AI in Sustainable Energy and Environment

Journal homepage: aisesjournal.com

Online

Employing Four Nature-Inspired Algorithms Hybridized with MLP for Accurate Occupancy Detection of an Office Room

Loke Kok Foong [1,2,3] Wan Amizah Wan Jusoh[4] Vellapandian Ponnusamy[5]

1. Institute of Research and Development, Duy Tan University, Da Nang, Vietnam
2. School of Engineering & Technology, Duy Tan University, Da Nang, Vietnam
3. Youth Skills Development Division, Ministry of Youth and Sport, Menara KBS, 62570, Putrajaya, Malaysia
4. Department of Civil Engineering Technology, Universiti Tun Hussein Onn Malaysia, 86400 Pagoh, Muar, Johor
5. Institute for Youth Research Malaysia (IYRES), Ministry of Youth and Sport, Menara KBS, 62570, Putrajaya, Malaysia

Article Info

Received 1 January 2025

Received in Revised form 1 March 2025

Accepted 5 March 2025

Published online 7 March 2025

DOI:

Keywords

Artificial intelligence

Metaheuristic algorithms

Occupancy detection

Smart building

Abstract

This paper investigates the performance of four optimization algorithms—Multi-Verse Optimization (MVO), Teaching-Learning-Based Optimization (TLBO), Wind-Driven Optimization (WDO), and Whale Optimization Algorithm (WOA)—in optimizing Multilayer Perceptron (MLP) models for occupancy detection in office environments. The algorithms were tested across a range of swarm sizes, and their performance was evaluated using statistical indices such as Area Under the Curve (AUC), with a particular focus on the training and testing phases. The results demonstrate that MVO-MLP, with a swarm size of 400, achieved the highest performance, yielding AUC scores of 0.9956 for training and 0.9955 for testing. TLBO-MLP (swarm size 300) followed closely, with AUC scores of 0.9929 for training and 0.9851 for testing. WOA-MLP, using a swarm size of 300, achieved AUC scores of 0.9942 for training and 0.9950 for testing. WDO-MLP, with the largest swarm size of 500, yielded AUC scores of 0.9829 for both training and testing. The optimization algorithms demonstrated strong generalization ability, with MVO-MLP ranking first in terms of total score and overall performance. This study highlights the potential of optimization algorithms in enhancing the accuracy of occupancy detection systems, providing a robust foundation for future developments in smart building technologies and energy management systems.

1. Introduction

The smart grid is necessary for creating innovative electrical networks with a flexible, effective, durable, and secure architecture. The smart grid paradigm may use the most recent developments in sensing, communication, and metering technologies to enable effective control of each power network unit. The building sector of the power system gets a lot of attention since it offers the most potential for energy savings and uses the most electricity. Numerous studies have demonstrated using occupancy data to enhance energy efficiency and lower building energy consumption [1, 2]. Agarwal, Balaji [3] suggested an HVAC (heating, ventilation, and air conditioning) control mechanism that monitors workplace occupancy to determine when to turn the system on or off. The simulation results showed that using the suggested approach decreased HVAC energy consumption by 10% to 15%. According to reports, utilizing a control ventilation strategy based on occupancy information can save the ventilation system up to 55% of its energy consumption [4, 5]. Leephakpreeda [6] suggested occupancy-based lighting management and demonstrated how the system's energy usage might be lowered by 35% to 75%. Data from PIR and RFID sensors were used by Scott, Brush [7] to estimate home occupancy trends for heating control. After that, the authors evaluated three different control strategies (based on the PreHeat, always-on, and scheduled algorithms) and calculated a possible 35% energy reduction.

