Islamic University of Technology

Enhancing Efficiency of Affordable Sensors: An Advanced Neural Network Paradigm with Metaheuristic Optimization Algorithms

by

Tanzila Arafin 180021311

Mahabub Alam Mridul 180021321

Sazid Sadman 180021112

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering

Submitted on Summer Semester 2023

DECLARATION

We hereby solemnly declare that the project report "Enhancing Efficiency of Affordable Sensors: An Advanced Neural Network Paradigm with Metaheuristic Optimization Algorithms" presented here is the outcome of our original work completed at Islamic University of Technology under the supervision of Dr. Ashik Ahmed. A portion of the criteria for the BSc in Electrical and Electronic Engineering were met through the project work.

We certify that this project report is a product of our own work and has not been submitted to any institution or university for the purpose of receiving a different degree or diploma. The information sources used and referenced in this study have all been properly recognized.

We are aware that plagiarism and other forms of academic misconduct are categorically incompatible with the values and moral norms of academic and research endeavors. We hereby declare that no instances of plagiarism or unlawful use of intellectual property have occurred in this project report.

We fully accept responsibility for all of the information contained in this project report, including any mistakes or omissions that may have been made. We are aware that we are responsible for any consequences associated with using this project report.

Tanzila Arafin, 180021311 Tanzila 07/06/23

Mahabub Alam Mridul, 120021321 07/06/23

Sazid Sadman, 18002112 Sazid 07/06/23

Enhancing Efficiency of Affordable Sensors: An Advanced Neural Network Paradigm with Metaheuristic Optimization Algorithms

Tanzila Arafin 180021311

Mahabub Alam Mridul 180021321

Sazid Sadman 180021312

Has been approved on 29th May 2023

By

Dr. Ashik Ahmed

Professor

Islamic University of Technology

BoardBazar, Gazipur, Dhaka

Ahmed 7/6/23

Dr. Ashik Ahmed

Professor

Islamic University of Technology

ACKNOWLEDGEMENTS

"All praise and gratitude be to Allah, the most beneficent, the most merciful."

We would like to offer our sincere gratitude and appreciation to everyone who helped our project, "Enhancing Efficiency of Affordable Sensors: An Advanced Neural Network Paradigm with Metaheuristic Optimization Algorithms," come to a successful conclusion. Without the help, direction, and encouragement we got from countless people and organizations, this project would not have been feasible.

We want to start by expressing our sincere gratitude to our project manager, Dr. Ashik Ahmed, for his essential advice, knowledge, and ongoing support. Their astute criticism, helpful recommendations, and persistent dedication were important in determining the course and development of this work.

In addition, we want to thank our peers, family, and friends for their help and support. Their support, compassion, and inspiration gave me the courage and inspiration to endure throughout the project's various phases.

Last but not least, we want to express our gratitude to the writers and members of the research community whose writings we have used as references and sources for our project. Their opinions and comments have significantly improved this report's validity and content.

Once again, I want to thank each and every one of you for being so crucial to our journey and for your steadfast support.

ABSTRACT

This project report gives a thorough investigation into how combining ANN models with metaheuristic optimization methods might improve the performance of low-cost sensors. Although low-cost sensors have a wide range of uses, they frequently lack the accuracy and dependability of their more expensive counterparts. The strength of ANN models is combined with optimization approaches in a novel way to address this restriction.

In order to optimize the parameters and design of the neural network, the project focuses on creating an ANN framework that uses metaheuristic techniques like Particle Swarm Optimization (PSO), Osprey Optimization Algorithm(OOA), Driving Training Based Optimization(DTBO), Salp Swarm Algorithm(SSA),Harris Hawk Optimization (HHO. By significantly enhancing the accuracy, precision, and robustness of low-cost sensors, this combination intends to enhance their overall performance.

Keywords: Metaheuristic, Optimization, PSO, SSA, Artificial Neural Network,HHO,DTBO Hidden Layer, RMSE .

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1
Introduction

Low-cost sensors have gained significant attention in various fields due to their affordability and potential for widespread deployment. However, one key challenge associated with these sensors, particularly weather sensors, is their inherent inaccuracy. Inaccurate weather data can have far-reaching implications across multiple domains, affecting fields such as agriculture, transportation, energy management, and urban planning, among others.

Weather sensors play a crucial role in providing real-time meteorological information, enabling decision-making processes, and ensuring public safety. However, low-cost weather sensors often suffer from inaccuracies in measurements, resulting in unreliable data outputs[1]. These inaccuracies can lead to erroneous predictions, inefficient resource allocation, and compromised safety measures.

As a result, for low-cost sensors to work at their best in many domains, precision, and dependability are essential. To fully realize the potential benefits of these sensors and to promote well-informed decision-making processes, it is crucial to address their limitations.

There is an increasing need for cutting-edge methods that improve the effectiveness of low-cost sensors to overcome their errors. The goal of this project is to increase the precision and dependability of low-cost sensors by using Artificial Neural Networks (ANNs) that have been meta-heuristically optimized. We hope to improve many domains affected by erroneous sensor readings by merging these computational tools in order to produce data that is more accurate and reliable.

The project involves several key stages, including data collection from the low-cost sensors, preprocessing and feature extraction, model training using the ANN architecture, and optimization using metaheuristic algorithms. These steps were chosen
The choice of each step in the project can be driven by specific reasons and considerations.

Data collection from low-cost sensors:Low-cost sensors offer a cost-effective solution for data collection in various domains.By utilizing low-cost sensors, it becomes feasible to deploy a dense network of sensors, providing a higher spatial coverage and enabling more comprehensive data collection.Low-cost sensors are increasingly accessible, allowing for broader data collection in real-world scenarios.[8]

Preprocessing and feature extraction: Preprocessing helps to clean and preprocess the raw sensor data, removing noise and inconsistencies.
Feature extraction is crucial for identifying relevant patterns and relationships in the sensor data.
These steps enable the extraction of meaningful information from the collected data, enhancing the subsequent modeling and optimization stages.[11]

Model training using the ANN architecture: Artificial Neural Networks (ANNs) are known for their ability to capture complex patterns and relationships in data.
ANNs can learn from the sensor data and generalize their understanding to make accurate predictions or classifications.The use of ANNs allows for the development of a data-driven model that can exploit the richness of the collected sensor data.[9]

Optimization using metaheuristic algorithms: Metaheuristic algorithms are well-suited for optimizing complex systems and finding near-optimal solutions.
These algorithms can efficiently explore the high-dimensional parameter space of ANNs to find optimal configurations.By utilizing metaheuristic algorithms, it is possible to improve the performance and accuracy of the ANN models, ultimately enhancing the reliability of the low-cost sensors.[10]

The performance of the proposed approach is evaluated through extensive experimentation and comparative analysis with traditional methods

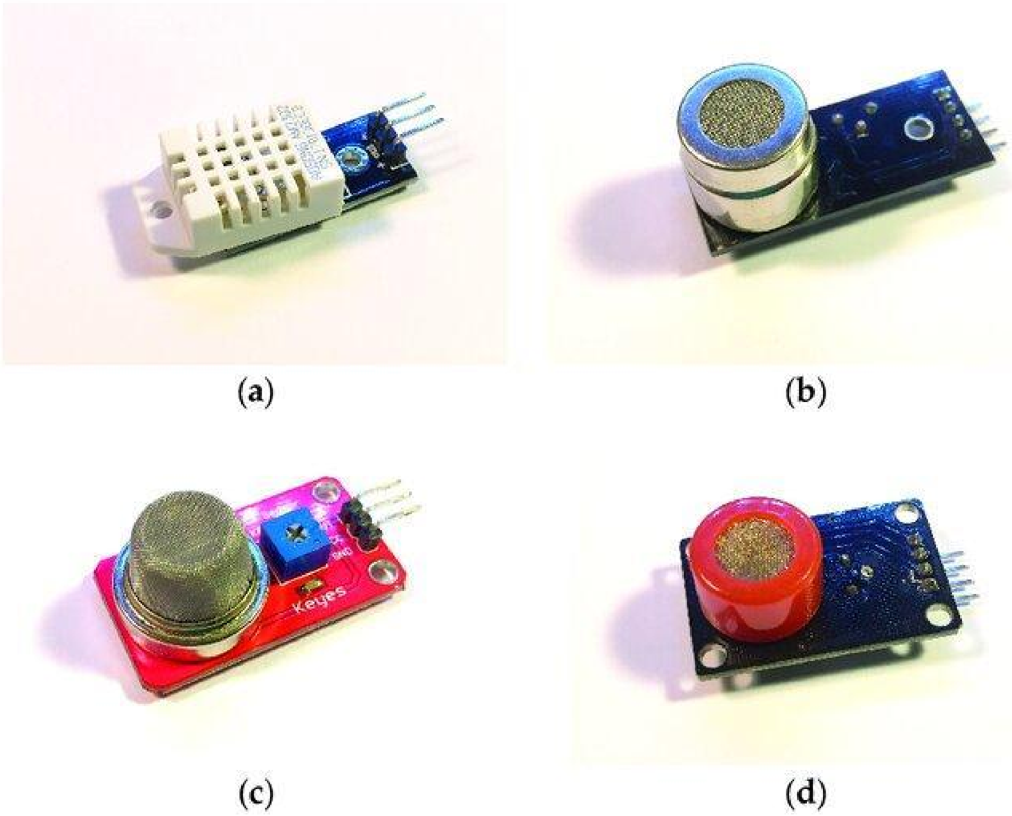

Figure 1.1 : Examples of Low-Cost Sensors - (a)the DHT22 sensor (b) MG-811 sensor(c) the MQ-2 sensor (d) the MQ-7 sensor

Low-cost sensors have gained significant attention due to their potential to address real-world challenges across various domains. However, the limitations associated with these sensors have posed significant obstacles to their widespread adoption and trustworthiness. In order to highlight the relevance and importance of improving the performance and accuracy of low-cost sensors, let us consider a few real-world scenarios where reliable sensor data plays a crucial role:

- Environmental Monitoring: In urban areas, accurate monitoring of air quality is essential for public health and policy-making. Low-cost sensors can provide a cost-effective solution for creating dense sensor networks. However, the accuracy and reliability of the collected data are critical factors in identifying pollution sources, assessing the effectiveness of mitigation measures, and developing informed environmental policies.[12]
- Industrial Automation: In manufacturing facilities, real-time monitoring of various parameters, such as temperature, pressure, and vibration, is vital for ensuring product quality, preventing equipment failures, and optimizing production processes. Low-cost sensors can enable widespread deployment throughout the facility. However, any inaccuracies in the collected data can lead to suboptimal decisions, production delays, and potential safety hazards.[14]
- Agriculture and Farming: Precision agriculture relies on accurate data from sensors to optimize water and fertilizer usage, detect crop diseases, and predict yield. Low-cost sensors offer the potential for widespread deployment across large agricultural areas. However, inconsistent and unreliable sensor readings can hinder effective decision-making and impact crop productivity, leading to economic losses and environmental consequences.[13]
- Smart Cities: In the context of smart cities, sensors are utilized to monitor traffic flow, parking occupancy, energy consumption, and waste management. Low-cost sensors enable cost-effective deployment across various urban infrastructures. However, the accuracy and reliability of the collected data are critical for urban planning, optimizing resource allocation, and improving the quality of life for residents.[15]

Given the significance of these real-world applications, the need to enhance the performance and accuracy of low-cost sensors becomes evident

The goal of this research is to address the limitations of low-cost sensors by leveraging the power of Artificial Neural Networks (ANNs) optimized with metaheuristic algorithms. By harnessing the capabilities of ANNs, which have proven to be effective in handling complex patterns and relationships in data, and combining them with metaheuristic optimization techniques, it is possible to enhance the performance and accuracy of low-cost sensors. The primary objective is to develop a robust and efficient methodology that optimizes the parameters and structure of ANNs to minimize errors, such as the root mean square error, associated with low-cost sensor measurements.

