Journal IJCSM

Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM

ISSN: 2788-7421 p-ISSN: 2958-0544

Classification of COVID-19 Symptoms Using Multilayer Perceptron

Nurulain Nusrah Mohd Azam[1], Mohd Arfian Ismail[1,2,*], Mohd Saberi Mohamad[3], Ashraf Osman Ibrahim[4], Shermina Jeba[5]

[1]Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600, Pekan, Malaysia.
[2]Center of Excellence for Artificial Intelligence & Data Science, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuhraya Tun Razak, Gambang 26300, Malaysia.
[3]Health Data Science Lab, Department of Genetics and Genomics, College of Medicine and Health Sciences, United Arab Emirates University, P.O. Box 17666, Al Ain, Abu Dhabi, United Arab Emirates
[4]Creative Advanced Machine Intelligence Research Centre, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu, Sabah, Malaysia.
[5]Department of Computing Muscat College Muscat,Oman.

*Corresponding Author: Mohd Arfian Ismail

DOI: https://doi.org/10.52866/ijcsm.2023.04.04.009
Received March 2023; Accepted December 2023; Available online October 2023

ABSTRACT: The COVID-19 virus had easily affected people worldwide through direct contact. Individuals diagnosed with positive COVID-19 virus may be affected with many symptoms, such as fever, tiredness, dry cough, difficulty in breathing, sore throat, chest pain, nasal congestion, runny nose, and diarrhea. An individual can also be diagnosed with COVID-19 even when he does not have any symptoms or be in contact with an infected person. Data classification was required due to the size of COVID-19 data that will be analyzed for future countermeasures determination. Some problems in data classification occurred due to unorganized data, such as time consumption, human error in complexity of symptom features and the diagnosis process data needed expert knowledge. This study aimed to use the artificial neural network (ANN) approach, which was multilayer perceptron (MLP) to classify the COVID-19 data by using patient symptom data. The MLP process involved data collection, data normalization, MLP design, MLP training, testing, and MLP verification. From the experiments, the MLP method was able to obtain an accuracy rate of 77.10%. In conclusion, the MLP method could classify the COVID-19 data and achieve a high accuracy rate.

Keywords: Multilayer Perceptron; Covid19; Artificial Neural Network; Machine Learning

1. INTRODUCTION

COVID-19 infection is a life-threatening disease which has claimed many human lives every day. The SARS-CoV-2 virus caused this and it usually spreads among people when a susceptible individual comes in direct contact with another. The virus can be transmitted via micro-liquid particles from an infected person's mouth or nose when he coughs, sneezes, speaks, sings or breathes heavily.

This life-threatening disease had affected not just one country but the entire globe . COVID-19 cases were increasing at a rapid pace all over the world. The World Health Organization (WHO) declared the condition to be a pandemic on March 11, 2020. According to Worldometers data, more than 130 million people globally contracted the COVID-19 disease between April 4, 2016 and April 4, 2021, with more than 2 million individuals dead.

As a result of this circumstance, a significant demand for such numerous data storage had developed and several data storage solutions were considered. The obvious rapid increase in cases and health data resulted in a valuable collection of data and information. These data were utilized for conducting virus research and development, as well as pandemic preparedness and countermeasures. Fever or chills, cough, chest tightness or difficulty in breathing, tiredness, muscle or body pains, headache, sudden loss of taste or smell, sore throat, blocked or runny nose, nausea or vomiting, and diarrhea are common symptoms in patients who were diagnosed positive.

These situations had led to the main problem, which was the massive COVID-19 infection data that may lead to unorganized data. Therefore, it was time-consuming when any operation was performed on these data. It was because of the complexity of patient symptom data which had many symptom features and number of infected people kept rising. Research experts needed more time to analyze the data, causing a slow process in diagnosing patients.

To manage COVID-19 data, the process of data classification needs to be implemented. Data classification is the process of grouping information into categories that make it easier to find, organize, and save information for later use. It contributes to a comprehensive understanding of the virus existence. The information gathered will be used to develop new preventive strategies in the future. Therefore, this study aims to utilise a deep learning approach called Multilayer Perceptron to classify the COVID-19 Data using patients' data symptoms. The contribution of this study is the development of a model and method to classify COVID-19 cases that can be used and facilitated by the medical practitioner in detecting COVID-19 cases.

2. REVIEW OF EXISTING METHODS

This section focuses on an existing classification method that has been widely used. A study on the artificial neural network (ANN) approach was also conducted as a direct consequence. The ANN is an artificial intelligence (AI) function that resembles the human brain processing of input and trend generation to produce selections. The ANN is an AI component of machine learning which utilizes neural networks to train unsupervised from unstructured or unlabeled data. In ANN, a neural network is a structure made up of machine learning algorithms in which artificial neurons serve as the central computing unit for discovering hidden patterns or relationships in a dataset. The input layer, hidden layer, and output layer are three layers which comprise a neural network. In ANN, there are many methods, such as recurrent neural networks (RNN), convolutional neural networks (CNN) and multilayer perceptron neural networks (MLP). RNN, CNN and MLP are three existing methods that were investigated in this research, whereby the research was started with the fundamentals of all approaches, including their benefits and drawbacks.

