Mechanical Engineering Project

[ME5001] – Data Mining for the Design of acoustic absorbers

Prediction of Sound Absorption Coefficient

by Artificial Neural Network, Radial Basis Function Neural Network and k-Nearest Neighbor

By

Jin Hoontae

Thesis supervisor:

Prof. Lee Heow Pueh

Department of Mechanical Engineering

National University of Singapore

Submitted on 20, April, 2022

Abstract

Machine Learning (ML) methods have recently paved a way for computing desired targets faster and accurately in the acoustic field, as compared with the phenomenological approach such as Hamet model. While the phenomenological approach significantly requires prior in-depth knowledge of acoustical concepts and the time spent for computing the target values by Finite Element Analysis (FEA) software is huge, ML does not usually require prior knowledge on the specific field and is capable of computing the target values precisely and quickly. This thesis delves into three conventional ML methods (k-Nearest Neighbor (kNN) and Artificial Neural Network (ANN), Radial Basis Function Neural Network (RBFN)) and presents their effectiveness for obtaining target values only using simple input parameters (thickness and hole size). Firstly, a dataset composed of 776 discrete sound absorption coefficients from 50 Hz to 1600 Hz at an interval of 2 Hz is analyzed and obtained by using the impedance tube. The processed dataset is then split into three datasets containing 200, 300 and 400 samples to examine how the prediction accuracy of the ML methods is improved with respect to the amount of dataset. The results show that the number of samples extremely affects the quality of the prediction performance. Moreover, the quality of the models can be further upgraded by conducting the hyperparameter optimization. However, although the ML methods are proven to be accurate for predicting new samples, they sometimes fail to predict with huge deviations, especially at mid-and high- frequencies. The complexity of the coefficient curve results from the stacking location of the disks. After discovering that the ANN model always works better than the RBFN and kNN models, the former is used to carry out the optimal configuration task. The correlation of sound absorption coefficients between the ANN-predicted target values and true target values is positively linear with the Pearson's correlation coefficients greater than +0.9 and significantly close to +1.0 in the frequency range of 600 Hz, which demonstrates that the ANN model can be effectively employed to predict the target values at low frequencies.

Key words: Machine Learning (ML), k-Nearest Neighbor (kNN), Artificial Neural Network (ANN), Radial Basis Function Neural Network (RBFN), Sound absorption coefficient, Impedance tube

Acknowledgment

This project was carried out at the Department of Mechanical Engineering in the National University of Singapore in Singapore, as a compulsory requirement for acquiring the Master of Science degree.

I would like to thank my supervisor Prof. Lee Heow Pueh for giving me this valuable opportunity to work on this project and providing me with excellent guidance and abundant resources to complete it successfully. His insight and guidance have indeed been of great value to me throughout the project of this year.

I would like to extend my special thanks to Dr. Sanjay Kumar and Mr. Jie Wei for sharing informative ideas and recommendations to improve the quality of the project, and for sourcing sample materials which was an enormous help to this thesis.

Heartfelt thanks are also extended to my parents because without their sincere support this project would not have been successful throughout my MSc degree.

Symbols and Abbreviations

Key Symbols

x_j	independent/input variables
y_j	dependent/output variables
b	bias
k	number of neighbors
w	weight
γ	learning rate

Operators

$\frac{d}{dw}$	derivative with respect to *w*
$\frac{\partial}{\partial w}$	partial derivative with respect to variable *w*
$\nabla f(x)$	gradient of function f(x)
Σ	sigma Operator

Abbreviations

ANN	Artificial Neural Network
BP	Back Propagation
FNN	Feedforward Neural Networks
GD	Gradient Descent
kNN	k-Nearest Neighbor
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MSE	Mean Square Error
NN	Neural Networks
RBFN	Radial Basis Function Neural Network
RMSE	Root Mean Square Error
SGD	Stochastic Gradient Descent
SLS	Selective Laser Sintering
SMM	Single Microphone Method
TMM	Two Microphone Method

Table of Contents

1. Introduction

One of the widespread environmental disturbances is noise, which enormously affects the lifestyle of humans and wildlife [1]. Noise emissions caused by aircraft, traffic, and air conditioners are within the range of low-frequency sound waves, and such emissions oftentimes attenuate more slowly with the long-distance transmission, which can adversely affect human health such as sleep disorders, stress, hearing loss, etc. [2-6]. Constructing an effective solution to mitigate low-frequency noise, hence, remains a great challenge in engineering acoustics. In fact, it is not sufficient to use natural structures such as trees or man-made structures such as walls to prevent the noise emissions of low-frequency from propagating further [7-8].

Designing perforated acoustic panel absorbers with materials that are excellent in absorbing sound is widespread due to their effectiveness to control noise levels [9]. Porous materials are comprised of numerous holes or cracks, and when used for acoustic performance, sound transmission loss can be achieved effectively as the travel of sound waves is persistently hindered by the viscosity of their complex structures [10-11]. Despite the advantages that porous materials hold, the performance can only be successful in the range of high frequencies [12]. Such a deficiency can be rectified by using the multi-layer stacking method [13]. Multi-layered materials specifically designed for noise control display two main features: An excellent sound transmission loss from an exterior to an interior area, and an ability to control the sound absorption coefficient [14]. Thus, it is especially important to devise methods to accurately predict the performance of such materials to properly design indoor noise systems. To date, multiple studies have been carried out to predict the transmission of sound waves in acoustical materials according to different material characteristics such as rigidity, elasticity [15-19], or elastic porosity [20-21]. To predict the behavior of the sound absorption coefficient within certain frequency ranges, acoustical or non-acoustical parameters need to be known.

To enhance the vibro-acoustic functionality of the materials being investigated, determining the right acoustic, mechanical, and physical parameters is crucial as its properties are significantly correlated to them [22-24]. The performance improvement of the systems is achieved based on the sound absorption coefficient, as it is known to be a fundamental attribute to the quality of sound absorption [25]. An extended phenomenological model suggested by Hamet et al. [26] is developed on two elements: viscosity and heat dissipation. Prior knowledge of three parameters (structure factor, resistivity, and porosity) is needed to proceed with the

Hamet method thereby computing the absorption coefficient of the sample [27]. Unlike the Hamet method in which developing complex mathematical models is required, which could display a limit as to simulating the acoustic performance of multiple materials, Machine Learning (ML) could be a solution to compute the output properly not being restricted by different natures of materials.

ML has allowed significant advances in extensive fields such as computer vision, speech processing, image processing, and physical science due to its excellent capabilities of automated pattern recognition and data processing [28-31]. ML has been of great interest in engineering acoustics as well because it has the significant potential of predicting the desired acoustic outputs by automatically analyzing patterns in data based on the given input parameters. Hence, statistical knowledge is paramount as it is the main factor that predicts precise patterns of the desirable outputs instead of using particular domain knowledge [32].

2. Outline

In this study, the feasibility of three ML techniques is investigated for predicting the absorption coefficient of multi-layered samples given only two non-acoustical input parameters (hole diameters and thicknesses). This paper begins by explaining the fundamental theories of ML techniques and a brief mechanism of the impedance tube in **Section 3**. In **Section 4**, the sample preparation and experimental procedure for computing the absorption coefficient are discussed. Following the experimental procedure, the measures used to examine the efficiency of the ML models and the construction of the models for determining the hyperparameters are explained in detail. In **Section 5**, the predicted sound absorption coefficient created by the ML techniques is compared with the measured/actual values based on the performance metrics and the predicted graphs, and, subsequently, the best model selected during the analysis is further examined for the optimal configuration based on the correlation plots and Pearson’s correlation coefficient values. Finally, in **Section** 6, the results discussed earlier are summarized with a possible improvement that can be made for better prediction.

3. Background

3.1. Introduction of Artificial Neural Networks

Utilizing the prediction model, widely known as neural networks, helps estimate the acoustic performance of multiple materials with different configurations, resulting in an enormous saving of resources and time to carry out acoustic measurements [33-35]. Analyzing the correlation between the material characteristics and its sound absorption coefficient can be performed by finding a suitable model geometry and structure due to the notable advantage of Artificial Neural Networks (ANN) where the acoustic behavior of the material can be predicted based on the parameters used [36]. The machine learning method efficiently operates a high level of industrial processes based on the data obtained during the training process [37-38]. Utilizing the dataset collected from the specimens, general specifics and models can be extracted, which is performed during the training phase and the data are subsequently used in the test phase in order to make predictions with the extracted knowledge [39]. Supervised learning provides a strategy to develop artificial intelligence, where an algorithm based on machine learning is continuously trained on input data paired with a specific output. The algorithms are trained until the models recognize inherent patterns and correlations between the input and the output data. Once successfully trained, accurate results can be predicted with the new dataset never seen before. An artificial neural network (ANN) can be defined as a mathematical model designed to simulate the functioning of a human brain and nervous system. The ANN model holds several features: its flexible adaptability to simultaneous changes and its ability to detect underlying patterns in complex systems. Hence, with these features combined, its unique self-learning capability is constructed [40-41]. The ANN has been extensively used for nonlinear systems to achieve pattern recognition, system identification and regression [42]. Following the pattern of neurons in the brain, ANN architecture consists of three layers: An input layer where the external data are received for pattern recognition, an output layer that produces the predictions and solutions, and a single or multiple hidden layer(s) located between the input and output layers. The parameters, results and complexity of the dataset determine the number of input, hidden and output neurons [43]. To date, multiple studies have been performed to estimate the aspect of sound energy travel in acoustic materials by developing elastic porous materials [44-45].

3.2. Input, Output and Hidden layers

Neural networks are modeled to duplicate the mechanism of the human brain, comprised of numerous groups of neurons interconnected by axons, leading to the formation of a network. [33]. The functioning of this technique can be exercised through a computational work between input variables and an activation function, which produces the output of the neuron [46]. The nature of neural networks enables non-linear mathematical transformation from a set of independent input variables, $x = (x_1, x_2, \dots, x_n)$ into a set of dependent output variables $y = (y_1, y_2, \dots, y_n)$. The results predicted are dependent on a set of values, $w = (w_1, w_2, \dots, w)$, named weights. The process can be formulated through the following equation:

$$y = f(\sum_j x_j \cdot w_j + b) \qquad [1]$$

where f is the activation function, x_j is the jth input, w_j is the jth weight, b is the bias, and y is the output.

The weights and bias include the information that the neurons obtain during the training process. To calculate accurate weight values in the neural networks, the Backpropagation (BP) algorithm technique should be adopted. During the Backpropagation, the gradient descent method is used to seek and minimize the error function dealing with the weights [47]. Hence, appropriate values of weights and biases can be computed through an iterative procedure, which often requires significant computational time and effort, leading to the minimization of the error values [48]. The activation function is a threshold function in which only neurons with valid signals that meet the threshold condition are activated, and the activated neurons progress to the next neuron(s). The representative activation functions are as follows: Sigmoid, Nonlinear stepped, or logistic functions [49]. Thus, the complexity of a neural network is accomplished by three types of layers: Input layer, Hidden layer(s), and Output layer.

The external signals are transmitted from the input layer to the output layer, through each neuron of the hidden layer(s), as demonstrated in **Fig 1**. Any received signal from one node to another node is constantly processed by the transfer function and the results are transmitted to the subsequent layers.

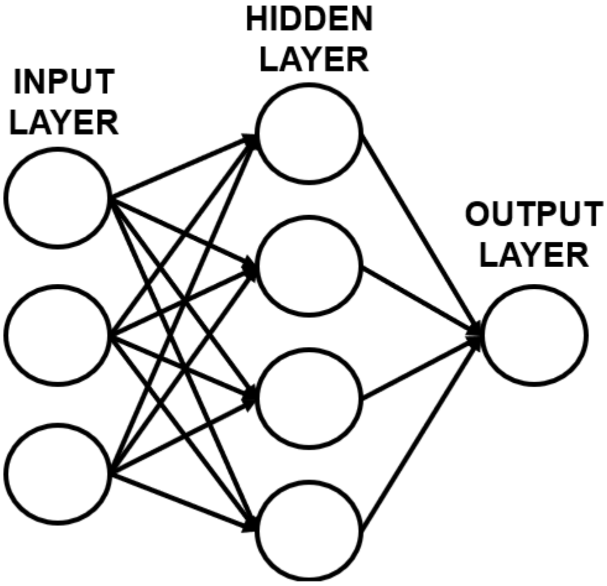


Figure 1. The architecture of neural networks with input, hidden, and output neurons/layers

Figure 2 [14] demonstrates a more detailed schematic diagram of the feedforward neural network, which shows how the previous neurons are transferred to the next nth neuron in the next layer. *Activation function*, *f*, will be discussed in more detail in **Section 3.5**. The feed-forwarded values of weights and bias will then be optimized using the gradient descent method, which will be explained in **Section 3.3**.

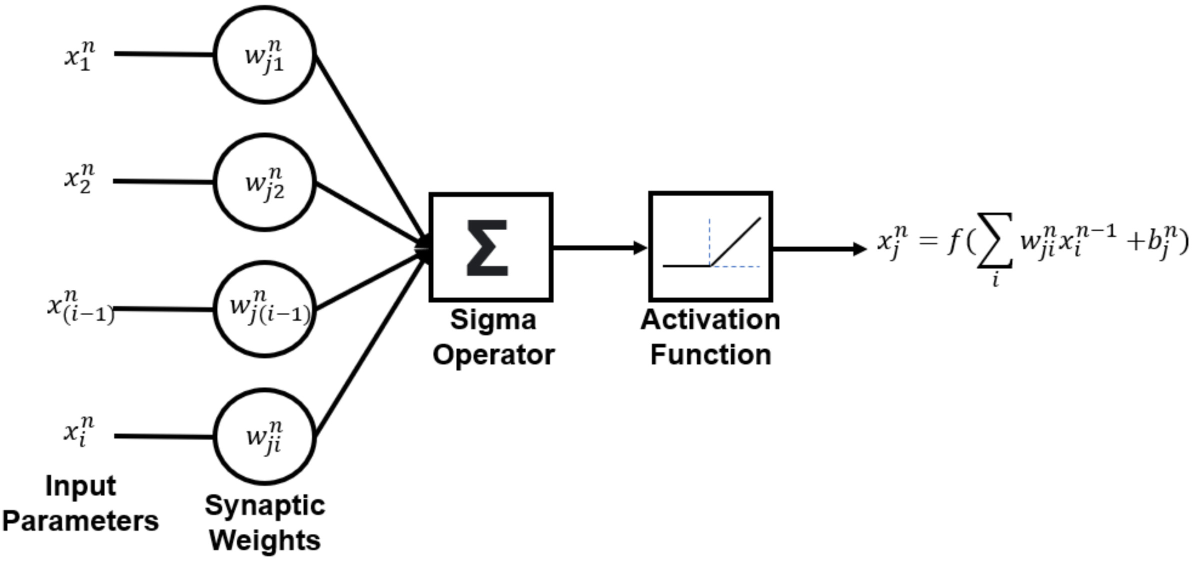


Figure 2. Detailed schematic diagram of the feedforward neural network

3.3. Gradient Descent

Minimization of the empirical risk $E_n(f_w)$ is often achieved using the gradient descent (GD) method [50]. Based on the gradient of $E_n(f_w)$, the weight values are updated iteratively, which can be expressed as:

$$w_{t+1} = w_t - \gamma \frac{1}{n} \sum_{i=1}^{m} \nabla_w Q(z_i, w_t) \qquad [2]$$

where γ is the learning rate ($\in R_+$), which controls the step size. The gradient of the cost is obtained using the derivative chain rule, which will be discussed in more detail in **Section 3.4**. When the initial estimate w_0 is adequately close to the optimality, and when the learning rate is properly chosen in accordance with the system, the algorithm realizes linear convergence [51]. The convergence can be found at the local minimum point of the function as shown in **Fig 3**. Each iterative step is taken continuously proportional to the negative direction of the gradient at the current coordinate until the local minimum is reached, which optimally minimizes the cost function given.

