# Artificial Neural Network-based Model for Classification Multiple Bacteria Types

**Ahmed Elhetsh , Nesrin Algaed , Faraj .S.f , Adnan Homrbtan**

**The Higher Institute of Engineering Technology-Tripoli**

Elhetsh@gmail.com

**Abstract:**

Data classification plays a crucial role in machine learning, as it entails assigning predefined labels or categories to a dataset. One highly effective approach for data classification is the utilization of artificial neural networks (ANNs), which mimic the structure and functionality of the human brain. ANNs possess the ability to learn from data, recognize patterns, and accurately classify new datasets through training. ANNs can be optimized for successful data classification processes. Moreover, ANNs exhibit remarkable adaptability and information processing capabilities, similar to the biological nervous system of the human brain. This makes them well-suited for handling complex data and adjusting to new tasks in order to achieve desired results. Particularly in the field of biosciences, where data complexity and sensitivity are prevalent, ANNs are advantageous for classifying diverse data such as different types of bacteria and human-body diseases. This paper presents a proposed Artificial Neural Network (ANN) model for the classification of five different types of bacteria using real collected data. The performance of the proposed ANN model is compared against various Machine Learning approaches, including Support Vector Machine (SVM) and Random Forest (RF). The results obtained at the conclusion of this research exhibit great promise, surpassing the current state-of-the-art methods.

**Key words:** ANNs Artificial Neural Networks, Data classification, MLP Multi Layered Perceptron.

**الملخص:**

يلعب تصنيف البيانات دورًا مهمًا في التعلم الآلي ، حيث يتوجب تعيين تسميات أو فئات محددة مسبقًا لمجموعة بيانات. أحد الأساليب الفعالة للغاية لتصنيف البيانات هو استخدام الشبكات العصبية الاصطناعية (ANNs)، التي تحاكي بنية ووظائف الدماغ البشري. تمتلك الشبكات العصبية الاصطناعية القدرة على التعلم من البيانات والتعرف على الأنماط وتصنيف مجموعات البيانات الجديدة بدقة من خلال التدريب. يمكن تحسين شبكات ANN لعمليات تصنيف البيانات الناجحة. علاوة على ذلك ، تُظهر الشبكات العصبية الاصطناعية قدرة ملحوظة على التكيف وقدرات معالجة المعلومات ، على غرار الجهاز العصبي البيولوجي للدماغ البشري. وهذا يجعلها مناسبة تمامًا للتعامل مع البيانات المعقدة والتكيف مع المهام الجديدة من أجل تحقيق النتائج المرجوة. على وجه الخصوص في مجال العلوم الحيوية ، حيث ينتشر تعقيد البيانات وحساسيتها، تعد الشبكات العصبية الاصطناعية مفيدة لتصنيف البيانات المتنوعة مثل الأنواع المختلفة من البكتيريا وأمراض جسم الإنسان. يقدم هذا البحث نموذجًا مقترحًا للشبكة العصبية الاصطناعية (ANN) لتصنيف خمسة أنواع مختلفة من البكتيريا باستخدام بيانات حقيقية تم جمعها. تتم مقارنة أداء نموذج ANN المقترح بمختلف مناهج التعلم الآلي ، بما في ذلك آلة المتجهات الداعمة (SVM) والغابة العشوائية (RF). النتائج التي تم الحصول عليها في ختام هذا البحث واعدة للغاية ، متجاوزة أحدث الأساليب الحالية.

## 1- Introduction:

In recent years, the use of Artificial Neural Networks (ANNs) has gained significant attention in the field of data classification. ANNs, inspired by the structure and functionality of the human brain, have proven to be powerful tools for handling complex datasets and achieving accurate classification results. One area where ANNs have demonstrated their potential is in classifying different types of bacteria. Figure 1, illustrates an artificial neural network modelled after a human neuron.

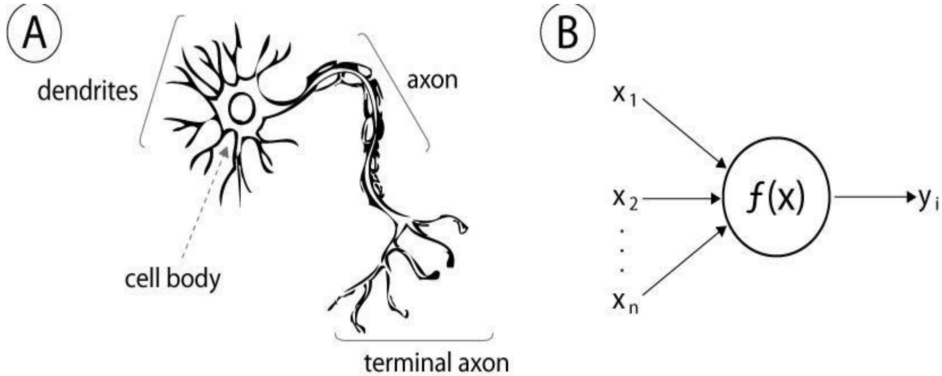



**Figure 1: The sells structure for A) human brain cell, B) an Artificial Neural cell**

The classification of bacteria is of utmost importance in various domains, including healthcare, environmental sciences, and biotechnology. Distinguishing between different types of bacteria can aid in identifying pathogens, understanding microbial diversity, and developing targeted treatments for bacterial infections. However, the classification of bacteria poses unique challenges due to their diverse characteristics, variations in data representation, and the presence of complex relationships between different bacterial species.