2.1 RECURRENT NEURAL NETWORK

The RNN is a form of ANN which works with sequence data or time-series data. The RNN instances include text conversion, natural language processing (NLP), speech detection and image tagging. The technique of context vectorizing is used to model sequential data with the time phase index t by using RNN. Context vectoring functions as memory capturing information on what has been computed so far, allowing the RNNs to recall past data and retain long and variable data sequences. RNNs can take one or more input vectors and consequently produce one or more output vectors. There are input vectors, weight vectors, hidden states, and output vector. The hidden state converts a sequence of patterns or meaning into a description variable. The outputs are affected not just by weights applied to the inputs, but instead by a hidden state vector which expresses context-based information from previous inputs, such that the same input might produce various outputs, depending on the context of subsequent inputs.

A work proposed in [1] used RNN to solve the COVID-19 classification. The dataset was collected from 60 healthy speakers and 20 COVID-19 patients. The participants were asked to record a sample of their cough, coughing, and speech effects. As a result, each researcher provided three samples. In this research, RNN architecture was applied by using long short-term memory (LSTM). Breathing sound had the highest quality, which accuracy increased to 98.2%. Then, for cough sounds, 97% accuracy was achieved. The system consistency with voices was just 88.2%. According to the findings, patient voices demonstrated such inconvenient precision as compared to cough and coughing noises. The explanation for these inefficient provisional findings was that the patients suffered from various types of respiratory illness.

Premature ventricular contraction (PVC) is the subject of research by [2]. When there are no major cardiovascular abnormalities, PVC is typically considered a benign symptom. In patients with systemic cardiac diseases and dysfunctions, premature heartbeats can lead to heart failure (HF), ventricular fibrillation (VF), as well as mortality. RNN showed great performance in detecting PVCs and may aid clinicians to interpret ECGs for arrhythmia diagnosis.

Based on [3], the work was on prediction of dengue. The proposed work utilized weather data (temperature, pressure, and humidity) from the TimeandDate.com website, rainfall records from Meteorological Department, air quality index from Central Pollution Control Board and number of dengue cases reported to the Health Department in Jodhpur. The information used was particular to Jodhpur. To solve the gradient problem and improve the model accuracy, RNN with LSTM was applied. The prediction accuracy of proposed method was 94% based on the results.

2.2 CONVOLUTIONAL NEURAL NETWORK

Yann LeCun initially introduced CNN, or ConvNets, in the 1980s [4]. Anomaly detection, object recognition and classification in medical field are commonly used in CNN. Convolutional, pooling, and fully connected layers are the three types of CNN layers. The core layer in CNN is convolutional layer which aims to extract features of inputs by applying filters. Filters are composed of small kernels. This layer utilizes a rectified non-linear unit (ReLU) for activation, which overcomes the vanishing gradient issue that occurs with sigmoid curves. Secondary features extraction is used in the pooling layer to minimize dimension feature maps and improve feature extraction robustness. Finally, in the fully connected layer, data from final feature maps are aggregated to generate the final categorization.

According to [5], the input data for CT scans are acquired from COVID-19 and nonCOVID-19 individuals, while the images are recognized by using the transfer learning model ResNet-50 (2020). The dataset was compiled from the Italian Society of Medical Radiology (SIRM), the Corona Virus open-source shared dataset. This dataset is accessible via https://github.com/ieee8023/covid-chestxray-dataset. The dataset was developed by assembling diagnosed images from several papers found on Radiopedia website (Radiopedia), and the dataset of X-ray scans on the chest from Kaggle [6]. Based on the research, the presented classification accuracy was high at 99.5%.

In [7], public source dataset was utilized in the research, whereby the data can be accessed from [8], [9]. This dataset had two types of data which were training data (174 images) and processing data (44 images). By using feature learning and classification process, the CNN model could dynamically train the X-ray picture feature. With a batch size of 100 and epoch numbers of 15, 20, 25, and 30, the first model utilized a 225x225 picture matrix, a 20x20 convolution matrix, and a 10x10 pooling matrix. It scored 94.99%, 95% and 95.47% in memory, precision, and accuracy, respectively. With a batch size of 100 and epoch numbers of 15, 20, 25, and 30, the second model utilized a 225x225 picture matrix, a 30x30 convolution matrix, and a 20x20 pooling matrix. It scored 97.73%, 95% and 96.59% in memory, precision, and consistency, respectively.

Based on [10], dataset was published by the Department of Montreal [8], [9] for COVID-19 patients and Kaggle dataset for bacteria infected pneumonic patients. The COVID-19 patients were detected by using a CNN model which has a training accuracy of 100% and a validation accuracy of 95.24%. Accuracy test 96.66% was also achieved.

2.3 MULTI-LAYER PERCEPTRON

Frank Rosenblatt is the inventor of multi-layer perceptron (MLP) neural networks. The MLP is an ANN with several layers. It consists of collections of perceptron. They comprise an input layer which accepts the data of an output layer that makes a judgment or inference about the input and an arbitrary number of hidden layers in between which are the MLP true computational engine. Signals are typically sent from input to output in a network. There is no loop since the output of each neuron has no influence on the neuron itself [11].