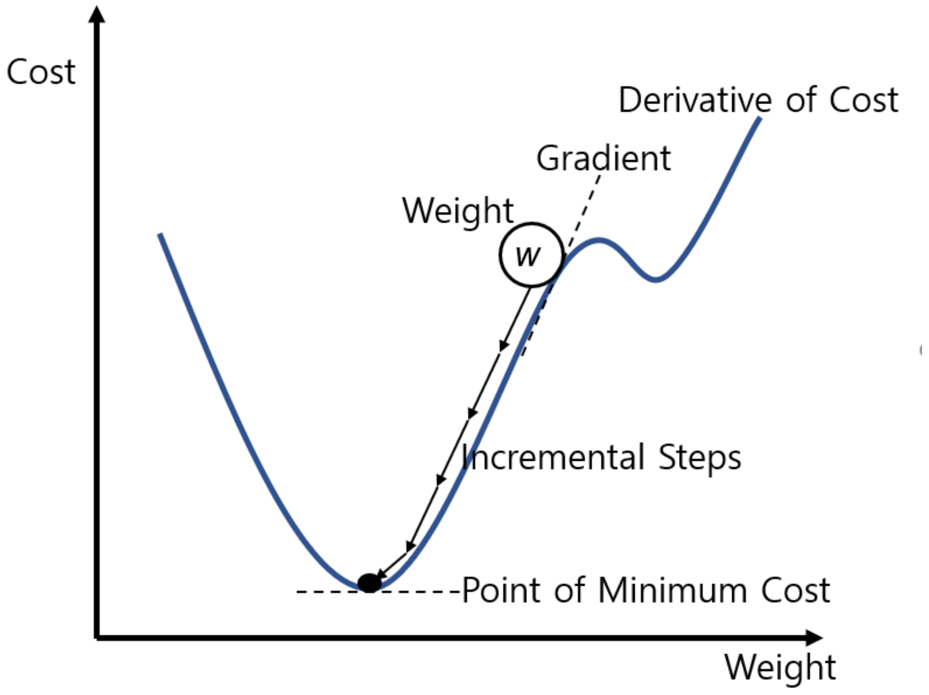


Figure 3. An example of the GD algorithm for finding the local minimum

The convergence speed is significantly affected by the learning rate γ as it directly determines the speed of tracing the path in the function space, and it is greatly advisable to select an appropriate learning rate [52]. For example, when the learning rate is significantly large, the algorithm traces its path with a big step downward iteratively, meaning that it has a high tendency of missing the local minimum as shown in **Fig 4 (b)**. Moreover, when the learning rate is overly small, the computation time would significantly increase, which can be a problematic issue when computing a large amount of data. Empirically, the learning rates of 0.1, 0.01 and 0.001 are known to perform well [52].

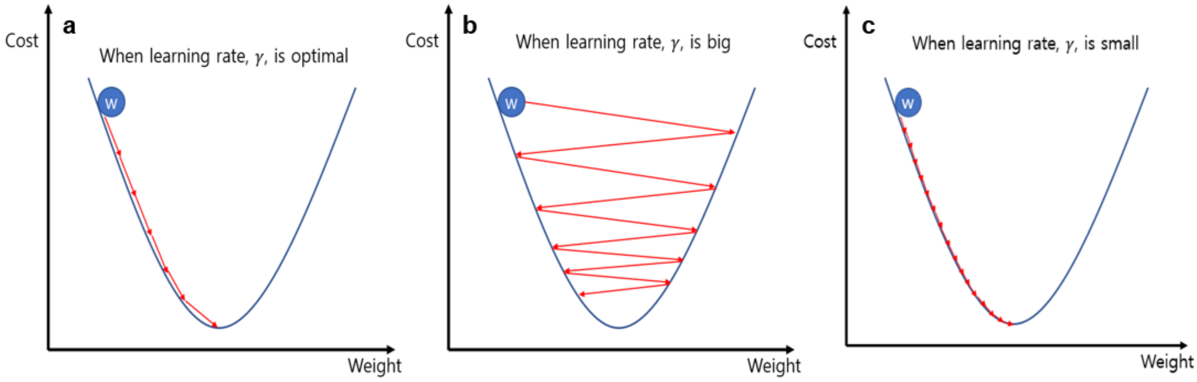

a
Cost
When learning rate, γ, is optimal
w
Weight
b
Cost
When learning rate, γ, is big
w
Weight
c
Cost
When learning rate, γ, is small
w
Weight

Figure 4. Three general cases of the minimum gradient tracking with respect to γ

3.3.1. Stochastic Gradient Descent

The stochastic gradient descent (SGD) method is a drastically simplified form of GD, which helps the model to decrease the computation time significantly, which can be expressed as:

$$w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t) \qquad [3]$$

Unlike the GD method in which the exact gradient of $E_n(f_w)$ needs to be computed, SGD approximates the gradient iteratively according to a randomly selected example of z_t [53]. Hence, this algorithm achieves a faster convergence rate, which can be an effective alternative for conventional GD algorithms. However, if the N-dimensional graph contains a complicated shape with numerous noisy curves, the computation speed of approximating the optimum convergence point would be adversely hindered. To rectify such a disadvantage that SGD holds against rather complicated graphs, it is important to control the learning rate γ_t appropriately since the estimate w_t is crucially affected by the learning rate. For example, if the learning rate decreases too marginally, the variance of w_t also decreases accordingly slowly. If the learning rate increases too drastically, it would take a considerable time for the expectation of the estimate w_t to reach the optimum point. It was proposed by Murata [54] that the optimal convergence speed can be realized using the learning rate $\gamma_t{\sim}t^{-1}$ under the assumption that sufficient regularity conditions are given. Therefore, in setting an appropriate learning rate $\gamma_t{\sim}t^{-1}$, the early iterations would take large step sizes to descend relatively quickly, and the late iterations would slow down in order to obtain the local minimum, which can significantly reduce the computation time as opposed to the conventional gradient descent method.

3.3.2. Adam Optimizer

Adam Optimizer is known to compute GD-related solutions much faster than the SGD [55]. During Adam's computation process, the first and second moments of the gradients are calculated. The two moments of the gradients are then used to obtain moving averages that are constantly updated with the gradually increasing data points. Its process can be expressed by the following equations [56]:

$$m_w^{(\boldsymbol{t+1})} = B_{\boldsymbol{1}} m_{\boldsymbol{w}}^{(\boldsymbol{t})} + (1 - B_{\boldsymbol{1}})\nabla_w Q(z_t, w_t) \tag{4}$$

$$v_w^{(\boldsymbol{t+1})} = B_{\boldsymbol{2}} v_w^{(\boldsymbol{t})} + (1 - B_{\boldsymbol{2}})\nabla_w Q(z_t, w_t) \tag{5}$$

$$\widehat{m_w} = \frac{m_w^{(\boldsymbol{t+1})}}{1-B_{\boldsymbol{1}}^{(\boldsymbol{t+1})}} \tag{6}$$

$$\widehat{v_w} = \frac{v_w^{(\boldsymbol{t+1})}}{1-B_{\boldsymbol{2}}^{(\boldsymbol{t+1})}} \tag{7}$$

$$w_{t+1} = w_t - \gamma_t \frac{\widehat{m_w}}{\sqrt{\widehat{v_w}}+\epsilon} \tag{8}$$

The moving average's damping ratio is adjusted by the hyperparameters of ML ($B_{\boldsymbol{1}}$ and $B_{\boldsymbol{2}}$). The first and second moments of the gradients are m_w and v_w, respectively, and ϵ is a small real-valued number to avoid division by zero.

3.4. Back-propagation

One of the most widely applied algorithms in machine learning is Back-Propagation, which is used for training the feedforward architecture of the neural networks (NN) [57]. In the feedforward neural networks (FNN), an output $\boldsymbol{y}$ is produced by an input $\boldsymbol{x}$. The initial information contained within the input $\boldsymbol{x}$ propagates forward to the units in the hidden layer(s) and eventually produces the output $\boldsymbol{y}$ [58]. BP algorithm is used to calculate the gradient of the loss function in terms of the weights $\vec{w}$ to better fit the NN, which, unlike a conventional approach of directly computing each gradient individually that only works in a single input-output system, enables the computation of the gradient for more complex, namely, multi-layer NN more efficiently. A combination of BP and various GD methods is often utilized to properly update weights and to minimize the loss. In BP, the gradient of the loss function is accordingly computed using the chain rule theory, and each iteration is processed backward from the output layer to the input layer to prevent unnecessary computations of intermediate terms in the chain rule [58].

To perform the BP task, three factors are generally required:

• Dataset: It needs to contain pairs of input and output $(\vec{x_\iota}, \vec{y_\iota})$, where $\vec{x_\iota}$ is the set of inputs and $\vec{y_\iota}$ is the set of outputs. The whole set of input-output pairs of size N can be written $X = \{(\vec{x_1}, \vec{y_1}), (\vec{x_2}, \vec{y_2}), \dots, (\vec{x_N}, \vec{y_N})\}$

• A feedforward neural network (FNN): The parameters of the neural network can be denoted θ with respect to the neural network's weights and biases, which is iteratively collected in gradient descent

• An error function, $E(X, \theta)$, is defined as the error between the true output $\vec{y_\iota}$ and the predicted output $\widehat{y_\iota}$ in the neural network with respect to the set of the input-output pairs and a particular value of the FNN parameters θ. It can be denoted:

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^{N} (\widehat{y_\iota} - \vec{y_\iota})^2 \qquad [9]$$

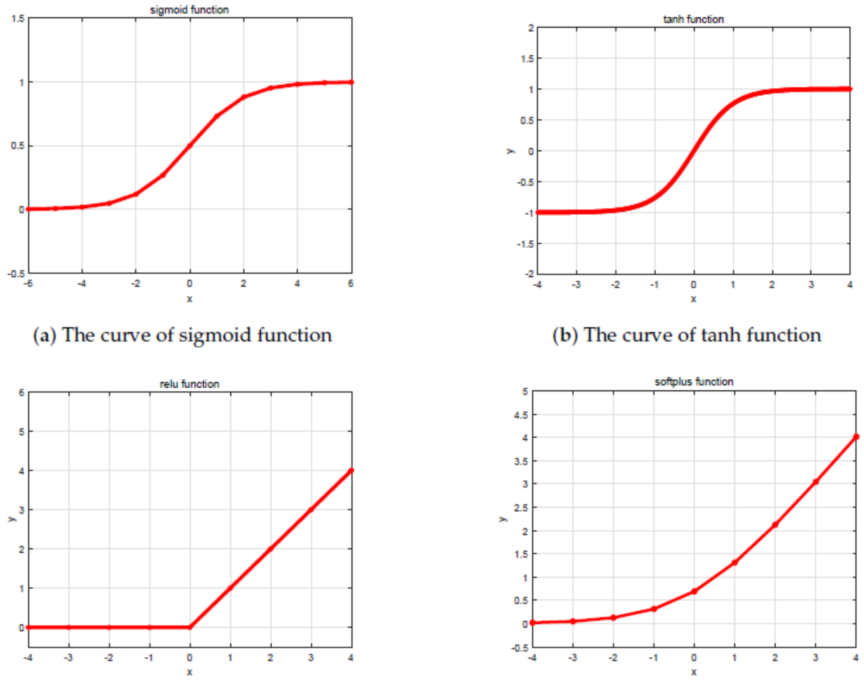
The sum of the partial derivatives of the error function for each input-output pair for weights can be written as:

$$\frac{\partial E(X, \theta)}{\partial w_{ij}^{k}} = \frac{1}{N} \sum_{d=1}^{N} \frac{\partial E_d}{\partial w_{ij}^{k}} \qquad [10]$$

where w_{ij}^{k} is a weight for neuron j in layer k for incoming neuron i. By minimizing the error function iteratively, hence, the values of weights and biases continue to be updated and the optimal relationship between each neuron can be constructed thereby computing the fine predicted target values eventually.

3.5. Activation Functions

The activation function, f, in **Eq.1**, is the core of the NN architecture for solving nonlinear problems by activating the feature of neurons [59], and the data-mapping performance in N-dimensions is significantly affected using a proper activation function [60, 61]. There are multiple activation functions being used widely due to the effectiveness of their non-linear problem-solving abilities; that is, Tanh, Sigmoid, ReLu and Softplus, which can be seen in **Fig 5**.



(a) The curve of sigmoid function

(b) The curve of tanh function

(c) The curve of ReLu function

(d) The curve of softplus function

Figure 5. The graphs of the four most common activation functions [59]

Of the activation functions introduced, in this paper, ReLu function is used as it has been proven to provide a faster computation time. Furthermore, unlike Sigmoid and tanh functions, the ReLu function does not encounter any gradient diffusion problem during processing; that is, the other two functions produce the gradient close to zero when the input value is significantly large or small. On the other hand, ReLu is a piecewise function in which the input value (≤0) will force the output to be zero, otherwise, the output value will be equal to the input value. The two major benefits of ReLu, which are 1) the reduced likelihood of the gradient to be zero (when $a > 0$) and 2) a sparse characteristic of forcing data to be zero (when $a \leq 0$), result in a significantly reduced computation time. The equation can be expressed as:

$$f(x) = \max\,(0, a) \qquad [11]$$

where $a = f(\sum_j x_j \cdot w_j + b)$

3.6. Radial Basis Function Neural Network (RBFN)

The radial basis function neural network (RBFN) is a three-layer feedforward network as shown in **Fig. 6**. Unlike the ANN, the RBFN cannot have more than one hidden layer and does not require the BP method due to its unique structural characteristic. The number of neurons in the input layer is determined by the number of input parameters. In the RBFN, the input values are transmitted straight to the hidden layer as there is no weight connection between them.