Traditional approaches for bacterial classification often relied on manual identification methods, which can be time-consuming, subjective, and error-prone. In recent years, the application of ANNs for bacterial classification has shown promising results. ANNs can learn from large datasets, recognize intricate patterns, and generalize their knowledge to accurately classify unseen bacterial samples. The adaptability of ANNs allows them to handle the complexity of bacterial data, which often includes multiple features such as genetic information, biochemical profiles, and phenotypic characteristics. By leveraging the interconnectedness of neural elements, ANNs can capture the intricate relationships between these features, leading to improved classification performance [1]. This paper aims to explore the application of ANNs for the classification of different types of bacteria. Real-world datasets containing diverse bacterial samples will be utilized to train, validate, and evaluate the performance of the proposed ANN model. The effectiveness of the ANN approach will be compared against other machine learning techniques to highlight its advantages in terms of accuracy, efficiency, and adaptability.

The findings of this research have the potential to significantly contribute to the field of bacterial classification, enabling more efficient and reliable methods for identifying and categorizing bacterial species. Furthermore, the insights gained from this study may have broader implications for other domains that deal with complex classification tasks. The aim of this paper is to create, implement, and assess a Neural Network for data classification. The paper defines an Artificial Neural Network (ANN) as an information processing framework inspired by the information processing methods of biological nervous systems, such as the human brain [2]. It highlights the distinctive feature of this paradigm, which is the interconnectedness of numerous processing elements (neurons) that collaborate to solve specific problems. Similar to humans, ANNs learn through examples. The aim of this paper is to create, implement, and assess a Neural Network for data classification.

This paper proposes an Artificial Neural Network-based model for the classification of multiple bacteria types. Real-world datasets encompassing a wide range of bacterial species will be utilized to train, validate, and evaluate the performance of the proposed model. The architecture and parameters of the ANN will be optimized to achieve the best classification accuracy. Additionally, comparative evaluations will be conducted to benchmark the performance of the ANN-based model against other popular machine learning algorithms, such as Support Vector Machines (SVM) and Random Forests (RF) [3]. The results obtained will provide insights into the superiority of ANNs in tackling the complexities of bacterial classification and highlight their potential as a state-of-the-art approach. The outcomes of this research are expected to contribute significantly to the field of bacterial classification. The development of an efficient and accurate ANN-based model for classifying multiple bacteria types can facilitate rapid and reliable identification of pathogens, enhance our understanding of microbial diversity, and support advancements in biotechnological and medical applications.

In conclusion, this study endeavours to leverage the capabilities of ANNs to propel the field of bacterial classification forward and foster the development of novel methodologies for identifying and categorizing multiple bacteria types. The subsequent sections of this paper are structured as follows: Section 2 presents a comprehensive review of related literature and previous work conducted in this area. Section 3 outlines the methodology employed in this study, encompassing aspects such as data collection, data pre-processing, feature extraction, and the proposed model's training and testing procedures. Section 4 shows the implantation and the training process of the proposed ANN-based model. The obtained results are presented in Section 5. Finally, Section 6 provides the conclusion drawn from the findings and discusses the potential implications and future directions for this research.

## 2- Related Works:

In recent years, the application of Artificial Neural Networks (ANNs) for bacterial classification has gained considerable attention. Numerous studies have explored the use of ANNs as a powerful tool for accurately and efficiently classifying different types of bacteria. This section provides a comprehensive review of the related literature and previous work conducted in this area, highlighting the advancements, methodologies, and key findings of these studies.

The research proposed in section 1, described an ANN-based model for bacterial classification using genomic data. They collected DNA sequences of various bacterial species and trained the ANN model to recognize patterns in these sequences. The results showed that the ANN achieved high accuracy in differentiating between different bacterial strains based on their genomic profiles. The study emphasized the importance of feature selection and data preprocessing techniques in improving classification performance. In a study represented in section 2 and 4, a Bayesian Neural Network model was developed for the purpose of multi-data classification using phenotypic and metabolic data. The researchers employed various machine learning algorithms, including ANNs, SVM, and RF, to compare their performance in classifying bacterial samples. The results demonstrated that the ANN model outperformed other approaches, showcasing its effectiveness in capturing complex relationships between phenotypic and metabolic features for accurate classification.

Another noteworthy contribution by Scientist focused on using different structure of ANNs for bacterial classification based on microscopic images. The researchers collected a large dataset of bacterial images and trained an ANN model to recognize distinct morphological features. The proposed model achieved remarkable accuracy in identifying different bacterial species solely based on their visual characteristics, highlighting the potential of ANNs in image-based classification tasks. In a comprehensive review by Expert various studies utilizing ANNs for bacterial classification were analysed and synthesized [4]. The review highlighted the importance of network architecture design, feature selection, and training methodologies in achieving optimal classification performance [5][6]. It also discussed the challenges faced in utilizing ANNs for bacterial classification, such as dataset size, feature representation, and interpretability of network decisions. Also, different Antifinance Intelligence (AI) approaches including Fuzzy Inference Systems (FIS), Deep Learning (DL), Machine Learning (ML) and Adaptive Fuzzy Inference System (ANFIS) have been employed for classification problems involving bacteria recognitions issues. Overall, the literature and previous work on using ANNs for bacterial classification showcase the effectiveness and potential of this approach [7]. The studies highlight the importance of selecting appropriate features, employing robust training methodologies, and optimizing network architecture to achieve accurate classification results. ANNs have demonstrated their capabilities in handling diverse types of bacterial data, including genomic sequences, phenotypic profiles, metabolic data, and microscopic images.