MLP was utilized in a study proposed by [12]. A database of 400 fingerprint pictures was used in the research. Fingerprints of 40 people were stored in this database. There were 10 samples in various positions for each person. The study calculated the pseudo zernik moment (PZM) by using a sub image from each image. The dataset was divided into two groups at random. The first set was for planning, while the second set was for verification. The PZM function vector was fed into the MLP. The MLP had the same amount of input neurons as the PZM, but there were 40 groups since there were 40 output neurons. The number of secret neurons was calculated in several experiments. After testing on the training package, the MLP was put to test on a test set that it has never seen before. If the input vector is accurately allocated to its associated class, the recognition is accurate; otherwise, it is inaccurate. According to this study, the proposed method was able to accurately classify all fingerprints in the database.

The Lung Image Database Consortium image collection (LIDC-IDRI) includes CT scan images for diagnosis of lung cancer [13]. The collection contains DICOM images of 1018 instances, including normal, benign, and malignant instances. The derived features of non-tumorous and tumorous images were fed into the classification algorithm. Training and testing are the two primary steps of the classification process. The classification of lung CT scan images into non-tumorous and tumorous classes was performed by using two classification algorithms, which were K-nearest Neighbor (KNN) and MLP. According to the research, the proposed methodology obtained a KNN accuracy of 98.30% and an MLP algorithm accuracy of 98.31%.

Based on studies in [14], two feature preference techniques and a MLP neural network were utilized to categorize human chromosomes. The first method was the "knock-out" procedure and the second was principal component analysis. As chromosomal categorization, the "knock-out" technique showed the importance of the centromeric index and chromosome length. On the other hand, the large training sets usage allowed tremendous data compression. Both algorithms had the gain of requiring only about 70% of the options available to provide almost the best classification results.

2.4 COMPARATIVE ANALYSIS OF EXISTING METHODS

The benefit of implementing RNN is that RNN has the ability to recall everything that happens over time. This is called LSTM. This happened on time series forecasting because it can recall recent inputs and has a function to remember previous inputs as well. LSTM is a type of memory in which data passes via a system known cell states, which is dependent on three different connections: the previous cell state, hidden state and input at the current time step. Other advantages of the RNN are it has a dynamic neural network which makes it perform better and computationally powerful in temporal processing models and applications [15]. Besides that, RNN can approximate any nonlinear dynamical system with arbitrary precision (given weak regularity constraints) by implementing intricate mappings from input to output sequences. Due to its ability the RNN is known as universal approximation property [16]. Another major advantage of RNN as compared to other ANN methods is the RNN size is substantially less for the problem [15]. Besides the many advantages, RNN suffers on its performance. Although the recurrent network is a straightforward and effective model, adequate training is challenging in practice [17]. Another deficiency in RNN can be seen in this situation. The recurrent inputs are all linked together [18]. In other words, this strategy will result in explosive gradients which will build up during an update, and thus result in a big gradient. The explosion happens due to exponential growth, which is achieved by continually multiplying gradients in network layers with values greater than 1.0 [19]. The vanishing gradient problem makes it impossible to learn some long duration correlations due to the inconsistent process, while the growing gradient leads terms to exponentially run to infinity; hence, their value can become NaN [20]. Moreover, the output of the previous layers is used as input for the next layers. As a result, planning for the time point is ongoing, and it is reliant on untrained layer inputs. The entire network is not fully trained because of the diminishing gradient. In addition, it is frequently challenging to determine the network stability because of the nonlinear nature of unit activation output characteristics and weight adjustment procedures [21]. The tanh or relu function also gives some troubles to RNN. If the tanh or relu function is used as the activation function, it cannot process very long sequences [15]. Lastly, the RNN tends to be stuck in local optimum problem, whereby the RNN is a constrained optimization algorithm. This is because the RNN operations are only optimized locally [22].

The CNN has advantage of having a high accuracy in recognizing important properties in images without the need for human input. This is because each filter has its own feature map [23]. These would be subsequently processed via an activation function, which determines if a certain characteristic is available in a particular position in the image. Another advantage of CNN is that CNN also performs better in computer vision domain, whereby CNN can avoid overfitting and decreases the number of trainable network parameters, which in turn aids in generalization improvement. This makes CNN easier to be used [24]. Besides that, CNN can also learn accurately by representing the important features as an object in image [25]. In addition, CNN is robust and performs well under challenging circumstances, such as the photo size and orientation, lighting, complicated background and various resolutions [26]. On the other hand, the disadvantage of this approach is that the targeted location and direction are not stored in CNN predictions. They lose all intrinsic features about object posture and orientation and send all data to the same neurons, which may or may not be able to process it. Designing the CNN is a difficult task. It involves many hyperparameters, such as number of convolution layer, type of pooling, type of activation function, number of fully connected layer in architecture, and many more. In addition, CNN is very sensitive, whereby small change on the value of hyperparameters can affect the performance of CNN [27], [28]. Moreover, CNN requires a lot of data for training to produce a good model [26], [27], [29]. Due to the complex structure of CNN networks architecture, CNN also suffers optimization problem with pre-trained model on small dataset [26].