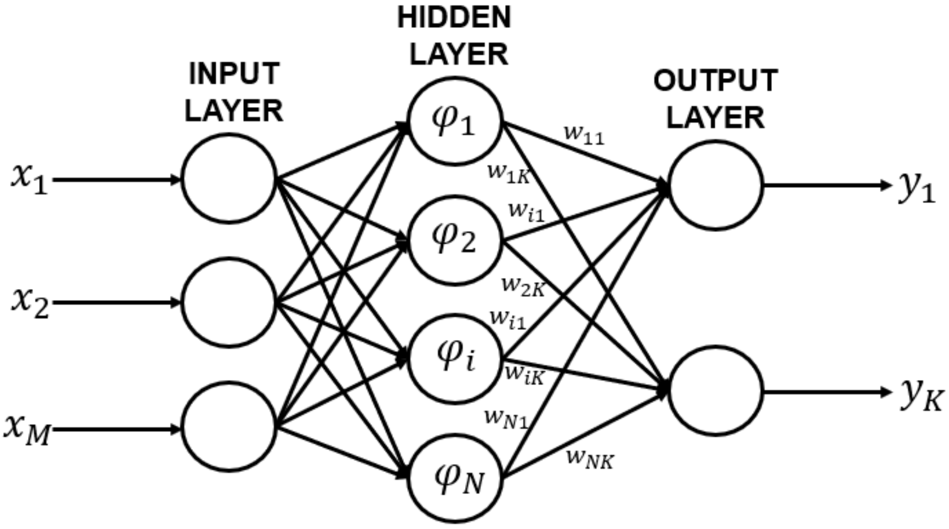


Figure 6. The structure of the RBFN

In the hidden layer, the number of neurons is significantly dependent of the complexity of the problem. The activation function, embedded in each hidden neuron, is the radial basis function. Its function is activated when the input neurons enter the hidden layer, and subsequently, it attempts to discover a surface in the hyper-dimensional space that corresponds to the problem. The found surface is then utilized to process the test data, which makes the linearly indivisible problem linearly separable [62]. Among many radial basis functions, a Gaussian function is mostly used and can be expressed as [63]:

$$\varphi(\left||X_k - X_i|\right|) = \exp\left(-\frac{1}{2\sigma^2}\left||X_k - X_i|\right|^2\right) \qquad [12]$$

where X_k is the input data of the kth sample, X_i is the center of the radial basis function of the ith neuron in the hidden layer, σ is the spread.

As the cost function and the weight update are not taken into consideration in this algorithm, the computation cost is extremely cheap. Once the RBFN structure has been established, only three factors are considered: the weights between the hidden and output layers, the center and the spread of the radial basis function. The role of the spread is to display the decaying speed of the function. The slope of the Gaussian function needs to be controlled desirably as it affects the prediction accuracy considerably, which can be denoted as:

$$\sigma = \frac{d_{max}}{\sqrt{2n}} \qquad [13]$$

where d_{max} is the maximum distance between the chosen centers, n is the number of hidden neurons.

Then, the pseudo-inverse method needs to be used to compute the weight matrix (w) located between the hidden layer and the output layer. The weight matrix can be expressed as below [64]:

$$w = \boldsymbol{G}^{+}d \qquad [14]$$

where d is the expected output, $\boldsymbol{G}^{+}$ is the pseudo-inverse matrix of $\boldsymbol{G}(=(\boldsymbol{\varphi}^{\boldsymbol{T}}\boldsymbol{\varphi})^{-1}\boldsymbol{\varphi}^{\boldsymbol{T}})$, which is the output value of the input vector computed by the radial function $\varphi(\left||X_k - X_i|\right|^2)$.

3.7. k-Nearest Neighbors (kNN)

The kNN (k-Nearest Neighbor) is considered an effective ML approach for both classification and regression problems. The algorithm goes through a process by which new data instances are classified according to the labels of the existing data [65]. Feature vectors are used as inputs for each data instance in kNN models, and a class can be formed with input data instances if they have similar features vectors. When new data instances are fed into the algorithm during the application, the model assigns the data instances to each desirable class that has the most similar feature vectors. The sensitivity of checking the similarity between classes and new data instances is determined based on a special distance measure set by the user. The number of neighbors, referred to as *k*, is also an important parameter that affects the classification/regression behavior of the model. Considering some anticipated noise or outliers in the data, it is essential to compare new data instances to multiple instances. That is to say, the number of classes being compared to a single new data instance is denoted *k*. The two features (the distance measure and *k*) are regarded as hyperparameters of the model. Hence, in order for the model to perform a successful ML task, it is crucial to determine appropriate settings of the hyperparameters. **Fig 7** demonstrates how new data instances are classified when there are two classes and *k* is 2.

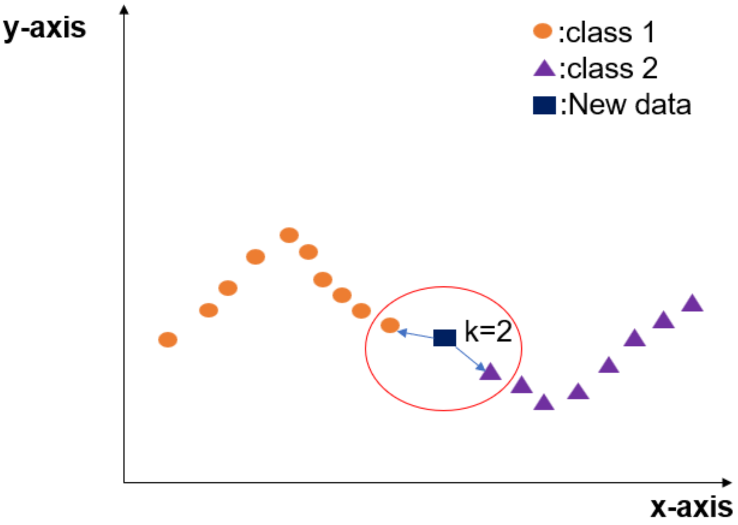


Figure 7. KNN Classification diagram when there are two classes and *k* is 2

3.8. Supervised Learning

In supervised learning, a dataset is required to contain labeled input parameters (features) and outputs (targets). The inputs are interpreted as independent variables that characterize each sample, whereas the outputs are dependent variables that define some property. A certain percentage of the dataset is fed into the training algorithm where an inferred function is produced, which can be subsequently used for mapping/predicting new targets. By comparing the predicted targets with the “known” targets in the testing phase, the statistical quality of an algorithm can be evaluated. Hence, the optimal algorithm needs to be able to produce correct target values with respect to new input parameters.

The input features can be formed as multiple types: Real numbers, integers, categorical, or nominal values. These features can also be arranged in a variety of ways such as a concatenated single vector containing all the features, 2D matrix for images, or multichannel 2D matrices comprised of complex tensors. On the other hand, categorical or numerical values are usually used for output targets, and, depending on the purpose of the experiment, there can be a single target or multiple targets. If target values are numerical, a regression method is utilized, whereas a classification method is for categorical targets. **Fig. 8** shows the general flowchart of supervised learning. As can be seen from the figure, its primary goal is to predict precise outputs by training the data given.

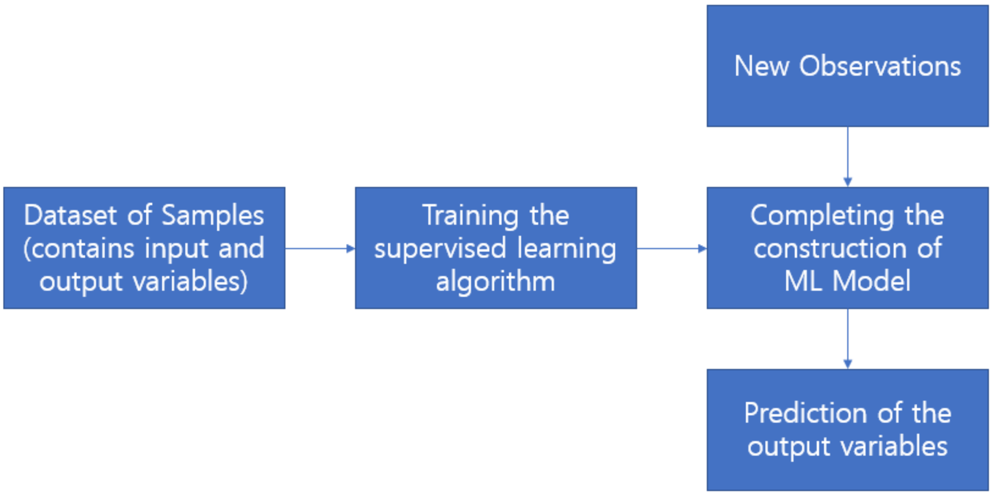

New Observations
Dataset of Samples (contains input and output variables)
Training the supervised learning algorithm
Completing the construction of ML Model
Prediction of the output variables

Figure 8. A brief flowchart of supervised learning

3.8.1. Regression

A statistical model, known as *regression*, is an effective statistical process used to analyze the relationship between a dependent variable (*output* or *target)* and independent variables (*input* or *features*) [66]. Its fundamental goal is to build an analytical prediction model that can approximate a function *f* so that, with respect to the input values, it is feasible to produce a desirable output $\hat{y}$ such that $\hat{y} = f(x)$ and $y = \hat{y}$ where y is the true target. Furthermore, multiple target values can also be predicted such that a brief formulation, $f: R^d = R^m$, can be made where d is the number of input features for each observation, and m is the dimensionality of the outputs.

3.10. Acoustic Absorption in Impedance Tube

An impedance tube is a device that can effectively measure acoustical properties of materials such as acoustic absorption coefficient, sound transmission loss and acoustic impedance. A loudspeaker mounted at one end of the closed tube is utilized to measure such properties by emitting a tone constantly for a short period of time. Acoustic absorption can be defined as a process by which multiple factors such as a material or structure of the object absorb the sound energy when sound waves propagate, instead of reflecting the energy. The absorbed energy is then either transformed into heat or penetrates through the object's body. The heat energy created during the process is considered 'lost' [67]. Typically, for the design of sound absorbers, porous materials are much preferred as they are capable of transforming a large amount of sound energy into heat energy. Hence, in order to analyze the efficiency of sound absorbers, the sound absorption coefficient is generally measured, which describes the ratio of the acoustic power absorbed by the object and the incident acoustic power [68]. It ranges from 0 to 1 where 0 represents a surface that can fully reflect the energy, and 1 represents a surface with the capability of full acoustic absorption. Since the impedance tube is built to an acoustically closed system, some amount of the incident wave propagating inside the system must be absorbed if not reflected by the material. The sound absorption measured by sound wave propagation can be formulated as [68]:

$$\alpha = \frac{P_\alpha}{P_I} = 1 - |r|^2 \qquad [15]$$

where P_α is the absorbed sound power, P_I is the incident sound power, and r is the reflection coefficient. For a 1-D acoustic wave that travels through the absorbing material in the normal direction, the reflection coefficient can be computed using the transform function.

$$H_{12} = \frac{p_2}{p_1} = \frac{e^{jkx_2} + re^{-jkx_2}}{e^{jkx_1} + re^{-jkx_1}} \qquad [16]$$

where p_1 is the sound pressure of a microphone located at the position x_1, p_2 is the sound pressure of the other microphone located at the position x_2 on the impedance tube, and k is the wave number. Computing **Eq. 16**, the reflection coefficient can be obtained as below:

$$r = \frac{H_{12} - H_I}{H_R - H_{12}} e^{2jkx_1} \qquad [17]$$

where $H_I = e^{jk(x_2-x_1)}$: The transfer function of the reflected wave, and $H_I = e^{jk(x_1-x_2)}$: The transfer function of the incident wave [69].

The maximum frequency created by the propagating 1-D planar waves in an impedance tube can be obtained by the equation below [69]:

$$f_u = 0.58\frac{c}{d} \qquad [18]$$

where c is the speed of sound in air, and d is the diameter of the tube. ISO 10534-2 [69] suggests that the minimum frequency of interest be set by determining the distance between microphones that needs to be at least 5% of the wavelength. Hence, the low frequency bound can be written as:

$$f_l = 0.05\frac{c}{x_1-x_2} \qquad [19]$$

3.11. Two microphone method (TMM)

Impedance measurement techniques have been developed significantly by the enhancement of computation capabilities and the advent of Fast Fourier Transform algorithms. With the mathematical development, several alternatives to the single microphone method (SMM) have been proposed by Singh [70], and TMM has initially been introduced by Seybert and Ross [71], which was further modified by Chung and Blaser [72,73]. The TMM can be implemented by flush-mounting and fixing two microphones onto the wall of an incidence acoustic impedance tube. Hence, with the microphones built on the impedance tube, the ratio of complex acoustic pressures can effectively be measured at each location of the microphones by forming a 1-D acoustic standing wave in a deterministic manner. The complex acoustic pressures are created in the process in which the incident and reflected acoustic waves interfere at the surface of the test sample. A noticeable advantage of the TMM, compared to the SMM, is that no movement of microphones is needed ensuring the faster and more efficient collection of data with no reconfiguration. Furthermore, in addition to utilizing discrete frequency distribution for excitation, the method can be used with random noise excitation like white noise to diminish the data collection data considerably. In order to implement the method properly, precise amplitude and phase calibrations of the two microphones are the paramount factors for enabling the accurate measurements of the transfer function. An innovative microphone switching technique, presented by Chung and Blaser [73], decreases the calibration time for the two microphones. Additionally, the distance between the microphones needs to be considered carefully as it can result in large errors when it is half the wavelength of the frequency.

The schematic diagram of the experimental apparatus, as shown in **Fig. 9**, consists of a hollow cylindrical tube with one end used as a sample holder and the other end for the loudspeaker (sound source) that generates the broadband noise, and the ports of the microphones flush-mounted onto the wall of the tube.

