SCMs: Systematic Conglomerated Models for Audio Cough Signal Classification

Sunil Kumar Prabhakar and Dong-Ok Won *

Department of Artificial Intelligence Convergence, Hallym University, Chuncheon 24252, Republic of Korea; sunilprabhakar22@gmail.com
* Correspondence: dongok.won@hallym.ac.kr

Abstract: A common and natural physiological response of the human body is cough, which tries to push air and other wastage thoroughly from the airways. Due to environmental factors, allergic responses, pollution or some diseases, cough occurs. A cough can be either dry or wet depending on the amount of mucus produced. A characteristic feature of the cough is the sound, which is a quacking sound mostly. Human cough sounds can be monitored continuously, and so, cough sound classification has attracted a lot of interest in the research community in the last decade. In this research, three systematic conglomerated models (SCMs) are proposed for audio cough signal classification. The first conglomerated technique utilizes the concept of robust models like the Cross-Correlation Function (CCF) and Partial Cross-Correlation Function (PCCF) model, Least Absolute Shrinkage and Selection Operator (LASSO) model, elastic net regularization model with Gabor dictionary analysis and efficient ensemble machine learning techniques, the second technique utilizes the concept of stacked conditional autoencoders (SAEs) and the third technique utilizes the concept of using some efficient feature extraction schemes like Tunable Q Wavelet Transform (TQWT), sparse TQWT, Maximal Information Coefficient (MIC), Distance Correlation Coefficient (DCC) and some feature selection techniques like the Binary Tunicate Swarm Algorithm (BTSA), aggregation functions (AFs), factor analysis (FA), explanatory factor analysis (EFA) classified with machine learning classifiers, kernel extreme learning machine (KELM), arc-cosine ELM, Rat Swarm Optimization (RSO)-based KELM, etc. The techniques are utilized on publicly available datasets, and the results show that the highest classification accuracy of 98.99% was obtained when sparse TQWT with AF was implemented with an arc-cosine ELM classifier.

Keywords: feature extraction; feature selection; audio cough classification; machine learning

1. Introduction

The most important information about the respiratory system can be obtained from the cough sounds of a particular patient [1]. The cough sounds of every respiratory disease are quite unique to each other, and so, the diagnosis of it can be made easily by the physician. The expelled air from the lungs is reflected by the waveforms of the cough sounds. The cough has a specific sound pressure, and its unique relationship to the respiratory condition of the patient can be demonstrated well [2]. The esophageal pressure and the cough flow have a good relationship with the power of cough sounds. The voiced phase, explosive phase and intermediate phase are the three main phases of the individual cough sounds. When COVID-19 occurred, Artificial Intelligence (AI) techniques helped a lot in the diagnosis of various medical conditions and provided good solutions for patient outcome prediction depending on various inputs of the data [3]. Machine learning algorithms with AI techniques helped a lot in developing a quick and inexpensive approach to diagnosing COVID-19 problems. The recording of human audio signals can be performed in an instantaneous manner. With the advent of cough segmentation, data segmentation, machine learning and deep learning techniques, audio signal classification...
has been performed in the past decade. A few important works conducted in the past decade and in the past few years include the following. For detecting abnormal pulmonary function, the classification of voluntary cough sounds and airflow patterns was conducted by Abaza et al. [4]. By internal sound analysis, the detection of voluntary cough sounds was implemented by Lucio et al. [5]. Asthmatic cough sounds were described quantitatively by Thorpe et al. [6]. For the rapid diagnosis of childhood pneumonia, wavelet-augmented cough analysis was examined by Kosasih et al. [7]. Cough sound recognition was used for automatic croup diagnosis by Sharan et al. [8]. Wet and dry coughs in pediatric patients were automatically identified and classified by Swarnkar et al. [9]. The audio-based assessment of cough analysis was performed by Shi et al. [10]. The analysis of cough sounds with objective correlation and spirometry was conducted by Rudraraju et al., where an accuracy of 91.97% was obtained [11]. A multi-modal deep learning method for the analysis of cough sounds was presented by Malik et al., where an accuracy of 99.01% was obtained [12]. In a real-world environment, automatic cough classification for tuberculosis diagnosis was performed by Pahar et al. [13]. The Mel Frequency Cepstral Coefficient (MFCC) features and a CNN were used for cough classification for COVID-19 by Bansal et al. [14]. A multi-branch deep learning model for COVID-19 detection from cough sounds was developed by Ulukaya et al. [15]. Scalogram image specification using deep learning models was performed for cough audio classification by Loey et al. [16]. An ensemble approach for detecting COVID-19 from cough sounds was developed by Chowdhury et al. [17]. The unique spectral fingerprints were identified in cough sounds for diagnosing respiratory ailments by Ghrabli et al. [18]. A deep neural network-based respiratory pathology classification using cough sounds was conducted by Balamurali et al. [19]. A Malaysian cough sound analysis and COVID-19 classification with deep learning was conducted by Kho et al. [20]. With the help of interpretable symptom embeddings, the early diagnosis of COVID-19 was performed by Pal et al. [21]. Cough classification using machine learning and global smartphone recordings was conducted by Pahar et al. [22]. An in-depth study of utilizing cough sounds and deep neural networks for the early detection of COVID-19 was conducted by Islam et al. [23]. The wavelet analysis of voluntary cough sounds in patients with respiratory diseases was performed by Knocikova et al. [24]. A Variable Markov Model and recurrence dynamics were used for cough sound detection by Mouawad et al. [25]. Cough sound analysis for pneumonia and asthma classification in a pediatric population were conducted by Amrulloh et al. [26]. The acoustic features were analyzed for the speech-sound-based classification of asthmatic and health subjects by Yadav et al. [27]. A proof-of-concept study was conducted for the development of machine learning for asthmatic and healthy voluntary cough sounds by Hee et al. [28]. In this work, three systematic conglomerated models (SCMs) are proposed for audio cough signal classification. Initially, the pre-processing aspect is given some importance in our work, as follows.

To pre-process and preserve the audio cough signal, a filter is applied so that the unnecessary and redundant detail coefficients are removed. To mitigate the noise with both non-white and uncertain sources, a time–frequency domain filter back dependent on the Wavelet Transform is used. Depending on a collection of recursive functions, the versatile form of a time series signal technique utilized is the Discrete Wavelet Transform (DWT) [29]. Wavelet functions aid greatly in the formation of the Wavelet Transform (WT). To mitigate the noise in the signal, Wavelet Transforms are used widely. A low-pass digital filter is utilized initially for the audio cough signal to pass through it efficiently. Most of the redundant frequency components of the signals are eliminated completely. A signal can be decomposed into various frequency sub-bands by DWT, which is expressed in the following equation:

$$\psi(t) = \frac{1}{\sqrt{s^*}} \phi \left( \frac{t-p}{s^*} \right)$$  (1)
where \( s^* \) denotes the scale and \( p \) denotes the translation parameter. At various frequencies, the correlation is expressed as follows:

\[
W_{s^*}(p, s^*) = \frac{1}{s^*} \int_{-\infty}^{\infty} x(t) \psi^* \left( \frac{t - p}{s^*} \right) dt
\]  

(2)

To define the signal frequency, the DWT coefficients are used, and this is represented as follows:

\[
\left( s^*_j, p_k \right) = \left( 2^j, 2^j k : j, k \in \mathbb{Z} \right)
\]  

(3)

Once the cough audio signals are pre-processed, the proposed models are implemented. The main innovation of this proposal is to select and choose the various available models in the literature and create a novel conglomerated or hybrid model and test its performance on the audio cough signal datasets. The following techniques are used:

(a) The first technique utilizes the concept of robust models with Gabor dictionary analysis and machine learning techniques with special focus on support vector regression (SVR) and ensemble models;

(b) The second technique utilizes the concept of stacked conditional autoencoders (SAEs);

(c) The third technique utilizes the concept of using some efficient feature extraction schemes like TQWT, sparse TQWT, MIC and DCC and some feature selection techniques like BTSA, aggregation functions, FA, EFA classified with machine learning classifiers, KELM, arc-cosine ELM, RSO-based KELM, etc.

2. Proposed Method 1: Robust Models with Gabor Dictionary Analysis and Machine Learning

The cough sound signals are projected as \( Z \) of length \( L \), and this is conducted in a very-high-dimensional subspace. Sometimes, the projection can be over an overcomplete dictionary also, which comprises column vectors arranged as normalized atoms and represented as \( C = [c_1, c_2, \ldots, c_N] \) of size \( L \times N \), with \( L < N \). With the help of an ordinary least square term, this can be conducted efficiently. A sparsity-inducing regularizer could also be designed efficiently using such a technique. By means of solving the following optimization issue, the best coefficient vector \( b \) of length \( N \) is attained:

\[
\min_b \| z -Cb \|_2 \text{ subject to } \|b\|_0 \leq K \ll N
\]  

(4)

where the energy norm is represented by \( \| \cdot \|_2 \) and the number of non-zero entries is represented by \( \| \cdot \|_0 \) in the vector \( b \). Therefore, the most descriptive collection of atoms is obtained by the model. A high-end analysis dictionary technique and a robust regularization scheme are required to model the atoms. With the help of greedy matching, the tackling of the issue is conducted efficiently. By means of projecting the residual onto the specific atoms of the dictionary, the coefficient vector \( b \) is computed iteratively. Atoms that mitigate the residual energy the most are chosen, and then, the residual is updated. Unless a termination criterion is met, the procedure is repeated often. Traditionally, the concept of matching pursuit was used with the Gabor dictionary; hence, in our proposed work, three different schemes were used to replace the matching pursuit concept [30]. The sparsity degree of the model is controlled, and the regularization parameter handles the multicollinearity very well.