As for MLP, it has the ability to generalize, which means it can classify a new pattern with other detected patterns that has comparable distinguishing properties. Besides that, noisy or partial inputs can be graded since they resemble pure and complete inputs. One of the main benefits of using MLP is its performance. Yu et al. demonstrated in their works that MLP performed well and robust when applied on imbalanced class composition and inaccurate class labels [30]. Moreover, MLP can predict and analyze the complex problem because it has some hidden layers that are able to learn the problem [31]. As opposed to these advantages, MLP necessitates long calculation times and a long training period since a hidden layer is necessary for computational purposes, increasing the necessary processing time [32]. In addition, one main difficult task in using MLP is to determine the suitable and correct network architecture that associate with input or hyperparameters in solving the problem [33]. Apart of that, another issue in MLP is to decide the input of data that must be connected correctly to its targeted variables [34]. Even though MLP has showed its ability in various domains, its performance really depends on the structure of data which are used in training and testing process. The predicting model produced by MLP is very sensitive and highly determined by parameters used in the training process. Due to that, care should be taken when choosing these parameters [35].

This section has discussed on the existing methods and comparative analysis between the existing methods, which are RNN, CNN MLP. The MLP method was chosen among these three methods. MLP has ability to classify unfamiliar pattern with other known patterns that have similar characteristics because it has ability to solve problems stochastically which allows approximate solutions for extremely complex problems [36].

3. METHODOLOGY

3.1 DATASET COLLECTION

Dataset of COVID-19 Symptoms Checker from Kaggle [37] was based on certain pre-defined standard symptoms by The World Health Organization. This dataset will aid in determining whether an individual has coronavirus disease or not. This dataset contains eleven major variables, which are fever, tiredness, dry-cough, difficulty-in-breathing, sore throat, no symptom, chest pain, nasal congestion, runny nose, diarrhea and information on whether the person has contact with positive a COVID-19 person.

3.2 DATA NORMALIZATION

The method of resizing data from its actual range such that all values lie between 0 and 1 is known as normalization. Normalization requires precise measurement of the minimum and maximum observable values. Data normalization is used to make the large amount of data to take up less space when stored in a database. Equation 1 gives the formula of data normalization used in this study.

$$v' = v - minAmaxA - minA((new\_maxA) - new\_minA) + new\_minA \tag{1}$$

where v is actual input data, $minA$ is minimum input data, $maxA$ is denoted as maximum input data, new_maxA is maximum target value and new_minA is minimum target value.

3.3 MULTI-LAYER PERCEPTRON DESIGN

The input layer and output layer of a MLP are fully coupled. Between the input and output layers, there are several hidden levels. The MLP algorithm is as follows:

i The inputs are shifted forward via the MLP by taking the dot product of input with weights that occur between input layer and hidden layer. This dot product produces a value at the concealed layer. In Figure 1, it is referred to as the input layer.

ii MLPs use activation functions at each of their measured layers. A rectified linear unit (ReLU) is a form of activation function which calculates output of the current layer if it receives a positive value and returns 0 if it receives a negative value.

iii After the hidden layer has calculated the output, push the dot product with corresponding weights to the next layer.

iv Repeat steps i and ii until you reach the output layer.

v The results will be employed in a backpropagation method which corresponds to the MLP activation function in the output layer.

3.4 MULTI-LAYER PERCEPTRON TRAINING

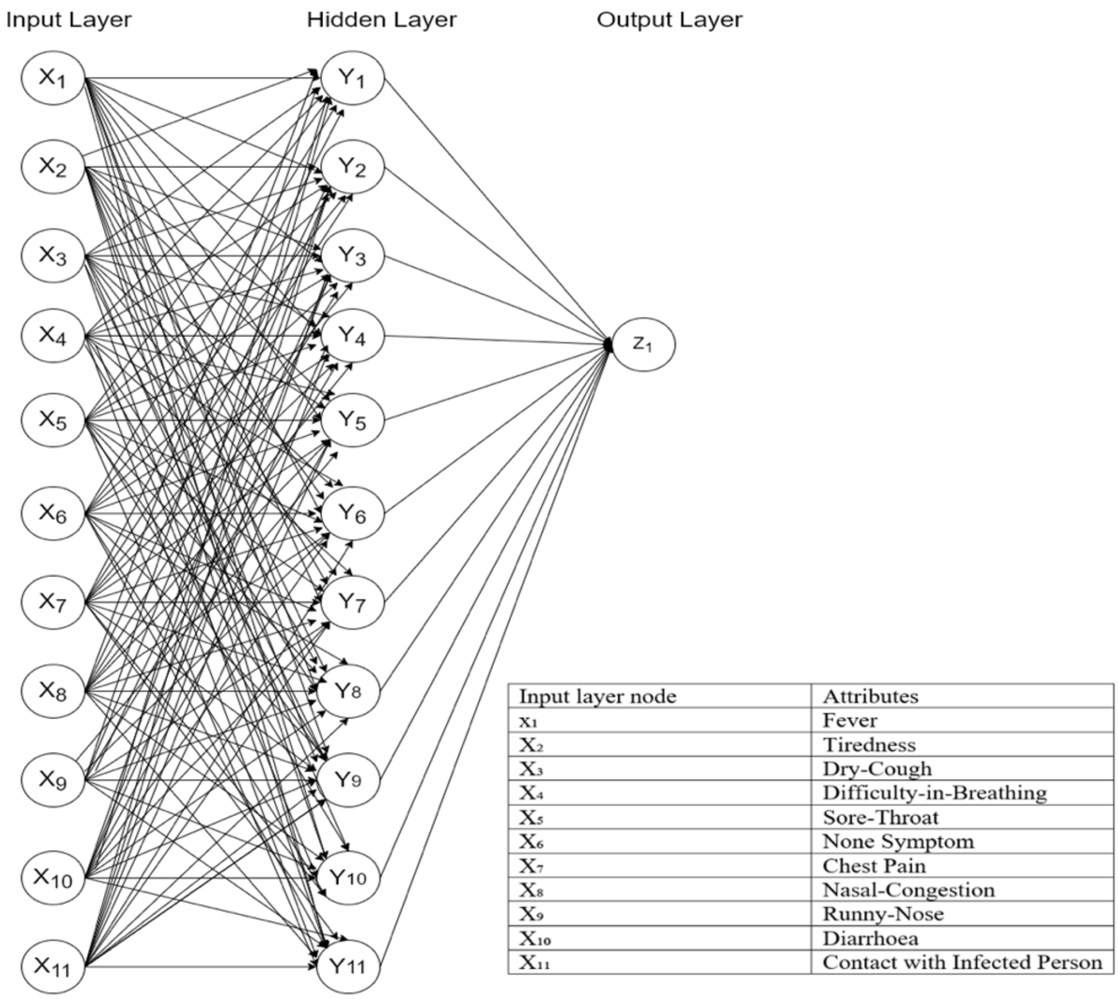
In a supervised manner, the backpropagation algorithm can be used to train a multilayer neural network. The feed-forward and backpropagation steps of BP algorithm can be divided into two categories. In the feedforward step, an input pattern is applied to the input layer, and its impact is carried across the network layer over layer until an output is generated. The actual output value of network is then compared to the predicted output value, and an error value is calculated for each of the output nodes. Because all hidden nodes contributed in some manner to output layer errors, the output error impulses are delivered reverse from the output layer to each hidden node that has contributed to the output layer.