## 3- Methodologies:

Designing an Artificial Neural Network (ANN) for a classification problem using a Multi-Layered Perceptron (MLP) involves several steps. An overview of the proposed model for classification 5 different types of bacteria is illustrating in Figure 2. The first step is to understand the problem to choose the best approach to resolve it. Also, determine the number of classes or categories you want to classify the data into, in this case is 5 different bacteria. Once the model is known, and the dataset collected, the pre-process step is started. In this step, the input features is elected to represent the collected data, in addition, the corresponding labels or classes is labelled. This step includes handling missing values/data, normalising or standardising features, and splitting the dataset into training and testing sets for 70% and 30%, respectively. In this step the structure and architecture of the MLP is decided. This includes determining the number of layers, the number of neurons in each layer, and the activation functions to be used. In this research, different number of nodes and hidden layers have been tested and evaluated to reach to the best results and avoid any model over-fit.

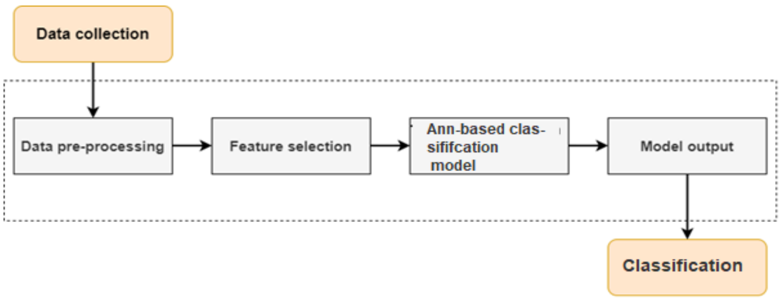



**Figure 2: Overview of the proposed ANN model for classification 5 different bacteria types.**

### 3.1- Data Collection and Pre-processing:

Each data class refer to different bacteria type and stored in separate file each file contains experimental data organized into a matrix (37 rows) by a variable number of columns. The column represents one full experimental data for that bacteria type which means that each bacteria data is made up of 37 counts, or 37 measurements. All data in a file belong to the same class and there are a total of 5 files or 5 classes which maps to five bacteria types. The data are provided in files for (Escherichia coli or e-coli), (Staphylococcus aureus), (Staph aureus) and (Serratia marcescens). As the artificial neural network design produced, these data files must be loaded straight into MATLAB and divided into the following data groups:

Training data: this data will be used to train the artificial neural network and adjust weights of neural network until it has been converged.

Testing data: this data will be used to verify the classification ability and measure the accuracy of data classification of the neural network after it has been trained and converged.

### 3.2- Feature Selection:

Data used for training network is the most important aspect of having a successful neural network. Data must have variety and express different features of the data, if the data collected for only specific state or one class (in case of classification neural network) then the final neural network will not be able to sort out all possible solutions. As mentioned before the data must be divided into two parts (training and testing data) mostly the data used for training will be of 75% of the total data, while data used for testing will be 25%. Important thing to point out is that the training and testing data must express the same variety in the original data. With that been said it is not reasonable to pick the 25% testing data as the first or last 25% part of the original data file, the data must be chosen at specific intervals. Moreover the data must be first plotted and any abnormal data must be identified and manipulated if possible (pre-processing data) as can be seen in the implementation part of this report. One type of data abnormality is an outlier which to outlier a value that "lies outside" (is much smaller or larger than) most of the other values in a set of data [8].

Also as the output data from the neural network will be around the range of 0 and 1 but not guaranteed that the output value will have exact value of either 0 or 1 so the output data must be manipulated (post-processing data). In order to achieve that, a threshold value must be set and used in order to map output values into either 0 or 1. For this report a threshold value of 0.05 will be used so any value within the range 0.95-1.05 will be mapped to 1 and any value within the range 0.05 to -0.05 will be mapped to 0. While any other values fall outside these ranges will be mapped to -1 which means undecided value.

### 3.3- Ann-based classification model:

Multi Layered Perceptron (MLP) trained with (error) Back Propagation (BP) method in particular has found wide spread acceptance in several classification, recognition and estimation applications. The reason is that the MLPs implement linear discriminants but in a space where inputs are mapped nonlinearly. As the data is not linearly separable then the network must have at least one hidden layer, adding more than one hidden layer is possible although this will add more complexity as coming next, while adding more than two hidden layers usually resulted in complexity without any observable enhancements in classifying data. For each layer of the neural network there are different number of nodes depending on the type and nature of input/output data as following:

- Input layer: Every neural network has exactly one input layer; number of nodes per this layer is dependent on the type and nature of input data. For this neural network input layer will contains 37 nodes as each column of data (experiment) contains 37 different reading (feature).
- Output layer: Like the Input layer, every NN has exactly one output layer. Determining its size (number of neurons) is simple; it is completely determined by the chosen model configuration. As this neural network will be used for data classification, so the output layer will consists of five nodes as the output should be one of five different possible classes (bacteria types). In this case each node will correspond to a different class of bacteria; also it is possible to use only three nodes at output layer to classify five different classes as will come next.
- Hidden layer: Although there is no rule of thumb to select the number of neurons at hidden layer or the number of hidden layers used, still there are some good practices that should be followed. Usually adding more than two hidden layers will add more computation complexity as mentioned before. It is more reasonable to add more neurons at hidden layers if needed than adding another third hidden layer. Also choosing too few neurons at hidden layer can lead to under-fitting problem while choosing too many neurons can lead to over-fitting problem. According to (The Clever Machine

2001) Under-fitting occurs when an estimator is not flexible enough to capture the underlying trends in the observed data. (The Clever Machine 2001) Also stated that over-fitting occurs when an estimator is too flexible, allowing it to capture illusory trends in the data, these illusory trends are often the result of the noise in the observations [9]. Figure 3 shows function f(x) black coloured that been estimated with g1(x) coloured red that under-fit, g10(x) coloured blue that over- fit and g3(x) coloured green that has a good fit (generalised).