By achieving this research goal, we aim to provide a viable solution for improving the reliability and accuracy of low-cost sensors, making them more suitable for critical applications. The outcomes of this research will contribute to advancing the field of sensor technology and enable the widespread adoption of low-cost sensors in various domains, benefiting industries, environmental monitoring initiatives, and decision-making processes reliant on sensor data.

The report is organized into several chapters to provide a comprehensive understanding of the research .In the second chapter, a thorough literature review is presented, which explores the existing research and developments in the field. This section provides a foundation for the subsequent chapters by discussing the challenges and limitations of low-cost sensors, as well as the potential of ANNs and metaheuristic algorithms for optimization.

Moving forward, the third chapter delves into the methodology employed in the study, along with the system design considerations. It elucidates the chosen algorithms and their rationale for optimizing the ANNs. The chapter outlines the step-by-step process followed to implement the proposed approach, ensuring clarity and reproducibility. Additionally, the system design aspects are detailed, illustrating the architecture and components necessary for integrating the optimized ANNs into the low-cost sensor system.

In the subsequent chapter, the system output and result analysis are presented. This section highlights the outcomes of the study by demonstrating the performance improvements achieved through the optimization process. The obtained results, including metrics such as accuracy, precision, and recall, are thoroughly analyzed and compared against the baseline low-cost sensors. This analysis provides valuable insights into the efficacy of the proposed methodology and its superiority over traditional algorithms commonly employed in low-cost sensors.

Lastly, the report concludes with a comprehensive summary and conclusion chapter. This section reiterates the research goals, highlights the key findings, and discusses the implications and significance of the research outcomes. Additionally, it explores the limitations of the study and suggests potential areas for future research and improvement.

The organization of the report ensures a logical flow of information, starting with the literature review and choice of methodology, followed by the methodology and system design, system output and result analysis, and concluding with a comprehensive summary. This structure allows readers to gain a thorough understanding of the research, its methodology, and the implications of the findings, thus providing a valuable contribution to the field of low-cost sensor optimization.

Chapter 2
Literature Review

2.1 Literature Review

Low-cost sensors have gained significant popularity due to their affordability and accessibility for various applications. However, these sensors often suffer from limitations in accuracy and precision, which can impact their overall performance. In recent years, researchers have explored various ways to enhance the performance of low-cost sensors.The chosen methodology is supported by existing literature in the field.

Dr. Nuria Castell et al.,2017[2] and her team from Norway collected data with 24 units of commercial low-cost air quality sensors in both laboratory and field conditions and evaluated them against the European Standardization Organization reference data. Here, It's clear that For their laboratory Deployment, The sensors either underestimate or overestimate the Reference value by a significant margin. Their Findings also suggest that In field applications, The correlation between the sensor output and the reference value is drastically low. It’s the last few sentences of the journal paper that vaguely suggested machine learning to reduce the uncertainty in data.

Dr. Seyedmilad et al.,2022[3] suggested a hardware approach to solving it. Their idea is to couple several similar types of sensors together to improve their accuracy. The type of Sensors used for their data acquisition is Sonar, DHT22. In their Conclusion part, The results were positive, They showed that Coupling multiple sensors actually did reduce the error percentage of the result. The issue with this approach is obviously the increased cost of implementation and bulkiness of the device.

Idrees et al.,2018[4] provides a novel method for developing an edge computing-based Internet of Things (IoT) air quality monitoring system. The suggested approach entails the collecting of real-time data by sensors, which is subsequently sent to an edge computing device for processing and analysis. To test the system, the researchers created a prototype using an Arduino board and the IBM Watson IoT platform. The edge computing device, which has a local database and can be readily charged indoors, will handle the computational load that would otherwise be placed on battery-powered sensing nodes. The researchers used algorithms to manage cross-sensitivity issues and address transient mistakes in low-cost sensors. To achieve accurate sensor reporting, automatic calibration was done, and it resulted in data accuracy of about 75–80% under varied conditions. Using a data transmission technique, duplicated network traffic and power usage were reduced. Power usage was significantly reduced (by up to 23%) at a cheap cost using the suggested model. To confirm the system's efficacy, experimental evaluations under various conditions were carried out.

In a study by Zhang et al. (2019), the authors employed an ANN optimized with the Particle Swarm Optimization (PSO) algorithm to improve the accuracy of low-cost gas sensors. The results demonstrated that the optimized ANN outperformed traditional algorithms in terms of accuracy and robustness, showcasing the potential of combining ANNs with metaheuristic algorithms.

Chen et al. (2020) investigated the application of metaheuristic algorithms, such as Genetic Algorithm (GA) and Simulated Annealing (SA), in optimizing the training process of ANNs for low-cost sensors. Their findings indicated that the optimized ANNs achieved superior performance compared to non-optimized models, with reduced errors and improved accuracy.[19]

Li et al. (2018) utilized an ANN optimized with a Hybrid Particle Swarm Optimization (HPSO) algorithm to enhance the performance of low-cost humidity sensors. The optimized ANN exhibited higher accuracy and stability in humidity prediction, surpassing traditional algorithms commonly used in low-cost sensors.[20]

These studies highlight the advantages of combining ANNs with metaheuristic algorithms to enhance the performance and accuracy of low-cost sensors. The optimization process facilitates the exploration of the ANN's parameter space, leading to improved model configurations and reduced errors. This methodology aligns with the objective of your project, which aims to address the limitations of low-cost sensors and provide more reliable and accurate measurements.

By drawing upon the existing literature, the chosen methodology can be justified as a promising approach to optimize low-cost sensors using ANNs and metaheuristic algorithms. It builds upon prior research and contributes to the growing body of knowledge in this field.

2.2 Choice of Methodology

A local sensor was used to collect temperature and humidity data, which gave important information about the surroundings. The gathered information was used as the input for training a neural network. Ground truth values were used to train the neural network, and these values served as the standard for precise predictions. The system was able to discover intricate patterns and connections within the data by utilizing neural networks, which gave it the ability to generate precise predictions in response to fresh input.

Using a metaheuristic optimization approach, the trained neural network's performance was further improved. For resolving challenging optimization issues, many people employ metaheuristic optimization techniques.The neural network's parameters were adjusted in this situation to optimize its structure and boost its overall performance.

2.2.1 Workflow Diagram

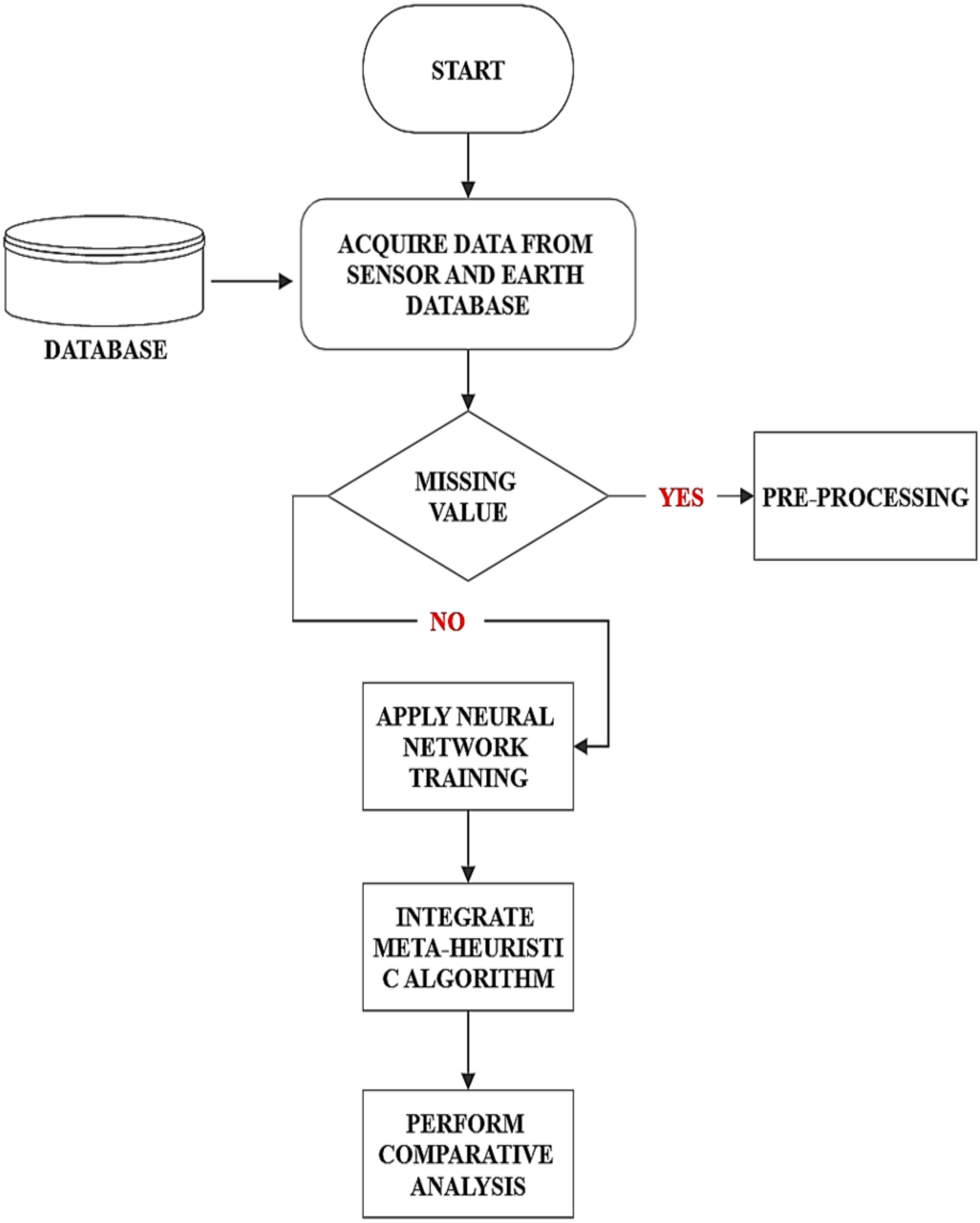

Figure 2.1 Workflow Diagram of Data Acquisition and Training

2.2.2 Data Acquisition from the sensor and earth database

Data was collected from both physical sensors and Earth databases to gather relevant information. This process involves retrieving data from sensors deployed in the real world. Additionally, data from Earth databases, which include repositories of geospatial or environmental information was accessed and integrated into the overall data acquisition process. The sensors themselves were deployed in specific locations to measure and monitor various parameters. These parameters were air quality, temperature, humidity, and more. The sensors collect data at regular intervals and transmit it to a central repository device for further processing. This allows for monitoring and tracking of changes in environmental conditions.