3.5 MULTI-LAYER PERCEPTRON TESTING AND VERIFICATION

The mean square error (MSE) is used to assess and evaluate the correctness of multi-layer perceptron. The formula of MSE is given by Equation 2 as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i) \tag{2}$$

where, n is number of test set, x is observed values and y is predicted values.



Input layer node	Attributes
x_1	Fever
X_2	Tiredness
X_3	Dry-Cough
X_4	Difficulty-in-Breathing
X_5	Sore-Throat
X_6	None Symptom
X_7	Chest Pain
X_8	Nasal-Congestion
X_9	Runny-Nose
X_{10}	Diarrhoea
X_{11}	Contact with Infected Person

FIGURE 1. Example multilayer perceptron with 13 neurons

3.6 IMPLEMENTATION OF MULTI-LAYER PERCEPTRON

The MLP in this study was implemented by using MATLAB script and can be downloaded from https://github.com/unpiye/MLP. To classify the covid symptoms dataset, *nprtool* GUI is use in classification process. Figure 2 shows the MATLAB script. The explanations of the script are as follows:

i In Line 9 and Line 10, the input data and target data are transposed and assigned as variable `x` and `t`.

ii In Line 17, `trainscg` is used in training function because this function can update weight and bias values according to the scaled conjugate gradient method.

iii In Line 20, `hiddenLayerSize` is assigned to 10 number of neurons and it can modify as desired by requirement.

iv Line 21 shows that the pattern recognition network is created according to size of hidden layers and training function.

v From Line 24 until Line 26 show that the data of input vector and target vector are divided for training with 70%, validation and testing with 15% each.
```
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
```

vi In Line 27, mean squared error (mse) is chosen as performance function to assess and evaluate the correctness of multi-layer perceptron.

vii Line 29 shows the data that plotted in `plotperform` is used to generate the plot performance graph during training, validating and testing data, `plottrainstate` is used to plot training state values, `plotterhist` is used to plot histogram error, `plotconfusion` is used to show the percentage of classified data after training, validating and testing, `plotroc` is used to show the performance of a classification model at all classification thresholds.

viii `[net,tr] = train(net, x, t);` : This command is used to train the network which can be reviewed in Line 32.

ix From Line 35 until Line 40, it shows the testing network process, where the performance of the network can be identified.

x Line 43 shows the network that has been created.

The network was tested after the training phase by using the backpropagation approach. Backpropagation finds the minimal error function in weight space by using the training method scaled conjugated gradient backpropagation (`trainscg`). The weights were first assigned in a general way. Weights were modified to reduce overall system inaccuracy. Starting with output layer, the weights are updated in reverse order. As a result, errors from the output node propagates back to the inner nodes. Because it minimizes and optimizes the error function, the number of neurons could be the solution to the learning problem. The network was given a few training cycles, as changing the amounts of neurons can produce good output. In this study, a network design with 19 various numbers of neuron was built.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Several experiments were conducted, and Table 1 displays the results of the first through fourth training sessions for 19 various numbers of neurons. The number of neurons with 25 provides the best result out of 19 various numbers of neuron. The confusion plot presents the confusion matrix in Figure 3. The confusion plot describes the detail results of the accuracy for training, testing and validation dataset by using the MLP.

On the first two diagonal cells (green and red) of the test confusion matrix (Figure 3), it showed no individual was accurately classified as not infected by the Corona virus. About 77.1 % of people were appropriately identified as being infected with the Coronavirus. In the red diagonal, 22.9 % of people were incorrectly identified as not infected by the Corona virus, whereas 0 % of individuals infected with the Corona virus are incorrectly classified. About 77.1 % of the whole testing dataset was correctly classified, whereas the remaining 22.9 % was incorrectly classified. The MLP must go through 48 iterations to recognize and learn the dataset. Figure 4 plots the best performance, whereby it occurred at epoch (iteration) 42. In the graph, the mse is reduced after more iterations of training. The graph shows the difference between target and the actual simulation of neural network design is presented in the graph.