For the cough audio signals, a dictionary along with a particular resolution based on its time–frequency atoms is considered, and then, the coefficient vector \( b \) is estimated so that the regularization parameter is explored well. This optimization step aids the accurate classification of cough audio signals. Then, using the estimated coefficient vector \( b \) of the respective cough audio signals, the time–frequency matrices are obtained so that it is useful for classification. The coefficient vectors are reshaped efficiently so that appropriate
mathematical functions are implemented correctly. The proposed framework is given in Figure 1, as follows.

**Figure 1.** Proposed work of robust models with Gabor dictionary analysis and machine learning.

### 2.1. Gabor Dictionary

For the conceptualization of basis functions, the analysis dictionary is commonly used. An overcomplete dictionary is generally used so that a better sparsity can be obtained by means of the versatile localization of atoms. The analytic Gabor dictionary is used here, where a Gaussian window function is used and can be translated, scaled, and modulated as follows

\[
de_{[\alpha, \beta, \omega, \theta]}[n] = \frac{\gamma_{[\alpha, \beta, \omega, \theta]}}{\sqrt{\alpha}} \sigma \left[ \frac{n - \beta}{\alpha} \right] \cos[w[n - \beta] + \theta]
\]  

(5)

where the scaling constant is expressed by \(\gamma_{[\alpha, \beta, \omega, \theta]}\) and the Gaussian factor is expressed by \(\sigma \left[ \frac{n - \beta}{\alpha} \right] = 2^{\frac{1}{2}} e^{-\pi \left(\frac{n - \beta}{\alpha}\right)^2}\), with \(n = 0, 1, 2, \ldots, 2^{n-1}\). The Gaussian window length is considered as \(L = 2^n\), and it should correspond to the audio signal length which is supposed to be decomposed. Initially, the scale parameter \(\alpha\) is chosen, and the translator parameter \(\beta\) and frequency \(\omega\) are expressed as follows:

\[
[a, \beta, \omega, \theta] = \left[2^h, 2^{h-1}, q, 2^{-h-1}\pi f, 0\right]
\]

(6)

where \(0 < h < N, q = 0 : 1 : 2^{N-h+1} - 1\) and \(f = 0 : 1 : 2^{h+1} - 1\).

The shift parameter is presumed to be zero here, and the total number of atoms are modelled accordingly. The \(h\)th Gabor dictionary \(C_h\) of size \(2^N \times 2^{N+2}\) is developed for a particular value of \(h\) in a matrix format. All the audio cough signals have a similar length, and the variables of the \(h\)th dictionary are assigned approximately. By means of altering \(h\), the tradeoff between the spectral resolution and temporal resolution of the dictionary is solved. There is a slow decay in the Gaussian function \(\sigma \left[ \frac{n - \beta}{\alpha} \right]\) as the value of \(h\) increases. Therefore, the time duration of the atom is enhanced, and the temporal resolution is severely degraded. So, there is an increase in the Gaussian function duration, and the overall bandwidth of atoms present around the frequency reduces, leading to the atoms attaining a higher spectral resolution.
2.2. Cross-Correlation Function (CCF) and Partial Cross-Correlation Function (PCCF) Model

The audio cough signal data are split into \( a_t \) and \( b_t \) with \( t = 1, 2, \ldots, n \) and a specific lag \( l \). Between the two time series \( a_t \) and \( b_{t+l} \), the general formula for the covariance function is represented as follows:

\[
c_{ab}(l) = \frac{1}{n} \sum_{t=1}^{n-l} (a_t - \bar{a})(b_{t+l} - \bar{b})
\]  

(7)

where the means of \( a_t \) and \( b_t \) are expressed as follows:

\[
r_{ab}(l) = \begin{cases} 
\frac{c_{ab}(l)}{\sigma_a \sigma_b}; & l \geq 0 \\
\frac{c_{ab}(-l)}{\sigma_a \sigma_b}; & l < 0
\end{cases}
\]  

(8)

where the standard deviations of \( a_t \) and \( b_t \) are indicated by \( \sigma_a \) and \( \sigma_b \), respectively. \( a_t \) and \( b_{t+l} \) are correlated if \( l \) is a positive integer. \( a_t \) and \( b_{t-1} \) are correlated if \( l \) is a negative integer. When \( l = 0 \), then \( r_{ab} = r_{ab}(0) \), indicating the correlation coefficient. It is assumed that the cough audio signal \( a_t \) is correlated with the time-delayed signal \( b_{t+l} \) and the audio signal \( q_t \) is highly correlated with both \( a_t \) and \( b_{t+l} \) (\( t = 1, 2, \ldots, n \)). Based on CCF \( r_{ab}(l) \), it can be assessed with the help of partial CCF (PCCF) and represented as follows \[32\]:

\[
r_{ab|q}(l) = \frac{r_{ab}(l) - r_{aq} \cdot r_{qb}(l)}{\sqrt{(1 - r_{aq}^2)} \sqrt{(1 - r_{qb}^2)}}
\]  

(9)

where the CCF at lag \( l \) is represented by \( r_{ab}(l) \) and \( r_{qb}(l) \). A discrete PCCF which corresponds to lag \( l \) is represented by \( r_{ab|q}(l) \). The elimination of the linear influence \( q_t \) is conducted from \( r_{ab}(l) \). The PCCF manages to indicate the cross-correlation among the signal residuals.

2.3. LASSO Model

LASSO is quite powerful in selecting variables and constructing regression techniques. A penalty term is added to the model’s residual sum of squares in the concept of LASSO \[33\]. If the penalty term is exceeded, then a greater number of variables will be penalized. A regularization process is added to the penalty term, and so, some of its coefficients are shrunk to zero. As the least correlated variables shrink, the interpretability of the regression model increases. The irrelevant variables can be eliminated, and so, only the explanatory variables are brought on board, thereby eliminating overfitting. During training, the model can fit well, but it cannot predict the testing data accurately. Finally, in between the variance and bias, a suitable tradeoff could be found, and so, the model can be less complex and well balanced. In this linear model, the mathematical formulation is as follows:

\[
G = H\beta + \epsilon
\]  

(10)

where the response variable is denoted as \( G \) and the explanatory variable is denoted as \( H \). \( H \) is assessed as an \( H_{n \times k} \) matrix and \( G \) is assessed as \( G_{n \times 1} \) matrix in a vector format. The RSS is minimized by LASSO where the upper bound is specified by \( t \). The formula for LASSO is expressed in the following equation:

\[
\min \sum_{i=1}^{n} \left( G_i - \sum_{j=1}^{k} h_{ij}\beta_j \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{k} |\beta_j| < t
\]  

(11)

The optimization of it in the vector form is expressed as follows:

\[
\min \left( \frac{||G - H\beta||_2}{n} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{k} |\beta_j| < t
\]  

(12)
The addition of a penalty term is conducted before the sum of the absolute value of the model parameters. If the penalty term is large, then the model will have more shrinkage.

The estimation of the parameters is represented as follows:

\[
\hat{\beta}(\lambda) = \arg\min_{\beta} \left( \frac{|G - H\beta|_2^2}{n} + \lambda \sum_{j=1}^{k} |\beta_j| \right)
\]  

(13)

The \( j^{th} \) coefficient is assumed to be zero, and it will be eliminated from the model. The penalty values can be altered so that the number of selected features can be changed.

2.4. Elastic Net Regularization Model

A slightly broader version of the LASSO technique/ridge regression is the elastic net regularization model [34]. Linear models have a lot of vector coefficients, and sometimes, there is an instability of estimates, and to prevent it, ridge regression is used. An accurate estimate of the coefficient is provided by the LASSO, as some explanatory atoms associated with it become zero. The regularization parameter denoted as \( \alpha \) is denoted between 0 and 1 and is non-negative in nature \( \lambda (\lambda \geq 0) \). It helps to solve the following issue and is represented as follows:

\[
\min_{a_{h,\alpha}} \left( \frac{1}{2N} \|Z - D_h a_{h,\alpha}\|_2^2 + \lambda \left( \frac{1 - \alpha}{2} \|a_{h,\alpha}\|_2^2 + \alpha \|a_{h,\alpha}\|_1 \right) \right)
\]  

(14)

The ridge regression and LASSO are obtained when the \( \alpha \) values are considered as 0 and 1. When the value of \( \alpha = 0.5 \), both the properties of ridge regression and LASSO are imbibed by it. Producing the lowest error is very important so that the coefficient vector is well estimated; thereby, the selection of non-zero entries is performed easily. For various values of \( \lambda \), a geometric sequence is used by the elastic net regularization models so that the feature shrinkage is controlled. Depending on the minimum mean square error, the assessment of the optimal value of \( \lambda \) is performed. With the help of the alternating direction method of multipliers, the optimization of Equation (14) is carried out. The termination of the iterative update of the vector \( a_{h,\alpha} \) is performed when there is a nice change in its size observed below a threshold. Some quantities of interest can be assessed at this convergence point so that the optimal dictionary can be identified well.