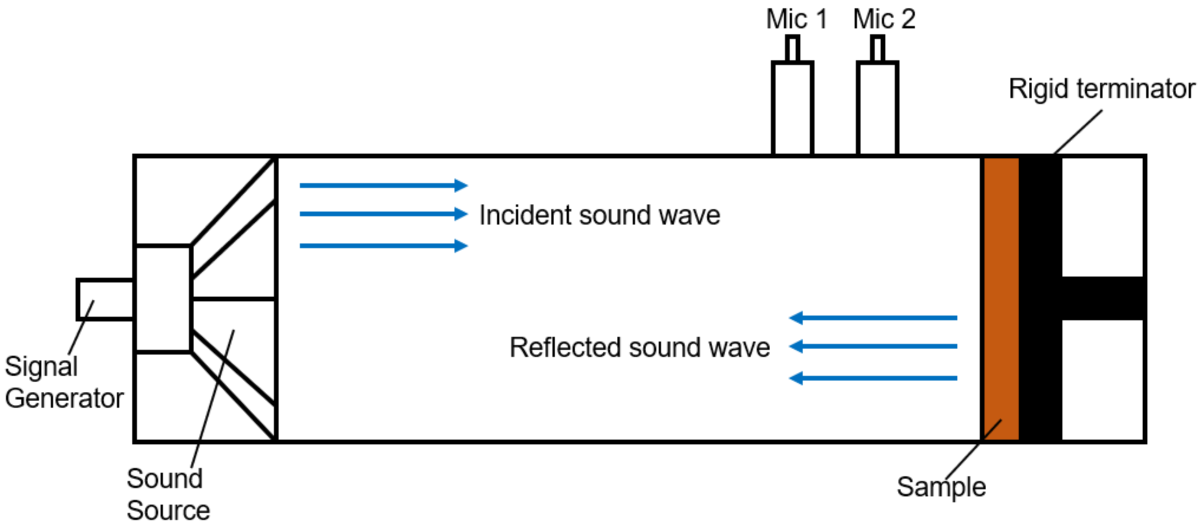


Figure 9. The schematic diagram of the impedance tube apparatus

4. Methodology

In this section, procedures such as sample preparation, data preparation, and coding for the ML algorithms are described in detail.

4.1. Sample Preparation

In order to fabricate samples within a short period of time, SLS (Selective Laser Sintering) 3-D printer was utilized. In SLS printing, the laser solidifies a thin layer of the preheated powder on the printing bed. The preheated temperature achieves the solidification process of the power exquisitely as it is set to be close to the melting point of the material. The desired object is formed layer by layer by the fused particles while the unsolidified power remains to hold the object firmly. The powder is dispersed onto the top of the previous layer after the printing bed lowers by the thickness of the layer. This process is completed when the object is fabricated.

Fig. 10 describes a schematic view and mechanism of SLS 3-D printer.

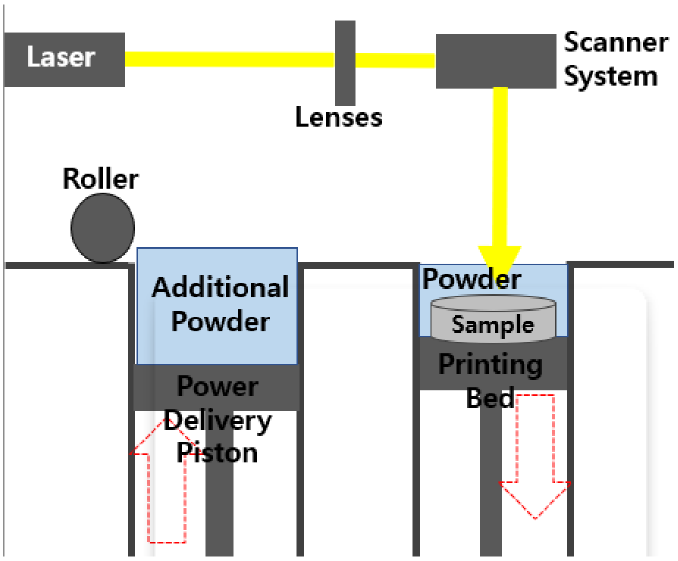


Figure 10. A schematic view and mechanism of SLS 3-D Printer

Initially, 4 disks, 2 rings and 1 bottom cap were produced with a diameter of 99.7mm allowing 0.3 mm gap of the inner diameter of the specimen mount part in the impedance tube. The reason for the tolerance of 0.3mm is to ensure that the samples do not get compressed or shaped convexly when placed inside the specimen mount. Each disk was designed to have a different hole size at the center to investigate the sound absorption coefficient trend according to different stacking locations. The rings were designed to be a hollow medium with two different heights to let the sound travel without restrictions. The bottom cap was designed to prevent the possible leakage of the sound waves. The printed samples are shown in **Fig. 11**.

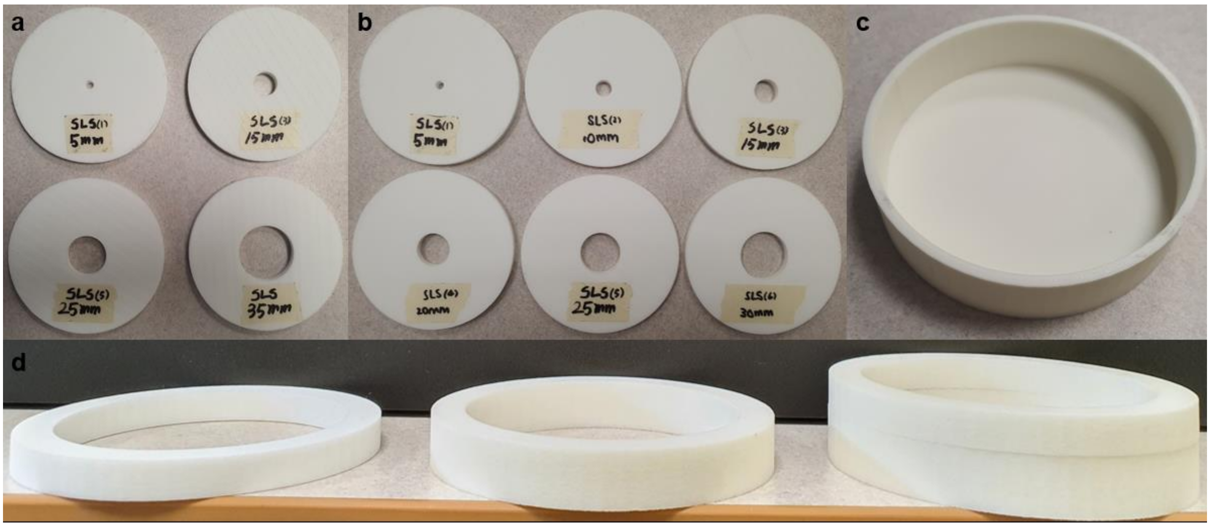


Figure 11. Samples prepared for the sound absorption coefficient (a: 4 disks with a hole

diameter of 5mm, 15mm, 25mm and 35mm, b: three previous disks and three new disks with a hole diameter of 10mm, 20mm and 30m, c: A bottom cap, and d: Rings with a height of 10mm (left), 15mm (middle) and 25mm (right, 10mm and 15mm combined))

After completing the absorption coefficient measurements for the 4-disk approach, three additional disks with a hole diameter of 10mm, 20mm and 30mm were produced to collect more data for better ML performance analysis as shown in **Fig. 11 (b)**. Based on the literature, three main factors (porosity, cavity size, and thickness of the materials) are known to affect the sound absorption coefficient. Hence, in this paper, samples with different hole sizes and thicknesses were produced to further analyze their different behaviors.

4.2. Impedance Tube Measurement Set-up

The setup of the two-microphone impedance measurement tube (*Brüel & Kjær Type 4206*) is illustrated in **Fig. 12**. The inner diameter of the external tube is 100mm, and two microphones were placed at 55 cm. The internal diameter determines the allowable maximum frequency, and the length of the tube and the microphones set the starting point of the frequency. The impedance tube with these specifications measures the absorption coefficient of the materials for the frequency range between 50 Hz and 1600 Hz. The loudspeaker placed at one end of the tube generates a noise inside the tube, and incident sound wave travels towards the reflecting surface of the specimen housed on the opposite side of the tube.

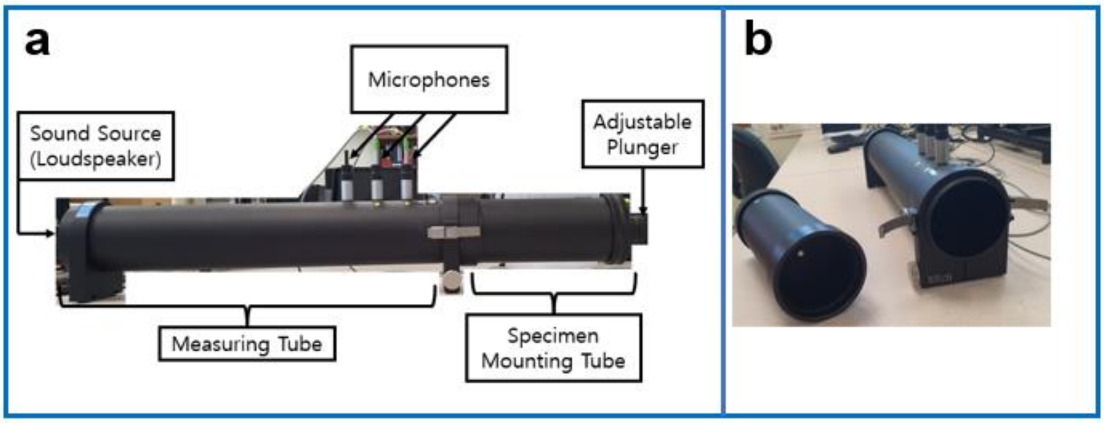


Figure 12. Two-microphone impedance tube parts for the measurement of sound absorption coefficient (a: General components, b: disassembled view)

The measurement process is as follows:

1) A sample is inserted inside the specimen mounting tube.

2) The sample is appropriately located using the adjustable plunger, ensuring that the front side of the sample is aligned with the boundary line.

3) Join the measuring tube and specimen mounting tube together and fix them by clicking hinges.

4) Once everything is set up, run each experiment three times and obtain the average coefficient values.

Before mounting each sample inside the specimen mount, a sample comprised of multiple disks, rings and a bottom cap is thoroughly taped to prevent the leakage of the sound waves completely.

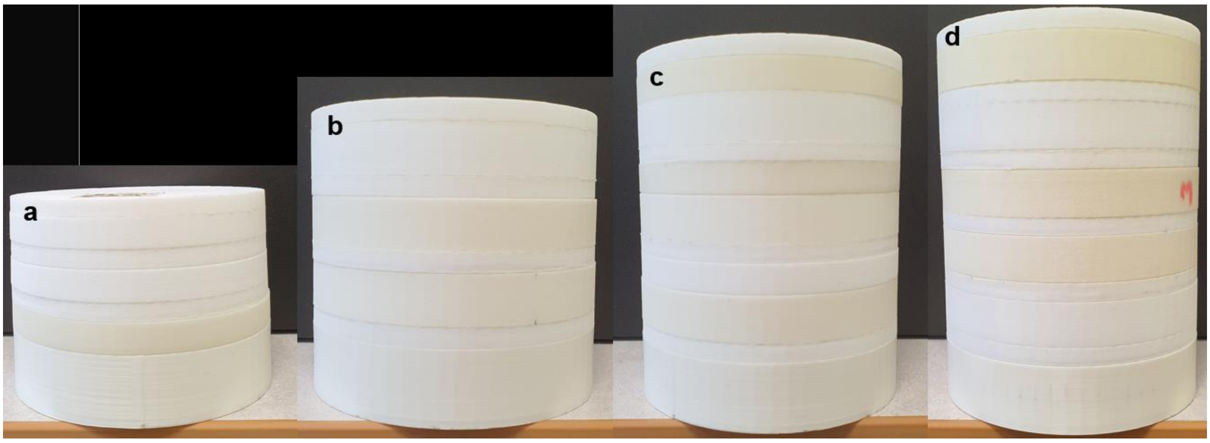


Figure 13. Samples with different ring heights before applying tapes (a: 10mm, b: 15mm, and c: 25mm)

The samples in **Fig. 13(a-c)** contain 4 disks, 3 rings and 1 bottom cap, and the other sample in **Fig. 13(d)** contains 6 disks, 5 rings and 1 bottom cap. For the measurement of the latter samples with 6 disks, although only one fixed ring height (15 mm) was used, 400 samples were obtained utilizing the disks with different hole sizes. Overall, from the variations of the stacking combinations given in **Table 1**, 57 and 400 measurements were completed for each sample type (4 disks and 6 disks, respectively).

Table 1. A brief summarization of samples measured for the absorption coefficient

	Number of Disks	**Disk Hole Diameter (mm)**	**Ring Thicknesses (mm)**	**Total number of measurements**	**Total number of possible measurements**
First Approach	**4**	**5, 15, 25, 35**	**10, 15, 25**	**57**	**72**
Second Approach	**6**	**5, 10, 15, 20, 25, 30**	**15**	**400**	**720**

* Per each measurement for the first approach, the same type of rings was used, meaning that only the location of disks was shuffled while keeping rings of the same thickness unchanged. For example, rings with the thickness of 10, 15, and 25mm cannot be used at the same time.

4.3. Data preparation

After obtaining the absorption coefficients ranging from 50 Hz to 1600 Hz at an interval of 2 Hz (equal to 776 coefficients in total per sample), the collected data need to be tidied up further to be used in the ML models. Hence, it is important to separate features as inputs and target(s) as outputs to create an appropriate data matrix. Normalizing the input values is also a crucial step as it can increase the efficacy of the model algorithms for producing higher predictions. Hence, the input values are all rescaled into the range of [0, 1]

As no error or outlier is found during the experiment for the samples, the process for outlier removal is not needed. The organized data is then split into two subsets: training set and testing set where 70% of the data is used for the former, and 30% for the latter. The ratio division (70% and 30%) is determined for the ML models based on the literature review. During the training phase, it learns the complicated patterns of target values that change with respect to different labeled inputs and computes numerous values of weights and biases of neurons in each layer. Then, during the testing phase, it examines the accuracy of the algorithm by comparing the computed target values with the ones that have not been experimented before. To analyze the accuracy of the algorithm, hence, it is important to use appropriate performance metrics.

4.4. Performance Metrics

In ML applications, it is paramount to define proper performance metrics to monitor and measure the quality of the solutions as the progress or accuracy of the models can be directly seen from them. Hence, regardless of ML types, any model needs metrics to judge performance.

For regression models used in this paper, the following four metrics are used to evaluate the performance of the models: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

4.4.1. Mean Squared Error (MSE)

Mean Squared Error (MSE) is considered as the most widely used metric for regression models. As its equation implies in **Eq. 20**, the average of the squared difference between the true target value and the predicted target value is computed.

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(\overrightarrow{y_j} - \widehat{y_j})^2 \qquad [20]$$

where $\overrightarrow{y_j}$: True target value, $\widehat{y_j}$: Predicted target value, and N : Total number of observations.

Hence, any value obtained from the MSE metric is always an absolute number and the values interpret how much the predicted target values deviate from the true value.