✉ *Corresponding author: lokekokfoong@duytan.edu.vn (Loke Kok Foong)*

According to research by Yokoishi, Mitsugi [8], installing a network of PIR and illumination sensors for occupancy detection would save 3.5 hours of power for lighting each day in a campus room. Using a learning approach, Peng, Rysanek [9] implemented a technique for controlling the cooling system in six workplaces. The findings indicated that the suggested management approach may save energy by 20.3%. Therefore, to reduce strain on the power grid and preserve user comfort, Building Energy Management Systems (BEMS) is advised to make occupancy data easier to use [10, 11]. Physical elements other than occupancy, such as weather, architectural characteristics, and equipment efficiency, can also affect a building's power consumption behavior [12, 13]. Occupancy, a crucial component of human factors, characterizes the presence of occupants, their consumption patterns, and the condition of the interior environment. "Occupancy" is regarded as the initial level of occupant behavior modeling, according to [14]. Occupancy information may be used to illustrate three main features of the intended applications at different resolution levels. However, these components are difficult for individuals to control or modify throughout the building's usage [15]:

- Occupancy resolution: describes the number, kind, and identity of people and their presence or absence within a building's zone.
- Temporal resolution: shows how frequently events happen (in hours, minutes, and seconds).
- Spatial resolution: specifies the building's measurements and other important details, such as the number of rooms.

The methods and procedures used to estimate building occupancy statistics have been discussed in the literature. [2, 14, 16, 17] provided an overview and comparison of several occupancy data-gathering methods. The advantages and disadvantages of occupancy sensing systems and occupancy modeling techniques used in institutional buildings were investigated by Yang, Santamouris and Lee [18]. Mane and Narasimha Rao [19] provided an overview of the patterns and deficiencies in the field of occupancy sensing research. Labeodan, Zeiler [20] reviewed the technologies used for office building occupancy evaluation. The authors base their categorization on the spatial-temporal features of the occupancy-detecting systems as comparison criteria. Popular methods for identifying, counting, and monitoring building residents were briefly covered in Saha, Florita [21]. The occupancy estimates and detection systems were studied in [22, 23] according to the kind of sensor used. They also contrasted each sensor to assess its benefits and drawbacks for occupancy detection.

Artificial neural networks, often known as ANNs, have been used in several studies. Building occupancy data was detected by stacking a multiclass neural network for occupancy detection. Zuraimi, Pantazaras [24] used feed-forward neural networks (FFNNs) with CO_2 data to predict a theater's occupancy. According to the study's findings, the ANN's average accuracy was 70%. In contrast, [25, 26] reported on a test bed for an environmental sensor network and its use for occupancy detection in an office building. The authors of these papers claim that a neural network can identify occupants with a 75% accuracy rate. Two occupancy-detecting techniques were reported by Kraipeerapun and Amornsamankul [27]. For occupancy detection, the first technique used stacking with a multiclass neural network, while the second method used stacking in conjunction with a dual-output neural network. The accuracy range found in the validation results was 68.87% to 91.18%. They used FFNNs with data from many sensors by Yang, Li [28] to calculate occupancy. In the first of these tests, the accuracy of occupancy estimation by each sensor was examined together with the performance of a single and multi-layered ANN. According to the test findings, the detection rate varied between 62.41% and 97.97%. In the second study, the researchers used artificial neural networks (ANNs) to count and identify the number of individuals inhabiting an office building. They reported an accuracy range of 92.5% to 97.1% for binary occupancy recognition and an RMSE of 0.139 to 0.758 for occupancy number estimate. Villariba [29] provided an example of how to apply and assess a feed-forward neural network to identify a home's occupancy using electricity usage data. Data from three residences was gathered to validate the experiment. The ANN was trained using a feature set that included temporal and statistical analyses of electrical data. The outcomes showed that this method could accurately detect 80.98% to 91.36% of house occupancy circumstances. Hobson, Lowcay [30] estimated the population of an academic office building using many sensors and ANN configurations, averaging an 83% coefficient of determination (R2). They illustrated how useful connected WiFi equipment may be in figuring out population density.