2.5. Support Vector Regression

When kernel range strategies are extended, SVR is formed, and it is used for regression analysis [35]. The qualities that help to explain SVM like sparsity, entropy and kernels are all supported by SVR. For predictive data analysis, SVR is a potential tool and is used in multiple applications. For the purpose of both non-linear and linear mapping, SVR is widely used, and different kernels can be predicted efficiently. Between the input and output, the mapping function is expressed as follows:

\[
g_i = w\zeta(y) + \alpha
\]  

where \( y \) is considered as the input \( (y^1, y^2, \ldots, y^{N_{train}}) \), and the output is considered as \( g_i \). The weight vector is assigned by \( w \in \mathbb{R} \). The training data number is expressed by \( N_{train} \), and the constant is expressed by \( \alpha \). The non-linear function is expressed by \( \zeta(\cdot) \). With the help of the following function, the estimation of the value of \( w \) is performed.

Minimize \[ \frac{1}{2} \left\| w \right\|^2 + B \sum_{i=1}^{N_{train}} (\theta_i - \theta_i^*) \]  

(16)

Subject to \[
\begin{align*}
    g_i - (w\zeta(y_i) + \alpha_i) & \leq \varepsilon + \theta_i \\
    (w\zeta(y_i) + \alpha_i) - g_i & \leq \varepsilon + \theta_i^* \\
    \theta_i, \theta_i^* & \geq 0
\end{align*}
\]  

(17)
where the slack variables are expressed as $\theta_i, \theta^*_i$. The box constraint is expressed as $B$, and the insensitive loss function is expressed by $\varepsilon$. With the help of following equation, the output can be computed as follows:

$$g_i = \sum_{i=1}^{N_{\text{train}}} (\theta_i + \theta^*_i)K(y_i, y) + \alpha$$  \hspace{1cm} (18)

where the kernel function is denoted as $K(y_i, y)$.

### 2.6. Learning through Ensemble Methods

To obtain good accuracy, many ensemble approaches are leveraged so that a good conceptual framework is developed. The overall performance can be improved by ensemble learning where several weak learners are bought together and then combined. When there are limited training data, this technique is implemented. The output of various classifiers is averaged, and so, the change in selecting an inadequate classifier is mitigated by the ensemble algorithm [36]. In the development of ensemble learning algorithms, bagging and boosting techniques are utilized. A famous regression algorithm that utilizes the bootstrapping technique is called the bagging ensemble learning technique. Here, a forest of decision trees is created randomly depending on various training sets. From every decision tree, an output is considered so that the final output is generated by averaging the equation below:

$$\hat{f}_{\text{average}} = \frac{1}{T} \sum_{t=1}^{T} f^T(y)$$  \hspace{1cm} (19)

where the individual prediction from $T$ training sets is expressed by $f^T(y)$. Another type of ensemble learning is boosting ensemble learning techniques where the bootstrap sampling method is not used. The creation of the models is conducted in a sequential manner, and hence, a good knowledge of the model is necessary before the generation of the model.

### 3. Proposed Technique 2: Stacked CAE Network for Audio Cough Classification

#### 3.1. CAE

An unsupervised feature learning neural network is autoencoder (AE), where the learned features are specified by the hidden layer [37]. The encoder network comprises the input and hidden layers, and the original input is transformed into hidden features by this network. The decoder network comprises the hidden and output layers, and the original inputs are reconstructed from the learned hidden features. For every input vector $p^d$ from datasets $\{p^d\}_{d=1}^{M}$, the hidden vector $h^d$ is represented as follows:

$$h^d = f\left( W^{(1)} p^d + b^{(1)} \right)$$  \hspace{1cm} (20)

For every input vector $p^d$ from datasets $\{p^d\}_{d=1}^{M}$, the reconstruction vector $\hat{p}^d$ is expressed as follows:

$$\hat{p}^d = f\left( W^{(2)} h^d + b^{(2)} \right)$$  \hspace{1cm} (21)

where the weight matrices are expressed by $W^{(1)}$ and $W^{(2)}$. The bias vectors are indicated by $b^{(1)}$ and $b^{(2)}$, and the activation function is indicated by $f$.

In this work, the sigmoid function is utilized, and its derivative function is expressed as follows:

$$f(\beta) = \frac{1}{1 + e^{-\beta}}$$  \hspace{1cm} (22)

$$f'(\beta) = f(\beta)(1 - f(\beta))$$  \hspace{1cm} (23)
The reconstruction error is expressed as follows:

$$L(p^d, \hat{p}^d) = \frac{1}{2} \| p^d - \hat{p}^d \|^2$$  \hfill (24)

To the reconstruction error of the AE, a penalty term is applied, and therefore, a CAE can be obtained easily [38]. With respect to the input vector, the Frobenius norm of the Jacobian matrix of the hidden activations corresponds to the penalty term and is expressed as follows:

$$\| J_f(p^d) \|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{e} \left( \frac{\partial h_j(p^d)}{\partial p_i} \right)^2$$  \hfill (25)

For the $j^{th}$ row and the $i^{th}$ column in the matrix, the $J_f(p^d)_{ji}$ element is specified as $J_f(p^d)_{ji}$ and is defined as follows:

$$J_f(p^d)_{ji} = \frac{\partial h_j(p^d)}{\partial p_i}$$  \hfill (26)

The computation of Equation (26) is conducted when the activation function is a sigmoid function and is represented as follows:

$$J_f(p^d)_{ji} = h_j(p^d) \left( 1 - h_j(p^d) \right) W_j^{(1)}$$  \hfill (27)

where the connection weight between neuron $j$ in the hidden layer and neuron $i$ in the input layer is expressed by $W_j^{(1)}$. Equation (27) can be further computed dependent on Equation (28) as follows:

$$\| J_f(p^d) \|_F^2 = \sum_{i=1}^{n} \left( h_j(p^d) \left( 1 - h_j(p^d) \right) \right)^2 \sum_{i=1}^{n} W_j^{(1)2}$$  \hfill (28)

For $S$ samples, the overall cost function of the CAE is expressed as follows:

$$J(W, b) = \frac{1}{2} \sum_{d=1}^{S} \left[ L(p^d, \hat{p}^d) + \lambda \| J_f(p^d) \|_F^2 \right]$$  \hfill (29)

where the penalty term parameters are specified by $\lambda$. The relative significance of the reconstruction error and the penalty term is controlled by $\lambda$.

The representative hidden features can be learnt automatically by training CAE so that all the useful information can be easily extracted. By utilizing the BP algorithm, the cost function $J(W, b)$ is minimized, and so, even to the small changes in the inputs, it can be invariant. Each element $W^{(1)}, W^{(2)}, b^{(1)}$ and $b^{(2)}$ is updated after every iteration as follows:

$$W_j^{(1)} = W_j^{(1)} - \varepsilon \frac{\partial J(w, b)}{\partial W_j^{(1)}} \quad W_j^{(2)} = W_j^{(2)} - \varepsilon \frac{\partial J(w, b)}{\partial W_j^{(2)}}$$  \hfill (30)

$$b_j^{(1)} = b_j^{(1)} - \varepsilon \frac{\partial J(w, b)}{\partial b_j^{(1)}} \quad b_j^{(2)} = b_j^{(2)} - \varepsilon \frac{\partial J(w, b)}{\partial b_j^{(2)}}$$  \hfill (31)

where the learning rate is expressed by $\varepsilon$. Thus, with the help of the training process, the simultaneous learning of the optimal parameters $W^{(1)}, W^{(2)}, b^{(1)}, b^{(2)}$ is determined.
3.2. Softmax Classifier

For multi-class classification, a commonly utilized classifier is the softmax classifier \([39]\). Assuming the description of training datasets as \(\{(a^{(1)}, b^{(1)}), \ldots, (a^{(N)}, b^{(N)})\}\), where one of the input vectors of the softmax classifier is expressed as \(a^{(s)} \in \mathbb{R}^q, s = 1, 2, \ldots, N\) and the corresponding label is represented as \(m = 1, 2, \ldots, N\). If the training datasets are assumed to belong to \(k\) different classes, then \(k\) different values are obtained by the labels, and \(b^{(s)} \in \{1, 2, \ldots, k\}\). The output vector \(o^{(s)}\) for every input vector \(a^{(s)}\) is expressed as follows:

\[
\begin{align*}
o^{(s)} &= \theta_a^{(s)}
\end{align*}
\]

where the learning rate is expressed as \(\theta \in \mathbb{R}^{k \times c}\) and is defined as

\[
\theta = \begin{bmatrix}
\theta_1^T \\
\theta_2^T \\
\vdots \\
\theta_k^T 
\end{bmatrix}
\]