The best result obtained in the experiment on COVID-19 data classification based on the symptoms dataset was 77.1 % and the worst was 72.6%. On the first training, the lowest results were obtained when the number of neurons was 38. This could be because the dataset size was small.The usage of MLP is effective in large dataset to prevent dataset form overfitting. This factor has impact on the outcome.

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by Neural Pattern Recognition app
%
% This script assumes these variables are defined:
%
%   covidInput - input data.
%   targetCovid - target data.

x = covidInput.';
t = targetCovid.';

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainscg';  % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = 10;% Adjust as desired
net = patternnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
net.performFcn = 'mse';

net.plotFcns = {'plotperform','plottrainstate','ploterrhist','plotconfusion','plotroc'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

% View the Network
view(net)
```

FIGURE 2. The MATLAB script of MLP

Table 1. Result of first training until forth training for 19 different numbers of neuron

Number of neurons	Accuracy			
	Result 1	Result 2	Result 3	Result 4
1	75.3	75.1	76.3	74
2	74.3	75.5	75.9	76.5
3	74.3	73.9	76.1	75.9
4	75.9	75.4	75.7	75.1
5	74.4	73.3	75	74.9
6	76.3	73.7	75.7	74.9
7	76.5	74.8	73.7	74.3
8	75.9	74.1	73.7	75.4
9	75.9	74.8	74.6	73.9
10	74.7	75	75.3	74.3
15	75.6	74.7	75.2	75.1
18	74.3	74.4	72.9	72.7
20	76.1	74.3	74.6	76.7
25	77.1	76.3	74.9	75.9
28	74.9	74.1	75.1	75.5
30	75	75.7	75.7	75.1
35	75.1	75.1	76.2	74.9
38	72.6	74.6	75.5	74.7
40	75.9	75.2	76.6	75.1

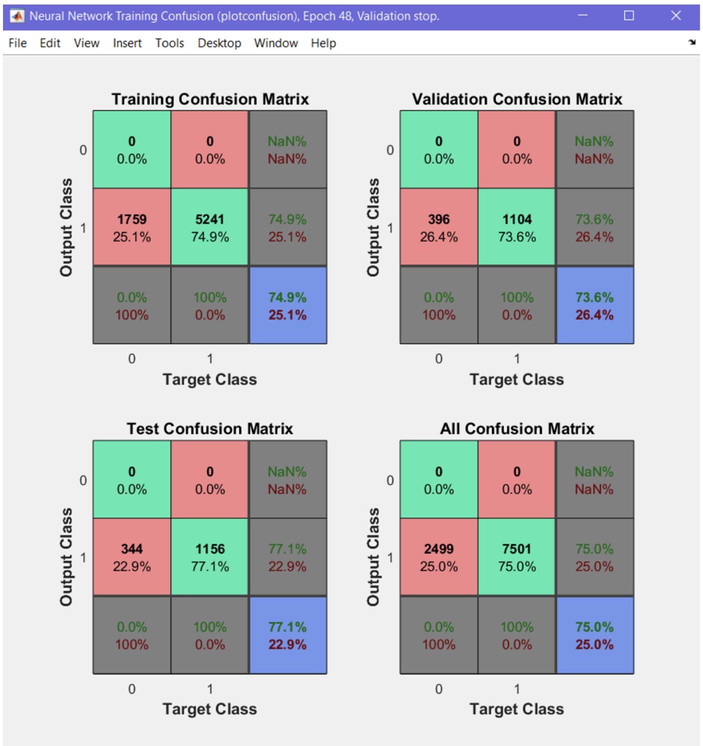


FIGURE 3. Confusion plot for first result with 25 neurons

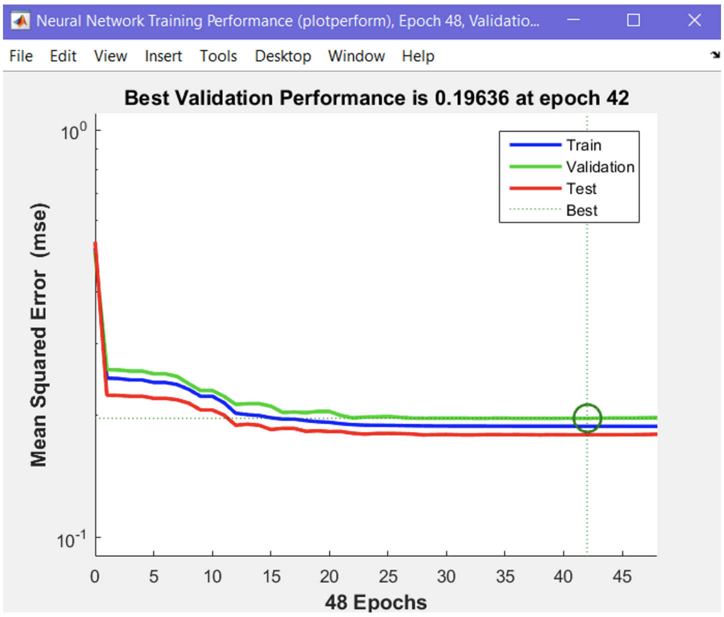


FIGURE 4. **Performance plot for first result with 25 neurons**

5. CONCLUSION

MLP is one of the methods in ANN. A variety of ANN methods are used widely to solve the classification problem. In this research, MLP method was used in the classification of COVID-19 symptoms. From the result obtained, the MLP method produced many iterations and mse was quite high, showing that the method was not suitable for a small dataset. Data preparation, network design, network configuration, network initialization, network training, and network validation were the six essential phases in MLP training and testing network designs. The training and testing basic procedures were applied using MATLAB Toolbox, nprtool. Data preparation was necessary in the processes to have a good result. The number of neurons in a design was then assigned to the newly created network, which was then configured. The network was created via the nprtool. After the network was first set up, the training process began. Finally, the network was evaluated to ensure that it will meet the requirements. The result was then generated to show the the accuracy of the model. Several tests were conducted by using a set range number of neurons to get better result for classifying the COVID-19 symptoms. The MLP was able to produce a good result. In the future, several improvements can be considered. Firstly, an improvement in identifying COVID-19 symptoms by adding new input data. A new patient can be added to the list without having a record in the dataset, allowing the MLP to classify, identify the result, and improve MLP accuracy. In addition, this study shows the benefit of using MLP to classify COVID-19 cases. But the results can be improved, and the process of MLP can be optimized. Applying the metaheuristic method seems to be a good choice to optimize the MLP process. Finally, it is hoped that the techniques used in this study can be applied to various fields such as in block chain [40], cloud [41] [38], cyber-security [45] [39], communication [46], data privacy [42] [43] and medical [44].

FUNDING

None.

ACKNOWLEDGEMENT