Masood, Soh and Chang [31] suggested using

wrapper model feature selection and extreme learning machines (ELM) to determine the number of persons in a classroom. The authors claim that their suggested approach performed better than filter approaches, with an accuracy range of 74.06% to 81.37%. Another comparable strategy was put out by Masood, Soh and Jiang [32]. Based on environmental factors, the authors assessed occupancy using a hybrid feature selection approach and wrapper. The accuracy of occupancy predictions for experimental data gathered in an office setting ranged from 75.63% to 79.17% for the WRANK-ELM approach and 76.88% to 77.92% for the RIG-ELM methodology. In an independent study, Masood, Jiang and Soh [33] suggested combining ELM with a hybrid filter-wrapper feature selection method to estimate the range of occupants. According to the experimental results, the HFS-ELM technique was performed with an accuracy ranging from 78.12% to 81.67% for the estimation of the occupancy range and from 94.37% to 98.12% for presence detection. Zou, Lu [34] introduced an online sequential extreme learning machine (OS-ELM) based indoor localization technique. The tests carried out by the researchers in a 580 m2 lab showed that OS-ELM was capable of 1.973 m object localization. Furthermore, WinOSS, a unique WiFi-based non-intrusive occupancy detection system, was introduced by Zou, Jiang [35]. WinOSS employed OS-ELM, a machine-learning localization technique based on fingerprinting. Installed over a 1500 square meter floor in a commercial building, the gadget achieved a 1.385-meter localization accuracy. It also showed that the system could successfully count and identify the occupants. Ertuğrul, Kaya and EminTağluk [36] demonstrated using a recurrent extreme learning machine (RELM) to determine building occupancy. The authors compared the effectiveness of RELM and other techniques from published research. The dataset created by Candanedo and Feldheim [37] was used for experimental validation. According to the comparison study's findings, RELM fared better than the other approaches, with an accuracy range of 99.02% to 99.58%. A feature-scaled extreme learning machine (FS-ELM) method was developed by Jiang, Masood [38] to determine the occupancy count in real time based on CO_2 data. Experimental data was collected in a 35-person office environment to evaluate the suggested strategy. The findings demonstrate an accuracy of up to 94% and a four-occupant tolerance.

The primary objective of this study is to propose a novel and inventive methodology that combines machine learning (ML) and artificial intelligence (AI) techniques in a way that complements each other to provide accurate occupancy recognition in office environments. Because we understand how crucial occupancy detection is for optimizing resource consumption, energy efficiency, and overall system performance, our study attempts to push the bounds of current technology. By employing evolutionary algorithms, we want to increase the system's ability for adaptability and optimization. The main goal is to efficiently anticipate and categorize occupancy situations using advanced machine learning algorithms and the power of data collected from several sensors, such as motion and infrared sensors. Accurate occupancy detection is crucial in smart settings, and our novel technique offers a practical way to develop intelligent building automation and management systems. We broaden our analysis to encompass four more methodologies—MVO, TLBO, WDO, and WOA—to further explore optimization techniques for occupancy detection. Every one of these approaches has a different angle for improving Multilayer Perceptron's (MLP) ability to accurately classify office space occupancy. MVO utilizes many search spaces to improve exploration, taking its cue from the multiverse idea. By simulating the teaching and learning process in a classroom, TLBO promotes information sharing among algorithm participants. While WOA mimics the social behaviors of humpback whales to enable effective exploration and exploitation of the solution space, WDO uses the mechanics of wind flow to direct the optimization process. The next sections thoroughly examine these techniques and highlight how they may be used to optimize MLPs for occupancy detection. We want to use these algorithms' special abilities to optimize the neural network for precise occupancy detection by combining them with MLP. The performance of each approach is thoroughly examined in the following sections, providing insight into the methods' usefulness and contributions to occupancy detection in offices.