(33)

The probability of \(a^{(s)}\) belonging to class \(q\) (i.e., the probability of \(b^{(s)} = q, q = 1, 2, \ldots, k\)) is defined as follows:

\[
\begin{align*}
p\left(b^{(s)} = q|a^{(s)}; \theta\right) &= \frac{e^{\theta_q^T a^{(s)}}}{\sum_{q=1}^{k} e^{\theta_q^T a^{(s)}}}
\end{align*}
\]

(34)

The probability of \(a^{(m)}\) belonging to each class is estimated by the hypothesis \(h_\theta(a^{(m)})\) and is expressed as follows:

\[
\begin{align*}
h_\theta\left(a^{(s)}\right) = 
\begin{bmatrix}
p\left(b^{(s)} = 1|a^{(s)}; \theta\right) \\
p\left(b^{(s)} = 2|a^{(s)}; \theta\right) \\
\vdots \\
p\left(b^{(s)} = k|a^{(s)}; \theta\right)
\end{bmatrix}
= \frac{1}{\sum_{q=1}^{k} e^{\theta_q^T a^{(s)}}}
\begin{bmatrix}
ee^{\theta_1^T a^{(s)}} \\
ee^{\theta_2^T a^{(s)}} \\
\vdots \\
ee^{\theta_k^T a^{(s)}}
\end{bmatrix}
\end{align*}
\]

(35)

In between the target and the predicted class labels, the error has to be described, and the cost function of the softmax classifier \(J(\theta)\) is defined as follows:

\[
\begin{align*}
J(\theta) &= \frac{1}{N} \left[ \sum_{i=1}^{N} \sum_{s=1}^{k} 1\{b^s = q\} \log \left(\frac{e^{\theta_q^T a^{(s)}}}{\sum_{q=1}^{k} e^{\theta_q^T a^{(s)}}}\right) \right] + \frac{\gamma}{2} \|\theta\|^2_2
\end{align*}
\]

(36)

To prevent overfitting, the second term, also called the weight penalty term, is used. To control the significance of the penalty term, the penalty term parameter used is \(\gamma\). To procure the optimal parameter matrix \(\theta\), the training of the softmax classifier is also performed like the training of CAE so that by means of mixing the art function with the back propagation (BP) algorithm a suitable transfer function is established between the input and the target labels. Every element of \(\theta\) is updated after each iteration as follows:

\[
\theta_{ji} = \theta_{ji} - \varepsilon_2 \frac{\partial J(\theta)}{\partial \theta_{ji}}
\]

(37)

where the learning rate is expressed as \(\varepsilon_2\). \(\theta_{ji}\) indicates the elements of \(\theta\) in the \(j^{th}\) row and \(i^{th}\) column.

3.3. Stacked CAE Network

To form a stacked CAE network with hidden layers, \(L\) CAEs and one softmax classifier are stacked together \([40]\). The encoder network of CAE1 comprises the input and the first
hidden layer. The encoder network of CAE2 comprises the first and second hidden layers. To the output layers of the stacked CAE network, the addition of a softmax classifier is performed so that the classification can be conducted efficiently. Pre-training and fine-tuning are the two important training processes of the stacked CAE network. Each CAE is pre-trained individually by the pre-training process, and to learn the intra-non-linear transformation relationship of every single CAE, a softmax classifier is used. The relationship between the target labels and the learned high-level features can be easily explored by this process. A better set of weight initialization is obtained by this process rather than utilizing random initialization. In between the layers, the inter-relationships are learned by the fine-tuning process, and the best suitable transformation relationships between the labels and the learned high-level intrinsic features are established well, and so, classification is enabled. In between the target and the predicted class labels, the error must be described, and the learned high-level intrinsic features are established well, and so, classification is fine-tuning process, and the best suitable transformation relationships between the labels and the learned high-level intrinsic features are established well, and so, classification is enabled.

A better set of weight initialization is obtained by this process rather than utilizing random initialization. In between the layers, the inter-relationships are learned by the fine-tuning process, and the best suitable transformation relationships between the labels and the learned high-level intrinsic features are established well, and so, classification is enabled. In between the target and the predicted class labels, the error must be described, and the learned high-level intrinsic features are established well, and so, classification is fine-tuning process, and the best suitable transformation relationships between the labels and the learned high-level intrinsic features are established well, and so, classification is enabled.

The output of the softmax classifier is the probability of each class label. It is computed as the input vector of the softmax classifier and is expressed as follows:

$$ h^{d}_{L} = f \left( W^{(1)}_{L} h^{d}_{L-1} + b^{(1)}_{L} \right) $$

Using the BP algorithm, CAE2 is also effectively trained, and the computation of the second hidden representation $h^d_2$ is conducted as the input vector of CAE3. Until the effective training of the $L^{th}$ CAE is carried out, this process is repeated, and the $L^{th}$ hidden representation $h^d_L$ is computed as the input vector of the softmax classifier and is expressed as follows:

$$ h^{d}_{L} = f \left( W^{(1)}_{L} h^{d}_{L-1} + b^{(1)}_{L} \right) $$

The calculation of both the hypothesis $Hyp_{\theta} \left( h^{(d)}_{L} \right)$ and the cost function $J(\theta)$ stacked is conducted as follows:

$$ Hyp_{\theta} \left( h^{(d)}_{L} \right) = \begin{bmatrix} p[q^{(d)} = 1 | h^{(d)}_{L}; \theta] \\ p[q^{(d)} = 2 | h^{(d)}_{L}; \theta] \\ \vdots \\ p[q^{(d)} = k | h^{(d)}_{L}; \theta] \end{bmatrix} = \frac{1}{\sum_{r=1}^{k} e^{\theta^{T} h^{(d)}_{L}}} \begin{bmatrix} e^{\theta^{T} h^{(d)}_{1}} \\ e^{\theta^{T} h^{(d)}_{2}} \\ \vdots \\ e^{\theta^{T} h^{(d)}_{k}} \end{bmatrix} $$

$$ J(\theta)_{stacked} = -\frac{1}{N} \sum_{d=1}^{M} \sum_{r=1}^{k} q^{(d)} = r \left\{ \log \left( \frac{e^{\theta^{T} h^{(d)}_{r}}}{\sum_{r=1}^{k} e^{\theta^{T} h^{(d)}_{r}}} \right) \right\} + \frac{\gamma}{2} \| \theta \|^{2} $$

As follows:

$$ J(\theta) = \frac{1}{N} \sum_{s=1}^{N} \sum_{q=1}^{k} 1 \{ b^{s} = q \} \log \left( \frac{e^{\theta^{T} a^{(s)}}}{\sum_{r=1}^{k} e^{\theta^{T} a^{(r)}}} \right) + \frac{\gamma}{2} \| \theta \|^{2} $$

(38)
Hence, by means of minimizing the cost function, the optimal parameter matrix $\theta$ is obtained, and ultimately, the training of the softmax classifiers is conducted with the help of supervised learning. Finally, to fine-tune the parameters of the entire network, the BP algorithm is used so that the most suitable relationship between the layers is well established, and thereby, a well-trained stacked CAE network is obtained. Figure 2 shows the illustration for the proposed stacked CAE for cough audio classification.

![Proposed Stacked CAE](image)

**Figure 2.** Proposed stacked CAE for audio cough classification.

### 4. Proposed Technique 3: Efficient Feature Selection and Machine Learning Classification

#### 4.1. Analysis of TQWT

Depending on the oscillation characteristics of the audio cough signal, an improvised version of the Wavelet Transform is proposed, known as TQWT [41]. With the aid of real-valued sampling factors, reversible oversampling filters are used by TQWT. Through the low-pass and high-pass filters, the passing of the input is performed in every decomposition stage. For the low-pass and high-pass filter in TQWT, the frequency responses are indicated as $G_0(w)$ and $G_1(w)$ and expressed as follows:

$$G_0(w) = \begin{cases} 
\theta \left[ \frac{w + (\sigma - 1) \pi}{\mu + \sigma - 1} \right] & |w| \leq (1 - \sigma) \pi \\
0 & (1 - \sigma) \pi < |w| < \mu \pi \\
\mu \pi & |w| \leq \pi 
\end{cases}$$

(44)

$$G_1(w) = \begin{cases} 
\theta \left[ \frac{\mu \pi - w}{\mu \pi - \sigma \pi} \right] & |w| \leq (1 - \sigma) \pi \\
0 & (1 - \sigma) \pi < |w| < \mu \pi \\
1 & \mu \pi \leq |w| \leq \pi 
\end{cases}$$

(45)
where the low-pass scaling factor is denoted by $\mu$ and the high-pass scaling factor is denoted by $\sigma$, and it is assumed that $0 < \mu < 1$, $0 < \sigma \leq 1$ and $\mu + \sigma > 1$. The Daubechies filter is considered as $\theta(w)$, which helps to construct the transition bands $G_0(w)$ and $G_1(w)$, giving rise to the following equation:

\[
\theta(w) = \frac{1}{2}(1 + \cos w)\sqrt{2 - \cos w}, \quad |w| \leq \pi
\]

The low-pass filter output $w_L$ of every stage is the preceding stage input in this multilayer decomposition technique. For each layer, the wavelet coefficient is $w_H$. To obtain a novel set, the processing of every coefficient is conducted after the decomposition is performed. With the help of TQWT inverse transformation, the reconstruction of the signal can be performed. The oversampling rate $r$, quality factor $Q$ and the decomposition level $D$ are the main parameters here. $Q = \frac{f_0}{B_w}$ is the ration between the center frequency $f_0$ and the bandwidth of the signal $B_w$. The number of wavelet coefficients can be expressed by $r$, and it assesses the spectral overlap occurring between the corresponding bandpass filters. In our work, $r = 5$ is considered. $D + 1$ sub-bands are produced by $D$-level decomposition. The filter’s bandwidth limitation is assessed, and the following is obtained:

\[
D_{\text{max}} = \left\lfloor \frac{\lg(\sigma N_s/8)}{\lg(1/\mu)} \right\rfloor
\]

Once the parameters are determined, $\sigma = \frac{2}{Q+1}$ and $\mu = 1 - \frac{2}{(Q+1)r}$ can be computed.