4.4.2. Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is a modified version of MSE that rectifies several disadvantages of the MSE. For the MSE, an overestimation of the badness of the model can be created due to its nature that even small errors are penalized by the squaring factor. Also, the squaring factor makes the metric behave more sensitive to outliers. On the other hand, the RSME offsets the penalization effect and the outlier sensitivity by square-rooting the MSE as represented in **Eq. 21** below:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\overrightarrow{y_j} - \widehat{y_j})^2} \qquad [21]$$

Hence, an easier and smoother interpretation can be achieved using the RMSE.

4.4.3. Mean Absolute Error (MAE)

Unlike the MSE, Mean Absolute Error (MAE) calculates the sum of the absolute difference between true and predicted target values, which is preferred when the dataset does not contain prominent outliers, and it also does not penalize large errors. The formula for the metric can be written as **Eq. 22**:

$$MAE = \frac{1}{N}\sum_{j=1}^{N}\left|\overrightarrow{y_j} - \widehat{y_j}\right| \qquad [22]$$

The disadvantage of the MAE is that the interpretation can become difficult as it does not consider the magnitude of the true target values. However, since the whole target values of the sound absorption coefficient are positive ranging from 0.0 to 1.0, the potential disadvantage can be mitigated, which makes it a good option for the regression models as well as MSE and RMSE.

4.4.4. Mean Absolute Percentage Error (MAPE)

The MAPE is defined as the average of errors of predictions expressed in percentage. Hence, its main advantage is that it requires a relatively easy level of interpretability for analyzing the results as it simply shows the average of the errors as a percentage. The MAPE can be expressed by the following equation [74]:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\widehat{y_i} - y_i}{y_i + \epsilon}\right| \qquad [23]$$

where n is the number of observations, ϵ is a user-defined value used to prevent division by zero, and $\widehat{y_i}$ and y_i are the predicted and true values of the ith observation, respectively.

4.5. Machine Learning Models

4.5.1. Baseline for Artificial Neural Network (ANN)

The baseline for the ANN model used in this paper varies depending on the approach mentioned in **Section 4.2**. For the first approach where four disks and three rings are used, the number of neurons in the first layer is seven neurons; that is, four different hole diameters of the disks and three different thicknesses (10, 15, and 25mm) of the rings. For the second approach where six disks and five rings are used, six neurons are used in the input layer since the hole diameters of the disks are the only factors that change while keeping the ring thickness (15mm) unchanged. Two hidden layers are used with numerous neurons in each layer for the first and second approaches. For the output layer, the sound absorption coefficient is the target output, and the output is measured continuously from 50 Hz to 1600 Hz at an interval of 2 Hz, which amounts to 776 coefficients, leading to 776 neurons eventually. The activation function used for each layer is ReLU, and the process for the gradient descent computation is performed using Adam optimizer with the learning rate of 1×10^{-3}. During the data training process, initially, values for weights, w, are randomized with the biases, b, set to zero. After numerous epochs, the values for weights and bias are updated and optimum values can eventually be obtained, which leads to the precise prediction of the target values. The number of epochs is determined according to the diversity of the dataset to allow the model to compute the fine results sufficiently. The summary of the hyperparameter for the model is explained in **Table 2**.

Table 2. A Summary of the hyperparameters for the ANN model

			First Approach	Second Approach
Hyperparameter/ values	Input Neurons		7	6
	Hidden Neurons	57 Samples	{500, 400}	
		200 Samples	{400, 300}	
		300 Samples	{500, 500}	
		400 Samples	{400, 400}	
	Output Neurons		776	
	Activation		ReLU	
	Gradient Descent		Adam Optimizer	
	Learning Rate		1×10^{-3}	
	Epoch	57 Samples	376	
		200 Samples	129	
		300 Sample	114	
		400 Samples	129	

The baseline described is utilized to analyze the prediction accuracy of the acoustical parameter (776 sound absorption coefficients from 50 Hz to 1600 Hz) based on the designated input variables.

4.5.2. Setup and Hyperparameter Optimization of ANN model

The training process for the ANN model developed in this paper is as follows. Each group of the dataset containing 57, 200, 300 and 400 samples experimented with the impedance tube is divided into two subsets (training set and test set). The test dataset contains 30% of the whole dataset and is used for examining the performance result once the training process is completed. The dataset used for the test process is selected randomly from the whole dataset to ensure the hypothesis in which the model predicts the target values with the dataset that has not been experimented before. For the optimization process, the hyperparameters (the number of hidden layers and neurons, gradient method, and activation function, etc.) are determined using the validation method, known as *Three-fold Cross Validation*, which is generally employed for optimizing ML hyperparameters. Hence, to perform the cross-validation process, the whole dataset is divided into three sub-parts where each part comprises 1/3 of the dataset. With the three sub-parts of equal size, training and validation processes are performed. The two parts are used for training the ANN model and the last one is for the validation process, which evaluates its performance. The prediction accuracy of target values (dependent variables) from labeled inputs (independent variables) is evaluated based on the MSE, RMSE, MAE and MAPE. Various hyperparameter values are tested during the cross-validation process. **Table 3** shows the details of the number of the test, training and sub-part datasets (300, 400 and 500 samples) and the hyperparameter values tested for the process.

Table 3. The details of the setup and hyperparameter tuning for the ANN model

		Test Set	Training Set	3 subparts of the dataset for Three-fold Cross Validation
Number of Samples	57	17	40	{13, 13, 14}
	200	60	140	{70, 70, 70}
	300	90	210	{93, 93, 94}
	400	120	280	{116, 117, 117}
Hyperparameter	Hidden Layer(s) & Neurons	1 Layer		{100}, {200}, {300}, {400}, {500}
		2 Layers		{100, 100}, {200, 100}, {200, 200}, {300, 100}, {300, 200}, {300, 300}, {400, 100}, {400, 200}, {400, 300},

			{400, 400}, {500, 100}, {500, 200}, {500, 300}, {500, 400}, {500, 500}
	Gradient Method	Stochastic Gradient, Adam Optimizer	
	Learning rate	0.1, 0.01, 0.001	
	Iterations	300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500	

Hence, 2340 different hyperparameter sets for each dataset are examined throughout the cross-validation process, and the optimal sets are selected. The summary of the hyperparameters chosen for the ANN model is illustrated in **Table 2** in **Section 4.5.1**.

4.5.3. Baseline and Hyperparameter Optimization for RBFN model

In a similar manner to the ANN hyperparameter optimization, the RBFN structure can also be optimized through the *Three-fold Cross Validation*. However, the RBFN model is built with different hyperparameters. **Table 4** summarizes the details of the optimal hyperparameters obtained during the process.

Table 4. A Summary of the hyperparameters for the RBFN model

			Second Approach
Hyperparameter/ values	Input Neurons		6
	Hidden Neurons	200 Samples	139
		300 Samples	209
		400 Samples	277
	Output Neurons		776
	Activation		*Gaussian Function*
	Width	200 Samples	1.0
		300 Samples	
		400 Samples	

4.5.4. Baseline and Hyperparameter Optimization for kNN model

The same datasets containing the input variables and the outputs are fed into the kNN algorithm for the performance analysis against ANN and RBFN. To predict the data points that are the closest to the true data points, a simple mathematical approach of calculating the distance, known as *Euclidean distance*, is utilized. The distance can be expressed by the following equation:

$$Euclidean(A,B) = \sqrt{\sum_{i=1}^{n}(fa_i - fb_i)^2} \qquad [24]$$

where A and B are the observation points, fa_i and fb_i are the ith feature of point A and point B, and n is the number of dimensions. The lower distance implies that there is a higher similarity between the objects. Hence, it is crucial to determine the value of *k* to discover the nearest neighbors (the true data points) of a point on which a prediction needs to be made.

With the given values from 1 to 50 for *k*, the performance of the algorithm is repeatedly tested by fitting kNN regressors on some of the data, and the value that produces the most reliable estimate of the predictive performance is determined. This process helps to discover the optimal performance metric values. The kNN algorithm, in this project, is used with *k*=4. A weighted average instead of a regular average is used to further improve the performance for predicting the target values more precisely. The summary of the hyperparameter for the model is explained in **Table 5**.

Table 5. A Summary of the hyperparameters for the kNN model

		Second Approach
Hyperparameter/ values	Input Variables	6
	Number of Outputs	776
	Distance	*Euclidean Distance*
	k	4
	Average	Weighted average

5. Results and Discussion

5.1. Analysis of ANN performance for 4-Disk Approach

Initially, the ANN performance for predicting sound absorption coefficients was done based on the dataset of 57 samples comprised of 4 disks and 3 rings. The performance metrics obtained through the computation are shown in **Table 6**.

Table 6. Performance metrics of ANN regression for 4-Disk Approach

Dataset	No. of Samples	MSE	RMSE	MAE	MAPE (%)
Training	40	0.045	0.22	0.16	44.10
Test	17	0.055	0.23	0.17	145.33

In general, the training set outperforms the test set in terms of performance metrics as more data (70% of the whole dataset) are fed into the system to learn patterns of target values with respect to input variables. As can be seen in the table, it is shown that all the metric values of the training set are better than those of the test set. An important finding to note from the table is the MAPE metric. For MAPE, the closer the value is to zero, the algorithm is expected to predict target values with higher accuracy; that is, the accuracy can be computed by subtracting 100 by the MAPE value. The accuracies (56.9% and -45.33%) were obtained for the training and test sets, respectively. In this case, ANN completely fails to predict the target values accurately due to the extremely insufficient amount of data provided. The graphs shown in **Fig. 14** below further demonstrate the erratic prediction performance of the algorithm for the 4-Disk data.

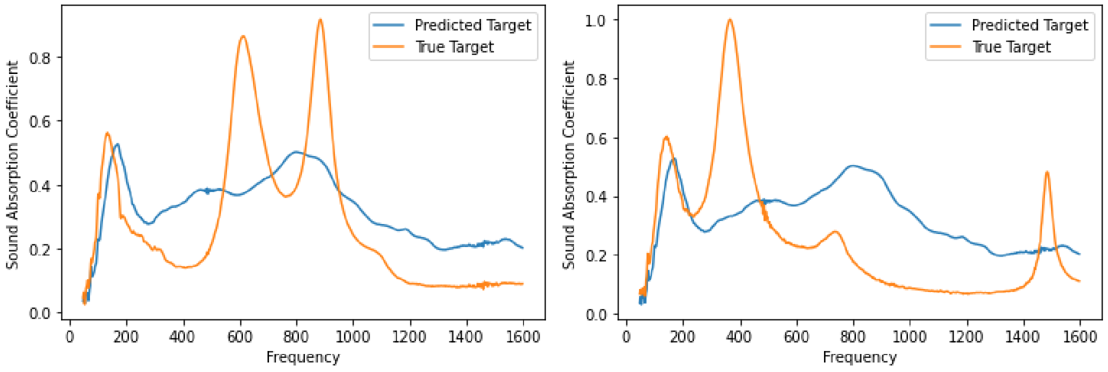


Figure 14. ANN-predicted and True Target graphs for comparison with 57 samples

As the main failure factor was a small amount of data for the algorithm to learn, the 6-Disk approach was conducted to resolve the issue and improve its performance.

5.2. Analysis of ANN performance for 6-Disk Approach

To improve the prediction performance, 400 samples were measured. Three different datasets containing 200, 300 and 400 samples were then created to investigate the prediction improvement along with the increasing data.

5.2.1. 6-Disk ANN Analysis with 200 samples

In **Table 7**, the performance metrics for a dataset of 200 samples are illustrated.

Table 7. Performance metrics of ANN regression for 6-Disk Approach with 200 samples

Dataset	No. of Samples	MSE	RMSE	MAE	MAPE (%)
Training	140	0.0011	0.073	0.025	9.487
Testing	60	0.0048	0.092	0.043	18.122

Even with 200 samples, the prediction performance was improved significantly. This can be explained by comparing the above table with **Table 6**. For the test set, the percentage improvements (98.4%↑, 65.22%↑, 76.47%↑ and 63.452%↑) were shown for MSE, RMSE, MAE and MAPE, respectively. Furthermore, the accuracies (90.51% and 81.88%) for both sets show great potential for prediction improvement. A graph comparison between ANN-predicted and true target values was done to examine the improvement, which can be seen in **Fig. 15**.

The ANN model seems to perform well in tracking the behavior of 776 absorption coefficients ranging from 50 Hz to 1600 Hz. Within the frequency regime (50 Hz to 300 Hz), most of the graphs were predicted accurately overlapping the true target values. However, certain faulty predictions were also made oftentimes in the higher frequency range. For example, in graph (a)**,** from 900 Hz to 1300 Hz, it completely fails the performance by overpredicting or underpredicting the true target values. Similar phenomena can also be observed in the graphs (b, c, e and f). The erratic prediction computation can be explained again by the insufficient amount of data as machine learning methods are designed to work well when given abundant data (typically, millions of data).

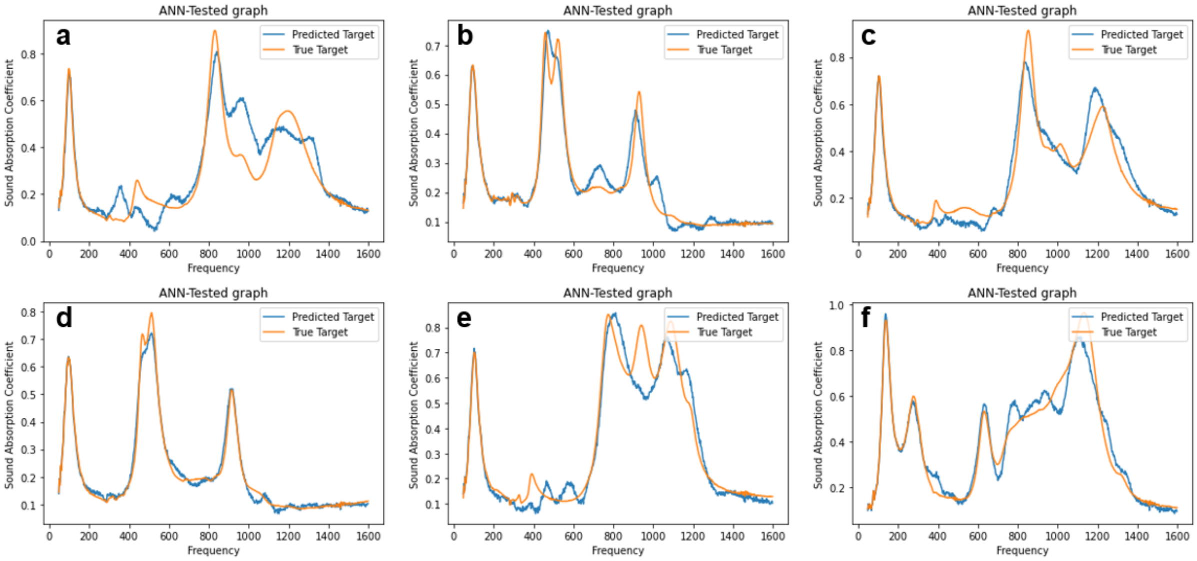


Figure 15. ANN-predicted and True Target graphs for comparison with 200 samples

5.2.2. 6-Disk ANN Analysis with 300 samples

Due to the nature of the ANN model, for the dataset of 300 samples, it computed better values for all the performance metrics as compared to those of 200 samples, which can be seen in **Table 8.** The percentage improvements (6%↑, 8.8%↑, 9.3%↑ and 4.3%↑) were achieved for MSE, RSME, MAE and MAPE, respectively, in the test set. The accuracy percentages for the training and test sets were 91.98% and 86.13%.

