International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.38, September 2025

GAs-based Weight Optimization of Multilayer Perceptron Neural Networks for Air Quality Prediction

Anthony Stone and Alaa Sheta

Department of Computer Science,
Southern Connecticut State University,
New Haven, CT, USA

ABSTRACT

Air pollution, particularly the concentration of particulate matter (PM10), poses significant risks to public health and environmental quality. Therefore, this study proposes a Multilayer Perceptron (MLP) optimized by a Genetic Algorithm (GA) to develop a predictive modeling approach for estimating PM_{10} levels, capturing the complex interactions between atmospheric conditions and pollutants. The model incorporates twelve key input variables, including meteorological conditions and pollutant indicators such as temperature, CO, NO, NO_2, NO_x, $PM_{2.5}$, O_3, RH, SO_2, wind direction (WD), wind speed (WS), and lagged PM_{10} values. The dataset was divided into 70% for training and 30% for testing. The recommended model demonstrated a remarkable capacity to identify complex patterns in the PM_{10} data by generating remarkably accurate predictions and a strong correlation with actual results during training. These results demonstrate the effectiveness of evolutionary optimization in enhancing FNN-based models for predicting air quality.

Keywords

Artificial Neural Networks, Multilayer Perceptron, Weight Optimization, Metaheuristics, Genetic Algorithm, Optimization

1. INTRODUCTION

It is widely acknowledged that air pollution poses a significant threat to global public health. Numerous studies have connected poor air quality to a wide range of dangerous medical conditions, such as heart attacks, lung damage, respiratory and cardiovascular disorders, and different types of cancer [1–5]. Furthermore, air pollution is associated with a higher risk of premature mortality and considerably worsens asthma symptoms in both adults and children [6].
To protect human health, especially in densely populated and industrialized urban areas, it is crucial to understand, monitor, and mitigate the effects of air pollution. The situation in China, where air pollution is thought to be responsible for more than 1.6 million deaths per year, serves as an example of how urgent it is to address this issue [7]. In addition to its detrimental effects on health, air pollution has a significant financial cost. When public health costs and agricultural yield losses are taken into consideration, researchers at the Chinese University of Hong Kong estimate that it causes an annual economic loss of 267 billion yuan (roughly US$38 billion) In Figure 1, we realize that China's PM_{10} emission trends from 2013 to 2020 demonstrate a notable decrease, with overall emissions falling by over 65% during that time. During Phase I (2013–2017), when strict air pollution control regulations were implemented, this decrease was most noticeable. The overall improvement was driven by significant declines in both the residential and industrial sectors, which were the main contributors in 2013. Emissions from all sectors had decreased by 2020, with the power sector making up a tiny portion of the total. These patterns demonstrate how well national laws work to reduce particulate matter, especially in cities and industrial regions.

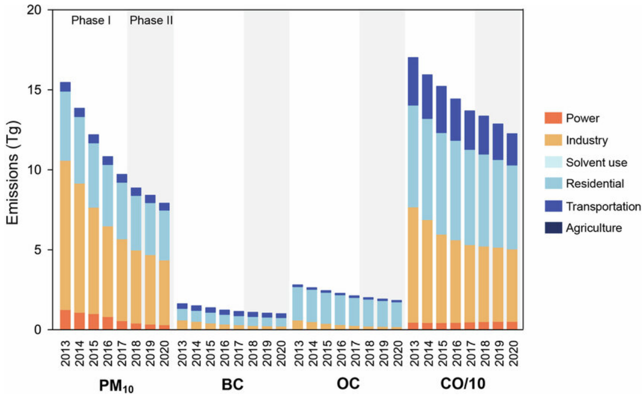

Fig. 1: Sectoral Trends in Anthropogenic Emissions in China (2013–2020): Supplementary Data for PM_{10}, Black Carbon (BC), Organic Carbon (OC), and Carbon Monoxide (CO) Emission Patterns [8]

In response to this crisis, China enacted the Ambient Air Quality Standard in 2012 and has since established a nationwide Air Reporting System, comprising 945 monitoring stations spread across 190 cities. These measures demonstrate the urgent need for stringent air quality regulations and effective environmental policies to mitigate the detrimental effects of air pollution on human health and economic growth.
In this study, we present a hybrid modeling framework that combines an Artificial Neural Network with a metaheuristic optimiza-

tion method, the GA, to address the complex problem of PM_{10} concentration prediction in Taiwan. This hybridization is motivated by the capacity of ANNs to model nonlinear relationships found in environmental data and the global search capabilities of GA, which effectively optimize the network's parameters and avoid convergence to local minima. The GA-ANN model was systematically developed and trained using a wide range of air quality and meteorological factors relevant to PM_{10} formation and dispersion. Extensive experiments were conducted to evaluate the performance of the proposed method, and the results indicate that the GA-optimized ANN significantly improves prediction accuracy in comparison to conventional training methods. This illustrates how complex hybrid intelligent systems can handle environmental modeling tasks.

This paper is formatted as follows: Section 2 provides a comprehensive review of the background and pertinent literature on metaheuristics (MHs), Artificial Neural Networks (ANNs), and hybrid ANN models. Section 3 gives a summary of metaheuristic algorithms, with a focus on Genetic Algorithms (GAs). Section 4 discusses the backpropagation (BP) learning process, the proposed hybrid methodology, and the fundamentals of ANNs. The dataset used in this study is described in Section 5, and the evaluation metrics are described in Section 6. A thorough explanation of the experimental setup and findings is given in Section 7. Finally, we provide the conclusion and future work.

2. BACKGROUND AND RELATED WORK

Air pollution is widely recognized as a major contributor to poor health outcomes and premature mortality worldwide. According to estimates from the World Health Organization (WHO), ambient air pollution alone results in approximately 4.2 million preventable deaths globally annually [9]. Several studies have found strong associations between short-term exposure to contaminated air and increased rates of cardiovascular, respiratory, and cerebrovascular death [10–12]. In response to these alarming statistics, considerable efforts have been made to monitor weather and atmospheric pollutant levels to ensure that air quality remains within safe limits [13]. Pollutants that are regularly measured include particulate matter (PM_{10}), carbon monoxide (CO), sulphur dioxide (SO_2), and nitrogen oxides (NO, NO_2, NO_x) [14]. Furthermore, meteorological variables such as temperature, wind speed, and relative humidity are routinely measured due to their significant influence on the concentration and dispersion of pollutants in the atmosphere.