4.2. Analysis of Sparse Coefficients of TQWT

The wavelet coefficients that assess the given audio cough signal are not distinctive as TQWT is oversampled. By means of basis pursuit, the sparse collection of TQWT coefficients can be assessed. The $(D + 1)$ wavelet coefficients are specified as \{\{w_d\}_{d=1}^{D+1}\}, where the number of coefficients of the $d^{\text{th}}$ set is represented by $H_d$. To solve the coefficients, the following optimization problem is utilized:

\[
\arg\min_{w} \left\| z - \text{TQWT}^{-1}(w) \right\|_2^2 + \sum_{d=1}^{D+1} \eta_d \|w_d\|
\]

where the regularization parameter for the $d^{\text{th}}$ sub-band is expressed as $\eta_d$. The TQWT reconstruction is assigned as $\text{TQWT}^{-1}(\cdot)$. With the help of Lagrangian shrinkage methods, the optimization problem can be solved easily.

4.3. Analysis of Maximal Information Coefficient

In between the two variables, the association strength can be easily measured by MIC where both the linear and non-linear relationships are considered [42]. The data are partitioned into a grid, and then, the dependence between the variables is measured in every cell. Assuming two variables $a_i$ and $a_j$, then, the computation of MIC is conducted as follows:

\[
\text{MIC} = \max \left\{ \log_{\min}\left\{ \frac{1}{n} \sum_{i \in a} a_i, \frac{1}{n} \sum_{j \in a} a_j \right\} \right\}
\]

where the data point is represented by $n$. The partition bins are expressed by $n_a$ and $n_j$. The quality of information about a particular variable is provided by this highest value of dependence. Detecting relationships is quite possible as the MIC is quite robust to outliers and the merits of maximal dependency are considered.

4.4. Analysis of Distance Correlation Coefficient

Between the two random variables, the interdependence is evaluated by means of analyzing both the linear and non-linear relationships [43]. Between the two variables, the
covariance between the distance matrices is normalized by means of distance standard deviations, and it is represented as follows:

\[
dCor(a_i, a_j) = \frac{dCor(a_i, a_j)}{dVar(a_i), dVar(a_j)}
\]  

(50)

The samples have a lot of observations, and the pairwise distance between them is indicated by a distance matrix. The correlation present between the scalar products is emphasized by the DCC. The proposed model of feature extraction/selection and classification for audio cough classification is shown in Figure 3.

![Figure 3](image)

**Figure 3.** Proposed integrated model of feature extraction, feature selection and classification for audio cough classification.

### 4.5. Feature Selection

For all the extracted features, the discriminative capabilities are not important or higher, and so, sometimes, this could lead to overfitting in the network. Only the essential features must be retained, and the redundant ones should be eliminated. By means of optimizing the fitness or cost function, the optimal features are identified by feature selection tasks by making use of nature-inspired algorithms. In this work, a simple and effective algorithm called the Binary Tunicate Swarm Algorithm (BTSA) is used for feature selection.

#### 4.5.1. BTSA

In the deep ocean, the food source search process of tunicates was evaluated to develop this famous swarm-based optimization algorithm. With the help of a jet propulsion scheme, the tunicates progress towards food sources, thereby exhibiting a higher amount of intelligence. To solve various numerical optimization issues, the mathematical modelling of these two behaviors is considered. In the BTSA algorithm, the tunicate position is converted to a binary form of a vector [44]. The tunicates have a swarm behavior, and so, collisions among them are prevented and progression is towards the direction of the best tunicate. While searching for food particles, the tunicates avoid colliding with one another. This strategy is defined as follows:

\[
\vec{p} = \frac{\vec{g}}{\vec{s}}
\]  

(51)

where the gravity forces among the agents are represented by \( \vec{g} \) and social forces among the agents are represented by \( \vec{s} \). Both of these forces regulate tunicate movement. The water flow (\( \vec{w} \)) is used to control the gravity force and is represented as follows:
\[ \vec{G} = c_1 + c_2 - \vec{W} \]  
\[ \vec{W} = 2c_3 \]  
where random variables are expressed as \( c_1, c_2 \) and \( c_3 \), located by \( \text{rand}(0, 1) \). The parameter \( \vec{S} \) is expressed as follows:

\[ \vec{S} = h_i + c_3(h_s - h_i) \]  
where \( h_i \) and \( h_s \) indicate the min and max speed so that a good social interaction is achieved. The tunicates trace the best position and progress towards it so that collisions are avoided between them. The average distance is computed as follows:

\[ D_k = \left| \vec{Q}_b - r_1 \cdot \vec{Q}_k(i) \right| \]  
where the distance of the present location of the \( k^{th} \) tunicate \( \vec{Q}_k(i) \) from the best position of tunicates \( \vec{Q}_b \) is represented by \( D_k \). The tunicate tries to remain near the best tunicate, and so, a good source position is tried out and expressed by the following equation:

\[ \vec{Q}_k(i) = \begin{cases}  
\vec{Q}_b + \vec{P} \cdot \vec{D}_k, & \text{rand} \geq 0.5 \\
\vec{Q}_b - \vec{P} \cdot \vec{D}_k, & \text{rand} \leq 0.5 
\end{cases} \]  

The next \( k^{th} \) tunicate position is updated as follows:

\[ \vec{Q}_k(i + 1) = \frac{\vec{Q}_k(i) + \vec{Q}_b}{2 + c_1} \]  

To match the feature selection issue, these positions are matched to a binary value. By means of applying transfer functions such as S-shaped and V-shaped, this conversion happens. In this work, a V-shaped transfer function (T) is used by the BTSA, so the positions are updated well from one iteration to another, and this is usually performed in binary space. The tunicate position in binary form is updated as follows:

\[ \vec{Q}_K(i + 1) = \begin{cases}  
\vec{Q}_k(i)', \text{rand}() < T(\vec{Q}_k(i + 1)) \\
\vec{Q}_k(i), \text{rand}() \geq T(\vec{Q}_k(i + 1)) 
\end{cases} \]  

where \( \vec{Q}_k(i)' \) indicates the complement of \( \vec{Q}_k(i) \).

The considered fitness function is as follows. Depending on the feature selection ratio and the classification error, the fitness function can be designed as follows:

\[ \text{Fitness}_1 = k_1 \cdot \text{error} + k_2 \cdot \text{FSR} \]  
where error represents the classification error. The total number of features chosen for the feature size is considered the feature selection ratio (FSR). The weighing factors are chosen as \( k_1 \) and \( k_2 \) in the range of \([0, 1]\). To mitigate the classifier burden, a small FSR is usually useful.

For the optimal feature selection technique, the BTSA along with the fitness function is utilized. Algorithm 1 shows the BTSA feature choosing process. The tunicates symbolize the search agent position vectors and are assigned initially with binary values which are
randomly generated. Every subset of a feature is selected by the agent where the feature from the initial vector is indicated by the binary value 1. By choosing the feature subset, every agent performs the classification task, and the fitness function is used to evaluate the result. The recording of the position vector which mitigates the fitness function is conducted. Every position of each agent is updated by the TSA. With the aid of the transformation function, the position is converted into binary values. Unless the termination criteria are met, the algorithm iterates continually by representing the novel position vectors. The optimal feature set is provided by the best position vector when the iteration ends.