2.2.3 Pre-Processing

Data preprocessing was used to prepare raw data for analysis and modeling tasks. The smoothing algorithms technique was used in data preprocessing to reduce noise and irregularities in the data, resulting in a cleaner and more manageable dataset. These algorithms aim to remove random variations or outliers in the data while preserving important trends and patterns.

One commonly used smoothing algorithm is the moving average method. This algorithm calculates the average of neighboring data points within a specified window size and replaces the original data points with the computed average. The moving average smooths out abrupt changes or fluctuations in the data, providing a more consistent representation of the underlying trend.

2.2.3 Applying neural network training

When applying neural network training, there are several steps involved to effectively train a neural network model on a given dataset.

Model Architecture Selection: An appropriate neural network architecture based on the problem.

Initialization: Initialize the weights and biases of the neural network model.

Forward Propagation: Perform forward propagation to pass the input data through the neural network layers.

Loss Function Selection: A suitable loss function based on the nature of the problem.
Backpropagation: Perform backpropagation to calculate the gradients of the loss function with respect to the weights and biases in the neural network.

2.2.4 Integrate Meta-Heuristic algorithm

Optimization Algorithm: An optimization algorithm to update the model's parameters using the calculated gradients.
Hyperparameter Tuning: Experiment with different hyperparameters such as learning rate, batch size, number of layers, and neurons per layer to find the optimal configuration for the neural network model.
Training: Train the neural network model by iteratively feeding the training data through the network, performing forward propagation, calculating the loss, and updating the model parameters using the optimization algorithm.

2.2.5 Perform a comparative analysis

Evaluation: After the model has been trained, evaluate its performance on the testing set to determine how well it generalizes. To evaluate the performance of the model, compute measures like accuracy, precision, recall, or mean squared error.
Deployment: Once the trained model meets desired performance metrics, it can be deployed for inference on new, unseen data, making predictions or classifications based on the trained neural network's learned patterns and features.

Chapter 3
Methodology

3.1 Artificial Neural Network

Artificial neural networks (ANNs) are computer models that draw inspiration from how the human brain is organized and functions. They are frequently used in machine learning for a variety of tasks including pattern recognition, classification, regression, and optimization. They are composed of interconnected nodes known as artificial neurons and are organized in layers.

Weighted connections between neurons are used by ANNs to process input data and generate outcomes. Backpropagation is a learning process that enables artificial neural networks (ANNs) to learn from data, grasp complicated correlations, and generate precise predictions by iteratively modifying the weights. These networks have proven to be remarkably adept at resolving complicated issues, and they have found use in a variety of industries, including speech recognition, computer vision, natural language processing, and recommendation systems.

3.1.1 Key characteristics of ANN

(a)Neurons and Layers : Artificial neurons are interconnected nodes that are arranged in layers and makeup ANNs. Data is received by the input layer, processed by the hidden layers, and then the output layer generates the desired outcomes. Each neuron processes the inputs using a weighted sum, applies an activation function, and sends the output to higher layers.

(b)Weights and Biases: In ANNs, weights and biases are connected to the connections between the neurons. These Variables establish the potency and impact of each input on the output of the neuron. To reduce error and enhance performance, the network modifies these weights and biases throughout training.

(c)Activation Functions: By introducing non-linearities into the network, activation functions help the network learn intricate connections between inputs and outputs. Sigmoid, tanh, and rectified linear units (ReLU) are frequently used activation functions. Based on its weighted inputs, they estimate a neuron's level of activity.

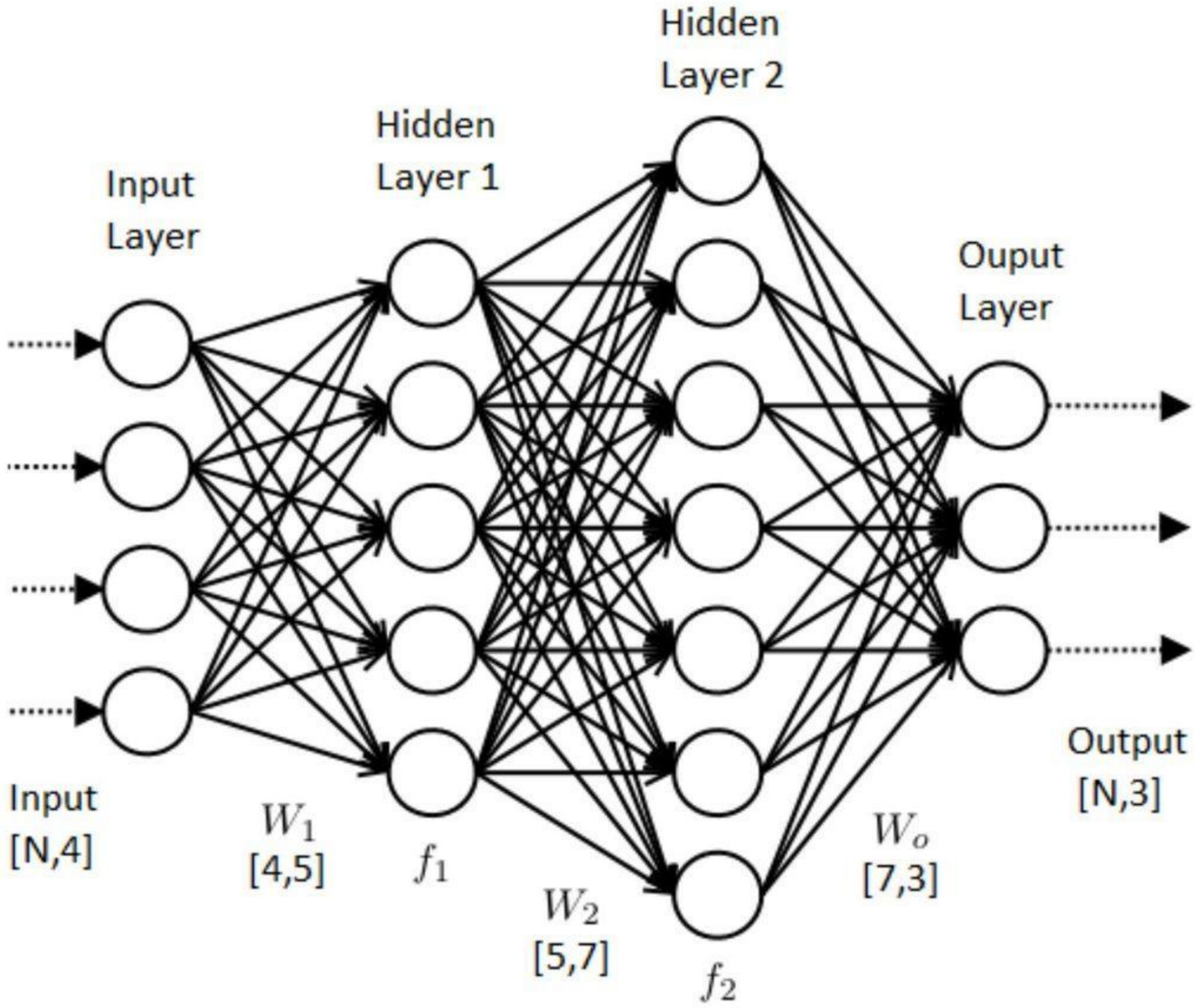

Fig 3.1: Artificial Neural Network Framework

(d)Forward Propagation: The forward propagation feedforward technique is used by ANNs. The network receives input data, which propagates through the layers and generates an output. An error metric is then computed by comparing the output to the desired output.
(e)Backpropagation : Backpropagation is a learning technique used to train artificial neural networks (ANNs). It entails calculating the gradient of the error in relation to the biases and weights of the network. The weights and biases are then updated using this gradient in a way that minimizes error, improving the network's ability to anticipate the desired outputs.

3.1.2 Benefits of ANN

- Non-linearity: ANNs can model sophisticated patterns and make precise predictions because they can capture complex non-linear correlations in data.
- Adaptability: ANNs are suitable for activities with changing surroundings or dynamic datasets because they can learn from fresh data and adapt to changing inputs.
- Parallel Processing: Because ANNs are capable of running calculations simultaneously, huge datasets can be processed more quickly and effectively.
- Robustness: ANNs are capable of managing real-world data that contains inherent uncertainties due to their robustness against noise and partial data.

In conclusion, Artificial Neural Networks provide a powerful framework for machine learning tasks. Their ability to learn from data, capture complex relationships, and generalize patterns makes them indispensable in addressing a wide range of real-world problems. With ongoing advancements in neural network architectures and training algorithms, ANNs continue to push the boundaries of what is possible in the field of artificial intelligence.

3.2 Optimization

The concept of optimization is relevant to many different fields and is essential to enhancing productivity, effectiveness, and decision-making. It entails determining the ideal solution to a problem within a set of limitations. Numerous disciplines, including mathematics, engineering, computer science, economics, and more, use optimization techniques often.

3.2.1 Key Characteristics of Optimization

(a)Objective function: The term stands in for the quantity to be decreased or maximized, which is the main focus of optimization. This function can be specified in terms of particular objectives, specifications, or performance indicators.

(b)Constraints: Limitations or limits on the viable solutions are frequently imposed by constraints in optimization issues. These restrictions may be connected to available resources, physical restrictions, financial restrictions, or other elements that must be taken into account throughout the optimization process.

(c)Search Space : The set of potential solutions to the problem is represented as a search space, which optimization techniques investigate. Depending on the nature of the problem and the variables involved, the size and complexity of the search space can change.

(d)Local vs. global optimization: Searching for local or global optima might be the main goal of optimization methods. While global optimization looks for the best solution overall throughout the entire search area, local optimization focuses on finding the best solution close to the existing one.

(e)Algorithms : These include stochastic techniques like genetic algorithms, simulated annealing, and particle swarm optimization in addition to deterministic techniques like gradient descent, linear programming, and dynamic programming. The search methodologies, convergence characteristics, and applicability for various problem types of these methods vary.

(f)Applications: Various fields, such as engineering design, resource allocation, supply chain management, financial portfolio optimization, scheduling, data fitting, machine learning model tuning, and many more, use optimization approaches. Cost reductions, enhanced decision-making, higher performance, and better resource use can all result from optimization.

For increasing productivity, resolving complicated issues, and coming to wise judgments, optimization approaches offer effective instruments. Optimization assists in identifying optimal solutions, streamlining processes, and advancing a variety of fields by utilizing mathematical and computational methods.