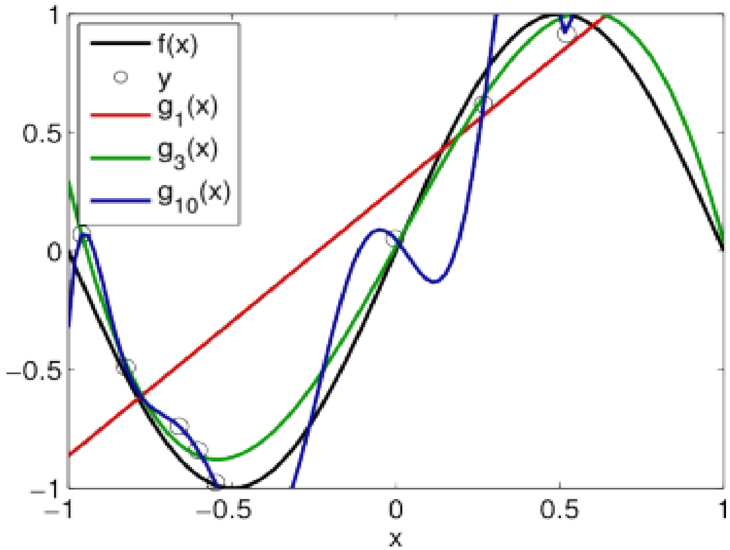



**Figure 3: over-fitting, under-fitting, and good fitting.**

In general over-fitting happens when the classifier can classify the training data more accurate than the test data while under-fitting happens when the classifier cannot classify the unseen data (test data) nor the seen data (training data). In order to solve this problem, there should be a trade-off between over-fitting and under-fitting so that one can get a good fit which known also as generalization [10].

One good practice that can be used to solve this problem is pruning technique which is: picking large random number of nodes for the hidden layer in order to get as accurate results as possible. This result will need a high training time mostly. Then one can start dropping some of the nodes and observe the difference in results accuracy since there are some nodes that have no effect on network performance and results accuracy but only add complexity to the network. Figure 4 shows the neural network suggested structure:

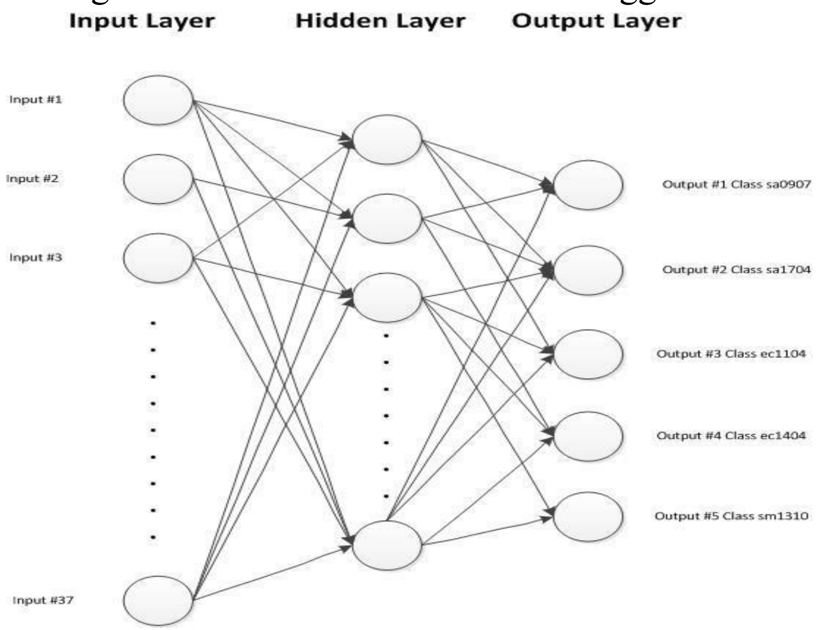



**Figure 4: Neural Network design**

Also, it is important when designing a neural network, choosing the appropriate activation functions for the neurons is a crucial step. Activation functions introduce non-linearity to the network and enable it to learn complex patterns and make non-linear predictions. In practice there is no rule of thumb for choosing the activation function for each layer. Although some suggest that using non-linear (sigmoidal function) with linear function (linear function) in classification problems can produce good results. They argue also that this function combination (non-linear and linear functions) will allow the network to converge within the minimum possible epoch. Epoch on the other hand is each cycle (i.e. each time the network is presented with a new input pattern) through a forward activation flow of outputs, and the backwards error propagation of weight adjustments [11][12].