**Algorithm 1: BTSA Algorithm**

<table>
<thead>
<tr>
<th>Algorithm 1: BTSA Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Feature Set</td>
</tr>
<tr>
<td><strong>Assign:</strong> Search agents (Tunicate population)</td>
</tr>
<tr>
<td><strong>Start TSA parameters</strong></td>
</tr>
<tr>
<td>For i = 1: no of search agents do</td>
</tr>
<tr>
<td>Choose feature with a position 1</td>
</tr>
<tr>
<td>Execute classification task with selected features</td>
</tr>
<tr>
<td>End</td>
</tr>
<tr>
<td><strong>Assess fitness function</strong></td>
</tr>
<tr>
<td><strong>Memorize the best agent position</strong></td>
</tr>
<tr>
<td>While (no of iterations) do</td>
</tr>
<tr>
<td>Update the tunicate positions</td>
</tr>
<tr>
<td>Convert to binary values</td>
</tr>
<tr>
<td>Execute classification using updated positions</td>
</tr>
<tr>
<td>Assess Fitness function</td>
</tr>
<tr>
<td>Memorize the global best position</td>
</tr>
<tr>
<td>If convergence reached then</td>
</tr>
<tr>
<td>Exit the process</td>
</tr>
<tr>
<td>End</td>
</tr>
<tr>
<td><strong>End</strong></td>
</tr>
<tr>
<td><strong>Output the optimal feature set.</strong></td>
</tr>
</tbody>
</table>

4.5.2. Choosing Aggregate Functions

The main goal of feature selection lies in aggregation function operations, Ref. [45]. In some techniques, every feature selection is collected and combined together to make a final decision. Aggregate functions based on basic operations such as average/minimum/maximum/median, etc., are usually considered. Depending on the voting strategies also, the aggregation technique can be developed easily.

**Aggregation Techniques Depending on Basic Operations**

Assuming there are \( k \) feature selection results, and every result comprises \( y \) features. The minimum value of the feature correlation is chosen by considering the minimum aggregation. The final score of all features is computed, and it is ordered to obtain the ultimate score so that various feature subsets can be aggregated. If the end score is high, the feature is more significant. The computation of the feature \( f_z(z = 1, 2, \ldots, y) \) end score is considered here. The max aggregation is nothing but the end score of \( f_z \), and it depends on the highest scores of each subset, and it is computed as follows:

\[
Score = \max(F_1(f_z), F_2(f_z), \ldots, F_k(f_z)) \tag{60}
\]

The minimum aggregation is nothing but the end score of \( f_z \), and it depends on the lowest scores of each subset, and it is computed as follows:

\[
Score = \min(F_1(f_z), F_2(f_z), \ldots, F_k(f_z)) \tag{61}
\]
The median aggregation is nothing but the end score of $f_z$, and it depends on the median scores of each subset, and it is computed as follows:

$$Score = \text{Median}(F_1(f_z), F_2(f_z), \ldots, F_k(f_z))$$  \hspace{1cm} (62)

The mean aggregation is nothing but the end score of $f_z$, and it depends on the mean score of each subset, and it is computed as follows:

$$Score = \frac{1}{k} \sum_{i=1}^{k} F_i(f_z)$$  \hspace{1cm} (63)

4.5.3. Factor Analysis

The relationships can be well summarized or reduced with the help of factor analysis so that a limited number of basic dimensions are obtained ultimately [46]. The dependency structure can be destroyed completely by means of mitigating the size. In between the variables, the relationships are explored so that some novel structures are formed and the number of variables is reduced. The selection of varimax is performed if there are independent factors in FA. In between the variables, the correlation is computed by means of assessing the ingredients of the obtained factors. Exploratory and confirmatory factor analysis are two types of factor analysis used generally and, in this work, explanatory factor analysis is used. The factor is assumed to be a simple linear hybridization of its variables. In between the observed variables, the covariance and correlation are linked, and so, it forms a variable reduction method. A collection of variables is completely reduced by means of utilizing a limited number of latent factors. The variables with very high correlations are considered by FA and then analyzed together as factors. With various names, FA is referred based on its utility. By means of utility correlation and covariance matrices of the data, independent new variables are formed by the EFA.

EFA

To provide a good visualization and analysis, the different variables are considered and correlated to each other, and then, the best few factors are considered after eliminating the total number of variables. The covariance or correlation matrix is used in this dimension reduction technique. The observed variables $(Z_i, i = 1, 2, \ldots, q)$ are assumed to be a linear combination of the factors. A factor analysis model with some observed variables and common factors is expressed as follows:

$$Z_1 - \mu_1 = c_{11}F_1 + c_{12}F_2 + \ldots + c_{1k}F_k + \epsilon_1$$  \hspace{1cm} (64)

$$Z_2 - \mu_2 = c_{21}F_1 + c_{22}F_2 + \ldots + c_{2k}F_k + \epsilon_1$$  \hspace{1cm} (65)

$$Z_q - \mu_q = c_{q1}F_1 + c_{q2}F_2 + \ldots + c_{qk}F_k + \epsilon_q$$  \hspace{1cm} (66)

where the coefficient is written as $c_{ij}$ and is expressed as factor loadings. The average vector is represented by $\mu$. The matrix form of the FA model is expressed as follows:

$$Z - \mu = \text{Load}_m \text{Factor}_v + \epsilon$$  \hspace{1cm} (67)

where the different vectors are denoted by $Z - \mu$, $\text{Load}_m$ is the factor loading matrix, $\text{Factor}_v$ is the factor vector and $\epsilon$ is the error vector. To assess the factors, likelihood techniques are used in this study.

4.6. KELM

The ELM is a commonly used machine learning technique comprising a single hidden layer feed-forward neural network [47]. A lot of parameters are present in the standard neural network, and adjusting the parameters becomes quite difficult, so back propagation is necessary. The inputs and hidden layer nodes are connected by means of initializing the weights and biases in a random manner by the ELM. Then, in an analytic manner, the
weights are estimated appropriately, and so, the performance and learning efficiency is highly improved. Sometimes, when choosing the hidden layer nodes and in the random start mechanism process, an unstable prediction may happen in the ELM, and to prevent these shortcomings, the KELM was proposed. The expression of SLFN is as follows:

$$f(p) = h(p)\beta = H\beta$$

where the input vector is denoted by \( p \). The weights to fulfil the transformation of the feature are denoted by \( \beta \), and the feature mapping matrix is indicated by \( H \).

\( \beta \) is calculated as follows:

$$\beta = H^T(\frac{1}{C} + HH^T)^{-1}T$$

where the regularization coefficient is denoted as \( C \), and the label data of the training dataset are denoted by \( T \). The expression of the ELM is as follows:

$$f(p) = h(p)\beta = h(p)H^T(\frac{1}{C} + HH^T)^{-1}T$$

The kernel function replaces the feature mapping of the hidden layer so that the KELM is obtained with generalization and stability. The definition of KELM is as follows:

$$f(p) = h(p)H^T(\frac{1}{C} + HH^T)^{-1}T = \begin{bmatrix} k(p, p_1) \\ \vdots \\ k(p, p_N) \end{bmatrix} \left( \frac{1}{C} + \Omega_{KELM} \right)^{-1}T$$

where \( \Omega_{KELM} \) is known as the kernel matrix and is expressed as follows:

$$\Omega_{KELM} = h(p_i)\cdot h(p_j) = k(p_i, p_j)$$

When compared to SVM [48], the generalization performance of the KELM is much better.

4.6.1. Arc-Cosine Kernel Function

Choosing the kernel function is quite important in the KELM. The most used kernel functions are sigmoid, wavelet, polynomial, RBF, etc. In this study, the arc-cosine kernel function is used. Depending on the cosine angle, the computation of vector input similarity has been conducted, and some good results have been obtained [49]. With the help of integral representation, the \( n^{th} \) order kernel is expressed as follows:

$$K_n(p, q) = 2\int dw e^{-\frac{\|w\|^2}{2}}(w\cdot p)\Theta(w\cdot q)(w\cdot p)^n(w\cdot q)^n$$

where the Heaviside step function is indicated as follows:

$$\Theta(l) = \frac{1}{2}(1 + \text{sign}(l))$$

With \( n \) degrees, an arc-cosine kernel family is expressed as follows:

$$k_n(p, q) = \frac{1}{\pi}||p||^n||q||^nA_n(\theta)$$
where the angular dependence is computed by \( A_n(\theta) \) and is expressed as follows:

\[
A_n(\theta) = (-1)^n (\sin \theta)^{2n+1} \left( \frac{\partial}{\sin \theta} \right)^n \left( \frac{\pi - \theta}{\sin \theta} \right), n = 1, 2, 3, \ldots
\]  

(76)

The first 3 expenses of \( A_n(\theta) \) are expressed as

\[
\begin{align*}
A_0(\theta) &= \pi - \theta \\
A_1(\theta) &= \sin \theta + (\pi - \theta) \cos \theta \\
A_2(\theta) &= 3 \sin \theta \cos \theta + (\pi - \theta) (1 + 2 \cos^2 \theta)
\end{align*}
\]

(77)

The similarity of vectors \( p \) and \( q \) is measured by the angle \( \theta \) and is expressed by the following:

\[
\theta = \cos^{-1} \left( \frac{(p \cdot q)}{\|p\| \|q\|} \right)
\]

(78)

When the activation factor has arbitrary values, it implies that various geometric properties are considered by the arc-cosine kernel. There are no corresponding kernel parameters for the arc-cosine kernel function, so no tedious and meticulous pre-setting is required here.