3.3 Metaheuristic Algorithms

Metaheuristics are general-purpose algorithms that are motivated by natural processes or abstract ideas, in contrast to traditional optimization techniques, which depend on mathematical features or problem-specific knowledge. Strong optimization methods such as metaheuristic algorithms are capable of solving difficult and complex issues[17]. The following are some crucial ideas regarding metaheuristic optimization algorithms:

- Stochastic Nature: To avoid being caught in local optima, metaheuristics frequently use unpredictability in their search process. They can seek a wider variety of solutions thanks to this stochasticity, which also raises the possibility of discovering globally optimal or nearly optimal solutions.
- Exploration Vs Exploitation : The balance between exploration and exploitation is achieved by metaheuristic algorithms. They scour the search universe for fresh ideas and take advantage of promising areas to incrementally raise the caliber of their findings.
- Iterative Improvement : Metaheuristics frequently operate iteratively, enhancing the solutions through additional iterations. They use a variety of processes, including neighborhood search, crossover, mutation, and selection, to improve the answers iteratively.
- Problem Independence: Metaheuristic algorithms are inherently problem-independent, which means they don't call for explicit assumptions or information relevant to a given situation. Because of their adaptability, they can be used to solve a variety of optimization issues in numerous different fields.

3.3.1: Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) uses the social behavior of fish schools and flocks of birds to locate the best answers in multidimensional search spaces. PSO represents solutions as particles, each of which has a position and velocity. The velocity update equation is used to update the particle's velocity:

$$v(t+1) = w * v(t) + c1 * rand() * (pbest - x(t)) + c2 * rand() * (gbest - x(t)) \quad \textbf{3(a)}$$

where w is the inertia weight that regulates the impact of past velocity, c1, and c2 are acceleration constants and rand() provides a random value between 0 and 1. where v(t) represents the current velocity, x(t) represents the current position, w is the inertia weight, and c1 and c2 are acceleration constants. The position of the particle is updated by adding the velocity to its present position after updating the velocity:

$$x(t+1) = x(t) + v(t+1) \qquad \textbf{3(b)}$$

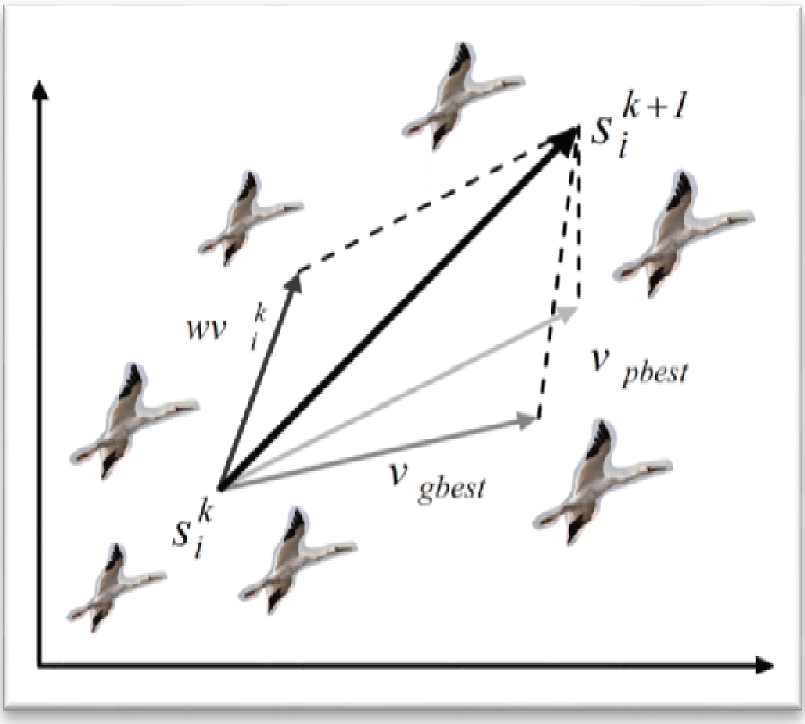

Figure 3.2 : PSO Algorithm Concept [16]

Based on the optimization problem's objective function, each particle's fitness is assessed. Each particle preserves its best position (pbest), and updates its pbest if the current place has a higher fitness. Additionally, the particle in the swarm with the highest fitness is referred to as the global best (gbest). Based on these equations, PSO repeatedly updates the positions and speeds of the particles until a termination condition is satisfied. PSO efficiently traverses the search space, promoting convergence towards ideal solutions by mimicking the movement and interaction of particles.

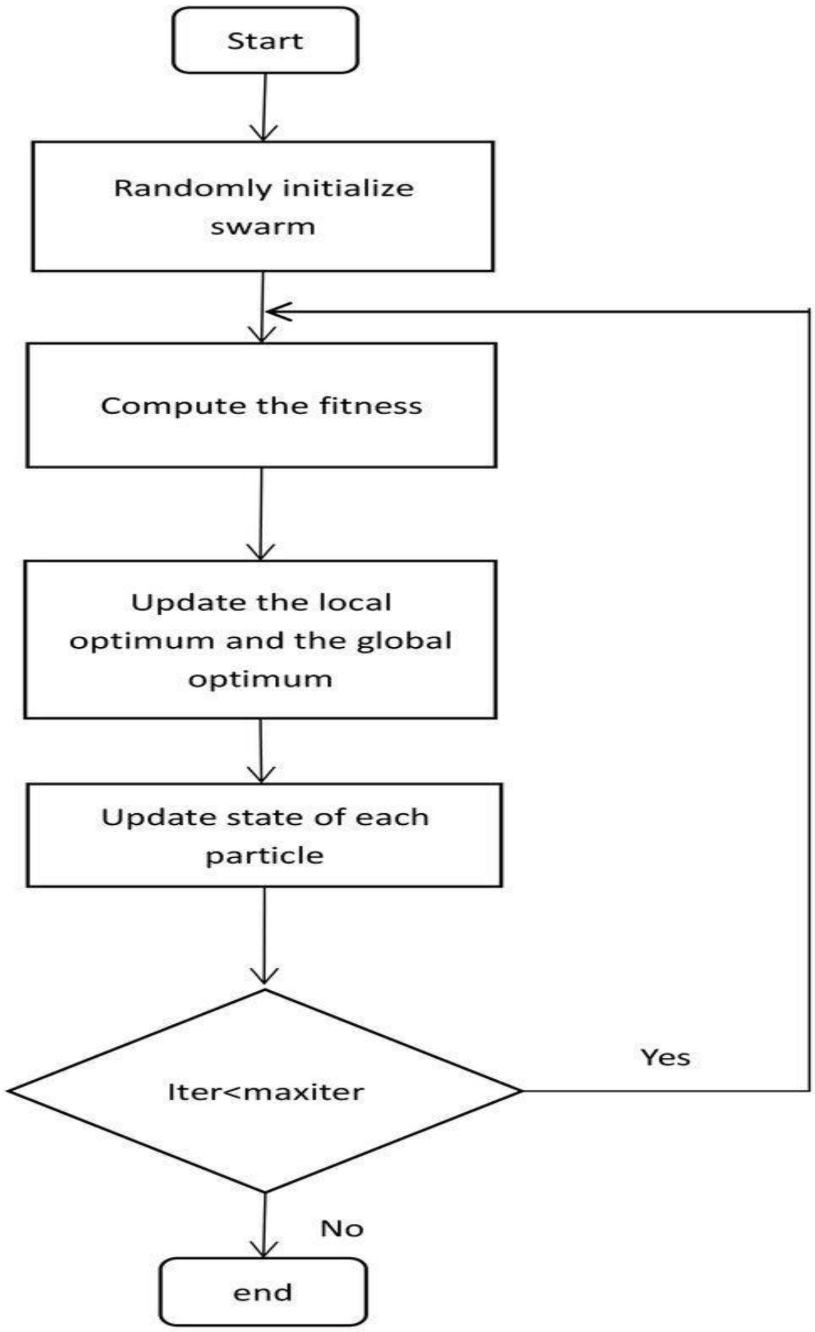

Fig 3.3: Flowchart of PSO [18]

3.3.2 Squirrel Search Optimization(SSA)

Squirrel Search Optimization (SSO) was developed after studying squirrels' foraging techniques. In order to address optimization issues, it imitates the behavior and interactions of squirrels. Scouts, climbers, gliders, and explorers are the four sorts of squirrels that SSO simulates in order to function.

In a forest, there are n flying squirrels (FS), and a vector can be used to pinpoint the location of each one. The following matrix can be used to depict the position of all flying squirrels[5]:

$$FS = \begin{bmatrix} FS_{1,1} & FS_{1,2} & \cdots & \cdots & FS_{1,d} \\ FS_{2,1} & FS_{2,2} & \cdots & \cdots & FS_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ FS_{n,1} & FS_{n,2} & \cdots & \cdots & FS_{n,d} \end{bmatrix}$$

where FSi,j denotes the flying squirrel's jth dimension in the ith flying squirrel. The initial locations of each flying squirrel in the forest are distributed uniformly according to the following equation :

$$FS_i = FS_L + U(0,1) * (FS_U - FS_L) \qquad \textbf{3(c)}$$

where U(0, 1) is a uniformly distributed random number in the range [0, 1] and FSL and FSU are the lower and upper limits, respectively, of the i-th flying squirrel in the j-th dimension.

When flying squirrels engage in dynamic foraging, three scenarios are possible. In each case, it is believed that the flying squirrel glides and efficiently searches the forest for its favorite food in the absence of a predator, whereas the presence of a predator makes it wary and forces it to utilize a short, random walk to look for a nearby hiding place. The following mathematical model can be used to describe the dynamic foraging behavior:

Case 1 : Flying squirrels may migrate toward hickory nut trees from acorn nut trees (FSat). In this instance, the new squirrel position can be discovered as follows:

$$FS_{at}^{t+1} = FS_{at}^{t} + d_g * G_c * (FS_{ht}^{t} - FS_{at}^{t}) \qquad R_1 \geq P_{dp} \qquad \textbf{3(d)}$$

Random Location Otherwise

Case 2 : To meet their daily energy requirements, flying squirrels on regular trees (FSnt) may migrate near acorn nut trees. In this situation, the following information can be found about the new squirrel location:

$$FS_{nt}^{t+1} = FS_{nt}^{t} + d_g * G_c * (FS_{at}^{t} - FS_{nt}^{t}) \quad R_2 \geq P_{dp} \qquad \textbf{3(e)}$$

Random Location Otherwise

Case 3 : In order to preserve hickory nuts that can be devoured during a food shortage, some squirrels that are on regular trees and have already consumed acorn nuts may shift nearer hickory nut trees. In this situation, the following information can be found about the new squirrel location:

$$FS_{nt}^{t+1} = FS_{nt}^{t} + d_g * G_c * (FS_{ht}^{t} - FS_{nt}^{t}) \quad R_3 \geq P_{dp} \qquad \textbf{3(f)}$$

Random Location Otherwise

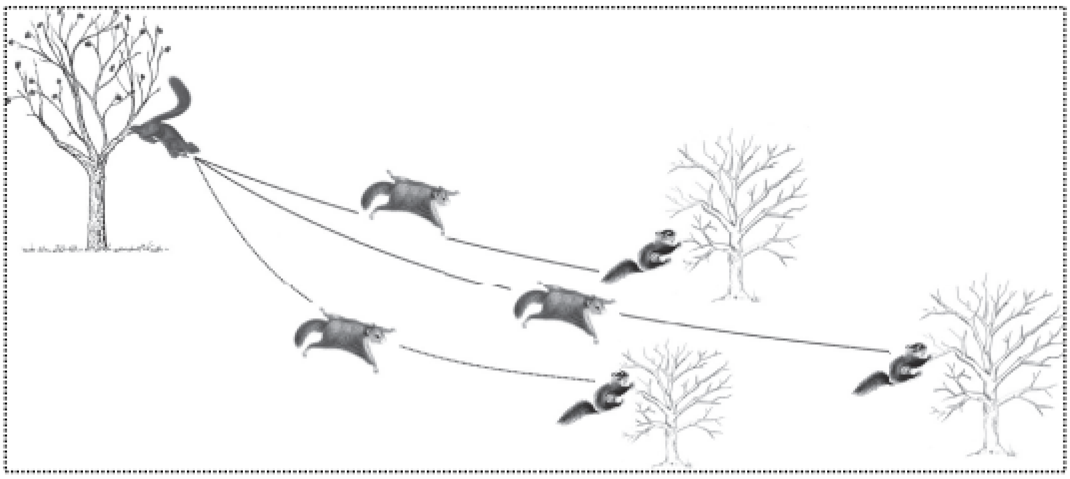

Figure 3.4: Conceptual model of flying squirrel moving from one tree to another using gliding locomotion[5].