## 4- Model implementation:

Implementation process of the proposed ANN-based model for classifying different bacteria was divided into the following stages:

### 4.1 Data preparation:

As mentioned before in the design part of this work, preparing data before training the neural network is a critical success factor for any classification project, including those using neural networks. Properly preparing the data ensures that it is in a suitable format and quality for training the neural network. Figure 5, shows a plot for the raw-data that representing the 5 different types of bacteria. As it can be seen from Figure 5, there are too many outliers, but notably two of them are too extreme and must be adjusted. These two extreme outliers occurred in (Staph aureus) file data as pointed out with arrow sign. There are many different methods that can be used for manipulating the outliers, the method used in this paper is taking the average value of the two adjacent points and replace it with the outlier value. Figure 8 below shows the outlier values located in data file (coloured red) before and after adjusting:

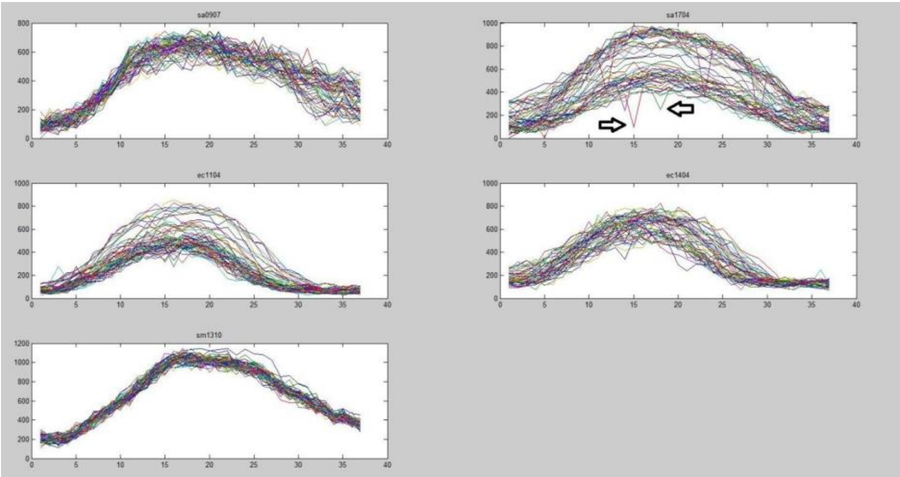



**Figure 5: Visualising and analysing the raw-data.**

As can be seen in above figure there are too many outliers, but notably two of them are too extreme and must be adjusted. These two extreme outliers occurred in (Staph aureus) file data as pointed out with arrow sign. There are many different methods that can be used for manipulating the outliers, the method used in this paper is taking the average value of the two adjacent points and replace it with the outlier value. Figure 6, shows the outlier values located in data file (coloured red) before and after adjusting:

| | |
|---|---|
| 338 | 391 |
| 412 | 350 |
| 418 | 380 |
| 446 | 93 |
| 470 | 438 |
| 412 | 445 |
| 246 | 426 |
| 419 | 479 |
| 473 | 446 |
| 445 | 436 |

| | |
|---|---|
| 338 | 391 |
| 412 | 350 |
| 418 | 380 |
| 446 | 409 |
| 470 | 438 |
| 412 | 445 |
| 415 | 426 |
| 419 | 479 |
| 473 | 446 |
| 445 | 436 |

**Figure 6: Adjusting outlier values.**

After adjusting the outlier values the plotted graph looked like in Figure 7:

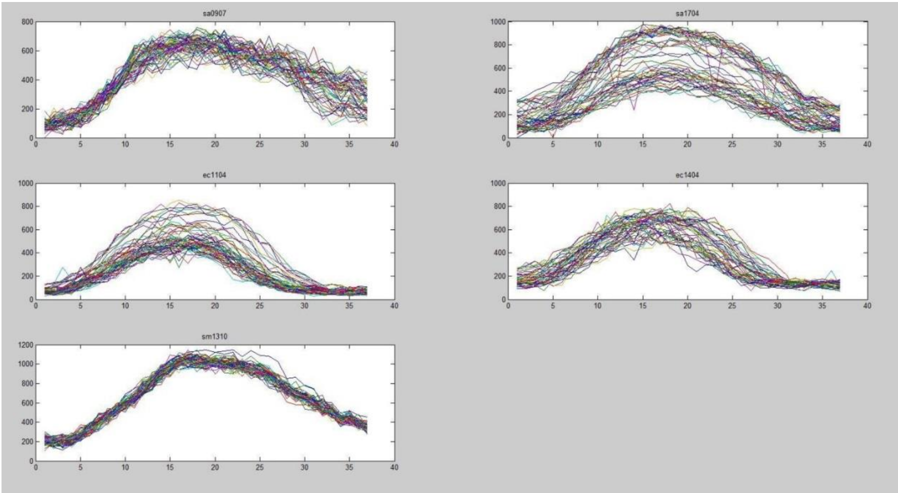



**Figure 7: Data after manipulating outliers.**

Next step is to compensate delay in the data by applying the compensate delay function, this function was provided during the MATLAB session. After applying this function, the new generated data is shown in Figure 8.