4.6.2. RSO-Based KELM

The kernel parameter and penalty coefficient are important in choosing the KELM algorithm. A better optimization capability is possessed by this RSO algorithm, and so, it is used to optimize the KELM [50]. The fitness function is nothing but the mean square error (MSE), and, in every iteration, the parameters with the minimum error are considered. This is conducted so that the best kernel parameters and penalty coefficient can be obtained. The MSE formula is considered as follows:

\[
MSE = \frac{1}{N} \sum_{t=1}^{N} \left( \hat{m}(t) - m(t) \right)^2
\]

(79)

where the original data are considered as \( m(t) \) and the predictor data are considered as \( \hat{m}(t) \). The predicted sample number is expressed by \( N \). The main procedure of RSO-KELM is expressed as follows:

(a) The features are input and divided into test and training data. The kernel parameter and penalty coefficient of the KELM are obtained as variables.

(b) The rat population size is set as \( ts \). The maximum iteration number is set as \( T \).

(c) The individual position of a rat is assigned as \( \vec{R}_i \), \( i = 1, 2, \ldots, n \).

(d) For every rat, the fitness value is computed.

(e) The best position (individual) of the rat \( \vec{R}_r \).

(f) Assume the present iteration number \( i = 0 \).

(g) The individual position information of the rat is updated.

(h) For every rat, the updated fitness value is computed.

(i) The present optimal position \( \vec{R}_r \) is saved if the present position is better than the previous position.

(j) Assume \( i = i + 1 \). The best position is output as \( \vec{R}_r \).

(k) The algorithm ends; otherwise, the individual position of the rat is updated and the process starts once again.

(l) The individual position of the best rat \( \vec{R}_r \) is given as an output which is nothing but the kernel parameter and best penalty coefficient.

5. Results and Discussion

This work was implemented on a publicly available dataset: http://web.firat.edu.tr/turkertuncer/acute_asthma_cough.rar [51]. From Firat University Hospital, 842 subjects
were considered, including healthy individuals and individuals with conditions such as heart failure, COVID-19 and acute asthma. For the acquisition of cough signals, smart phone microphones were utilized, and signals were saved at a sampling rate of 48 KHz. To avoid redundancy, some of the speech signals were removed manually. A smaller number of segments was obtained when every cough sound signal was subdivided. The parameter settings for the experiments followed in this work are as follows. As far as the BTSA is concerned, the weighing factor was considered as 0.6. The iteration limit was considered as 500, the search space was considered within [0, 10] and the population size was considered as 50. When the ELM is considered, the regularization coefficient was set to 4, and the kernel parameter was set to 200. The gamma value was set to 0.5, and the error tolerance in kernel matrix factorization was set to a trial-and-error approach. The input dimension was present, and the model structure selection was based on it. For the LASSO model, the grid search approach is utilized widely, and it ranges between 0 and 20. As far as SVR is concerned, the input dimension was present, the constant value was set as 0.5 and the gamma value was set as 0.8. When using the concept of autoencoders, the learning rate used in every layer was 0.4, and the learning rate of the fine-tuning was set at 0.2. As far as the hyperparameters in the bagging ensemble learning are concerned, the number of trees was specified as 10, the minimum leaf size of the tree was set as 50 and the number of predictors in each split was set as 75. A 10-fold cross-validation method was used in this work for all the experiments. This work was implemented on Matlab 2020a version, windows 10.1 operating system, 512 GB SDD hard disk, intel i7 CPU and 32 GB RAM.

Table 1 shows the proposed work of robust models with Gabor dictionary analysis and machine learning. Table 2 shows the proposed stacked CAE for audio cough classification. Table 3 shows the proposed model of TQWT, feature selection and classification for audio cough classification. Table 4 shows the proposed model of sparse TQWT, feature selection and classification for audio cough classification. Table 5 shows the proposed model of MIC, feature selection and classification for audio cough classification. Table 6 shows the proposed model of DCC, feature selection and classification for audio cough classification. On examining Table 1, it is evident that a high classification accuracy of 98.78% is obtained when the Gabor dictionary is utilized with an elastic net model and ensemble models. A low classification accuracy of 92.05% is obtained when the Gabor dictionary is utilized with the LASSO model. On examining Table 2, it is evident that a high classification accuracy of 96.35% is obtained when using SCAE, and a low classification accuracy of 91.02% is obtained when using AE. On examining Table 3, it is evident that a high classification accuracy of 98.34% is obtained when AF is utilized with the arc-cosine ELM classifier. A low classification accuracy of 90.31% is obtained when FA is utilized with the LASSO classifier. On examining Table 4, it is evident that a high classification accuracy of 98.99% is obtained when AF is utilized with the arc-cosine ELM classifier. A low classification accuracy of 91.23% is obtained when FA is utilized with the ELM classifier. On examining Table 5, it is evident that a high classification accuracy of 98.86% is obtained when EFA is utilized with the arc-cosine ELM classifier. A low classification accuracy of 90.11% is obtained when FA is utilized with the ELM classifier. On examining Table 6, it is evident that a high classification accuracy of 98.93% is obtained when EFA is utilized with the arc-cosine ELM classifier. A low classification accuracy of 92.23% is obtained when the BTSA is classified with the ELM classifier.

Table 1. Proposed work of robust models with Gabor dictionary analysis and machine learning.

<table>
<thead>
<tr>
<th>Model Configuration</th>
<th>SVR</th>
<th>Ensemble Models</th>
<th>NBC</th>
<th>SVM</th>
<th>Adaboost</th>
<th>RF</th>
<th>DT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gabor dictionary + CCF</td>
<td>96.76</td>
<td>97.64</td>
<td>93.45</td>
<td>94.56</td>
<td>93.25</td>
<td>94.37</td>
<td>93.45</td>
</tr>
<tr>
<td>Gabor dictionary + PCCF</td>
<td>97.21</td>
<td>98.03</td>
<td>94.35</td>
<td>95.32</td>
<td>94.24</td>
<td>95.21</td>
<td>94.22</td>
</tr>
<tr>
<td>Gabor dictionary + LASSO</td>
<td>96.24</td>
<td>97.99</td>
<td>95.21</td>
<td>94.34</td>
<td>94.92</td>
<td>96.01</td>
<td>92.05</td>
</tr>
<tr>
<td>Gabor dictionary + elastic net</td>
<td>95.77</td>
<td>98.78</td>
<td>97.09</td>
<td>97.22</td>
<td>94.67</td>
<td>95.78</td>
<td>93.78</td>
</tr>
</tbody>
</table>
Table 2. Proposed stacked CAE for audio cough classification.

<table>
<thead>
<tr>
<th></th>
<th>AE</th>
<th>CAE</th>
<th>SCAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>91.02</td>
<td>93.45</td>
<td>96.35</td>
</tr>
</tbody>
</table>

Table 3. Proposed model of TQWT, feature selection and classification for audio cough classification.

<table>
<thead>
<tr>
<th>ELM</th>
<th>Kernel ELM</th>
<th>Arc-Cosine ELM</th>
<th>RSO-ELM</th>
<th>SVM</th>
<th>Adaboost</th>
<th>NBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTSA</td>
<td>89.45</td>
<td>92.35</td>
<td>96.89</td>
<td>97.44</td>
<td>95.34</td>
<td>93.23</td>
</tr>
<tr>
<td>AF</td>
<td>91.03</td>
<td>93.45</td>
<td>98.34</td>
<td>98.23</td>
<td>96.34</td>
<td>92.24</td>
</tr>
<tr>
<td>FA</td>
<td>90.31</td>
<td>94.21</td>
<td>97.34</td>
<td>98.01</td>
<td>95.45</td>
<td>93.45</td>
</tr>
<tr>
<td>EFA</td>
<td>92.34</td>
<td>94.99</td>
<td>96.23</td>
<td>98.11</td>
<td>96.22</td>
<td>91.45</td>
</tr>
</tbody>
</table>

Table 4. Proposed model of sparse TQWT, feature selection and classification for audio cough classification.

<table>
<thead>
<tr>
<th>ELM</th>
<th>Kernel ELM</th>
<th>Arc-Cosine ELM</th>
<th>RSO-ELM</th>
<th>SVM</th>
<th>Adaboost</th>
<th>NBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTSA</td>
<td>90.23</td>
<td>93.33</td>
<td>97.11</td>
<td>98.03</td>
<td>97.23</td>
<td>95.34</td>
</tr>
<tr>
<td>AF</td>
<td>92.34</td>
<td>94.21</td>
<td>98.99</td>
<td>98.22</td>
<td>96.23</td>
<td>94.56</td>
</tr>
<tr>
<td>FA</td>
<td>91.23</td>
<td>95.11</td>
<td>98.02</td>
<td>98.01</td>
<td>95.35</td>
<td>93.56</td>
</tr>
<tr>
<td>EFA</td>
<td>93.23</td>
<td>95.01</td>
<td>97.34</td>
<td>97.99</td>
<td>97.01</td>
<td>95.01</td>
</tr>
</tbody>
</table>

Table 5. Proposed model of MIC, feature selection and classification for audio cough classification.