Particulate matter with a diameter of 10 micrometres or less, or PM_{10}, originates from a range of natural and man-made sources, including waste burning, industrial operations, construction, vehicle emissions, and wildfires. Because of its detrimental effects on health and respiratory penetration, PM_{10} is one of the most significant indicators used to evaluate the health impacts of air pollution [15]. Accurately forecasting PM_{10} levels can support proactive public health efforts and provide valuable data on regional trends in air quality.

Over time, a variety of mathematical and computational models have been developed to predict air pollution levels [16]. Due to their ability to capture complex, non-linear relationships between pollutant concentrations and input features, Artificial Neural Networks (ANNs) have garnered attention among these researchers [17, 18]. Despite their potential, ANNs are still challenging to train efficiently, particularly when dealing with nonlinear and high-dimensional problems. Backpropagation (BP) and other traditional gradient descent-based training techniques are sensitive to initial weight configurations and often converge to local optima.

To overcome these limitations, researchers have investigated metaheuristic (MH) search algorithms as a different training strategy. MHs offer powerful global search capabilities and have proven effective in resolving complex optimization problems. In the context of air pollution modeling, MHs have been utilized to develop empirical models that forecast pollutants, such as CO_2 [16], using exponential regressions. Moreover, MHs have demonstrated promising outcomes in ANN training by improving convergence behavior and avoiding suboptimal solutions [19].

This combination, sometimes known as the hybrid approach, combines the predictive power of ANNs with the optimization capabilities of MHs. By hybridizing ANNs with MHs, researchers can overcome local optima, enhance training efficacy, and improve the generalization performance of models [20]. This hybrid methodology has been successfully applied in the field of air quality forecasting on multiple occasions. For instance, [21] hybridised a Recurrent Fuzzy Neural Network (RFNN) with the Grey Wolf Optimiser (GWO) to predict PM_{10}, $PM_{2.5}$, and ozone (O_3) concentrations. The proposed GWO-RFNN model outperformed the standalone RFNN in terms of predictive accuracy and generalization ability.

Moreover, hybrid approaches have also outperformed in other domains. In [22], a Multi-Layer Perceptron (MLP) was trained using a range of metaheuristic algorithms and compared with a traditional backpropagation-based MLP. The results demonstrated that for high-dimensional and complex datasets, hybrid models consistently outperformed conventional BP-MLP models, particularly in terms of prediction accuracy and training stability.

These findings together demonstrate the potential of hybrid intelligent systems, especially those that integrate ANNs and MHs, for accurate and dependable modeling of air pollution phenomena. Building on this growing body of work, this study optimizes an ANN for PM_{10} concentration prediction in Taiwan using a GA, demonstrating the ongoing relevance and efficacy of hybrid methods in environmental modeling.

3. METAHEURISTIC SEARCH ALGORITHMS

MHs are robust and effective due to their ability to *explore* and *exploit* the solution space [23–25]. Exploration is the process of globally searching for new candidate solutions within the solution space. Exploitation is the process of utilizing known candidate solutions to search for a better one locally [26]. MHs achieve these global and local search methods by modeling natural phenomena such as evolution, biology, physics, and swarm behavior to optimize for a given problem. MHs are a broad category of search algorithms, the hierarchy of which is depicted in Figure 2. These search algorithms were utilized to handle many real-world problems [27–29].

While MHs do not guarantee optimal solutions, they can determine good approximations to complex problems within a reasonable amount of time. This is particularly useful in real-world applications, where it is often the case that the solution is complex and poses difficulties for other methodologies.

3.1 Genetic Algorithms

The GA is an evolution-based algorithm that draws inspiration from Charles Darwin's theory of evolution. GAs optimize for a solution the same way a species evolves to survive its environment better. A species that is better suited for its environment is more likely to survive, mate, and pass on its genetics to its offspring. In the GA, the more fit an individual is, the better it solves a given problem, and

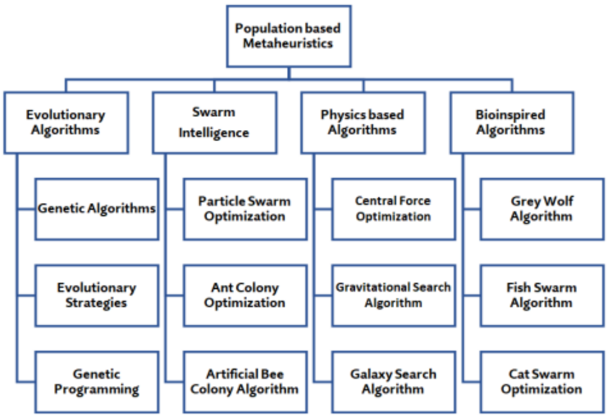

Fig. 2: Population-based Metaheuristic Search Algorithms [30]

the more likely it is to survive to reproduce offspring that inherit some of its traits.

GAs consist of a *population* of *individuals*. Each individual represents a candidate solution to the problem, represented by a *chromosome*. The chromosome serves as the base for the various GA operations to be applied, through a process known as *Reproduction*. A goal of the GA is to maintain variety within the population, to avoid getting stuck in local optima and early convergence. The operations in the GA aim to strike a balance between exploration and exploitation, while maintaining diversity within the population. The first operation that takes place is determined with the *selection* operation. This determines which individuals will be the parents who produce the next generation. Several selection methods have been utilized to select the parents [19]. These methods employ a combination of the parents' fitness and randomness to ensure variety within the next generation.

Just like how two biological mates produce offspring that inherit their genetics, selected parents in the GA populate the next generation with offspring that are a combination of their chromosomes. The process of combining these chromosomes is known as *crossover*. There are several methods of crossover [19]. These methods primarily involve dividing two parent chromosomes at one or more selected points and then swapping corresponding segments to generate new chromosomes. Individuals of the new population may then be subject to *mutation*. Mutation occurs at a predefined rate, where some form of alteration is made to that individual's chromosome. This once again increases variety in the GA, avoiding early convergence. The combination of crossover and mutation rates in the GA helps create a balance between exploration and exploitation, thereby avoiding being trapped in local optima.