This study introduces a novel integration of four advanced optimization algorithms—Multi-Verse Optimization (MVO), Teaching-Learning-Based Optimization (TLBO), Wind-Driven Optimization (WDO), and Whale Optimization Algorithm (WOA)—with Multilayer Perceptron (MLP) models to optimize occupancy detection in office environments. Unlike traditional methods, this approach leverages nature-inspired algorithms to efficiently tune MLP hyperparameters, resulting in higher accuracy and robustness in predicting

occupancy patterns. The use of MVO-MLP, with its superior performance, showcases the potential of metaheuristic optimization in improving the reliability of occupancy detection systems. The study is significant for smart building and energy management systems, as accurate occupancy prediction can optimize energy usage, enhance comfort, and reduce operational costs in real time. By exploring multiple optimization algorithms, this work offers a comparative analysis, revealing insights into the strengths and weaknesses of each method, thereby paving the way for more efficient and adaptable occupancy detection solutions in the future.

2. Established database

A workspace of around 5.85 x 3.50 x 3.53 meters (W x D x H) was used to assess the temperature, humidity, light, and CO_2 levels (Figure 1). The data was obtained using a microcontroller. A ZigBee radio attached to the recording station was used to transmit the data. A digital camera was utilized to ascertain whether the room was inhabited. The camera took a time-stamped photo every minute, which was manually examined to confirm the data. Using the timestamp from the data, this work calculates the amount of seconds that pass between midnight every day (NSM). Another way to utilize the date stamp is to classify it as a weekend (0) or a weekday (1); this variable is called Week Status (WS). An extra feature or variable in the data model is the humidity ratio (W). The recorded temperature and relative humidity were used to determine the humidity ratio, represented in kgw/kgda. The data was collected in Mons, Belgium, on February 2-3, during the winter. Hot water radiators kept the room's temperature above 19°C and supplied warmth. Examined data sets with open and closed office doors show the range of occupancy detection accuracy provided by the models. The measurements were taken three or four times a minute or every 14 seconds. The pertinent minute's average of the findings was calculated. The following repository will house the data sets and data processing scripts so that researchers may replicate the findings and eventually enhance model comparison or accuracy detection:

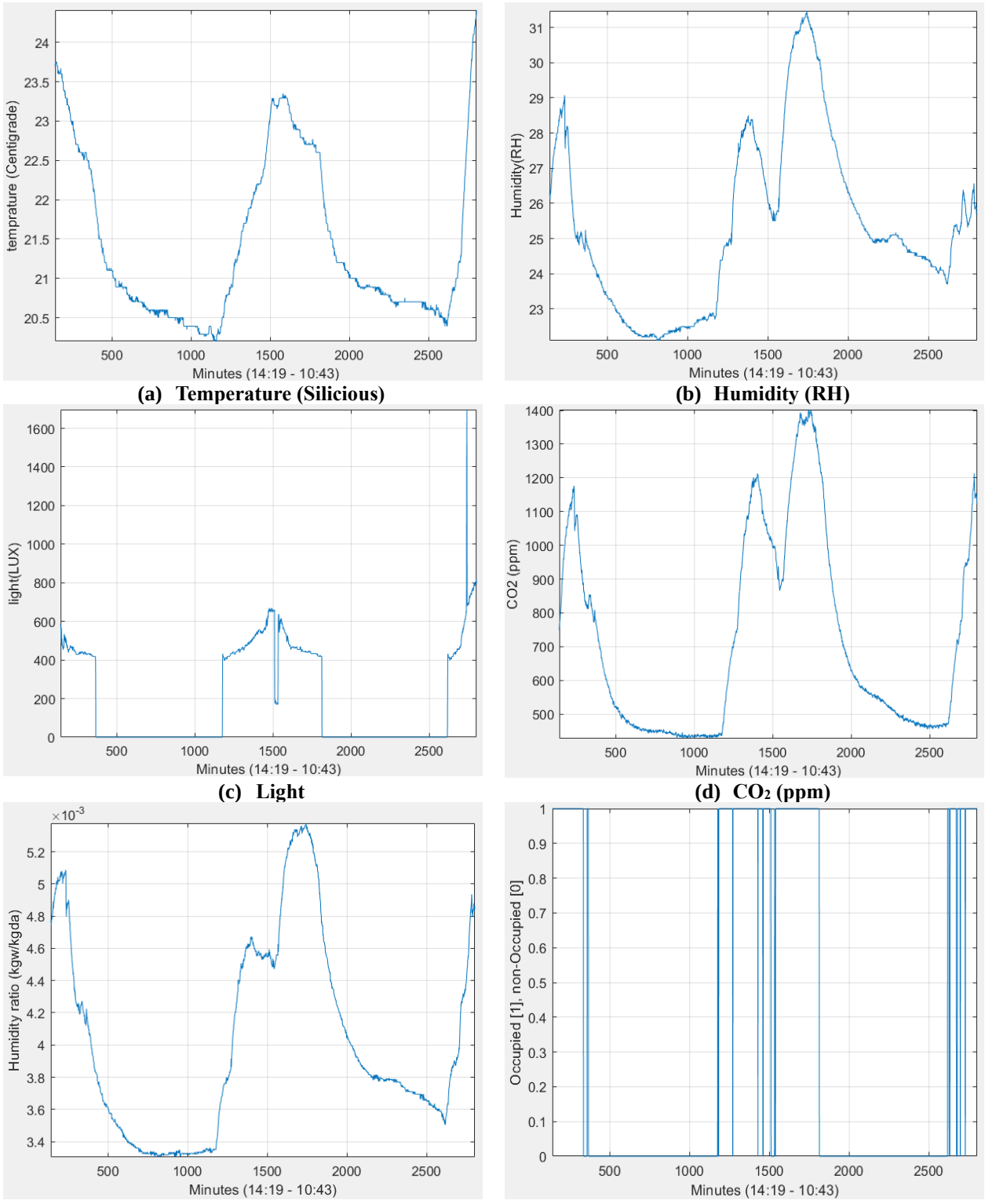
https://github.com/LuisM78/Occupancy-detection-data