<table>
<thead>
<tr>
<th>ELM</th>
<th>Kernel ELM</th>
<th>Arc-Cosine ELM</th>
<th>RSO-ELM</th>
<th>SVM</th>
<th>Adaboost</th>
<th>NBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTSA</td>
<td>91.32</td>
<td>94.08</td>
<td>95.22</td>
<td>97.34</td>
<td>96.87</td>
<td>96.09</td>
</tr>
<tr>
<td>AF</td>
<td>93.21</td>
<td>92.66</td>
<td>97.34</td>
<td>96.67</td>
<td>95.55</td>
<td>95.67</td>
</tr>
<tr>
<td>FA</td>
<td>90.11</td>
<td>96.78</td>
<td>95.67</td>
<td>95.87</td>
<td>97.67</td>
<td>96.89</td>
</tr>
<tr>
<td>EFA</td>
<td>92.99</td>
<td>93.92</td>
<td>98.86</td>
<td>98.11</td>
<td>96.93</td>
<td>94.22</td>
</tr>
</tbody>
</table>

Table 6. Proposed model of DCC, feature selection and classification for audio cough classification.

<table>
<thead>
<tr>
<th>ELM</th>
<th>Kernel ELM</th>
<th>Arc-Cosine ELM</th>
<th>RSO-ELM</th>
<th>SVM</th>
<th>Adaboost</th>
<th>NBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTSA</td>
<td>92.23</td>
<td>95.09</td>
<td>97.11</td>
<td>98.22</td>
<td>95.11</td>
<td>95.09</td>
</tr>
<tr>
<td>AF</td>
<td>94.46</td>
<td>94.77</td>
<td>98.24</td>
<td>98.36</td>
<td>97.23</td>
<td>94.97</td>
</tr>
<tr>
<td>FA</td>
<td>95.77</td>
<td>97.67</td>
<td>98.75</td>
<td>97.87</td>
<td>96.78</td>
<td>95.56</td>
</tr>
<tr>
<td>EFA</td>
<td>95.89</td>
<td>95.83</td>
<td>98.93</td>
<td>97.98</td>
<td>95.99</td>
<td>93.88</td>
</tr>
</tbody>
</table>

On examining Figure 4, it is evident that a high classification accuracy is obtained for the Gabor dictionary and elastic net combination classified with the ensemble models. On examining Figure 5, it is evident that a high classification accuracy is obtained for the TQWT concept with AF analysis and the arc-cosine ELM classifier. On examining Figure 6, it is evident that a high classification accuracy is obtained from the DCC concept with EFA analysis and classified with the arc-cosine ELM classifier.
Figure 4. Proposed work of robust models with Gabor dictionary analysis and machine learning—Comparative Analysis.

Figure 5. Comparative analysis of TQWT, feature selection and classification for audio cough classification.

Figure 6. Comparative analysis of DCC, feature selection and classification for audio cough classification.
**Performance Comparison with Other Works**

A generalized comparison is given in Table 7 for the reader’s understanding where different techniques have been utilized with various subjects reporting a classification accuracy for various datasets utilized for audio cough classification. However, this work mainly compares its results with the Firat University Hospital dataset [51] that contains 110 individuals with acute asthma, 247 healthy individuals, 241 individuals with COVID-19 and 244 individuals with heart failure, and the results are provided.

Table 7. Performance comparison with other works.

<table>
<thead>
<tr>
<th>Reference Number</th>
<th>Method</th>
<th>Subjects</th>
<th>Accuracy (%) or Other Performance Metric</th>
<th>Mentioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>[17]</td>
<td>MFCC and multi-criteria decision making</td>
<td>622 COVID-19 and 1610 healthy</td>
<td>AUC = 78.00%</td>
<td></td>
</tr>
<tr>
<td>[21]</td>
<td>MFCC and DNN</td>
<td>150 COVID-19, bronchitis, healthy and asthma</td>
<td>90.8% for cough data</td>
<td></td>
</tr>
<tr>
<td>[22]</td>
<td>CNN + LSTM</td>
<td>92 COVID-19 and 1079 healthy</td>
<td>95.3%</td>
<td></td>
</tr>
<tr>
<td>[23]</td>
<td>MFCC and DNN</td>
<td>50 COVID-19 and 50 healthy</td>
<td>97.5%</td>
<td></td>
</tr>
<tr>
<td>[24]</td>
<td>Wavelet Transform + LDA</td>
<td>26 healthy, 17 asthma and 22 chronic obstructive pulmonary disease</td>
<td>90%</td>
<td></td>
</tr>
<tr>
<td>[25]</td>
<td>MFCC and Recurrence Quantification Analysis and XGBoost</td>
<td>20 COVID-19 and 1468 healthy</td>
<td>97%</td>
<td></td>
</tr>
<tr>
<td>[26]</td>
<td>Neural networks</td>
<td>9 asthma and 9 pneumonia</td>
<td>Kappa = 89.00%</td>
<td></td>
</tr>
<tr>
<td>[27]</td>
<td>MFCC + SVM</td>
<td>47 asthma and 48 healthy</td>
<td>77.8%</td>
<td></td>
</tr>
<tr>
<td>[51]</td>
<td>Directed Knight Pattern</td>
<td>110 acute asthma, 247 healthy, 241 COVID-19 and 244 heart failure</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Proposed work</td>
<td>Gabor dictionary with elastic net and ensemble model</td>
<td>110 acute asthma, 247 healthy, 241 COVID-19 and 244 heart failure</td>
<td>98.78%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCAE</td>
<td>110 acute asthma, 247 healthy, 241 COVID-19 and 244 heart failure</td>
<td>96.35%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TQWT + AF + arc-cosine ELM</td>
<td>110 acute asthma, 247 healthy, 241 COVID-19 and 244 heart failure</td>
<td>98.34%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sparse TQWT + AF + arc-cosine ELM</td>
<td>110 acute asthma, 247 healthy, 241 COVID-19 and 244 heart failure</td>
<td>98.99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MIC + EFA + arc-cosine ELM</td>
<td>110 acute asthma, 247 healthy, 241 COVID-19 and 244 heart failure</td>
<td>98.86%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DCC + EFA + arc-cosine ELM</td>
<td>110 acute asthma, 247 healthy, 241 COVID-19 and 244 heart failure</td>
<td>98.93%</td>
<td></td>
</tr>
</tbody>
</table>

On examining Table 7, the proposed work gave very good results when compared to previous works, especially in reference number [51]. Though the previous work proposed a classification accuracy of 100 percent, only a single idea was concentrated on, whereas in this work, three unique ideas were implemented, and a comprehensive analysis was performed and reported. The highest classification accuracy of 98.99% was obtained when sparse TQWT with AF was implemented with the arc-cosine ELM classifier. The second highest classification accuracy of 98.93% was obtained when DCC with EFA was implemented with the arc-cosine ELM classifier. The third highest classification accuracy of 98.86% was obtained when MIC with EFA was implemented with the arc-cosine ELM classifier. The advantage of the proposed model is that it utilizes the intrinsic properties of all the hybrid models, thereby taking advantage of all the wonderful properties of each algorithm, and a good result is produced every time. The disadvantage of the proposed model is that one must be careful in choosing the various models or algorithms, as certain combinations can produce extremely good results, and some combinations can produce
very poor results. As a result, a lot of trial-and-error combination possibilities must be tried out, which might require a lot of time and energy from the researchers.

6. Conclusions and Future Works

In machine learning, one of the hot research issues is the cough-based detection of diseases. As the COVID-19 pandemic affected the lives of people very badly, the innate signs of COVID-19 may resemble other medical conditions such as kidney problems, lung problems, heart failure problems, etc. With the advent of AI techniques, many wonderful applications have been implemented in the field of medicine. In this work, cough sound signals are used for disease detection with the help of some algorithms and machine learning techniques. Various computer-aided diagnostic methods are used for the automated detection of many disorders, and so, in this work, these nice strategies are developed using systematic conglomerated models. The best results were produced when sparse TQWT with AF was implemented with the arc-cosine ELM classifier, producing a classification accuracy of 98.99%. The second highest classification accuracy of 98.93% was obtained when DCC with EFA was implemented with the arc-cosine ELM classifier. The use of advanced AI technique chip-based diagnostic models can be developed in the future too. This work can also be extended to telemedicine and cloud-based applications by means of applying it directly to embedded systems. This work can be useful for pulmonology clinics where it can largely assist the physician. The work strategy can also be implemented with other signal processing techniques in the future. Work can also be planned to implement large audio cough datasets in the future.

Author Contributions: S.K.P.—Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Resources, Data Curation, Writing—Original draft preparation; D.-O.W.—Validation, Formal Analysis, Writing—review and editing, visualization, supervision, project administration, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A5A8019303) and partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-02068, Artificial Intelligence Innovation Hub).

Data Availability Statement: The data is publicly available online at http://web.firat.edu.tr/turkertuncer/acute_asthma_cough.rar.

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

References


35. Kandiri, A.; Shakor, P.; Kurda, R.; Deifalla, A.F. Modified Artificial Neural Networks and Support Vector Regression to Predict Lateral Pressure Exerted by Fresh Concrete on Formwork. *Int. J. Concrr. Struct. Mater.* 2022, 16, 64. [CrossRef]


42. Shuliang, W.; Surapunt, T. Bayesian Maximal Information Coefficient (BMIC) to reason novel trends in large datasets. *Appl. Intell.* 2022, 52, 10202–10219. [CrossRef]


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.