Through these operations, the GA aims to model the process of evolution over a population of candidate solutions to a specific problem. This allows the GA to navigate complex solution spaces and produce a reasonable solution (see Algorithm 1).

Algorithm 1 Genetic Algorithm

Input: Population size n, mutation rate mr, total generations G
Initialize: population of n individuals randomly
Evaluate: Determine the fitness of each individual in the initial population
for generation g to G **do**
    Selection: Select parents of the next generation
    Crossover: Create offspring with parent chromosomes
    Update population with the new offspring
    for individual in population **do**
        if $mr > r \sim U(0,1)$ **then**
            Mutate: Alter chromosome
        end if
    end for
    Evaluate: Determine the fitness of each individual
end for
Return: The most fit individual

4. METHODOLOGY

Using evolutionary algorithms, such as Genetic Algorithms (GA) [31, 32], Differential Evolution (DE) [33], Genetic Programming (GP) [29,34–36], or Evolution Strategies (ES) [37], to optimize the network's connection weights is known as evolutionary weight optimization of ANNs [38]. Evolutionary approaches can efficiently explore complex, multi-modal search spaces without the need for derivatives, in contrast to conventional gradient-based training techniques, such as backpropagation. To reduce prediction error, these algorithms iteratively evolve a population of weight solutions by mimicking natural selection processes, such as reproduction, mutation, and selection. This method enhances the performance of ANNs, particularly when working with noisy, non-differentiable, or highly nonlinear data, which are challenging for standard optimization techniques to handle. The following describes the algorithms used in this paper for predicting air pollutants.

In the following section, we shall describe the main architecture of a feedforward ANN.

4.1 Artificial Neural Networks

ANNs are made up of fundamental processing units called *neurons*, just like the brain. Figure 3 shows an illustration of a neuron. Each neuron has a *weight* (w) and a *bias* (b) that affect the input signals' threshold and strength, respectively. Additionally, each neuron has an *activation function* that adds non-linearity and bounds the output values to control the neuron's output y [19]. ANNs can model intricate relationships in the data thanks to this mechanism.

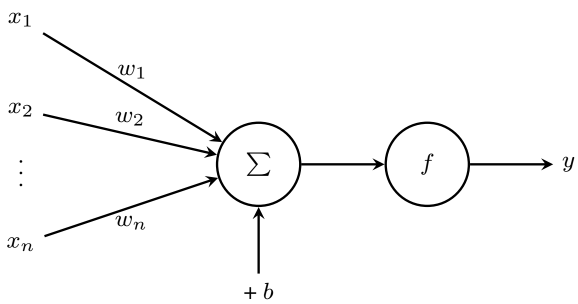

Fig. 3: A perceptron model showing weighted inputs, bias b, summation, and activation function f.

The output of a perceptron can be mathematically expressed as:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \quad (1)$$

Where:

—y: Output of the perceptron.
—x_i: The i^{th} input feature, where $i = 1, 2, \ldots, n$.
—w_i: The weight associated with the i^{th} input.
—b: The bias term.
—$f(\cdot)$: Activation function (e.g., step function, sigmoid, $tanh$).
—n: The total number of input features.

The underlying architecture of an ANN is a key component of its information processing. Typically, an input layer, one or more hidden layers, and an output layer make up the structured layers in which neurons are arranged. There are no feedback loops in a feed-forward ANN; information moves unidirectionally from the input layer to the output layer via the hidden layers [39]. While each neuron in a layer is completely connected to every other neuron in the layer below it, it is not connected to any neurons in its layer.
Figure **??** shows an example of a fully connected feed-forward ANN [40]. It is crucial to remember that there is no hard-and-fast rule for determining the best ANN architecture for a given issue; instead, empirical tuning and experimentation are often necessary.

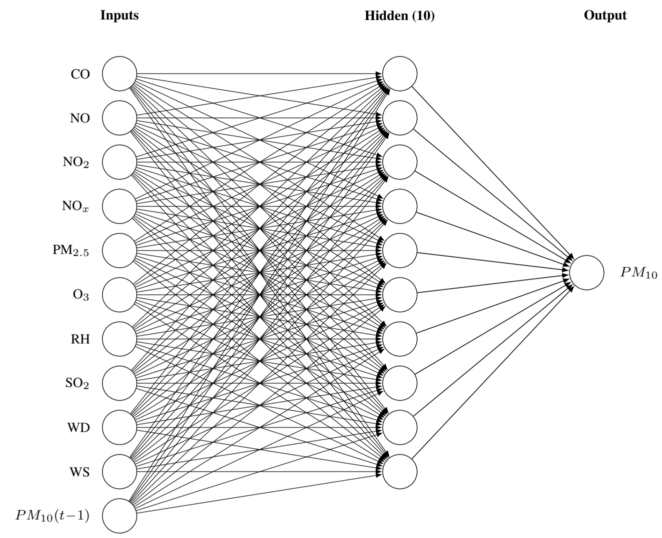

Fig. 4: Fully Connected MLP with 11 input features, one hidden layer of 10 neurons, and a single output predicting PM_{10}.

To achieve the intended result, the weights and biases of an ANN need to be adjusted after it has been built. Typically, a training procedure that aims to minimize the error between the expected and actual outputs achieves this. The Backpropagation (BP) algorithm, a gradient-based optimization technique, is the most commonly used training method (see Algorithm 2).
To propagate error backward through the network, BP uses the chain rule to calculate the gradient of a given *loss function* concerning the weights [39]. The weights are then iteratively updated using these gradients in a way that minimizes the loss. A predefined parameter known as the *learning rate* controls the size of these updates and determines how quickly or slowly the network adapts during training. An *epoch* is a complete run through the entire training dataset, and training usually continues for a predetermined number of epochs until convergence or a stopping criterion is satisfied.