Table 8. Performance metrics of ANN regression for 6-Disk Approach with 300 samples

Dataset	No. of Samples	MSE	RMSE	MAE	MAPE (%)
Training	210	0.00046	0.0589	0.018	8.024
Testing	90	0.00448	0.0839	0.039	13.87

With the improved performance metrics, the ANN model could predict the target values more accurately. As shown in the graphs (a and e in **Fig. 16**), the model predicted the samples almost perfectly with the predicted target values overlapping the true values. However, the same issue of the model over-or under-predicting the target values occurred for the dataset of 300 samples as well. Depending on the stacking location of the disks with different hole sizes, it was observed that the behavior of the absorption coefficient changed dramatically. Hence, the prediction improvement could have been made again if the model was learned with more samples.

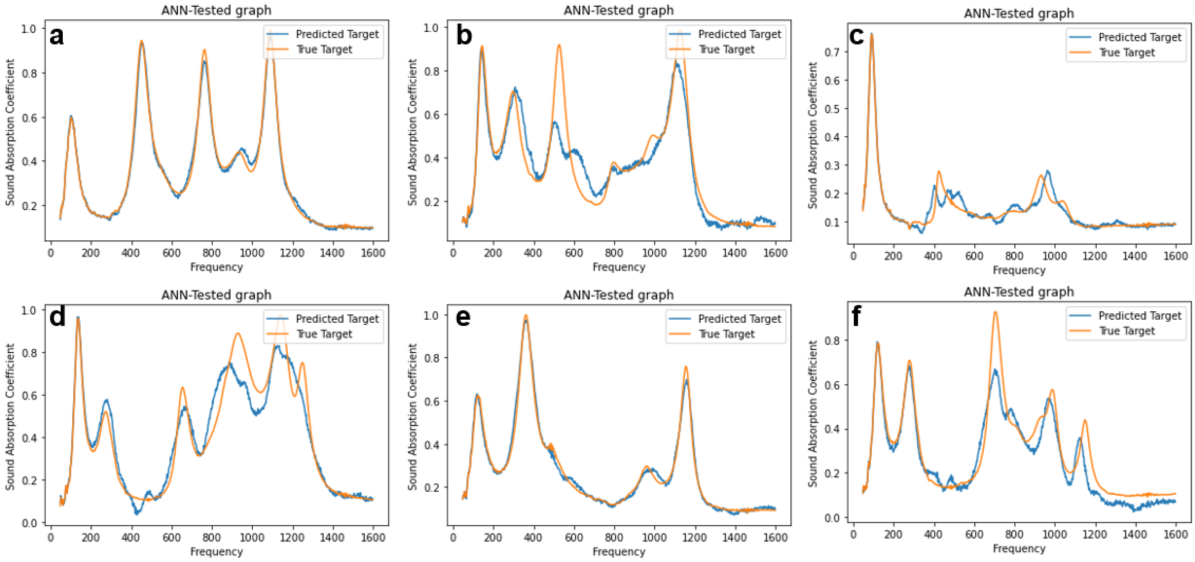


Figure 16. ANN-predicted and True Target graphs for comparison with 300 samples

5.2.2. 6-Disk ANN Analysis with 400 samples

The performance metrics were further improved with 100 additional samples. As compared to the dataset with 300 samples, the metrics (MSE, RMSE, MAE and MAPE) improved by 14.5%, 8.7%, 12.8% and 2.4%, resulting in an accuracy of 88.57% for the test set. The details are described in **Table 9**.

Table 9. Performance metrics of ANN regression for 6-Disk Approach with 400 samples

Dataset	No. of Samples	MSE	RMSE	MAE	MAPE (%)
Training	280	0.00041	0.0573	0.017	7.310
Testing	120	0.00383	0.0766	0.034	11.434

As 400 samples only accounted for 64.5% of the whole possible data ($_{6}C_{5}$=720 samples), the model could not predict target values perfectly for some samples as shown in **Fig. 17**.

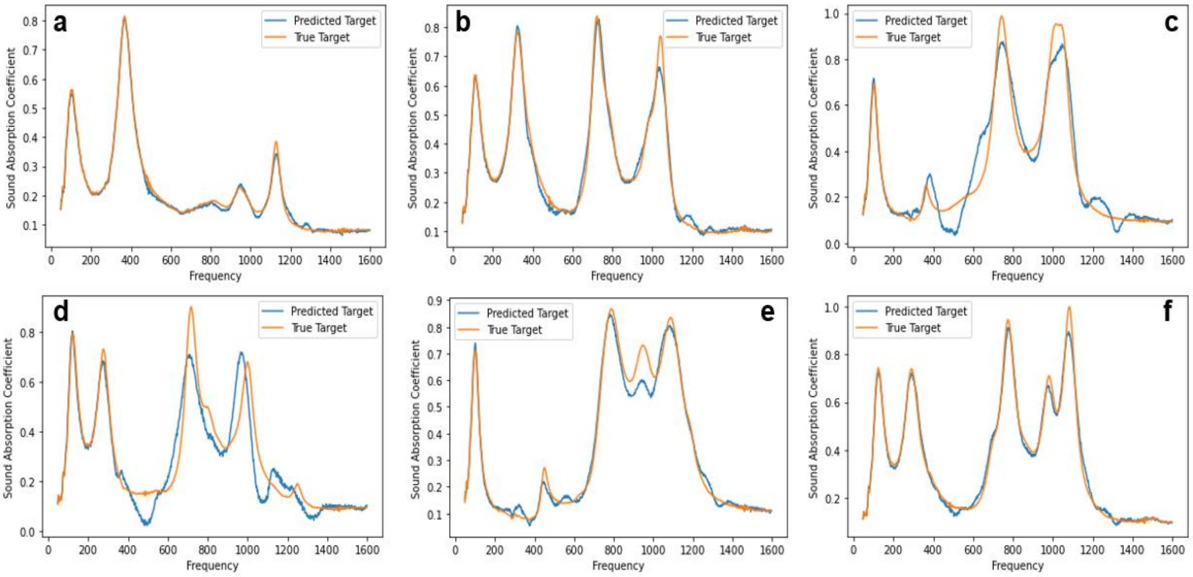


Figure 17. ANN-predicted and True Target graphs for comparison with 400 samples

5.3. Analysis of RBFN performance with 400 samples

Another machine learning model, RBFN, was employed to investigate if it would be able to predict as precisely as the ANN model. As it was demonstrated previously that the insufficient amount of the dataset leads to the poor performance in **Section 5.1**, the 6-disk approach was only considered. As shown in **Table 10**, the performance metrics obtained by the RBFN model were not as precise as those by the ANN model. The main reason could be because, unlike the ANN model, any mathematical equation for minimizing the error function (between the true output and the predicted output) is not involved in the RBFN model. Moreover, even though it was observed that the three-performance metrics (MSE, RMSE and MAE) were improved with the increasing number of samples, a slight increase in the other metric (MAPE) was observed with 300 samples. **Fig. 18** shows how the prediction performance differs depending on the number of samples for the RBFN model. As can be seen in the red boxes in the figure, it created relatively huge fluctuations, failing to predict the true outputs completely, which was hardly observed in the ANN model. Therefore, an assumption could be drawn from the results that the RBFN model is rather inferior in terms of predicting multiple outputs in this study because it frequently failed to take the correlation between the target outputs into account.

Table 10. Performance metrics of RBFN with 200, 300 and 400 samples

No. of Samples	MSE	RMSE	MAE	MAPE (%)
200	0.0150	0.1225	0.0773	30.77
300	0.0163	0.1277	0.0767	31.61
400	0.0088	0.0939	0.0596	24.14

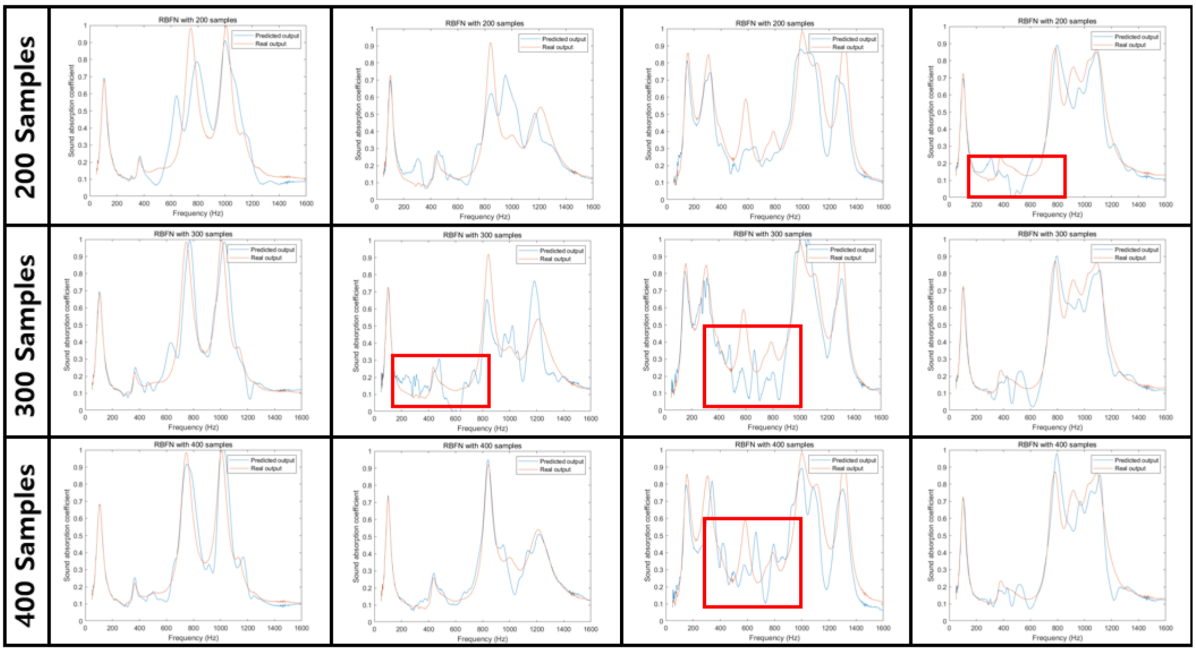


Figure 18. A comparison between RBFN-predicted samples and true target samples

In conclusion, despite the fact that ANN and RBFN originate from the neural network background, as the highest prediction of the ANN model was 88.57%, which was far greater than that of the RBFN model (75.56%), the ANN model was chosen for the optimal configuration process, which will be discussed in **Section 5.6**.

5.4. Analysis of kNN performance with different dataset

The basic machine learning model, kNN, was used to evaluate its functionality of predicting target values. The three different datasets (200, 300 and 400 samples) were again used to examine the improvement of the model along with the increasing dataset.

5.4.1. 6-Disk kNN Analysis with 200, 300 and 400 samples

As the kNN model employs simple mathematical concepts as opposed to the ANN model, the computation time was much shorter. However, in terms of the performance metrics, it could not obtain desirable values. **Table 11** shows the overall metric performance of the kNN model used for the three datasets.

Table 11. Performance metrics of kNN with 200, 300 and 400 samples

No. of Samples	Dataset	MSE	RMSE	MAE	MAPE (%)
200	Training	0.0105	0.1026	0.0665	25.02
	Test	0.0166	0.1288	0.0812	29.58
300	Training	0.0094	0.0973	0.0624	22.48
	Test	0.0134	0.1156	0.0752	28.70
400	Training	0.0058	0.0761	0.0449	16.07
	Test	0.0123	0.1109	0.0702	24.09

As can be seen in **Table 11**, the performance metrics improved with respect to the amount of data, which demonstrates that the prediction accuracy of the kNN model is also significantly affected by the amount of the dataset. However, the performance quality was proven to be not as efficient as the ANN model. The highest accuracy percentage of the training set obtained by the kNN model was 75.91% whereas the ANN yielded 88.57% with the same number of samples, resulting in the prediction difference of 12.66% between the models. Hence, it can be said that much larger number of samples would be required for the kNN to predict the target values as precisely as the ANN.

In **Fig. 19**, the graphs demonstrate that the kNN model better-predicts the behavior of sound absorption coefficients when a larger number of samples is stored in the dataset. For the graphs in the first column, the model nearly matches all the coefficient values from 50 Hz to 1600 Hz. However, for the rest of the graphs in the columns (2,3 and 4), even though the model attempts to better-predict to match the true target values with respect to the number of samples, it still fails due to the multiple peak absorption coefficients in which the complexity of the target behavior is dramatically increased. Hence, it can be assumed, from the graph analysis, that the kNN model exhibits the inability to extrapolate the target curve anymore when the peak sound absorption coefficients occurs several times in the measured sample.