The process of splitting the dataset into training and testing sets is a crucial step in machine learning, ensuring that the model is evaluated fairly and accurately. In this study, the dataset used for occupancy detection in an office environment was divided into two parts: the training set and the testing set. The training set is used to train the model, i.e., to allow the model to learn the relationships between the input features and the output labels (occupancy status). The testing set, on the other hand, is kept separate and is used to evaluate the model's performance on unseen data. This ensures that the model does not overfit the training data and can generalize well to new, unseen data.

For the data split, a common approach was followed where a significant portion (usually around 70%-80%) of the total dataset is allocated to training the model, while the remaining portion (typically 20%-30%) is reserved for testing. The split ensures that the model has enough data to learn the patterns while also providing an unbiased evaluation. The training and testing sets are typically selected randomly to prevent any bias, though techniques such as k-fold cross-validation could also be used for more robust evaluation. In this study, the training and testing sets were selected using standard random splitting, ensuring that each set had a representative sample of the data for accurate model evaluation. The use of separate training and testing datasets is critical for assessing the true performance of the model and ensuring its effectiveness in real-world applications.

3. Methodology

Processing data is the first step in the proposed technique. Multi-collinearity analysis is the second step. Feature selection is the third step. Evolutionary algorithm optimization is the fourth step. Building the occupancy detection map is the fifth step. Testing and model comparison (linear regression analysis) is the sixth step. To facilitate the use of evolutionary algorithms, maps and geographical data about the susceptibility of landslides were created using ArcGIS [39].