This study was supported by Fundamental Research Grant (RDU) with vot No. RDU220304 from Universiti Malaysia Pahang Al-Sultan Abdullah. The authors would like to thank the reviewers for providing useful suggestions, allowing for the improved presentation of this paper.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] A. Hassan, I. Shahin, and M. B. Alsabek, "Covid-19 detection system using recurrent neural networks," pp. 1–5, 2020.
[2] X. Zhou, X. Zhu, K. Nakamura, and N. Mahito, "Premature ventricular contraction detection from ambulatory ecg using recurrent neural networks," pp. 2551–2554, 2018.
[3] M. Chovatiya, A. Dhameliya, J. Deokar, J. Gonsalves, and A. Mathur, "Prediction of dengue using recurrent neural network," pp. 926–929, 2019.
[4] B. Dickson, "What are convolutional neural networks (cnn)?," 1 2020.
[5] Z. KARHAN and F. AKAL, "Covid-19 classification using deep learning in chest x-ray images," pp. 1–4, 2020.
[6] P. Mooney, "Chest x-ray images (pneumonia)," 7 2018.
[7] D. Nurtiyasari and D. Rosadi, "Covid-19 chest x-ray classification using convolutional neural network architectures," pp. 667–671, 2020.
[8] J. P. Cohen, P. Morrison, and L. Dao, "Covid-19 image data collection," 2020.
[9] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, "Covid-19 image data collection: Prospective predictions are the future," 2020.
[10] Y. Khan, P. Khan, S. Kumar, J. Singh, and R. M. Hegde, "Detection and spread prediction of covid-19 from chest x-ray images using convolutional neural network-gaussian mixture model," pp. 1–6, 2020.
[11] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS*, vol. 8, pp. 579–588, 2009.
[12] I. El-Feghi, A. Tahar, H. Aboasha, and Z. Xu, "Efficient features extraction for fingerprint classification with multi layer perceptron neural network," pp. 1–4, 2011.
[13] S. Potghan, R. Rajamenakshi, and A. Bhise, "Multi-layer perceptron based lung tumor classification," pp. 499–502, 2018.
[14] B. Lerner, M. Levinstein, B. Rosenberg, H. Guterman, L. Dinstein, and Y. Romem, "Feature selection and chromosome classification using a multilayer perceptron neural network," vol. 6, pp. 3540–3545 vol.6, 1994.
[15] A.-N. Sharkawy, "Principle of neural network and its main types: Review," *Journal of Advances in Applied Computational Mathematics*, vol. 7, pp. 8–19, 8 2020.
[16] P. A. N. D. S. E. D. M. Wolfgang and Joshi, "Computational aspects of feedback in neural circuits," *PLOS Computational Biology*, vol. 3, pp. 1–20, 4 2007.
[17] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," vol. abs/1211.5063, 2013.
[18] A. Sarangan, "Recurrent neural network: A complete guide in 5 easy steps," 12 2020.
[19] J. Brownlee, "A gentle introduction to exploding gradients in neural networks - machinelearningmastery.com," 8 2019.
[20] B. Or, "The exploding and vanishing gradients problem in time series," 10 2020.
[21] Y. M. Chiang, L. C. Chang, and F. J. Chang, "Comparison of static-feedforward and dynamic-feedback neural networks for rainfall–runoff modeling," *Journal of Hydrology*, vol. 290, pp. 297–311, 5 2004.
[22] Y. Li, S. Li, and B. Hannaford, "A novel recurrent neural network for improving redundant manipulator motion planning completeness," *IEEE International Conference on Robotics and Automation : ICRA : [proceedings]. IEEE International Conference on Robotics and Automation*, vol. 2018, p. 2956, 9 2018.
[23] A. Dertat, "Applied deep learning - part 4: Convolutional neural networks," 11 2017.
[24] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, p. 53, 2021.
[25] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," *Computation*, vol. 11, 2023.
[26] A. Kamilaris and F. X. Prenafeta-Boldú, "A review of the use of convolutional neural networks in agriculture," *The Journal of Agricultural Science*, vol. 156, pp. 312–322, 2018.
[27] M. A. Saleem, N. Senan, F. Wahid, M. Aamir, A. Samad, and M. Khan, "Comparative analysis of recent architecture of convolutional neural network," *Mathematical Problems in Engineering*, vol. 2022, p. 7313612, 2022.