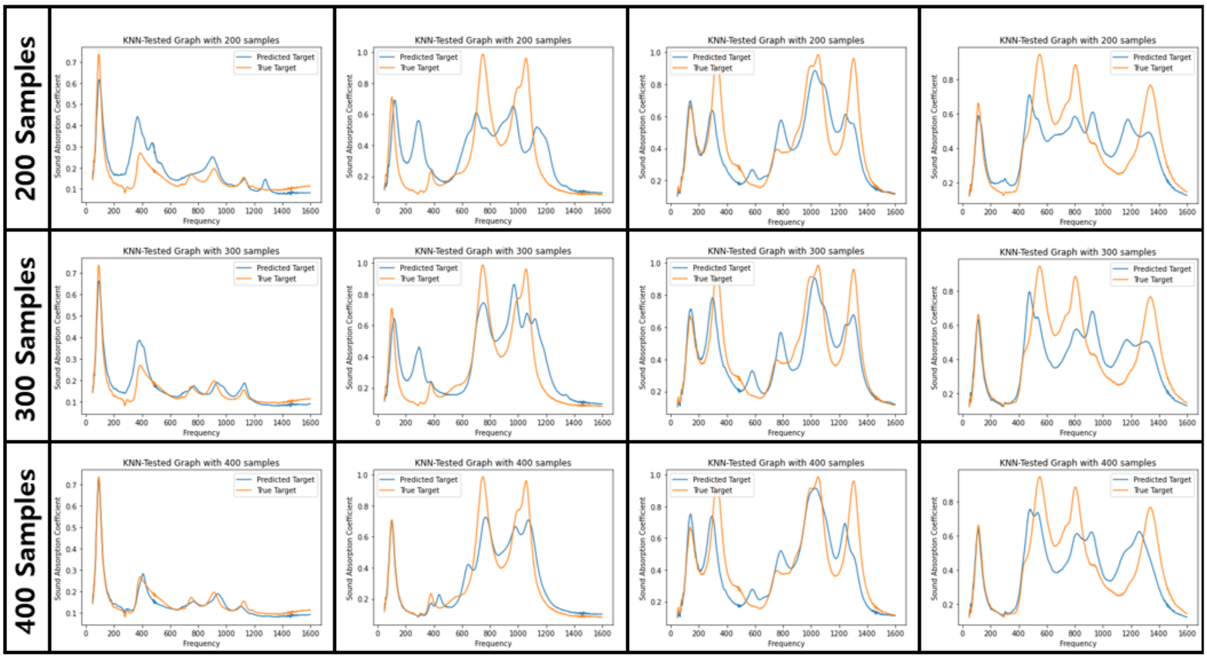


Figure 19. A comparison between kNN-predicted samples and true target samples

5.5. Performance Comparison between ANN and kNN

A further comparison between the ANN and kNN models was done to examine the performance difference in more detail. The dataset of 400 samples was only considered as the most accurate prediction was made with 400 samples for both models. **Fig. 20** shows how the two models extrapolated the target graphs in which the graphs in each column were generated based on the same samples. From the observation, it is obvious that the ANN model has the ability to extrapolate complicated target graphs with multiple peak absorption coefficients effectively to some extent from the trained data, whereas the kNN model fails to predict new samples beyond the trained data.

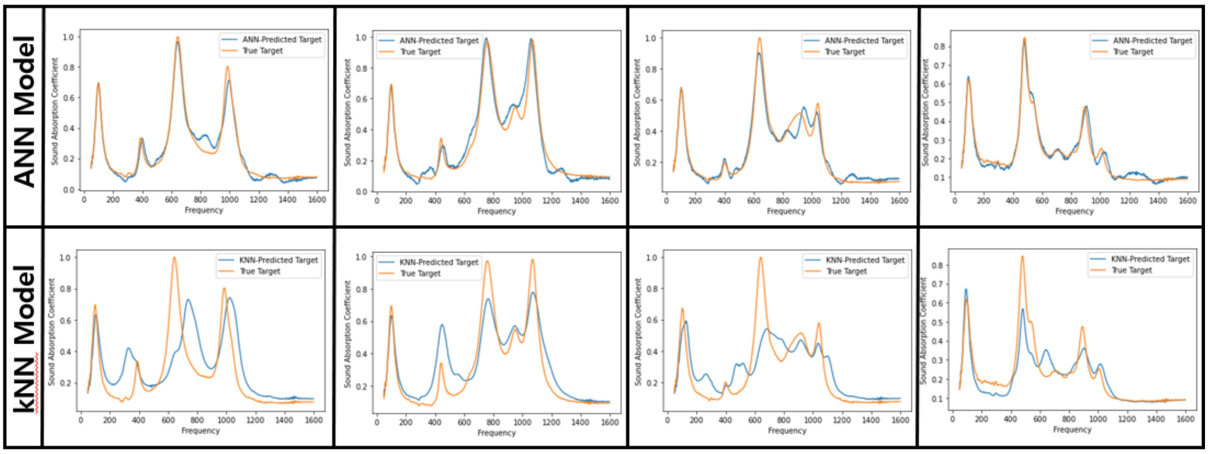


Figure 20. An extrapolation comparison between ANN and kNN models

Table 12 shows that the performance metrics of the ANN model outperform those of the kNN model in all aspects. Hence, for the optimal configuration process, the ANN model was chosen and analyzed further to examine the prediction accuracy of peak sound absorption coefficients in the low-frequency range (<600 Hz), which is discussed in **Section 4.5**.

Table 12. A comparison of the performance metrics between the ANN and kNN models

Dataset	MSE	RMSE	MAE	MAPE (%)	Accuracy (%)
ANN	0.0038	0.0766	0.034	11.434	88.566
kNN	0.0123	0.1109	0.0702	24.09	75.91

5.6. Optimal Configuration for ANN

As it was proven that ANN works better than kNN for all discrete frequency points, the optimal configuration task was done using the ANN. The target value of the coefficient for the optimal configuration was determined to be greater than 0.8; that is, only the samples with the peak coefficients (>0.8) within the frequency range of 600 Hz were investigated. The accuracy of the model 1) to find the target graphs that meet the requirement for the optimal configuration and 2) to predict the target graphs correctly was greater than 80%. **Fig. 21** shows the graphs that meet the requirement for both ANN-predicted and true target graphs. Although the ANN model was able to extrapolate the absorption coefficients fairly accurately except for the graph (e) within the low-frequency range (<600 Hz), huge erratic deviations could be observed within the mid-and high- frequency ranges for some samples (graph b, e, f, h and i in the figure).

Therefore, it can be assumed that, to improve the prediction behavior of the ANN model for the high-frequency range (>600 Hz in this case), it is crucial to collect more samples.

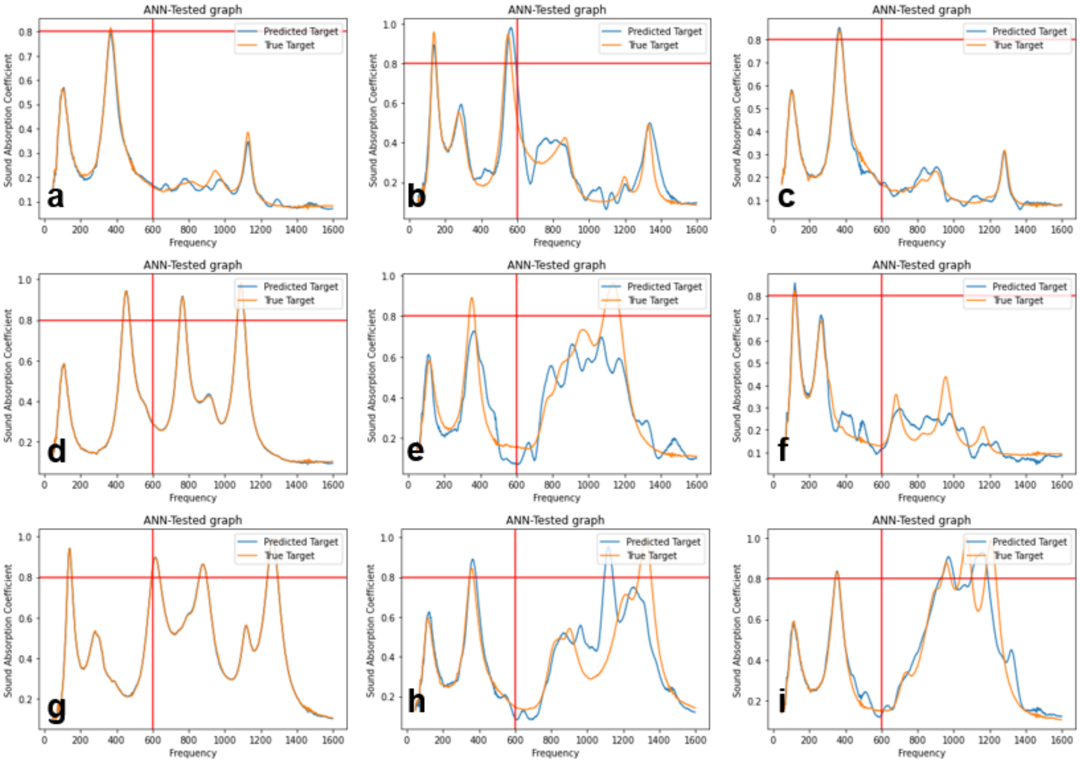


Figure 21. A comparison for the optimal configuration between ANN-predicted samples and true target samples

The accuracy analysis for the low-frequency range was further done using the correlation approach as shown in **Fig. 22**. As can be observed, most of the ANN target points are linearly fit with the true target points, showing a sign of the strong correlation, despite some outliers at 300 Hz, 400 Hz and 500 Hz. The Pearson's correlation coefficient for each frequency is also illustrated in **Table 13**. The degrees of all the coefficients were extremely close to +1, which can be said to be *a perfect positive correlation*. That is to say, the ANN model can effectively be employed to discover new samples that meet the configuration requirement.

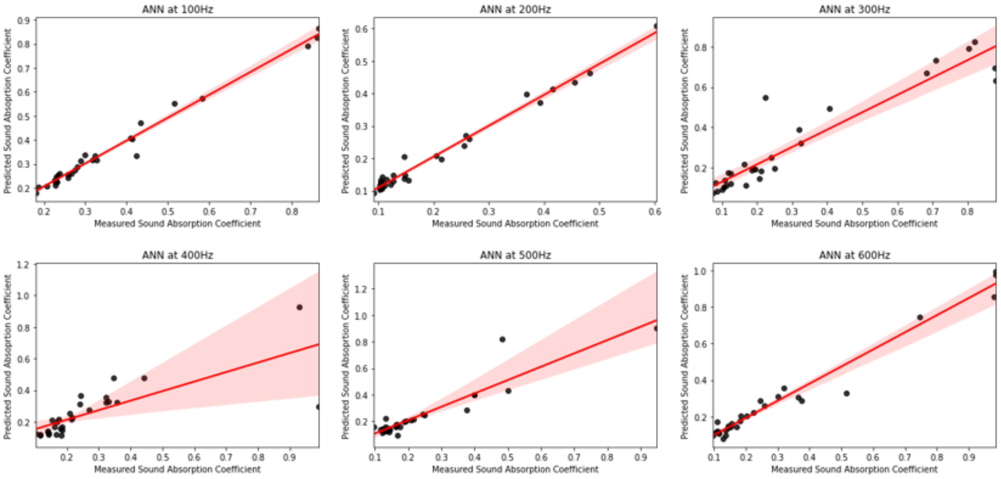


Figure 22. Correlation graphs for ANN vs True target values

Table 13. Pearson's Correlation Coefficients for the low-frequency range by the ANN

Frequency	100Hz	200Hz	300Hz	400Hz	500Hz	600Hz
Pearson’s r	0.992	0.912	0.959	0.928	0.993	0.957

Moreover, the superiority of the ANN model against the KNN model can be observed clearly in **Fig. 23**. While a strong correlation can be observed for the ANN model, the kNN model tends to produce a weaker correlation. The Pearson’s correlation coefficients obtained by the kNN model also demonstrate that it is not able to predict target values as precisely as the ANN model as illustrated in **Table 14.** Although the model computes strong correlation coefficients (>0.9) at 400Hz and 500Hz, all the coefficient values obtained by the ANN are in more favor of the optimal configuration task.

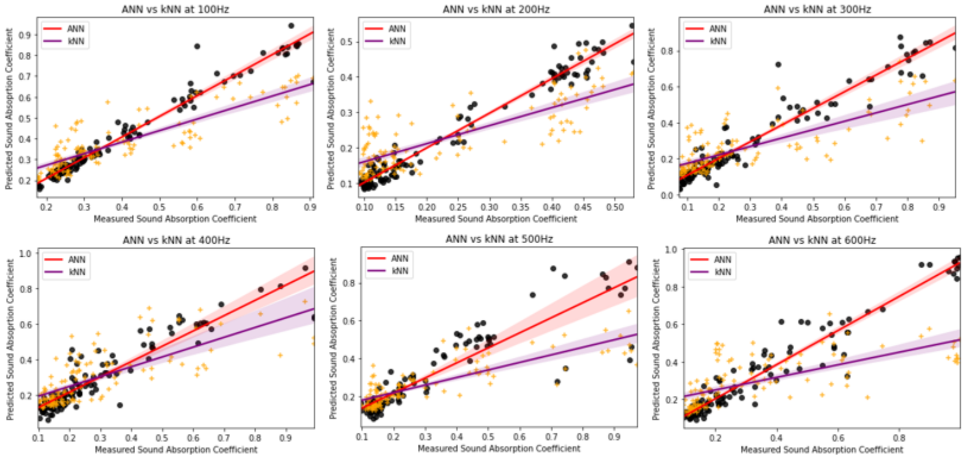


Figure 23. A correlation comparison between ANN and kNN for the low-frequency range

Table 14. Pearson's Correlation Coefficients for the low-frequency range by the kNN

Frequency	100Hz	200Hz	300Hz	400Hz	500Hz	600Hz
Pearson’s r	0.793	0.695	0.752	0.920	0.913	0.857

Fig. 24 shows that the correlation between the actual and predicted coefficients is improved as more samples are used for training the ANN model. As can be observed, for the 400 samples, the Pearson coefficient values are mostly greater than 0.9 and the minimum value observed is 0.89 at 502Hz. As previously discussed, the ANN model occasionally fails to predict the desired outputs with rather big deviations at high frequencies; that is, the results obtained for the 200 samples display such a shortcoming with poor Pearson values from 1400Hz to 1600Hz. The lowest value observed is 0.68 at 1600Hz. Nevertheless, as the primary goal of the optimal configuration task is to ensure that the model predicts sound absorption coefficients accurately from 100Hz to 600Hz, the datasets of 200 and 300 samples can also be used as their minimum Pearson values are close to 0.85 within the frequency range, which is considered a fairly strong linear correlation in general.