(a) Temperature (Silicious) **(b) Humidity (RH)**

(c) Light **(d) CO_2 (ppm)**

(e) Humidity ratio (kg_w/kg_{da}) **(f) Occupied [1], non-Occupied [0]**

Figure 1: Parameters in the dataset

3.1. Artificial neural network

The best way to characterize ANNs is as supervised machine learning models that can use a collection of inputs and known data points to potentially categorize unknown data into groups. An information processing system (ANN) is thought to provide knowledge based on a learning mechanism akin to human learning [40-42]. ANNs acquire new skills by analyzing historical examples and storing the results in a data structure of created probability-weighted network correlations. The most popular ANN design consists of layers of

neurons linked by correlation weights that convert input data into predefined outputs using a sophisticated non-linear function. The feedforward Multilayer Perceptron neural network model that we used in our experiment consisted of three layers of neurons: an input layer, a hidden layer, and an output layer. Each input occupancy-related parameter was represented by a single neuron in the input layer; several neurons in the buried layer made the input neurons more complex. One neuron in the output layer represented the conclusion of the process, which, in our instance, decided whether an office space was occupied. An ANN can learn new information by determining the difference between the intended result and its forecast. Three layers of neurons made up the feedforward Multilayer Perceptron neural network model that we employed in our investigation: an input layer, a hidden layer, and an output layer. A single neuron in the input layer represented every input occupancy-related parameter. The input neurons' increased complexity results from many neurons in the buried layer. The output layer consisted of a single neuron representing the process completion, which, in our case, determined if an office space was occupied. Finding the discrepancy between the expected and predicted results can be used to train an artificial neural network.

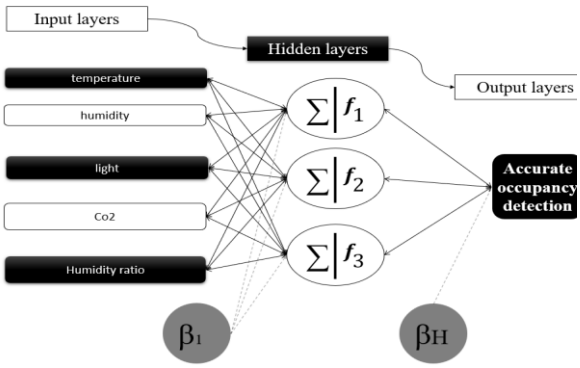


Figure 2. Three layers: input, hidden, and output of an MLP.

3.2. Hybrid model development

Thirty percent of the data were randomly allocated to the training phase, and thirty percent were assigned to the testing phase to assess the models' performance. The mean absolute error (MAE) and mean squared error (RMSE) of the models' performance indices are then used. The following is an explanation of certain statistics indicators:

$$\mathrm{MSE} = \frac{\sum_{k=1}^{S}(P_k - T_k)^2}{S} \quad (1)$$

$$\mathrm{MAE} = \frac{\sum_{k=1}^{S}|P_k - T_k|}{S} \quad (2)$$

S is the total number of training or testing samples, and *P* and *T* are the anticipated and target values.

Another crucial thing to remember is that MATLAB was used to build all the code.