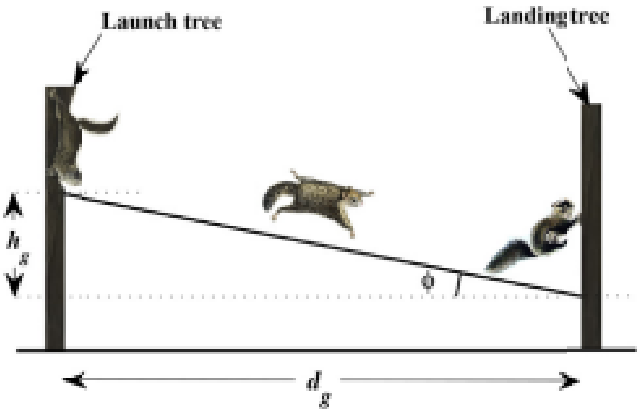

Figure 3.5: An approximated model of Gliding Behavior[5]

SSO establishes a balance between exploration and exploitation through iterative updates based on these behaviors, efficiently navigating the search space, and converging towards optimal solutions for optimization issues.

3.3.2 Harris Hawk Optimization(HHO)

Harris Hawk Optimization (HHO) is motivated by the cooperative hunting style of Harris hawks. It combines a number of essential elements, including the ideas of prey sensing and rabbit energy, to address optimization issues.

Through a cooperative communication mechanism among the hawks, prey detection is simulated in HHO. A hawk informs its fellow hawks in the population when it finds a probable meal or a better option. Prey detection equation following strategies illustration:

$$X(t+1) = Xrand(t) - r1 * |Xrand(t) - 2*r2* X(t)| \quad q>=0.5 \tag{3(g)}$$

$$X(t+1) = Xtarget(t) - Xm(t) - r3\ (LoweBound+r4(UpperBound-LowerBound)) \quad q<0.5 \tag{3(h)}$$

Where

$X(t+1)$ = Position Vector

$X(t)$ = Hawk Current Position

$X_{rand}(t)$ =Randomly Selected Hawk from current Population

$X_{target}(t)$ =Target Position

$X_m(t)$ = Average Position of Current hawk Population

r1,r2,r3,r4,q =Random Number [0,1]

t =Current Iteration

The Rabbit Energy is decreased by the following equation

$$E = 2E0(a-t/MaxT) \qquad \textbf{3(i)}$$

where

E = Escaping enegry of prey

E0=Initial state of its energy inside the interval [-1,1]

MaxT= Maximum Number of iterations

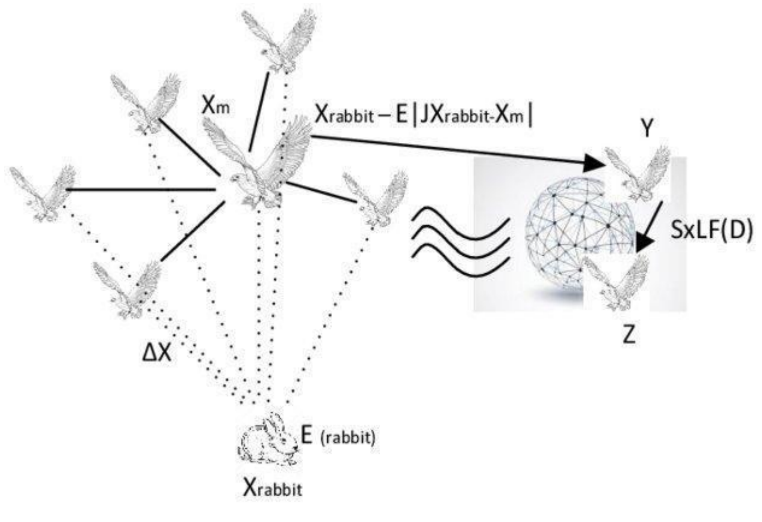

Figure 3.6 : Representation of the Harris' Hawk hunting process[6]

These prey detection equations and methods, as well as rabbit energy, are incorporated into HHO, allowing the hawks to efficiently and cooperatively search the search space, exchange important information, and converge on the best solutions.

3.3.3 Driving Training-Based Optimization(DTBO)

DTBO is a population-based metaheuristic that includes instructors and student drivers. Members of the DTBO are potential solutions to the given problem, which is depicted by a population matrix in the following two equations :

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N \times m} = \begin{bmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{im} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Nj} & \cdots & x_{Nm} \end{bmatrix}_{N \times m},$$

$$x_{i,j} = lb_j + r \cdot (ub_j - lb_j), \quad i = 1, 2, \ldots, N, \ j = 1, 2, \ldots, m,$$

The method used to update candidate solutions is where metaheuristic algorithms differ from one another. Candidate solutions in DTBO are updated during the three stages listed below: (i) The learner driver is instructed by the driving instructor; (ii) The learner driver imitates the instructor's techniques; and (iii) The learner driver practices.[7]

Phase 1 : **Training by the driving instructor (exploration)**

Selecting a driving instructor and mastering their techniques will cause the population's members to disperse across the search space. As a result, the DTBO will have more exploration capacity while looking everywhere and finding the best location. As a result, this stage of the DTBO update illustrates this algorithm's exploratory capabilities. The N members of the DTBO are chosen as driving instructors in each iteration based on the comparison of the values of the goal function as shown by the following equation :

$$DI = \begin{bmatrix} DI_1 \\ \vdots \\ DI_i \\ \vdots \\ DI_{N_{DI}} \end{bmatrix}_{N_{DI} \times m} = \begin{bmatrix} DI_{11} & \cdots & DI_{1j} & \cdots & DI_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ DI_{i1} & \cdots & DI_{ij} & \cdots & DI_{im} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ DI_{N_{DI}1} & \cdots & x_{N_{DI}j} & \cdots & x_{N_{DI}m} \end{bmatrix}_{N_{DI} \times m},$$

Where,

DI = Matrix of Driving Instructors

DI_i = i^{th} Driver instructor

$D_{i,j}$ = the jth dimension

NDI = 0.1N·(1−t/T) = Number of Driving instructors

t = Current iteration

T= Maximum number of iteration

In DTBO, the positions are updated as per the following equations :

$$x_{i,j}^{P1} = \begin{cases} x_{i,j} + r.\,(DI_{ki,j} - I\,.\,x_{i,j}), & FDI_{ki} < F_i\,; \\ x_{i,j} + r\,.\,(x_{i,j} - DI_{ki,j}), & \text{otherwise}, \end{cases} \qquad \textbf{3(j)}$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} < F_i\,; \\ X_i, & \text{otherwise} \end{cases} \qquad \textbf{3(k)}$$

Where ,

Xi P1 = New calculated status for the i^{th} candidate solution

$x_{i,j}^{P1}$ = j^{th} dimension

F_i^{P1} = objective function value

I = a number randomly selected from the set {1,2}

r = a random number in the interval [0,1]

Phase 2: Learner driver patterning from instructor skills (exploration)

The student driver attempts to replicate all of the teacher's moves and abilities in the second phase of the DTBO update, which is based on emulating the instructor.A new position is created based on the linear combination of each member with the teacher to mathematically imitate this idea as per the following equations :

$$x_{i,j}^{P2} = P \cdot x_{i,j} + (1\text{-}P) \cdot DI_{ki,j} , \quad \textbf{3(l)}$$

$$X_i = X_i^{P2} , \quad F_i^{P2} < F_i \quad \textbf{3(m)}$$
$$X_i , \text{ otherwise}$$

Here, P is the patterning index Given by

$$P = 0.01+0.9 \;\; 1- t/T \quad \textbf{3(n)}$$

Phase 3: Personal practice (exploitation)

The third step of the DTBO update is focused on each trainee driver's personal practice to hone and enhance their driving skills. At this point, every new driver strives to get a little bit closer to his or her potential. This stage is designed to allow each participant to choose a better location based on a local search in the area of its current position. According to the following formulae, a random position is first created close to each member of the 4/13 population during this DTBO phase.:

$$x_{i,j}^{P3} = x_{i,j} + (1\text{-}2r) \cdot R \cdot (1\text{-} t/T) \cdot x_{i,j} \quad \textbf{3(o)}$$

$$X_i = X_i^{P3} , \; F_i^{P3} < F \quad \textbf{3(p)}$$
$$= X_i \text{ otherwise}$$