Algorithm 2 Backpropagation Learning Algorithm

Input: Training data $(x^{(i)}, d^{(i)})$, learning rate η, number of epochs E
Initialize: Weights and biases to small random values
for epoch $= 1$ to E **do**
    for each training sample (x, d) **do**
        Forward Pass:
        Compute activations of all hidden and output neurons
        Compute Output Error:
        $\delta_o = (d - y) \cdot f'(net_o)$ ▷ Error at output layer
        Backward Pass:
        Compute error for hidden neurons:
        $\delta_h = f'(net_h) \cdot \sum \delta_o \cdot w_{ho}$
        Update Weights:
        $w_{ij} \leftarrow w_{ij} + \eta \cdot \delta_j \cdot x_i$
        $b_j \leftarrow b_j + \eta \cdot \delta_j$
    end for
end for
Return: Final weights and biases

4.2 The Backpropagation Algorithm's Limitations:

Notwithstanding its widespread use, the BP has several drawbacks that may affect the robustness and performance of ANNs:

—**Sensitivity to Initial Weights:** Slow convergence or entrapment in local minima can result from poorly initialized weights.
—**Gradient Exploding/Vanishing:** Gradients in ANN can get incredibly small or large, which makes training challenging.
—**Reliance on Variable Activation Functions:** The selection of activation functions is restricted by BP's requirement that they be differentiable.
—**Overfitting Risk:** When training data is scarce, BP can result in overfitting if it is not properly regularized.
—**Adjusting** the Learning Rate: Selecting the correct learning rate is essential; too low leads to slow training, while too high could result in divergence.
—**Caught in Saddle Points or Local Minima:** Depending on the error surface, the algorithm might converge to less-than-ideal solutions.

4.3 Evolutionary MLP-ANN

For an MH, a hybridized ANN replaces conventional ANN training methods. In this process, the ANN is somehow represented within the MH. The weights and biases of a particular ANN are typically arranged to form its representation. By doing this, the MH can manipulate them and start to maximize their values. Determining the fitness function is another essential step in the hybridized approach. This statistical measure is typically employed to assess the performance of an ANN. The MH is striving for a more effective ANN by optimizing it for this purpose.
Figure 5 illustrates the structure of a typical feedforward ANN consisting of three main components: an input layer, a hidden layer, and an output layer. This architecture demonstrates how input signals are transformed through weighted connections and activation functions to produce output values. The complete set of weights and biases of the MLP model is encoded as a single linear vector, known as an individual or chromosome, when training such a network using GAs. Throughout the evolutionary optimization process, this vector serves as the candidate solution.

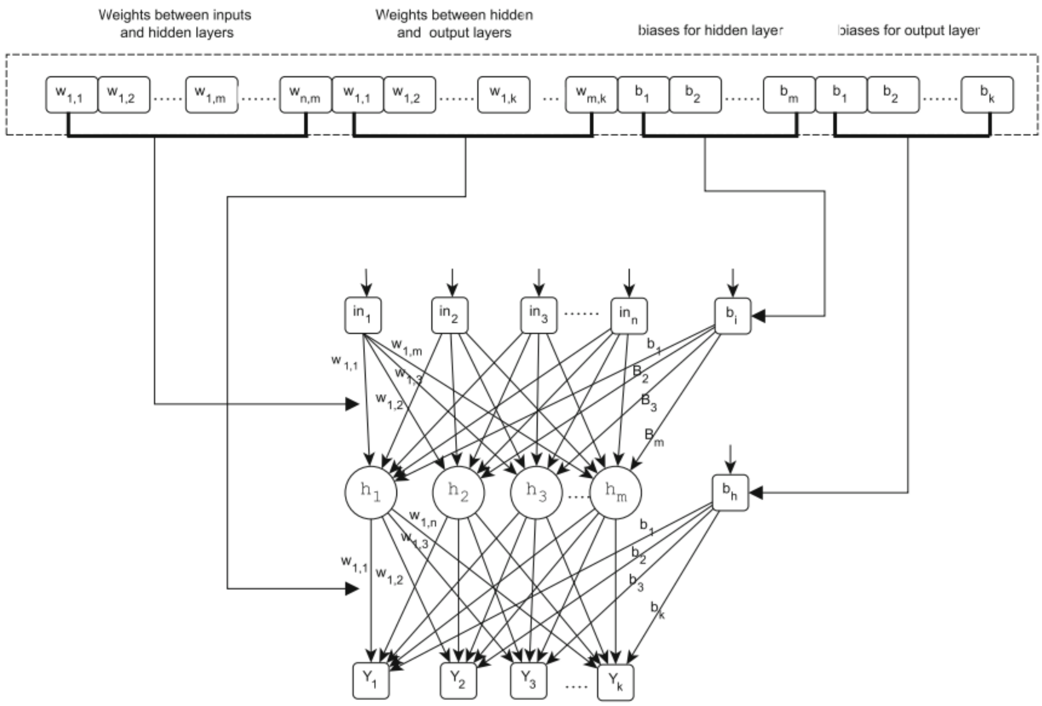

Fig. 5: Mapping a GA individual to a MLP neural network [41]

4.4 MLP Weight Optimization Network

Assuming a Multilayer Perceptron (MLP) network composed of n input neurons, h hidden neurons, and o output neurons, the total number of parameters—including both weights and biases—can be calculated as follows. The number of weights connecting the input layer to the hidden layer is $n \times h$, and the hidden layer includes h bias terms. Likewise, the number of weights from the hidden layer to the output layer is $h \times o$, with an additional o bias term in the output layer. Thus, the total number of parameters in the MLP is:

$$\text{Total Parameters} = (n \times h) + h + (h \times o) + o \tag{2}$$

—**Input-to-Hidden Weights:** $W^{(1)} \in \mathbb{R}^{h \times n}$

—**Hidden Biases:** $\mathbf{b}^{(1)} \in \mathbb{R}^{h}$

—**Hidden-to-Output Weights:** $W^{(2)} \in \mathbb{R}^{o \times h}$

—**Output Biases:** $\mathbf{b}^{(2)} \in \mathbb{R}^{o}$

In the GA, each ANN has a chromosome representation, which consists of its associated weights and biases; an example is shown in Table 1. This chromosome representation serves as the basis for the various GA operations.

Table 1. : ANN Chromosome Representation.

w_1	w_2	$\cdots$	w_n	b_1	b_2	$\cdots$	b_n

The *k-point crossover* method was used. This means that k points along two parent chromosomes were selected, and the corresponding segments were swapped to generate new offspring.