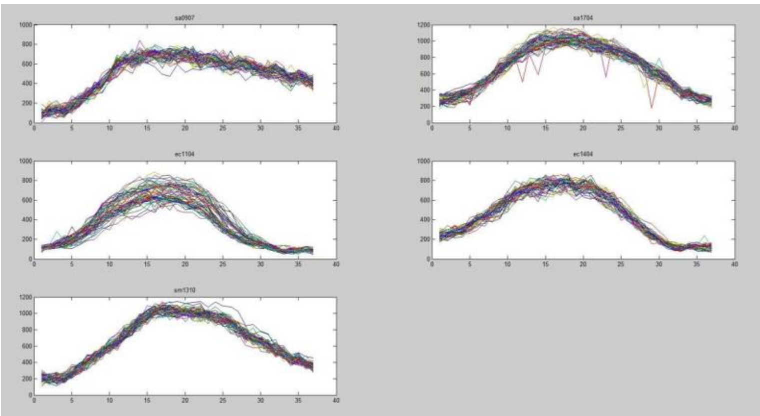



**Figure 8: plot data after compensating delay.**

The compensate delay function was used because plotting a 3D graph of the data showed that over time there was decay in the quantity of bacteria that was collected. This was due to some deterioration in equipment throughout the day, or some other factor. The compensate function calculates a linear formula that describes the decay over time for each data point. It then raises all the data up by the relevant amount to normalise it with the first data set of the day, as it is shown in Figure 9.

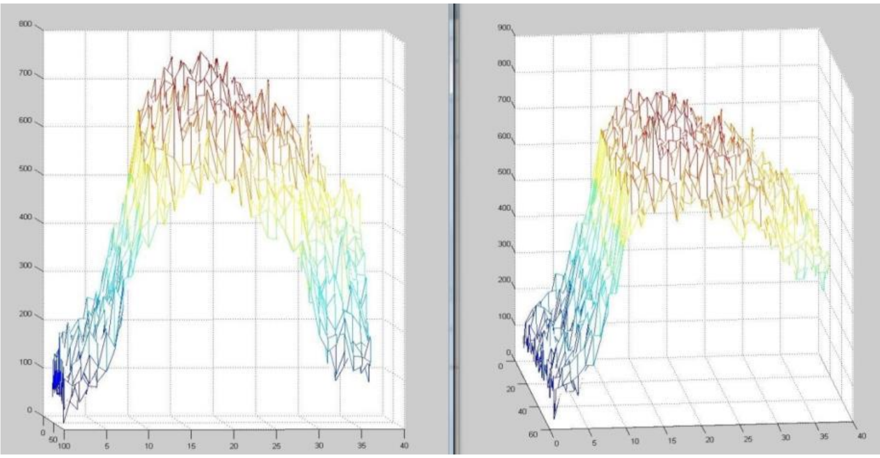


**Figure 9: Data Before and After Compensating Delay.**

The new generated data after compensating delay introduced new outliers that must be manipulated same way as before. Figure 9, above shows that (Staph aureus) produced four new extreme outliers that after manipulation are illustrated.

### 4.2Training and Testing using the ANN-based model:

Before training the proposed ANN-based model for classifying the bacteria types with the real collected data, dividing the pre-prepaid data into training and testing datasets is an essential step before training and evaluating the proposed ANN model. This division ensures that the model's performance is assessed on unseen data, providing a reliable estimate of its generalization capabilities. Here's an overview of the process. Now the data is divided into training and testing datasets with 70% and 30% for training and texting, respectively. Randomly divide the prepared dataset into two subsets: the training set and the testing set. The training set is used to train the ANN model, while the testing set is kept separate and used for evaluating the model's performance.

Next step is to train the model with the training data set. The training data must be concatenated in order to produce one large file of data that contains all the training data for all the different classes. Also, the target data must be generated, as mentioned before in the design phase the following values will be used for each bacteria class as it shown in Table 1.

| **Bacteria Class** | **Assigned Value** |
|---|---|
| Staphylococcus aureus | 10000 |
| Staph aureus | 01000 |
| Escherichia coli or e-coli | 00100 |
| e- coli | 00010 |
| Serratia marcescens | 00001 |

Table 1: The name of the 5 different types of bacteria and their assigned value**.**

At this point all the required data (training, testing and target data) have been produced and ready to use, so the next step is to implement and train the neural network itself. The implementation of neural network will use neural network toolbox in MATLAB environment. Using above assigned values to classify the bacteria types, many different settings and parameters were implemented and performance for each of them was measured and compared until the best solution was identified as will come in test part Also the following function was implemented which will accept matrix of values and threshold value and return a matrix for values mapped from the range of 0 to 1 into 0,1 or -1. As mentioned before -1 will mean that the value is undecided (fall outside the threshold value ranges), also will return the number of undecided values (-1 values).

To improve the accuracy of the model in classification, different numbers of hidden layers have been tested to consider the possible number of hidden layers that must be used in designing the neural network. As mentioned before, choosing too many hidden layers or neurons will result in an over-trained network that is: network will perform perfect for training data (seen data) but not the test data (unseen data). To calculate the best number of hidden layers and neurons following test cases implemented and POSTREG function used [13]. POSTREG function will perform regression analysis between the network actual output for the test data and the desired output for the test data. Table 2, shows different consideration of the number of hidden layers and their classification accuracy.