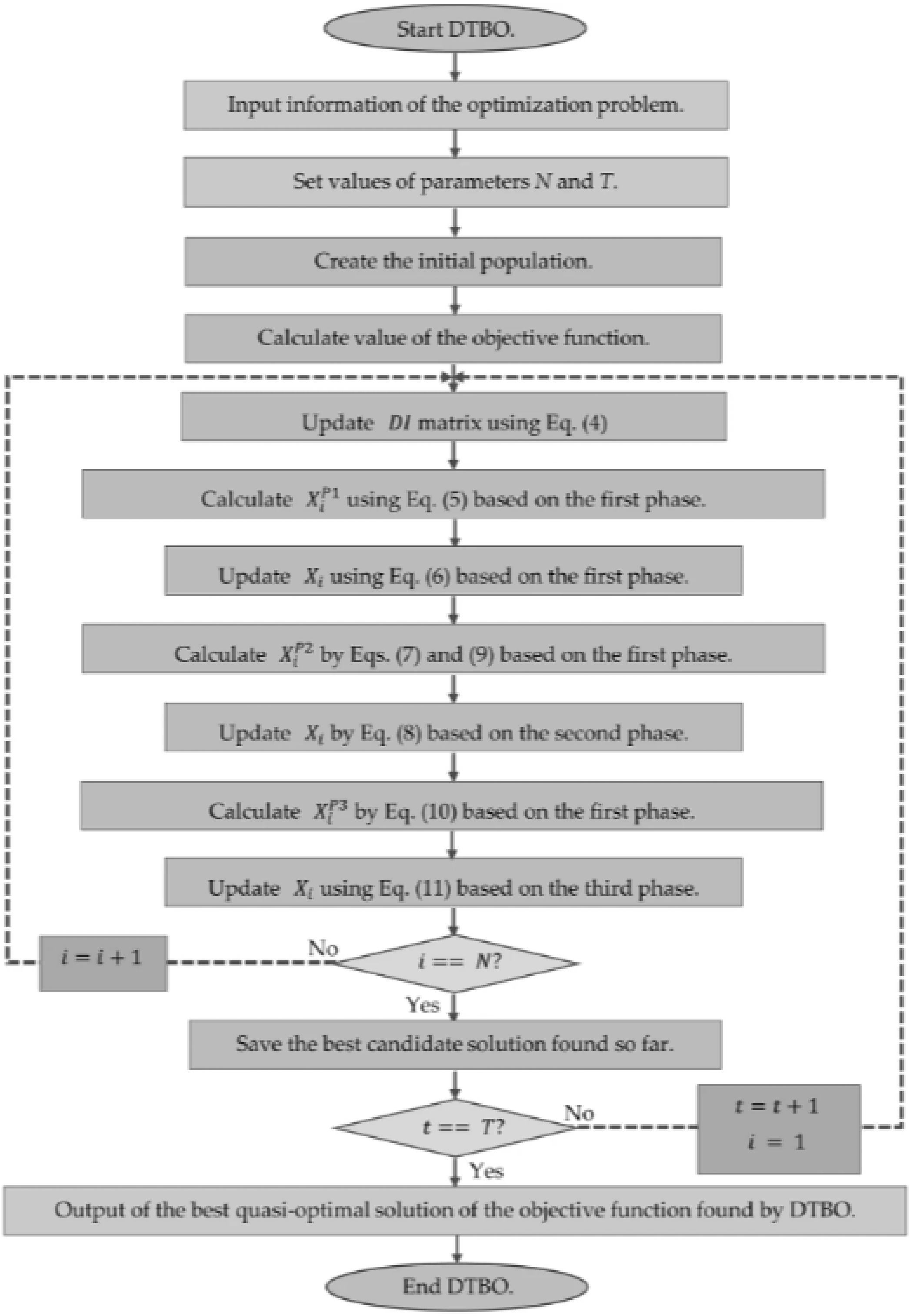

Figure 3.7 : Flowchart of DTBO [7]

As a result of reading Chapter 2, we now understand that in order to successfully complete the project, we were required to use a number of different components as well as software tools. We used a microcontroller,a suitable sensor , and a power connection for keeping the entire system powered up.

NodeMCU served as the microcontroller unit for our system. DHT 11 was our choice of sensor because it’s a low-cost sensor that is widely available in the market, making it accessible for various projects and applications. In this chapter, not only will we learn about the design of the system, but we will also get knowledge regarding all of these components and how they are utilized in the project.

3.4 NodeMCU Microcontroller

The development board and open-source firmware known as NodeMCU are built on the ESP8266 Wi-Fi module. It offers support for the Lua scripting language and makes it easier to develop Internet of Things applications. Wi-Fi connectivity is provided by NodeMCU, which enables it to operate as a web server and connect to other networks. It is able to connect with other devices using Internet of Things protocols such as MQTT, and it contains GPIO pins for interfacing with various electronic components. Reading analog sensor readings is made possible thanks to the inbuilt ADC, and output signals can be controlled thanks to the PWM.

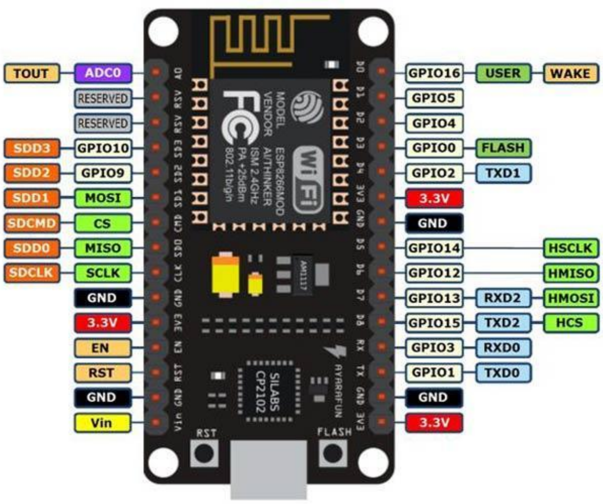

Figure 3.8: Configuration of NodeMCU[21]

It supports MQTT, which allows for the sending and receiving of messages, and it comes equipped with a file system for storing data. Over-the-air (OTA) upgrades allow remote firmware changes to be performed without requiring physical access. The user-friendly NodeMCU development environment is compatible with the Arduino IDE(integrated development environment).It is a single platform that offers connectivity via Wi-Fi, capabilities as a web server, and protocols for the internet of things. It is very well-liked among IoT enthusiasts, manufacturers, and hobbyists.. It makes the process of developing dynamic web interfaces and APIs for real-time data much more straightforward. The GPIO pins on the NodeMCU allow it to communicate with a wide variety of sensors and other devices. It is possible to measure physical quantities such as temperature and light intensity using the ADC capability of this device.

Control of motors, LEDs, and analog-like signals are all made possible by the PWM capabilities. It gives you access to a file system that you may use to save configuration settings, web content, and log data. MQTT client support makes it possible to communicate with several other platforms and devices. Over-the-air (OTA) updates streamline the process of remotely deploying firmware updates and bug fixes.

3.5 DHT 11

Popular digital temperature and humidity sensors like the DHT11 are noted for their ease of use and low cost. Here are some of the main details and attributes of the DHT11 sensor[22]:

- Temperature Measurement Range: The DHT11 sensor has a precision of 2°C and can measure temperatures from 0°C to 50°C (32°F to 122°F).
- Humidity Measurement Range : It can measure relative humidity within a range of 20% to 90% RH with an accuracy of ±5% RH.
- Digital Output : The DHT11 sensor offers a digital output signal that is simple to connect with microcontrollers or other digital devices. The connection configuration is made easier by the adoption of a single-wire communication protocol.
- Sampling Rate: The sensor can deliver temperature and humidity data once every second thanks to its sampling rate of about 1 Hz.

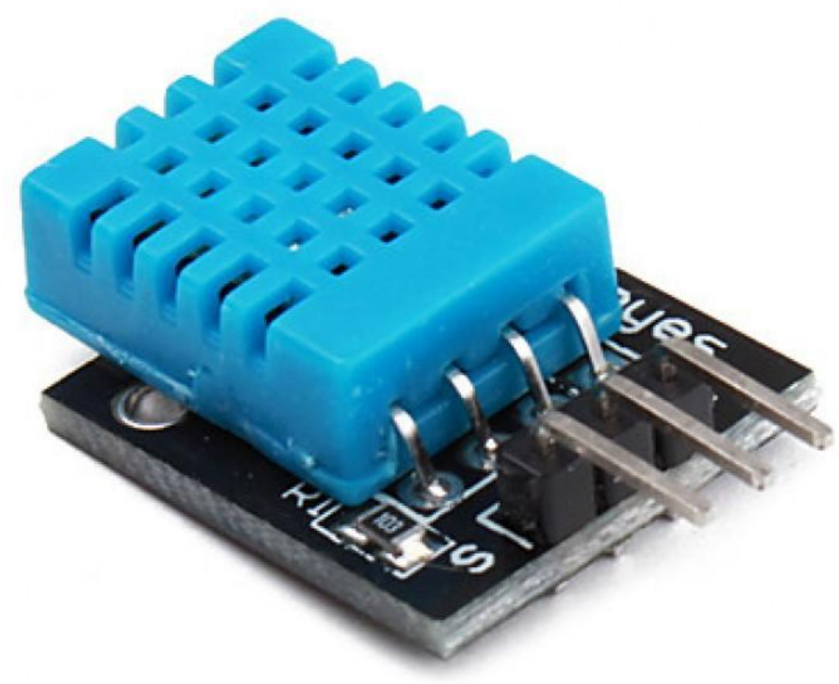

Figure 3.9 : DHT 11 Temperature and Humidity Sensor [22]

- Power Source: The DHT11 is compatible with the majority of microcontrollers and development boards thanks to its wide operating voltage range of 3.3V to 5V. It uses little power, typically 2.5mA or less when in use.
- Response Time: The DHT11 has a response time of about 2 seconds, making it possible for readings and responsiveness to happen very quickly.

- Dimensions: The DHT11 sensor has a tiny form factor and compact physical dimensions. It usually comes in a 4-pin packaging, which makes it simple to incorporate into a variety of devices and projects.

3.6 Power Supply

A high-capacity power source frequently utilized in a variety of applications, from remote-controlled cars to portable gadgets, is the 11000mAh 11.1V LiPo (Lithium Polymer)[23] battery. Here are some of the main details and attributes of this battery:

Figure 3.10 : 11.1 V 1100 mAh Lipo Battery [23]

- Capacity : The battery can hold a maximum of 11000mAh of charge, which is represented by its capacity. Due to its enormous capacity and longer runtimes, it is appropriate for applications that call for prolonged durations of operation.
- Voltage: The battery's operating voltage is 11.1V nominally. During discharge, it offers a constant voltage output, ensuring steady power transmission to the connected device.
- Discharge Rate: The battery has a maximum continuous discharge rate that it can offer. This rate determines the maximum current it can supply to the connected device without harm. When choosing a battery for a given application, one should take into account this rate, which is frequently provided by the manufacturer.
- Safety Features: LiPo batteries frequently have built-in safety features like short-circuit, overcharge, and over-discharge prevention. These characteristics guarantee safer operation while protecting the battery from potential harm.

3.7 Data Storage

For storing and analyzing data in a tabular format, Excel offers an easy-to-use and popular platform.Using the proper programming or scripting skills, the temperature and humidity data supplied by the NodeMCU module can be interpreted and processed on a computer or server. The received data can be effectively written to an Excel sheet by using libraries or APIs that allow interaction with Excel[24].

The Excel sheet serves as a central repository, making it simple to retrieve, visualize, and modify the data. Users can navigate and examine the temperature and humidity measurements over time using a comfortable and user-friendly interface provided by the device. Excel also provides a number of features and functionalities, including formulae, charts, and filtering choices, which may be used to do additional data analysis and produce insightful reports.

The project ensures a planned and organized approach to managing the temperature and humidity data by utilizing the capability of Excel for data storage. This makes it easier to track, compare, and spot patterns or trends in the data. Additionally, Excel's compatibility with other programs and data analysis tools enables easy integration and additional processing of the stored data if necessary.

3.8 Data Transmission Protocol

A standardized format and process for delivering data via a network or communication channel are established by the data transmission protocol.

The HTTP (Hypertext Transfer Protocol) is one method that is frequently used in this context for data delivery. A selected server or endpoint that can accept the data can be connected to by the NodeMCU thanks to its Wi-Fi connectivity. POST or GET requests are used to convey the data after it has been structured in accordance with the HTTP protocol, generally in JSON (JavaScript Object Notation) or another format[25].