If a chromosome is subject to mutation, weights are randomly selected at the defined mutation rate, to which their values are randomly updated, as follows:

$$w = w + r \sim N(0, 1) \tag{3}$$

The GA individual $\mathbf{x}_{\text{GA}}$ during evaluation is the

(1) Reformed back into $W^{(1)}, \mathbf{b}^{(1)}, W^{(2)}, \mathbf{b}^{(2)}$,
(2) Used in a typical forward pass:

$$\begin{aligned} \mathbf{h} &= \sigma\left(W^{(1)} \cdot \mathbf{in} + \mathbf{b}^{(1)}\right), \\ \mathbf{y} &= \sigma\left(W^{(2)} \cdot \mathbf{h} + \mathbf{b}^{(2)}\right) \end{aligned} \tag{4}$$

(3) The fitness score is calculated by comparing the ground truth with the predicted output $\mathbf{y}$.

With a basic MLP configuration consisting of $n = 4$ input neurons, $h = 3$ hidden neurons, and $o = 2$ output neurons, the chromosome used by the GA represents all trainable parameters of the network, including weights and biases. The total number of parameters is calculated as follows:

$$\begin{aligned} \text{No of parameters} &= (4 \times 3) + 3 + (3 \times 2) + 2 \\ &= 12 + 3 + 6 + 2 \\ &= 23 \end{aligned} \tag{5}$$

This encoding allows the GA to optimize the entire set of MLP parameters simultaneously. Unlike traditional gradient-based methods, the GA navigates the search space through evolutionary operations, making it well-suited for optimizing neural networks in complex or non-differentiable environments.

5. DATA SET

As part of its national *Air Reporting System* and *Ambient Air Quality Standard*, China set up 945 monitoring sites in 190 cities in 2012. Meteorological factors such as wind direction, temperature, relative humidity, wind speed, and solar radiation intensity have a significant impact on the photochemical processes that govern the production and degradation of ozone [7]. Similarly, over the past three decades, Taiwan's air pollution patterns have undergone significant changes, shifting from point-source pollution in the 1980s to predominantly urban pollution in the 2000s. The map of monitoring stations spanning Taiwan is shown in Figure 6.

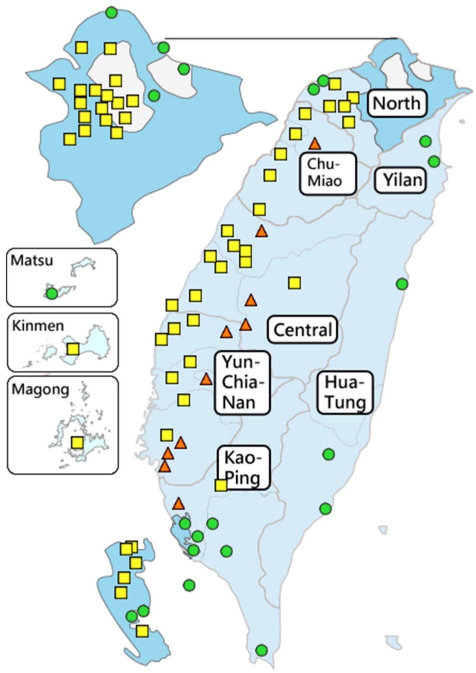

Fig. 6: Monitoring stations at TAQMN, Taiwan. Published March 12, 2020.

While particulate matter concentrations exhibit considerable variability, with an average of about 19 μg/m^3 but a maximum of 101.7 μg/m^3, the mean temperature is roughly 22.8°C, with a range of 9.2°C to 32.3°C. The average ozone level is 26.3 μg/m^3, with a wide range from 2.8 to 59.2. Significant variations are also observed in relative humidity and wind speed, reflecting the diverse atmospheric conditions recorded during the monitoring period. For a more thorough examination of air quality trends and their possible health effects, these statistics offer a fundamental understanding of the environmental conditions and pollutant levels.
Temperature (T), carbon monoxide (CO), nitric oxide (NO), nitrogen dioxide (NO_2), nitrogen oxides (NO_x), delicate particulate matter ($PM_{2.5}$), ozone (O_3), relative humidity (RH), sulphur dioxide (SO_2), wind direction (WD), wind speed (WS), and lagged PM_{10} [PM_{10}(t–1)] are included in the model's input features (see Figure 7). These features were selected due to their availability in the dataset and their demonstrated impact on particulate matter dynamics. The main summary statistics for the various meteorological and air quality variables measured in the dataset are presented in Table 2.

6. EVALUATION CRITERION

Different performance metrics were utilized to assess the quality of the developed model. The mean squared error (MSE), the root mean squared error (RMSE), the Pearson correlation coefficient (r), and the variance accounted for (VAF) were measured. These metrics are presented mathematically in the following equations. here, y is the true value from the recorded data, and $\hat{y}$ is the value predicted by the model.

—**Mean Squared Error (MSE):**

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6}$$

—**Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{7}$$

—**Pearson Correlation Coefficient (r):**

$$r = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{8}$$

—**Variance Accounted For (VAF):**

$$\text{VAF} = \left(1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)}\right) \times 100 \tag{9}$$

7. EXPERIMENTAL SETUP AND RESULTS

In this study, we used a feedforward neural network optimized by a Genetic Algorithm to create a predictive model for PM_{10} concentration. The goal was to utilize an evolutionary search strategy to optimize the network's weights and biases, thereby increasing its predictive accuracy. By using the previous PM_{10} value as an input, the model was specifically built to leverage temporal dependencies and capture the nonlinear relationships between air quality variables and PM_{10} levels.
Ten neurones with a nonlinear activation function (such as `tanh`) make up the neural network's single hidden layer. The predicted PM_{10} concentration is represented by a single continuous value generated by the output layer. To assess generalization performance, the dataset was divided into 70% for training and 30% for testing.
We used a GA setup with the following parameters to optimize the neural network: a mutation rate of 0.001, a population size of 500 individuals, and a maximum of 100 generations. The mean squared error between the actual and predicted PM_{10} values served as the basis for the fitness function. To reduce prediction error over generations, the GA iteratively developed candidate solutions using crossover, mutation, and selection operations. By avoiding local minima and utilizing GA's global search capabilities, we effectively trained the neural network and enhanced the model's capacity to identify intricate relationships in the air quality data.
Figure 8 describes the convergence behavior of the Genetic Algorithm used for model optimization. It demonstrates a sharp decline in MSE in the first few generations, followed by stabilization, indicating that the algorithm effectively identified a near-optimal solution early on and refined it throughout subsequent generations.