| Test Case: BacteriaClassification-001 | | POSTREG output | Classification Accuracy |
|---|---|---|---|
| **Test Description:** | Finding best hidden layers number and neurons | | |
| **Activaion Functions Used:** | Different types of activation functions | | |
| **Training Algorithm Used:** | Different types of training algorithm | | |
| **Input Layer Size:** | 37 inputs | | |
| **Output Layer size:** | 5 outputs | | |
| **Number of hidden layers:** 3 | **Neurons number:** 20, 20, 20 | M=0.7808, B=0.0274, R= 0.7306 | 73.97 |
| **Number of hidden layers:** 3 | **Neurons number:** 10, 10, 10 | M=0.9932, B=0.0342, R= 0.8422 | 83.78 |
| **Number of hidden layers:** 2 | **Neurons number:** 20, 20 | M=1.0034, B=0.0377, R= 0.8455 | 82.19 |
| **Number of hidden layers:** 2 | **Neurons number:** 10, 10 | M=1.0068, B=0.0068, R= 0.96505 | 94.25 |
| **Number of hidden layers:** 1 | **Neurons number:** 20 | M=1.0205, B=0.1164, R= 0.6750 | 76.45 |
| **Number of hidden layers:** 1 | **Neurons number:** 15 | M=0.9966, B=0.0171, R= 0.9242 | 93.35 |
| **Number of hidden layers:** 1 | **Neurons number:** 10 | M=0.9932, B=0.0205, R= 0.9041 | 90.8 |

**Table 2: choosing hidden layers specifications test cases summary.**

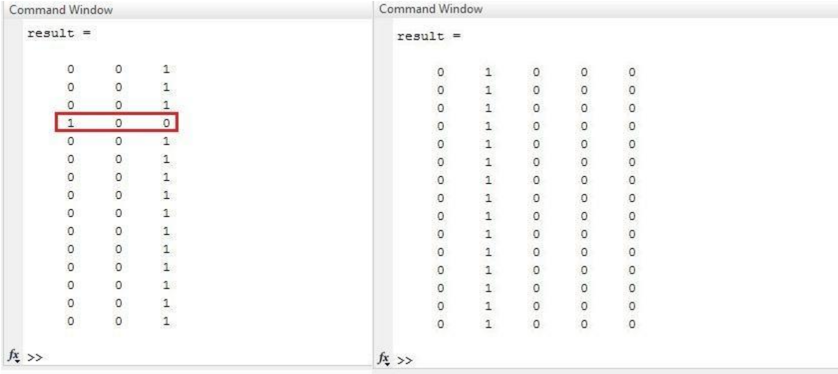
As can be seen from Table 2, although that one hidden layer neural network with (15) neurons can give good performance. The best results can be achieved using two hidden layers with (10) neurons each, experiments showed that although both have too close classification accuracy (94.25 and 93.35) two hidden neural network will require less epochs to converge, more on this in the coming test cases.

Although output layer has been designed as five nodes it also can have three nodes as mentioned before, so using neural network with two hidden layers each with 10 neurons the difference will be measured if there is any. Table 3, shows the expected output for both three and five neurons output layer:

| Bacteria Class | Five Nodes Output | Three Nodes Output |
|---|---|---|
| Staphylococcus aureus | 10000 | 000 |
| Staph aureus | 01000 | 001 |
| Escherichia coli | 00100 | 010 |
| e- coli | 00010 | 011 |
| Serratia marcescens | 00001 | 100 |

**Table 3: The expected output for both three and five neurons output layer.**

Empirical study showed that using only three output nodes resulted in misclassification for some of the bacteria types while using five output nodes minimizes these misclassifications into their minimum if none. Table 4, shows two trained neural networks first one with three output nodes while second one with five output nodes. Although both of them trained to get the best possible result, 3-output nodes network misclassified one bacteria from class (Staphylococcus aureus) while 5-output nodes produced the correct output:


Command Window

result =

| | | |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |

fx >>

Command Window

result =

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |

fx >>

**Table 4: 3-nodes against 5-nodes output layer network.**

As an analysis of the result, is that 3-nodes output layer network will mean less weights values than 5-nodes output layer, these less weights in 3-nodes network must be adjusted to generate all the desired output while for 5-nodes output the network will have a more expressive space to find a function as it has more weights to learn.

## 5- Results and Discussion:

Throughout the paper, it becomes evident that determining the optimal design for a neural network relies on trial and error methods, as there are no general rules to ascertain the best possible solution. Based on the results obtained, two neural network designs emerged as the most effective: one with a single hidden layer comprising 15 nodes and another with two hidden layers, each containing 10 neurons. Both of these networks employed the "Trainlm" training algorithm, utilized the "tansig" activation function in the hidden layer(s), and employed the "purelin" activation function in the output layer.

However, it is important to note that these conclusions are specific to the dataset and problem under consideration. The No Free Lunch Theorem cautions against generalizing these network designs as universally superior for all classification problems. Additionally, it is crucial to acknowledge that the high classification accuracy achieved in the experiments was only possible after manipulating the data through processes such as outlier removal and compensation for delay. Notably, when using raw, unprocessed data, the classification accuracy dropped significantly.

This observation does not imply that data manipulation, such as outlier removal and delay compensation, is universally the correct approach. In reality, outliers may carry important information and could be valuable features of the data that warrant special consideration. It is essential to carefully assess the nature of the outliers and determine their relevance in the specific context of the problem at hand.

In conclusion, while the study identifies specific neural network designs and data manipulation techniques that yielded favorable results for the given dataset, it is crucial to approach each classification problem with careful consideration of the data characteristics, potential outliers, and the specific requirements of the problem domain. The optimal network design and data preprocessing strategies may vary significantly depending on the unique attributes and complexities of each classification task.