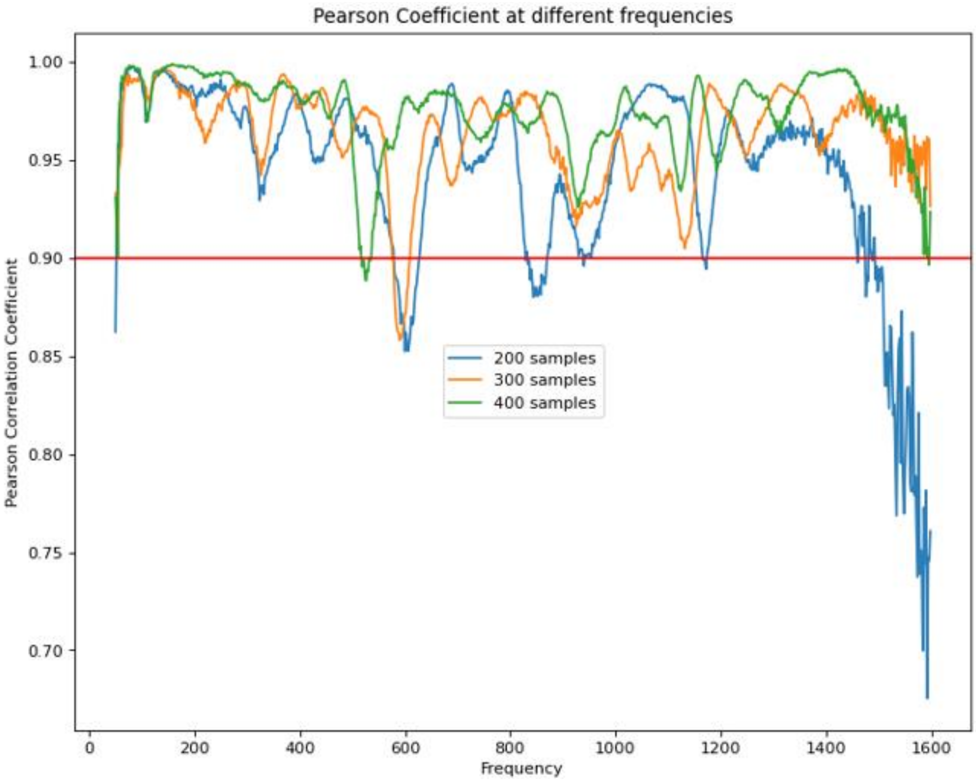


Figure 24. Pearson correlation coefficients from 50Hz to 1600Hz for different amount of samples

6. Conclusion

In this thesis, utilizing machine learning methods to estimate sound absorption coefficients based on physical input parameters is presented. Artificial Neural Network (ANN), Radial Basis Function Neural Network (RBFN) and k-Nearest Neighbor (kNN) methods were employed to predict the target values, and the results of all the models showed the potentiality of improvement with respect to the number of samples. Although the ANN worked better than the other two ML methods for all the analytical approaches, huge deviations could often be observed generally within the high-frequency range (>600Hz). This limitation could be due to the small number of samples used for the algorithm since only 64.5% of all the possible stacking combinations (720 samples) were collected, which implies that a large amount of dataset is the most crucial factor to accurately predict target values. In terms of the absorption coefficient curve from 50 Hz to 1600 Hz in each graph, the peak absorption coefficients were significantly affected by the location of the disks. This demonstrated that the hole size was an important input parameter closely correlated with the absorption coefficient. For the optimal configuration task, the ANN model displayed a strong prediction accuracy, which was validated by plotting the correlation graphs. The Pearson's correlation coefficients of the target values in the range [100Hz, 600Hz] were all greater than +0.9 and close to +1.0, which proved that the ANN model can effectively be used for discovering new samples that meet the optimal configuration requirement in practice.

7. References

1. H. J. Ju., S. Y. Sung, J. K. Yeon, Estimation of sound absorption coefficient of layered fibrous material using artificial neural networks, Applied Acoustics, Volume 169, 2020, 107476

2. I. van Kamp and F. van den Berg, Health effects related to wind turbine sound, including low-frequency sound and infrasound, Acoust. Aust. 46, 31–57, 2018

3. C. Roberts, Low frequency noise from transportation sources, in Proceedings of the 20th International Congress on Acoustics, Sydney, Australia (International Congress on Acoustics, 2010), 23–27

4. H. G. Leventhall, Low frequency noise and annoyance, Noise Health 6, 59–72, 2004, available at https://www.noiseandhealth.org/text.asp?2004/6/23/59/31663

5. K. Waye, Effects of low frequency noise and vibrations: Environmental and occupational perspectives, in Encyclopedia of Environmental Health, edited by J. Nriagu (Elsevier, Burlington, 240–253, 2011

6. B. Berglund, P. Hassmen, and R. S. Job, Sources and effects of low-frequency noise, J. Acoust. Soc. Am. 99, 2985–3002, 1996

7. V. Tyagi, K. Kumar, and V. K. Jain, A study of the spectral characteristics of traffic noise attenuation by vegetation belts in Delhi, Appl. Acoust. 67, 926–935, 2006

8. I. Ekici and H. Bougdah, A review of research on environmental noise barriers, Build. Acoust. 10, 289–323, 2003

9. M. Lin, K. Tsai, B. Su, Estimating the sound absorption coefficients of perforated wooden panels by using artificial neural networks, Applied Acoustics, 70(1), 31, 2009
10. U. Berardi and G. Iannace. Acoustic characterization of natural fibers for sound absorption
applications. Building Environments, 94, 840–852, 2015
11. X. Xinzhao, L. Guoming, L. Dongyan, S. Guoxin, and Y. Rui. Electrically conductive graphene coated polyurethane foam and its epoxy composites. Composite Communications, 7, 1–6, 2018
12. J.-P. Groby, B. Nennig, C. Lagarrigue, B. Brouard, O. Dazel, et al. Using simple shape

three-dimensional inclusions to enhance porous layer absorption. Journal of Acoustical Society of America, 136(3), 1139–1148, 2014

13. Y. Yang, B.R. Mace, and M.J. Kingan. Wave and finite element method for predicting sound transmission through finite multi-layered structures with fluid layers. Computers and Structures, 204, 20-30, 2018

14. Jeon J. H., Yang S. S., Kang Y. J., Estimation of sound absorption coefficient of layered fibrous material using artificial neural networks, Applied Acoustics, 169, 1, 2020

15. ME. Delany, EN. Bazely, Acoustical properties of fibrous absorbent materials. Appl Acoust, 3(2), 105–16, 1970

16. Y. Miki, Acoustical properties of porous materials - modifications of Delany- Bazley models. J Acoust Soc Japan, 11(1), 19–24, 1990

17. T. Komatsu, Improvement of the Delany-Bazley and Miki models for fibrous sound-absorbing materials. Acoust Sci Technol, 29(2), 121–9, 2008

18. DL. Johnson, J. Koplik, R. Dashen, Theory of Dynamic Permeability and Tortuosity in fluid-saturated porous media. J Fluid Mech, 176, 379–402, 1987

19. Y. Champoux, JF. Allard, Dynamic tortuosity and bulk modulus in air-saturated porous media. J Appl Phys, 70(4), 1975–9, 1991

20. MA. Biot, Theory of propagation of elastic waves in a fluid-saturated porous solid. I. Low-frequency range. J Acoust Soc Am, 28, 168–78, 1956

21. MA. Biot, Theory of propagation of elastic waves in a fluid-saturated porous solid. II. Higher frequency range. J Acoust Soc Am, 28, 179–91, 1956

22. J.H. Rindel, The use of computer modeling in room acoustics. J. Vibroengineering, 3, 219–224, 2000

23. L. Savioja, Modeling techniques for virtual acoustics. Simulation, 45, 10, 1999

24. M.D. Egan, Architectural Acoustics; McGraw-Hill: New York, NY, USA, 21, 1988

25. G. Ciaburro, G. Iannace, V. Puyana-Romero, A. Trematerra, A Comparison between Numerical Simulation Models for the Prediction of Acoustic Behavior of Giant Reeds Shredded. *Applied Sciences*. 10(19), 6881, 2020

26. G, Iannace, G. Ciaburro, A. Trematerra, Modelling sound absorption properties of broom

fibers using artificial neural networks. Appl. Acoust. 163, 2020

27. J.F. Hamet, M. Berengier, Acoustical Characteristics of Porous Pavements: A New Phenomenological Model, Internoise, Leuven, Belgium, 93, 641–646, 1993

28. M. I. Jordan and T. M. Mitchell, “Machine Learning: Trends, Perspectives, and Prospects,” Science 349(6245), 255–260, 2015

29. Y. LeCun, Y. Bengio, and G. E. Hinton, “Deep learning,” Nature 521(7553), 436–444, 2015

30. Q. Kong, D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, and P. Gerstoft, “Machine learning in seismology: Turning data into insights,” Seismol. Res. Lett. 90(1), 3–14, 2018

31. K. J. Bergen, P. A. Johnson, M. V. de Hoop, and G. C. Beroza, “Machine learning for data-driven discovery in solid earth geoscience,” Science 363, eaau0323, 2019

32. Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Trans. Pattern Anal. Mach. Intell. 35(8), 1798–1828, 2013

33. G. Ciaburro, G. Iannace, J. Passaro, A. Bifulco, D. Marano, M. Guida, F. Marulo, F. Branda, Artificial neural network-based models for predicting the sound absorption coefficient of electrospunpoly (vinyl pyrrolidone)/silica composite. Appl. Acoust., 169, 2020

34. G. Ciaburro, G. Iannace, M. Ali, A. Alabdulkarem, A. Nuhait, An Artificial neural network approach to modelling absorbent asphalts acoustic properties. J. King Saud Univ. Eng. Sci. 2020

35. G. Iannace, G. Ciaburro, Modelling sound absorption properties for recycled polyethylene terephthalatebased material using Gaussian regression. Build. Acoust. 2020

36. G, Iannace, G. Ciaburro, A. Trematerra, Modelling sound absorption properties of broom fibers using artificial neural networks. Appl. Acoust. 163, 2020

37. Alpaydin, E., 2020. Introduction to machine learning. MIT press.

38. M. Mohri, A. Rostamizadeh, A. Talwalkar, 2018. Foundations of machine learning. MIT press.

39. G. Ciaburro, G. Iannace, M. Ali, A. Alabdulkarem, A. Nuhait, An artificial neural networks approach to modelling absorbent asphalts acoustic properties, Journal of King Saud University – Engineering Sciences, 33(4), 215, 2021

40. JC. Spall, JA. Cristion, A neural network controller for treatment. IEEE Trans Syst, Man Cybernet Part B: Cybernet, 27(3), 369–75, 1997

41. DS. Lee, JM. Park, Neural network modeling foe on-line estimation of nutrient dynamics in a sequentially operated batch reactor. J Biotechnol, 75, 229–39, 1999

42. Fu C, Poch M. System identification and real-time pattern recognition by neural networks for an activated sludge process. Environ Int, 21(1), 57–69, 1995

43. Z. Wang, CD. Massimi, MT. Tham, AJ. Morris, A procedure for determining the topology of multilayer feed-forward neural networks. Neural Networks, 7, 291–300, 1994

44. MA. Biot, Theory of propagation of elastic waves in a fluid-saturated porous solid. I. Low-frequency range. J Acoust Soc Am, 28, 168–78, 1956

45. MA. Biot, Theory of propagation of elastic waves in a fluid-saturated porous solid. II. Higher frequency range. J Acoust Soc Am, 28, 179–91, 1956

46. BD. Ripley, Pattern Recognition and Neural Networks. Cambridge 1996.

47. M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Proceedings of the IEEE international conference on neural networks, San Francisco, CA, USA, 28 March–1 April 1993; Volume 1993, pp. 586

48. MF. Møller, A scaled conjugate gradient algorithm for fast supervised learning. Neural networks, 6(4), 525–33, 1993

49. WN. Venables, BD. Ripley, Modern Applied Statistics with S. Fourth edition. Springer; 2002.

50. D. E. RUMELHART, G. E. HINTON, R. J. WILLIAMS, Learning internal representations by error propagation. In Parallel distributed processing: Explorations in the microstructure of cognition, 1, 318-362, 1986

51. J. E. DENNIS, R. B. SCHNABEL, Numerical Methods For Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, 1983

52. E. Bisong, Optimization for Machine Learning: Gradient Descent. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, CA, 2019

53. L. Bottou, Large-Scale Machine Learning with Stochastic Gradient Descent. In: Lechevallier Y., Saporta G. (eds) Proceedings of COMPSTAT'2010. Physica-Verlag HD, 2010

54. N. MURATA, A Statistical Study of On-line Learning. In Online Learning and Neural Networks, Cambridge University Press, 1998

55. N. Buduma, Fundamentals of deep learning. O’Reilly Media; 2017

56. DP. Kingma, JL. Ba, Adam: a method for stochastic optimization, ArXiv e-print; 2014

57. D. Rumelhart, G. Hinton, R. Williams, Learning representations by back-propagating errors. Nature, 323, 533–536, 1986

58. J. Ian, Bengio, Yoshua; Courville, Aaron (2016), Deep Learning, MIT Press, 200–220. ISBN 9780262035613.

59. Y. Wang, Y. Li, Y. Song, Rong, X. The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition. *Appl. Sci. 10,* 1897, 2020

60. G. Maguolo, L. Nanni, S. Ghidoni, Ensemble of Convolutional Neural Networks Trained with Different Activation Functions, 2019

61. A.K. Dubey, V. Jain, Comparative Study of Convolution Neural Network's ReLu and Leaky-ReLu Activation Functions. In Applications of Computing

62. C. Ming, MATLAB Neural Network principle and case refinement. Beijing China: Tsinghua University Press; 2013.

63. L. Liang, H. Mi, W. Guo, Y. Zhang, H. Ma, Z. Zhang, L. Li, Estimation of sound absorption coefficient of composite structured aluminum foam by radial basis function neural network, Applied Acoustics, 85, 1-14, 2022

64. L. Liang, W. Guo, Y. Zhang, W. Zhang, L. Li, X. Xing, Radial Basis Function Neural Network for prediction of medium-frequency sound absorption coefficient of composite structure open-cell aluminum foam. Appl. Acoust. 170, 2020

65. S. Kuschmitz, T. Ring, H. Watschke, S. Langer, T. Vietor, Design and Additive Manufacturing of Porous Sound Absorbers-A Machine-Learning Approach. Materials (Basel), 14(7), 1747, 2021

66. A. Claudia, Regression Analysis, Encyclopedia of Bioinformatics and Computational Biology, Academic Press, 722-730, 2019

67. M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, C. A. Deledalle, Machine learning in acoustics: Theory and applications, The journal of the Acoustical Society of America, 146(5), 3590-3628, 2019

68. T. Cox, P. D'Antonio, Acoustic Absorbers and Diffusers: Theory, Design and Application, 2009

69. S. Merten, A. Christian, W. Mathies, G. Antje, H. Nobert, Explainable machine learning determines effects on the sound absorption coefficient measured in the impedance tube, The Journal of the Acoustical Society of America 149(3), 1932-1945, 2021

70. ISO 10534-2 (1998). “Determination of sound absorption coefficient and impedance in impedance tubes—part 2: Transfer-function method,” International Organization for Standardization, Geneva, Switzerland.

71. R. Singh, Acoustic impedance measurement methods. Shock Vib. Inf. Center Shock Vib. Digest, 14, 3–10, 1982

72. A.F. Seybert, D.F. Ross, Experimental determination of acoustic properties using a two-microphone random-excitation technique. J. Acoust. Soc. Am., 61, 1362–1370, 1977

73. J. Chung, D. Blaser, Transfer function method of measuring in-duct acoustic properties. I. Theory. J. Acoust. Soc. Am., 68, 907–913, 1980

74. R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *International Journal of Forecasting*, 22(4), 679 – 688, 2006.