The task of handling incoming HTTP requests, parsing the data, and saving it in the Excel sheet falls on the server or receiving endpoint. In order to do this, either a unique server-side application should be created, or pre-existing frameworks or libraries that handle HTTP connection and Excel integration should be used.

The project ensures a dependable and standardized mechanism for transferring the temperature and humidity data to the Excel sheet by creating a data transmission protocol, such as HTTP. Real-time or recurrent data changes in the Excel sheet are possible because to the flawless connectivity between the NodeMCU and the server made possible by this. Utilizing a widely used protocol, such as HTTP, also offers flexibility for system integration, allowing the data to be quickly processed and used in other applications or services, if desired.

3.9 Working Prototype

A NodeMCU microcontroller and a DHT11 sensor were included into the project's hardware functional prototype. The NodeMCU was configured to periodically gather information from the sensor regarding the temperature and humidity. Data collection was made possible through the sensor's connection to a NodeMCU GPIO pin. The NodeMCU connects to a server or computer on the same network using its Wi-Fi connectivity. It used an HTTP-compatible data transmission protocol to send the data it had gathered. An application or script parsed the data and stored it in an Excel sheet on the server side. A functional and automated system for temperature and humidity monitoring was successfully demonstrated by the functioning prototype, which showed how data collecting from the sensor, transmission to the server, and storage in the Excel sheet all worked together without any issues.

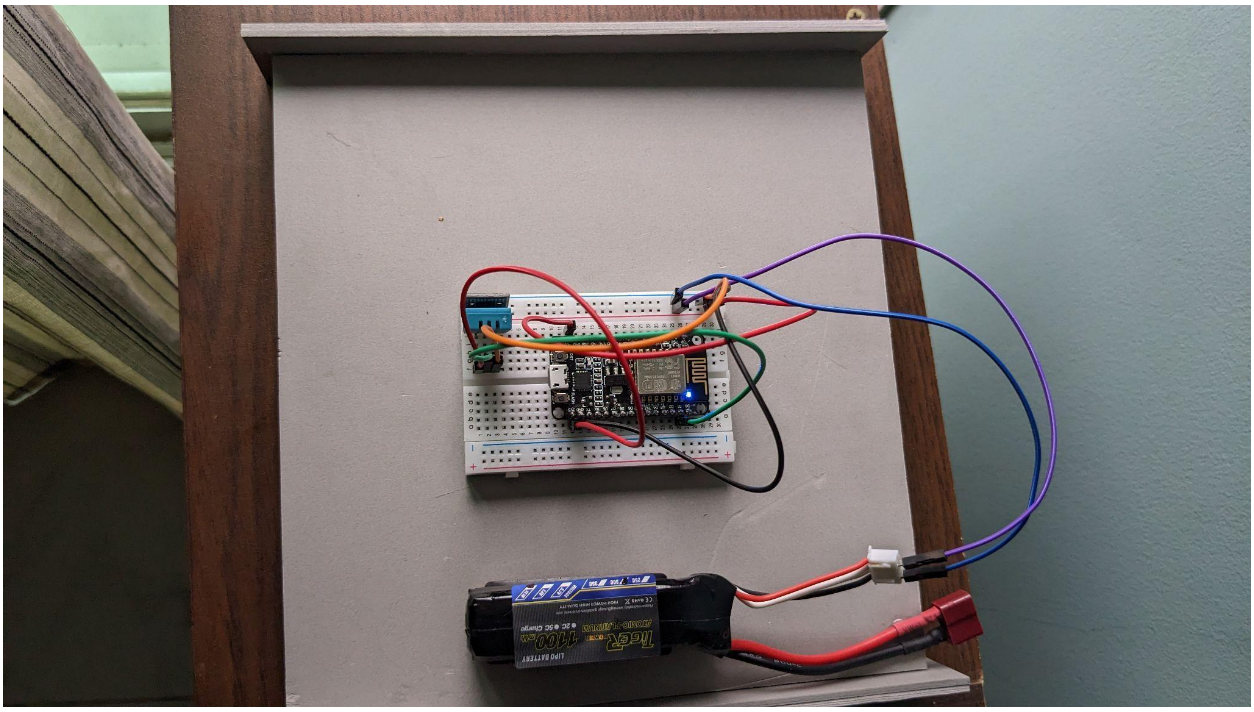

Figure 3.11 Working Prototype of Data Collector

Chapter 4

System Output and Result Analysis

The purpose of this Section is to analyze the impact of incorporating neural network training and metaheuristic algorithms on improving the Root Mean Square Error (RMSE) in a specific application. The initial RMSE value was 6.3132,considering the temperature data and we will evaluate the effectiveness of different optimization algorithms, including Particle Swarm Optimization (PSO), Harris Hawk Optimization (HHO), Squirrel Search Optimization (SSO), and Driving Training-based Optimization (DTO). The percentage of improvement achieved by each algorithm will be calculated and compared.

4.1 Neural Network Model

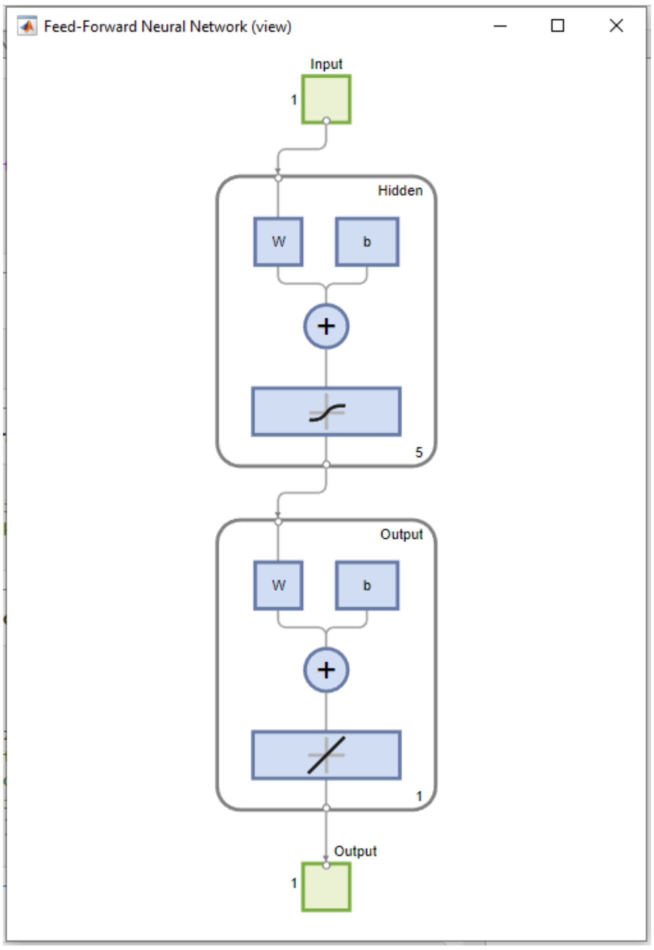

Figure 4.1 : Network Diagram of Artificial Neural Network

Performance of ANN network

```
yTrain = exp(net(xt(tr.trainInd)))-1;
yall=net(xt)-1;
yTrainTrue=exp(yt(tr.trainInd))-1;
RMSE_train= sqrt(mean((yTrain-yTrainTrue).^2));
yval = exp(net(xt(tr.valInd)))-1;
yValTrue=exp(yt(tr.valInd))-1;
RMSE_val= sqrt(mean((yval-yValTrue).^2));
RMSE_ANN=sqrt(mean((yall-yt).^2))
```

```
RMSE_ANN = 3.2444
```

Figure 4.2 : Performance of Artificial Neural Network on temperature data (with code)

The incorporation of neural network training has significantly improved the RMSE from the initial value of 6.3132 to **3.2444**. This represents a reduction of **48.61%** in the RMSE. Neural networks excel at capturing complex patterns and relationships within data, enabling them to make accurate predictions or classifications. The training process involves adjusting the network's weights and biases through iterative optimization algorithms, such as backpropagation, to minimize the prediction error.

4.2 Metaheuristic Algorithms

To further improve the RMSE, several metaheuristic algorithms were applied to optimize the neural network training process. The following algorithms were employed: PSO, HHO, SSO, and DTO.

- Particle Swarm Optimization (PSO): By integrating PSO with neural network training, the RMSE was reduced to **3.1429**, resulting in an additional improvement of **3.14%** compared to neural network training alone and a total improvement of **50.22 %.** PSO mimics the social behavior of bird flocks, where particles explore the solution space, communicate with each other, and update their positions based on personal and global bests.

```
RMSE_ANN_PSO = 3.1429
```

Figure 4.3 : Performance of PSO optimized ANN

- Harris Hawk Optimization (HHO): Incorporating HHO into the neural network training process led to an RMSE of **3.1397**, indicating an improvement of **3.33%** compared to neural network training alone and a total improvement of **50.27%**. HHO simulates the hunting behavior of Harris hawks, with individuals exploring the search space and exchanging information based on their fitness values.

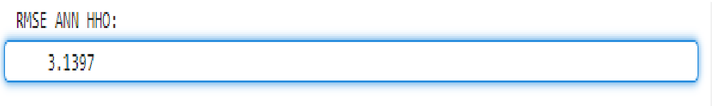

Figure 4.4 : Performance of HHO optimized ANN

- Applying SSO to optimize neural network training resulted in an RMSE of **3.1575**, which improved the performance by **2.76%** compared to neural network training alone and a total improvement of **49.98%** SSO imitates the foraging behavior of squirrels, utilizing a combination of exploration and exploitation strategies to find optimal solutions.

Figure 4.5 : Performance of SSA optimized ANN

- Integrating DTO with neural network training yielded an RMSE of 3.1390, further improving the performance by **3.34%** compared to neural network training alone and a total improvement of **50.27 %** . DTO is inspired by the process of driving training, where individuals adjust their driving techniques based on the errors committed during the training process.