The performance metrics compiled in Table 3 quantitatively support this visual evaluation. The training data produced a Mean

Table 2. : Summary Statistics of Dataset Variables

Statistic	T	CO	NO	NO_2	NO_x	$PM_{2.5}$	O_3	RH	SO_2	WD	WS	PM_{10}
Mean	22.836	0.388	4.537	11.242	15.782	18.991	26.296	80.041	2.624	190.358	1.702	37.992
Std Dev	5.262	0.126	2.679	3.526	5.245	11.189	10.453	8.443	1.583	44.456	0.676	21.232
Min	9.225	0.038	0.279	0.824	1.103	0.917	2.763	46.292	0.263	52.464	0.780	3.833
Max	32.250	0.939	29.380	27.203	43.938	101.667	59.208	97.083	22.750	302.298	8.610	377.500

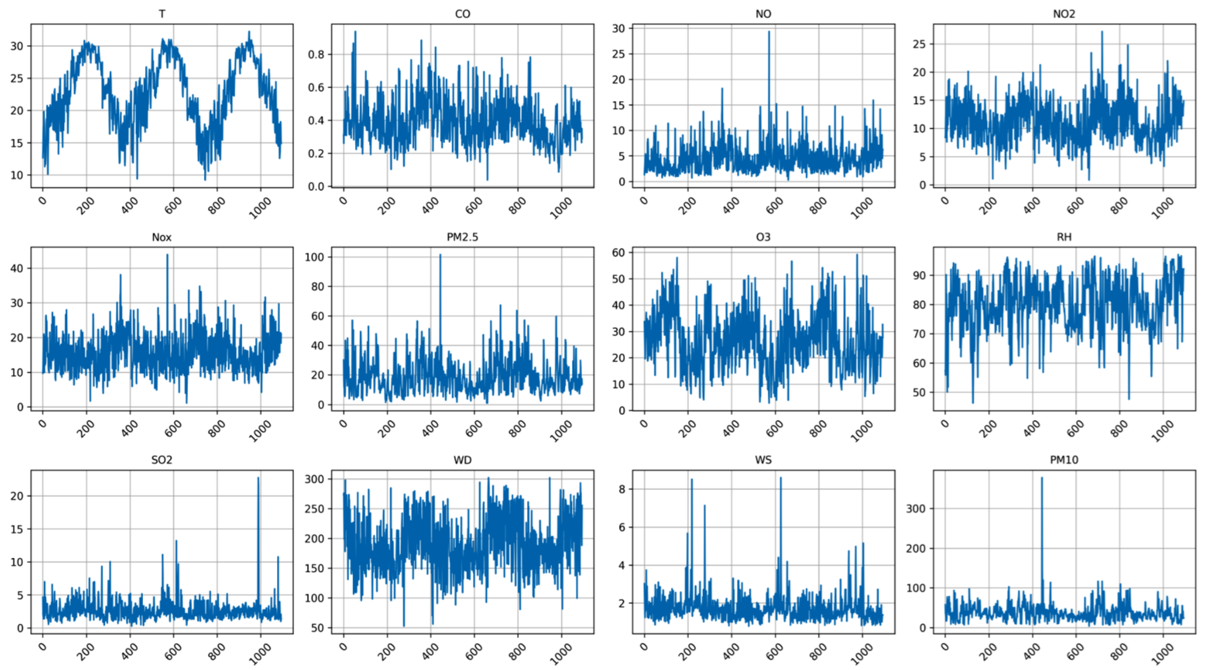

Fig. 7: Time series plots of all variables in the dataset utilized to predict PM_{10}

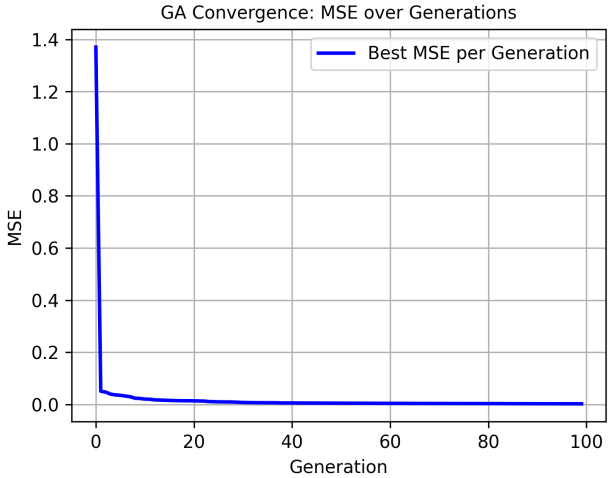

Fig. 8: Best convergence curve of GAs showing Best MSE over generations.

Squared Error (MSE) of 51.37 and a Root Mean Squared Error of 7.17, along with a Pearson correlation coefficient of 0.947 and a Variance Accounted For of 0.892. A robust and predictive model is confirmed by the testing data, which also show high accuracy with an MSE of 60.25, RMSE of 7.76, Pearson r of 0.922, and VAF of 0.849.

Table 3. : Performance Metrics for Training and Testing

Metric	**MSE**	**RMSE**	**Pearson r**	**VAF**
Training	51.36901	7.16722	0.94728	0.89180
Testing	60.24747	7.76193	0.92168	0.84850

The comparison of actual and predicted PM_{10} concentrations for both training and testing datasets is shown in Figure 9. Strong fit is indicated by the predicted values in the top subplot closely matching the real training data, and the bottom subplot suggests good generalization ability, with similarly close alignment on the testing dataset.