## 6- Conclusion:

The research presented in this paper addresses the inherent challenge in designing an effective artificial neural network (ANN), namely the absence of general rules for selecting a specific network configuration. Consequently, the process of determining an optimal network design necessitates a trial-and-error approach. Given the multitude of factors influencing neural network performance, numerous combinations and settings must be thoroughly tested. While the study explored a broad range of neural network configurations, it is important to acknowledge that there are still additional variables that can be manipulated. For instance, one potential variable is the learning rate (LR) factor, which determines the magnitude of adjustments applied to the network's weights. Adjusting the learning rate to a higher value facilitates faster learning, but in the presence of significant input data variability, the network may struggle to learn effectively or may fail altogether. In essence, the most successful solutions thus far have been derived through comprehensive experimentation, exploring as many variations in neural network design as possible. It is crucial to note, however, that the absence of a standardized approach does not imply that the presented solutions represent the definitive best options.

**Reference:**

1- Francis H Shajin, Salini P, P. Rajesh & Venu Kadur Nagoji Rao (2023). Efficient Framework for Brain Tumour Classification using Hierarchical Deep Learning Neural Network Classifier, Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 11:3, 750-757, DOI: 10.1080/21681163.2022.2111719.

2- Jospin, L. V., Laga, H., Boussaid, F., Buntine, W. & Bennamoun, M (2022). Hands-on Bayesian neural networks-a tutorial for deep learning users. IEEE Comput. Intell. Mag. 17(2), 29–48.

3- Corso, M.P., Perez, F.L., Stefenon, S.F., Yow, K.C., García Ovejero, R.,
Leithardt. V.R.Q (2021): Classification of contaminated insulators using k-nearest neighbors based on computer vision. Computers 10(9), 112.

4- Wang, H., Kang, S., Zhao, X., Xu, N., Li, T (2021): Command filter-based adaptive neural control design for nonstrict-feedback nonlinear systems with multiple actuator constraints. IEEE Trans. Cybern. 1–10.

5- Ribeiro, M.H.D.M., Stefenon, S.F., de Lima, J.D., Nied, A., Marini, V.C,
Coelho, L.d.S (2020).: Electricity price forecasting based on self-adaptive decomposition and heterogeneous ensemble learning. Energies 13(19), 5190.

6- Stefenon, S.F., Branco, N.W., Nied, A., Bertol, D.W., Finardi, E.C., Sartori, A, et al (2020).: Analysis of training techniques of ANN for classification of insulators in electrical power systems. IET Gener., Transm. Distrib. 14(8),1591–1597.

7- G, M., Lotfi, A., & Pourabdollah, A. (2018). Human activities recognition based on neuro-fuzzy finite state machine. Technologies, vol. 6, no. 4, p. 110.

8- Mariey, L., et al. "Discrimination, classification, identification of microorganisms using FTIR spectroscopy and chemometrics." Vibrational spectroscopy 26.2 (2001): 151-159.

9- G. Mohmed, A. Lotfi, , and A. Pourabdollah, “Long short-term memory fuzzy finite state machine for human activity modelling,” in PETRA ’19, 2019.

10- A. N. Aicha, G. Englebienne, and B. Krose (2017). “Unsupervised visit ¨ detection in smart homes,” Pervasive and Mobile Computing, vol. 34, pp. 157–167.
11- T.-H. Tan, M. Gochoo, F.-R. Jean, S.-C. Huang, and S.-Y. Kuo (2017), “Frontdoor event classification algorithm for elderly people living alone in smart house using wireless binary sensors,” IEEE Access.
12- Duvenaud, D.K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru Guzik, A., Adams, R.P (2015): Convolutional networks on graphs for learning molecular fingerprints. In: Proceedings of the 29th Conference on Neural Information Processing Systems, pp. 2224–2232.
13- Huang, W., Song, G., Hong, H., Xie, K (2014): Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. 15(5), 2191–2201.
14- Krizhevsky, A., Sutskever, I., Hinton, G.E (2012): Imagenet classification with deep convolutional neural networks. In: Proceedings of the 26th Conference on Neural Information Processing Systems, pp. 1097–1105.
15- Ferreira, P. M., Faria, E., and Ruano, A (2002). Neural network models in greenhouse air temperature prediction. Neurocomputing 43, 1-4, 51–75.
16- M. Gochoo, T.-H. Tan, F.-R. Jean, S.-C. Huang, and S.-Y. Kuo, “Devicefree non-invasive front-door event classification algorithm for forget event detection using binary sensors in the smart house,” in IEEE International Conference on Systems, Man, and Cybernetics, 2017, pp.

405–409.

17- Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, “A hierarchical fused fuzzy deep neural network for data classification,” IEEE Transactions on Fuzzy Systems, vol. 25, no. 4, pp. 1006–1012, 2017.

18- I. N. Yulita, M. I. Fanany, and A. M. Arymurthy, “Fuzzy clustering and bidirectional long short-term memory for sleep stages classification,” in International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT), 2017, pp. 11–16.

19- G. Acampora, P. Foggia, A. Saggese, and M. Vento, “A hierarchical neuro-fuzzy architecture for human behavior analysis,” Information Sciences, vol. 310, pp. 130–148, 2015.

20- K. Wongpatikaseree, M. Ikeda, M. Buranarach, T. Supnithi, A. O. Lim, and Y. Tan, “Activity recognition using context-aware infrastructure ontology in smart home domain,” in 2012 Seventh International Conference on Knowledge, Information and Creativity Support Systems. IEEE, 2012, pp. 50–57.

21- S. Ali, S. Khusro, I. Ullah, A. Khan, and I. Khan, “Smartontosensor: ontology for semantic interpretation of smartphone sensors data for context-aware applications,” Journal of Sensors, vol. 2017, 2017.