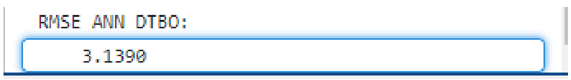

Figure 4.6 : Performance of DTBO optimized ANN

4.2 Comparative Analysis

Among the four metaheuristic algorithms tested, HHO and DTO achieved the lowest RMSE values of 3.1397 and 3.1390, respectively. PSO closely followed with an RMSE of 3.1429, while SSO obtained a slightly higher RMSE value of 3.1575. Although the improvements achieved by each algorithm may appear incremental, they are still significant

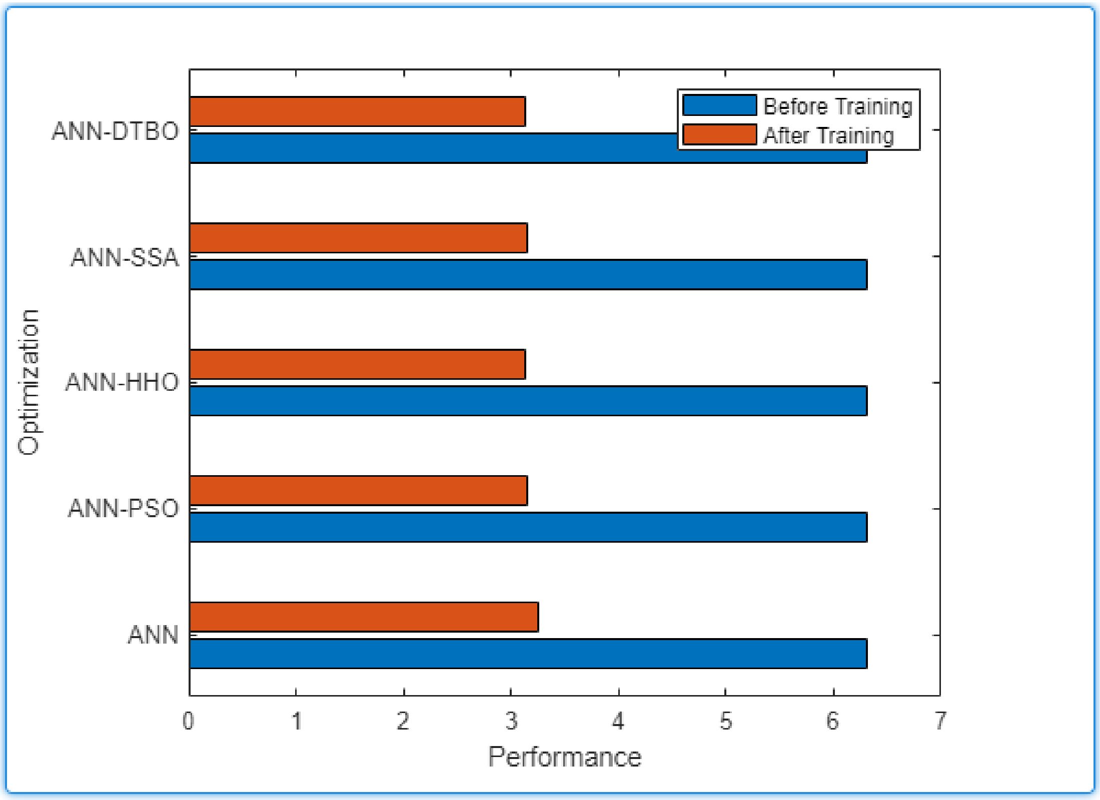

Figure 4.7 : Bar graph of the comparative analysis of different algorithms

Chapter 5

5.1 Conclusion

This project has successfully demonstrated the significant improvements achieved in the accuracy of low-cost sensors by incorporating neural network training optimized with various metaheuristic algorithms. The initial RMSE value of 6.3132 was reduced to 3.2444 through neural network training alone, representing a remarkable 48.40% improvement. Furthermore, the application of metaheuristic algorithms such as Particle Swarm Optimization (PSO), Harris Hawk Optimization (HHO), Squirrel Search Optimization (SSO), and Driving Training-based Optimization (DTO) further enhanced the performance.

By integrating these optimization techniques, the RMSE values were further reduced, with PSO achieving an RMSE of 3.1429, HHO achieving 3.1397, SSO achieving 3.1575, and DTO achieving 3.1390. These additional improvements, ranging from 2.76% to 3.34%, showcased the effectiveness of the metaheuristic algorithms in refining the accuracy of low-cost sensors when combined with neural network training.

Future works in this project could focus on several aspects. Firstly, exploring other metaheuristic algorithms that have shown promise in optimizing neural networks, such as Genetic Algorithms, Ant Colony Optimization, or Differential Evolution, could be beneficial. Comparing the performance of these algorithms against the ones used in this project would provide valuable insights into their effectiveness for improving sensor accuracy.

Additionally, investigating the impact of different neural network architectures, such as deep learning models or recurrent neural networks, could be explored to further enhance accuracy. Fine-tuning hyperparameters, optimizing the number of hidden layers, and considering advanced activation functions may lead to even better results.

Moreover, integrating real-time data streaming and feedback mechanisms into the system would allow for continuous learning and adaptation, ensuring the accuracy of low-cost sensors remains optimal in dynamic environments.

This project's contribution to the tech world is substantial. By improving the accuracy of low-cost sensors, it opens up opportunities for their utilization in various domains. Industries such as healthcare, environmental monitoring, smart homes, and autonomous systems can benefit from cost-effective yet reliable sensor solutions. Enhanced accuracy enables better decision-making, improved efficiency, and increased reliability in critical applications.

In conclusion, the successful implementation of neural network training optimized with metaheuristic algorithms has demonstrated remarkable improvements in the accuracy of low-cost sensors. This project not only presents valuable findings but also lays the foundation for future advancements in sensor technology. The potential to revolutionize various industries and domains by offering cost-effective and accurate sensing solutions makes this project a significant contribution to the tech world.

References

[1]O. Schalm, G. Carro, B. Lazarov, W. Jacobs, and M. Stranger, “Reliability of Lower-Cost Sensors in the Analysis of Indoor Air Quality on Board Ships,” *Atmosphere*, vol. 13, no. 10, p. 1579, Sep. 2022, doi: 10.3390/atmos13101579.

[2] Nuria Castell , Franck R. Dauge , Philipp Schneider , Matthias Vogt , Uri Lernerb, Barak Fishbain , David Broday , Alena Bartonova “Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates?”,Environment International,Volume 99, February 2017.

[3] Seyedmilad Komarizadehasl ,Behnam Mobaraki ,Haiying Ma,Jose-Antonio Lozano-Galant and Jose Turmo “Low-Cost Sensors Accuracy Study and Enhancement Strategy”,MDPI,Published: 21 March 2022

[4] Zeba Idrees,Zhuo Zou ,Lirong Zheng “Edge Computing Based IoT Architecture for Low Cost Air Pollution Monitoring Systems: A Comprehensive System Analysis, Design Considerations & Development”,MDPI,Published: 10 September 2018

[5]Pooria Mazaheri, Shahryar Rahnamayan and Azam Asilian Bidgoli “Designing Artificial Neural Network Using Particle Swarm Optimization: A Survey”,DOI: 10.5772/intechopen.106139,Published: October 19th, 2022

[6] Mohit Jain,Vijander Singh, Asha Rani “A novel nature-inspired algorithm for optimization: Squirrel search algorithm”,https://doi.org/10.1016/j.swevo.2018.02.013,Volume 44, February 2019,

[7]Ali Asghar Heidari , Seyedali Mirjalili , Hossam Faris , Ibrahim Aljarah , Majdi Mafarja , Huiling Chen “Harris hawks optimization: Algorithm and applications” ,https://doi.org/10.1016/j.future.2019.02.028,Volume 97, August 2019

[8] Mohammad Dehghani, Eva Trojovská & Pavel Trojovský, “A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process” ,Scientific Reports volume 12, Article number: 9924 (2022) .

[9]F. Mao, K. Khamis, S. Krause, J. Clark, and D. M. Hannah, “Low-Cost Environmental Sensor Networks: Recent Advances and Future Directions,” Frontiers, Aug. 13, 2019. https://www.frontiersin.org/articles/10.3389/feart.2019.00221/full

[10]Hindawi, M. Madhiarasan, and M. Louzazni, “Analysis of Artificial Neural Network: Architecture, Types, and Forecasting Applications,” Analysis of Artificial Neural Network: Architecture, Types, and Forecasting Applications, Apr. 18, 2022. https://www.hindawi.com/journals/jece/2022/5416722/

[11]A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, “Metaheuristic Algorithms in Modeling and Optimization,” *Metaheuristic Applications in Structures and Infrastructures*, pp. 1–24, 2013, doi: 10.1016/b978-0-12-398364-0.00001-2.

[12]D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso, “Preprocessing techniques for context recognition from accelerometer data,” Personal and Ubiquitous Computing, vol. 14, no. 7, pp. 645–662, Mar. 2010, doi: 10.1007/s00779-010-0293-9.

[13]J. Lozano, C. Apetrei, M. Ghasemi-Varnamkhasti, D. Matatagui, and J. P. Santos, “Sensors and Systems for Environmental Monitoring and Control,” Journal of Sensors, vol. 2017, pp. 1–2, 2017, doi: 10.1155/2017/6879748

[14]Chetan Dwarkani M, Ganesh Ram R, Jagannathan S, and R. Priyatharshini, “Smart farming system using sensors for agricultural task automation,” 2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR), Jul. 2015, Published, doi: 10.1109/tiar.2015.7358530.

[15] H. Ramamurthy, B. S. Prabhu, R. Gadh, and A. M. Madni, “Smart Sensor Platform for Industrial Monitoring and Control,” IEEE Sensors, 2005., Published, doi: 10.1109/icsens.2005.1597900.

[16]M. A. Ramírez-Moreno et al., “Sensors for Sustainable Smart Cities: A Review,” Applied Sciences, vol. 11, no. 17, p. 8198, Sep. 2021, doi: 10.3390/app11178198.

[17] J. Kennedy and R. Eberhart, “Particle swarm optimization,” Proceedings of ICNN’95 - International Conference on Neural Networks, Published, doi: 10.1109/icnn.1995.488968.

[18]V. Sharma and A. K. Tripathi, “A systematic review of meta-heuristic algorithms in IoT based application,” Array, vol. 14, p. 100164, Jul. 2022, doi: 10.1016/j.array.2022.100164.

[19]G. Venter and J. Sobieszczanski-Sobieski, “Particle Swarm Optimization,” AIAA Journal, vol. 41, no. 8, pp. 1583–1589, Aug. 2003, doi: 10.2514/2.2111.

[20]S. H. Chagas, J. B. Martins, and L. L. de Oliveira, “Genetic Algorithms and Simulated Annealing optimization methods in wireless sensor networks localization using artificial neural networks,” 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), Aug. 2012, Published, doi: 10.1109/mwscas.2012.6292173.

[21]N. Thi My Binh, A. Mellouk, H. Thi Thanh Binh, L. Vu Loi, D. Lam San, and T. Hai Anh, “An Elite Hybrid Particle Swarm Optimization for Solving Minimal Exposure Path Problem in Mobile Wireless Sensor Networks,” Sensors, vol. 20, no. 9, p. 2586, May 2020, doi: 10.3390/s20092586.

[22]“NodeMCU ESP8266 Specifications, Overview and Setting Up,” Make-It.ca, Sep. 15, 2021. https://www.make-it.ca/nodemcu-details-specifications/

[23]“DHT11–Temperature and Humidity Sensor,” Components101. https://components101.com/sensors/dht11-temperature-sensor

[24]“A Guide to Understanding LiPo Batteries — Roger’s Hobby Center,” Roger’s Hobby Center. https://www.rogershobbycenter.com/lipoguide

[25] Raubenheimer, “Excel-lence in Data Visualization?,” Data Visualization and Statistical Literacy for Open and Big Data, pp. 153–193, doi: 10.4018/978-1-5225-2512-7.ch007.

[26]E. Wilde, “Hypertext Transfer Protocol (HTTP),” *Wilde's WWW*, pp. 53–135, 1999, doi: 10.1007/978-3-642-95855-7_4.

[27]M. Ródenas García et al., “Review of low-cost sensors for indoor air quality: Features and applications,” Applied Spectroscopy Reviews, vol. 57, no. 9–10, pp. 747–779, Jun. 2022, doi: 10.1080/05704928.2022.2085734.