8. CONCLUSION AND FUTURE WORK

Using a set of twelve atmospheric and pollution-related input variables, an MLP NN optimized by a GA was successfully used in this study to predict PM_{10} concentrations. The model's high correlation coefficients and low prediction errors on both training and

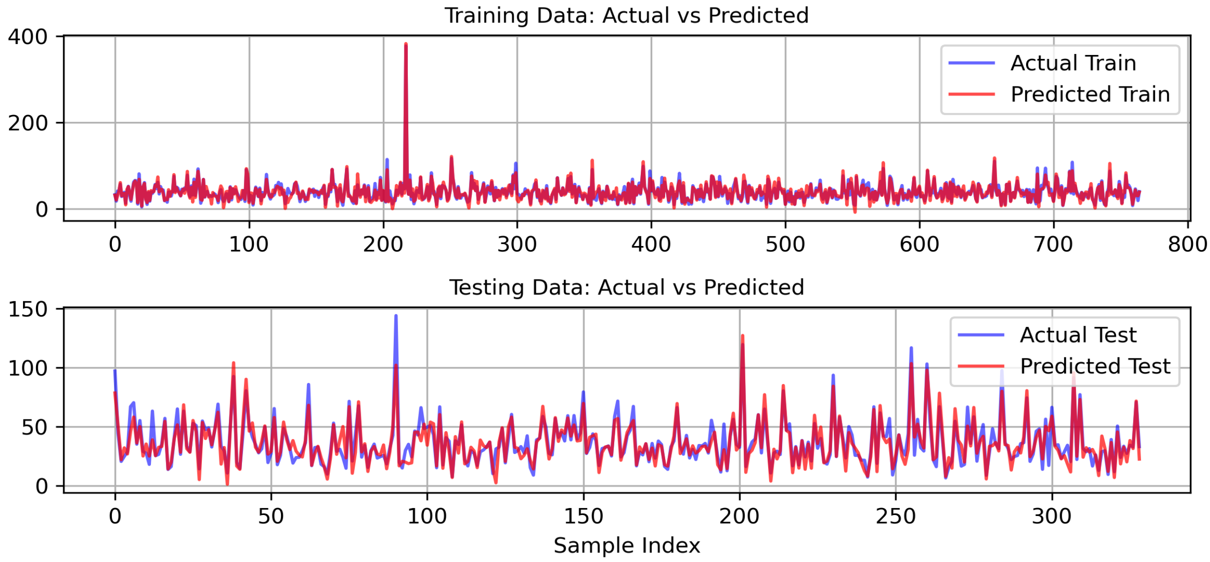

Fig. 9: Actual vs. Predicted PM_{10} using evolutionary MLP-ANN weights: top — training case, bottom — testing case.

testing datasets confirmed that evolutionary optimization is a suitable method for improving air quality prediction models.

Future research could explore additional methods to enhance the model's applicability and accuracy further. These include comparing GA with other metaheuristic optimization algorithms, such as Particle Swarm Optimization or Differential Evolution, utilizing more complex network architectures like deep or recurrent neural networks, and incorporating additional meteorological or temporal features. Furthermore, the model's usefulness for environmental management and public health planning would be increased by expanding it to multi-step forecasting or integrating it with a real-time air quality monitoring system.

Acknowledgment

Prof. Sheta would like to sincerely thank the Department of Computer Science at Southern Connecticut State University for supporting this research during Summer 2025.

9. REFERENCES

[1] M. R. Miller and D. E. Newby, "Air pollution and cardiovascular disease: Car sick," *Cardiovascular Research*, vol. 116, no. 2, pp. 279–294, Feb. 2020.

[2] P. Petropoulou, I. Artopoulou, I. Kalemikerakis, and O. Govina, "Environment and public health: Air pollution and chronic diseases," *Environmental Sciences Proceedings*, vol. 26, no. 1, 2023.

[3] R. Roy and A. D'Angiulli, "Air pollution and neurological diseases: current state highlights," *Frontiers in Neuroscience*, vol. 18, Mar. 2024.

[4] U. Sagheer, S. Al-Kindi, S. Abohashem *et al.*, "Environmental pollution and cardiovascular disease: Part 1 of 2: Air pollution," *JACC: Advances*, vol. 3, no. 2, Feb. 2024.

[5] Y.-J. Zhang, T.-T. Xu, Y.-L. Luan, H.-M. Shen, and Y. Guo, "Navigating the complexity of exposure to multiclass organic pollutants in respirable size-resolved particles and implications for oxidative potential," *Environment International*, vol. 202, p. 109646, 2025.

[6] M. Krishan, S. Jha, J. Das, A. Singh, M. K. Goyal, and C. Sekar, "Air quality modelling using long short-term memory (lstm) over nct-delhi," *Air Quality, Atmosphere Health*, vol. 12, no. 8, pp. 899–908, 2019.

[7] R. Rohde and R. Muller, "Air pollution in china: Mapping of concentrations and sources," *PLoS ONE*, vol. 10, no. 8, pp. 1–14, 2015.

[8] G. Geng, Y. Liu, Y. Liu, S. Liu, J. Cheng, L. Yan, N. Wu, H. Hu, D. Tong, B. Zheng, Z. Yin, K. He, and Q. Zhang, "Efficacy of china's clean air actions to tackle pm2.5 pollution between 2013 and 2020," *Nature Geoscience*, vol. 17, pp. 987–994, 09 2024.

[9] World Health Organization, "Ambient (outdoor) air quality and health," Fact sheet, World Health Organization, Oct. 2024, last updated 24 October 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health

[10] A. Abdelfatah, S. A. Mokhtar, A. Sheta, and B. Solaiman, "Forecast global carbon dioxide emission using swarm intelligence," *International Journal of Computer Applications*, vol. 77, no. 12, pp. 1–5, 2013.

[11] P. Orellano, J. Reynoso, N. Quaranta, A. Bardach, and A. Ciapponi, "Short-term exposure to particulate matter (PM10 and PM2.5), nitrogen dioxide (NO2), and ozone (O3) and all-cause and cause-specific mortality: Systematic review and meta-analysis," *Environment International*, vol. 142, p. 105876, 2020.

[12] M. Alkasassbeh, A. F. Sheta, H. Faris, and H. Turabieh, "Prediction of PM10 and TSP air pollution parameters using artificial neural network autoregressive, external input models: A case study in salt, jordan," *Middle East Journal of Scientific Research*, vol. 14, no. 7, pp. 999–1009, 2013.

[13] H. Turabieh, A. F. Sheta, M. Braik, and E. Kovač-Andrić, "A layered recurrent neural network for imputing air pollutants missing data and prediction of NO_2, O_3, PM_{10}, and $PM_{2.5}$," in *Forecasting in Mathematics*, A. A. Jaoude, Ed. Rijeka: IntechOpen, 2020, ch. 3.

[14] A. F. Sheta and H. Faris, "Influence of nitrogen-di-oxide, temperature and relative humidity on surface ozone modeling process using multigene symbolic regression genetic programming," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 6, 2015.