[28] T. Hinz, N. Navarro-Guerrero, S. Magg, and S. Wermter, "Speeding up the hyperparameter optimization of deep convolutional neural networks," *International Journal of Computational Intelligence and Applications*, vol. 17, p. 1850008, 6 2018. doi: 10.1142/S1469026818500086.
[29] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," *Neurocomputing*, vol. 323, pp. 37–51, 1 2019.
[30] H. Yu, D. C. Samuels, Y. yong Zhao, and Y. Guo, "Architectures and accuracy of artificial neural network for disease classification from omics data," *BMC Genomics*, vol. 20, p. 167, 2019.
[31] D. E. Cahyani, E. P. Gunawan, and L. Gumilar, "Classification of scholarship applicants using multilayer perceptron neural network," *AIP Conference Proceedings*, vol. 2639, p. 030004, 11 2022.
[32] V. T. Le, N. H. Quan, H. H. Loc, N. T. T. Duyen, T. D. Dung, H. D. Nguyen, and Q. H. Do, "A multidisciplinary approach for evaluating spatial and temporal variations in water quality," *Water*, vol. 11, 2019.
[33] J. Isabona, A. L. Imoize, S. Ojo, O. Karunwi, Y. Kim, C.-C. Lee, and C.-T. Li, "Development of a multilayer perceptron neural network for optimal predictive modeling in urban microcellular radio environments," *Applied Sciences*, vol. 12, 2022.
[34] J. Isabona and A. I. Osaigbovo, "Investigating predictive capabilities of rbfnn, mlpnn and grnn models for lte cellular network radio signal power datasets," *FUOYE Journal of Engineering and Technology*, vol. 4, 3 2019.
[35] R. Barzegar and A. A. Moghaddam, "Combining the advantages of neural networks using the concept of committee machine in the groundwater salinity prediction," *Modeling Earth Systems and Environment*, vol. 2, p. 26, 2016.
[36] L. S. Thota and S. B. Changalasetty, "Optimum learning rate for classification problem with mlp in data mining," *International Journal of Advances in Engineering Technology*, vol. 6, pp. 35–44, 2013.
[37] B. Hungund, "Covid-19 symptoms checker," 2020.
[38] M. Aljanabi, S. N. Abd-Alwahab, R. D. Saedudin, H. R. Ebraheem, Defni, R. Hadi, and M. A. Ismail, "Cloud computing issues, challenges, and needs: A survey," *JOIV : International Journal on Informatics Visualization*, vol. 5, pp. 298–305, 9 2021.
[39] M. Aljanabi, M. A. Ismail, and A. H. Ali, "Intrusion detection systems, issues, challenges, and needs," *International Journal of Computational Intelligence Systems*, vol. 14, pp. 560–571, 2021.
[40] L. KADHUM, A. Firdaus, M. F. B. Zolkipli, S. I. Hisham, L. Bayuaji, and M. F. A. Razak, "Auto–transition in radg based on chaotic system," *Iraqi Journal For Computer Science and Mathematics*, vol. 4, pp. 97–108, 3 2023.

[41] F. D. Ahmed and M. A. Majid, "Towards agent-based petri net decision making modelling for cloud service composition: A literature survey," *Journal of Network and Computer Applications*, vol. 130, pp. 14–38, 2019.
[42] S. Jeba, M. Binjubier, M. A. Ismail, R. Krishnan, S. Nair, and G. Narasimhan, "Classifying and evaluating privacy-preserving techniques based on protection methods: A comprehensive study," *Journal of Theoretical and Applied Information Technology*, vol. 15, 2022.
[43] S. Jeba, M. BinJubier, M. A. Ismail, R. Krishnan, S. Nair, and G. Narasimhan, "A hybrid protection method to enhance data utility while preserving the privacy of medical patients data publishing," *International Journal of Advanced Computer Science and Applications*, vol. 13, 2022.
[44] N. F. Idris and M. A. Ismail, "The study of cross-validated bagging fuzzy-id3 algorithm for breast cancer classification," *Journal of Intelligent Fuzzy Systems*, vol. 43, pp. 2567–2577, 1 2022.
[45] N. S. Nordin and M. A. Ismail, "A hybridization of butterfly optimization algorithm and harmony search for fuzzy modelling in phishing attack detection," *Neural Computing and Applications*, vol. 35, pp. 5501–5512, 3 2023.
[46] Sutarman, M. B. A. Majid, J. B. M. Zain, and A. Hermawan, "Recognition of malaysian sign language using skeleton data with neural network," pp. 231–236, 2015.