, F.; Kraemer, J.F.; Wessel, N.; Tufik, S.; Bittencourt, L.; Cistulli, P.A.; de Chazal, P.; et al. Heart rate variability during wakefulness as a marker of obstructive sleep apnea severity. *Sleep* **2021**, *44*, zsab018. [[CrossRef](#)]
19. Cheng, M.; Sori, W.J.; Jiang, F.; Khan, A.; Liu, S. Recurrent neural network based classification of ECG signal features for obstruction of sleep apnea detection. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 21–24 July 2017; Volume 2, pp. 199–202.
20. Marhon, S.A.; Cameron, C.J.; Kremer, S.C. Recurrent neural networks. In *Handbook on Neural Information Processing*; Springer: Berlin/Heidelberg, Germany 2013; pp. 29–65.
21. Nguyen, H.D.; Wilkins, B.A.; Cheng, Q.; Benjamin, B.A. An online sleep apnea detection method based on recurrence quantification analysis. *IEEE J. Biomed. Health Inform.* **2013**, *18*, 1285–1293. [[CrossRef](#)]
22. Hassan, A.R. Computer-aided obstructive sleep apnea detection using normal inverse Gaussian parameters and adaptive boosting. *Biomed. Signal Process. Control* **2016**, *29*, 22–30. [[CrossRef](#)]
23. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [[CrossRef](#)]
24. Wang, T.; Lu, C.; Shen, G. Detection of sleep apnea from single-lead ECG signal using a time window artificial neural network. *BioMed Res. Int.* **2019**, *2019*. [[CrossRef](#)] [[PubMed](#)]
25. Shen, Q.; Qin, H.; Wei, K.; Liu, G. Multiscale Deep Neural Network for Obstructive Sleep Apnea Detection Using RR Interval From Single-Lead ECG Signal. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13.
26. Chang, H.Y.; Yeh, C.Y.; Lee, C.T.; Lin, C.C. A sleep apnea detection system based on a one-dimensional deep convolution neural network model using single-lead electrocardiogram. *Sensors* **2020**, *20*, 4157. [[CrossRef](#)]
27. Thompson, S.; Fergus, P.; Chalmers, C.; Reilly, D. Detection of Obstructive Sleep Apnoea Using Features Extracted from Segmented Time-Series ECG Signals Using a One Dimensional Convolutional Neural Network. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
28. Mashrur, F.R.; Islam, M.S.; Saha, D.K.; Islam, S.R.; Moni, M.A. SCNN: Scalogram-based Convolutional Neural Network to Detect Obstructive Sleep Apnea using Single-lead Electrocardiogram Signals. *Comput. Biol. Med.* **2021**, *134*, 104532. [[CrossRef](#)] [[PubMed](#)]
29. Faußer, S.; Schwenker, F. Ensemble methods for reinforcement learning with function approximation. In *Lecture Notes in Computer Science, Proceedings of the International Workshop on Multiple Classifier Systems, Naples, Italy, 15–17 June 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 56–65.
30. Glodek, M.; Schels, M.; Schwenker, F. Ensemble Gaussian mixture models for probability density estimation. *Comput. Stat.* **2013**, *28*, 127–138. [[CrossRef](#)]
31. Chakraborty, A.; De, R.; Chatterjee, A.; Schwenker, F.; Sarkar, R. Filter Method Ensemble with Neural Networks. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019*; Springer: Cham, Switzerland, 2019; pp. 755–765.
32. Kächele, M.; Amirian, M.; Thiam, P.; Werner, P.; Walter, S.; Palm, G.; Schwenker, F. Adaptive confidence learning for the personalization of pain intensity estimation systems. *Evol. Syst.* **2017**, *8*, 71–83. [[CrossRef](#)]
33. Dey, S.; Bhattacharya, R.; Schwenker, F.; Sarkar, R. Median Filter Aided CNN Based Image Denoising: An Ensemble Approach. *Algorithms* **2021**, *14*, 109. [[CrossRef](#)]
34. Bellmann, P.; Thiam, P.; Schwenker, F. Multi-classifier-systems: Architectures, algorithms and applications. In *Computational Intelligence for Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 83–113.
35. Kundu, R.; Basak, H.; Singh, P.K.; Ahmadian, A.; Ferrara, M.; Sarkar, R. Fuzzy rank-based fusion of CNN models using Gompertz function for screening COVID-19 CT-scans. *Sci. Rep.* **2021**, *11*, 1–12. [[CrossRef](#)] [[PubMed](#)]

36. Angelov, P.; Almeida Soares, E. SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification. *MedRxiv* **2020**. [[CrossRef](#)]
37. Soares, E.; Angelov, P. A large dataset of real patients CT scans for COVID-19 identification. *Harv. Dataverse* **2020**, *1*. [[CrossRef](#)]
38. Sedghamiz, H.; Santonocito, D. Unsupervised detection and classification of motor unit action potentials in intramuscular electromyography signals. In Proceedings of the 2015 E-Health and Bioengineering Conference (EHB), Iasi, Romania, 19–21 November 2015; pp. 1–6.
39. Murofushi, T.; Sugeno, M. An interpretation of fuzzy measures and the Choquet integral as an integral with respect to a fuzzy measure. *Fuzzy Sets Syst.* **1989**, *29*, 201–227. [[CrossRef](#)]
40. Pacheco, A.G.; Krohling, R.A. Aggregation of neural classifiers using Choquet integral with respect to a fuzzy measure. *Neurocomputing* **2018**, *292*, 151–164. [[CrossRef](#)]
41. Tahani, H.; Keller, J.M. Information fusion in computer vision using the fuzzy integral. *IEEE Trans. Syst. Man Cybern.* **1990**, *20*, 733–741. [[CrossRef](#)]
42. Narukawa, Y.; Murofushi, T. Decision modelling using the Choquet integral. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Modeling Decisions for Artificial Intelligence, Barcelona, Spain, 1–3 August 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 183–193.
43. Mishra, M.; Srivastava, M. A view of artificial neural network. In Proceedings of the 2014 International Conference on Advances in Engineering & Technology Research (ICAETR-2014), Unnao, India, 1–2 August 2014; pp. 1–3.
44. Bebis, G.; Georgiopoulos, M. Feed-forward neural networks. *IEEE Potentials* **1994**, *13*, 27–31. [[CrossRef](#)]
45. Wilson, E.; Tufts, D.W. Multilayer perceptron design algorithm. In Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, Ermioni, Greece, 6–8 September 1994; pp. 61–68.
46. Bikku, T. Multi-layered deep learning perceptron approach for health risk prediction. *J. Big Data* **2020**, *7*, 1–14. [[CrossRef](#)]
47. Tripathy, R. Application of intrinsic band function technique for automated detection of sleep apnea using HRV and EDR signals. *Biocybern. Biomed. Eng.* **2018**, *38*, 136–144. [[CrossRef](#)]
48. Feng, K.; Liu, G. Obstructive sleep apnea detection based on unsupervised feature learning and hidden markov model. In Proceedings of the BIBE 2019; The Third International Conference on Biological Information and Biomedical Engineering, Hangzhou, China, 20–22 June 2019; pp. 1–4.