[15] World Health Organization, "Types of pollutants," Web page, WHO– Air Quality and Health (Health Impacts section), 2025, accessed 25 July 2025. [Online]. Available: https://www.who.int/teams/environment-climate-change-and-health/air-quality-and-health/health-impacts/types-of-pollutants

[16] A. Abdelfatah, S. Mokhtar, A. Sheta, and B. Solaiman, "Forecast global carbon dioxide emission using swarm intelligence," *International Journal of Computer Applications*, vol. 77, pp. 1–5, 09 2013.

[17] R. Selvaraj, R. Samuel, G. R., E. Krishnamoorthy, and S. Jeyakumar, "A neural network model for short-term prediction of surface ozone at tropical city," *International Journal of Engineering Science and Technology*, vol. 2, oct 2010.

[18] A. Afzali, M. Rashid, B. Sabariah, and M. Ramli, "Pm10 pollution: Its prediction and meteorological influence in Pasir-Gudang, johor," *IOP Conference Series: Earth and Environmental Science*, vol. 18, no. 1, p. 012100, feb 2014.

[19] A. Azzini, "A new genetic approach for neural network design and optimization," Ph.D. Thesis, Università degli Studi di Milano, Milano, Italy, 2007, defended March 8, 2007; academic year 2005/06.

[20] H. Faris, I. Aljarah, and S. Mirjalili, "Training feedforward neural networks using multi-verse optimizer for binary classification problems," *Applied Intelligence*, vol. 45, no. 2, pp. 322–332, 2016.

[21] M. Braik, A. Sheta, and H. Al-Hiary, "Hybrid neural network models for forecasting ozone and particulate matter concentrations in the republic of china," *Air Quality, Atmosphere & Health*, vol. 13, pp. 839–851, 2020.

[22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Let a biogeography-based optimizer train your multi-layer perceptron," *Information Sciences*, vol. 269, pp. 188–209, 2014.

[23] T. Thaher, M. Awad, M. Aldasht, W. A. Deabes, A. F. Sheta, and H. K. Chantar, "An enhanced evolutionary-based feature selection approach using grey wolf optimizer for the classification of high-dimensional biological data," *Journal of Universal Computer Science*, vol. 28, no. 5, pp. 499–539, 2022.

[24] M. Braik, A. F. Sheta, and H. Al-Hiary, "A novel meta-heuristic search algorithm for solving optimization problems: Capuchin search algorithm," *Neural Computing and Applications*, vol. 33, no. 7, p. 2515–2547, 2021.

[25] M. S. Braik, A. F. Sheta, H. Turabieh, and H. Al-Hiary, "A novel lifetime scheme for enhancing the convergence performance of salp swarm algorithm," *Soft Computing*, vol. 25, no. 104, pp. 8459–8477, 2021.

[26] B. Al-Himyari, H. Al-khafaji, and N. Hussain, "Exploration-exploitation tradeoffs in metaheuristics: A review," *Asian Journal of Applied Sciences*, vol. 12, 01 2025.

[27] A. Sheta and A. Abdel-Raouf, "Stock market price forecasting using metaheuristic search algorithms: A comparative analysis," *International Journal of Computing*, vol. 23, no. 4, p. 702–708, 2024.

[28] A. Sheta, H. Faris, M. Braik, and S. Mirjalili, "Nature-inspired metaheuristics search algorithms for solving the economic load dispatch problem of power system: A comparison study," in *Applied Nature-Inspired Computing: Algorithms and Case Studies*, ser. Springer Tracts in Nature-Inspired Computing, N. Dey, A. S. Ashour, and S. Bhattacharyya, Eds. Springer, Singapore, 2019.

[29] A. F. Sheta, H. Faris, and E. Oznergiz, "Improving production quality of a hot rolling industrial process via genetic programming model," *International Journal of Computer Applications in Technology*, vol. 49, no. 3/4, p. 239–250, 2014.

[30] A. Sheta, A. Ali, A. Baareh, and S. Aljahdali, "Meta-heuristic search algorithms for solving the economic load dispatch problem," in *2022 3rd International Conference on Artificial Intelligence, Robotics and Control (AIRC)*, 2022, pp. 87–92.

[31] A. Sheta, M. S. Braik, and S. Aljahdali, "Genetic algorithms: A tool for image segmentation," in *2012 International Conference on Multimedia Computing and Systems*, 2012, pp. 84–90.

[32] A. Sheta, "Restoration of medical images using genetic algorithms," in *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. IEEE, 2017, pp. 1–6.

[33] A. F. Sheta, "Analogue filter design using differential evolution," *International Journal of Bio-Inspired Computation*, vol. 2, no. 3, pp. 233–241, 2010.

[34] A. F. Sheta and A. Al-Afeef, "A gp effort estimation model utilizing line of code and methodology for NASA software projects," in *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA 2010)*. Cairo, Egypt: IEEE, Nov. 2010, pp. 1003–1008.

[35] ——, "Software effort estimation for NASA projects using genetic programming," *Journal of Intelligent Computing*, vol. 1, no. 3, p. 146–156, 2010.

[36] Z. Al-Rahamneh, M. Reyalat, A. F. Sheta, S. Bani-Ahmad, and S. Al-Oqeili, "A new software reliability growth model: Genetic-programming-based approach," *Journal of Software Engineering and Applications*, vol. 4, no. 8, p. 476–481, 2011.

[37] A. F. Sheta and A. H. Abdel-Wahab, "Identification of nonlinear communication channel using evolutionary volterra time-series," in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, vol. 1. IEEE, 1999, pp. 425–431.

[38] D. Whitley, "Genetic algorithms and neural networks," in *Genetic Algorithms in Engineering and Computer Science*, J. Periaux and G. Winter, Eds. John Wiley & Sons, Jul. 1998.

[39] M. Sazli, "A brief review of feed-forward neural networks," *Communications Faculty of Science, University of Ankara*, vol. 50, pp. 11–17, 01 2006.

[40] A. Mert, N. Kjljç, E. Bilgili, and A. Akan, "Breast cancer detection with reduced feature set," *Computational and Mathematical Methods in Medicine*, vol. Article ID 265138, p. 11 pages, 05 2015.

[41] I. Aljarah, H. Faris, and S. Mirjalili, "Evolving neural networks using bird swarm algorithm for data classification and regression applications," *Cluster Computing*, vol. 22, no. 1, pp. 299–317, 2019.