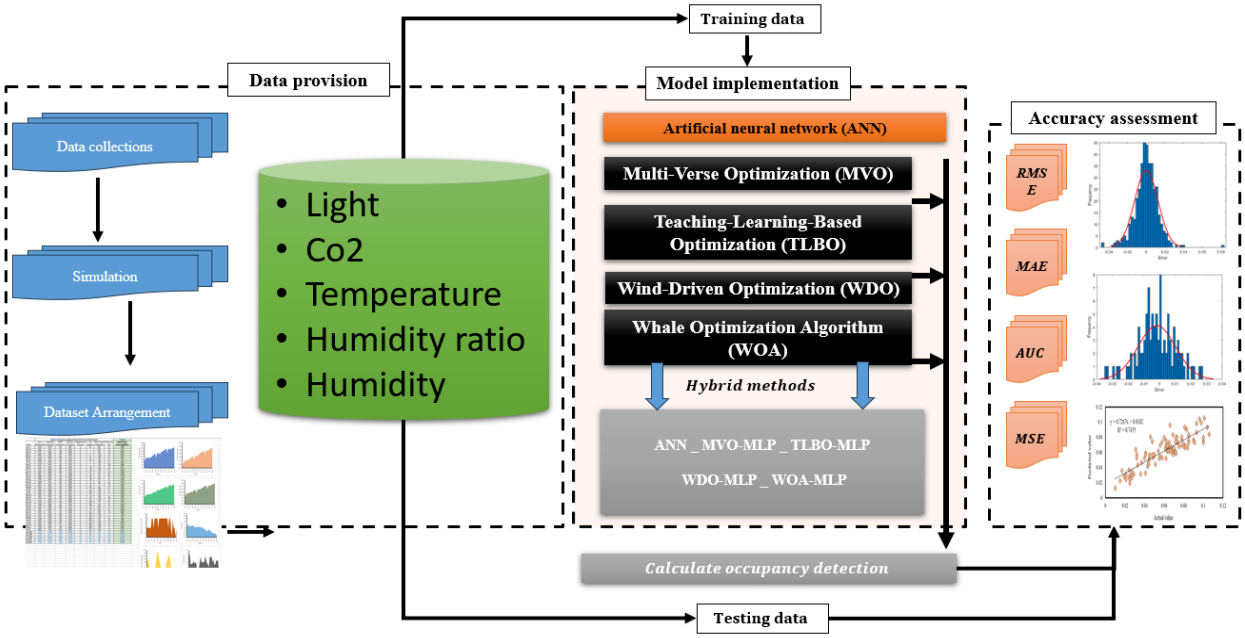


Figure 3. An overview of the process for hybrid modeling.

3.3. Multi-Verse Optimization (MVO)

Over the past few decades, metaheuristic algorithms have gained popularity for solving difficult combinatorial optimization problems [43-49]. A novel metaheuristic algorithm called the MVO has been presented based on the ideas of the multiverse in physics. The academic community has taken notice of this new trend [50]. In the context of the MVO framework, each element represents a variable that helps determine the optimal solution, and the solution to the optimization problem is found when all the components are added together. An inflation rate that is precisely proportionate to the total value of the objective function that the solution represents has been assigned to each component. The notion of swapping components in several locations, comparable to cosmological wormholes, black and white holes, inspires the MVO search method. Three factors are taken into account by MVO while searching: the population size, represented by Q; the likelihood that wormholes exist, represented by p_w; and the trip distance rate, represented by r. Further information in greater detail may be found in [50].

Initializing the algorithm's parameters— Q, p_w, $\tilde{r}$, and the halting condition—is the first stage in the process. After that, a population of universes is generated, each examined in detail. When i = 1, 2, . . ., Q, the expression $X_i = (x_{i1}, x_{i2}, \dots, x_{iM})$ is used to denote a particular universe inside the population.

$$x_{ij} = L_j + (U_j - L_j) \cdot r, \quad j = 1, 2, \dots, M \tag{3}$$

Both the "upper" and "lower" boundaries of the "jth variable" are designated inside the parameters of this particular research project by the notations "U_j" and "L_j," respectively. M is the total number of variables, while r is a randomly selected integer from the interval [0, 1]. It's crucial to update the universe set before moving on to Step 3 by completing the following sub steps in the right order: In Step 3.1 of the process, objects are transferred via white and black hole tunnels using an algorithm called "Object exchange in S.F.". In Step 3.2 of the algorithm, named "S.F. Object exchange through a wormhole tunnel," objects can be exchanged through a wormhole tunnel. The universe must be the most advantageous thus far, updated in Step 4. Not to mention, Step 5 only allows for creating the finest possible world under specific halting conditions. The procedure has to be started from Step 3, and further updates must be applied if these requirements are not satisfied.

3.4. Teaching-Learning-Based Optimization (TLBO)

The TLBO method for teaching and learning optimization was first presented in [51-55]. The way the TLBO algorithm works is by taking into account how a teacher affects the academic achievement of pupils in a classroom. This algorithm replicates the instructional procedure involving students and a teacher. This algorithm's two main learning modes are the teacher phase, where students learn from their instructor, and the learner phase, where students learn through peer interactions. The two basic components of this algorithm are learners and a teacher. The results of this optimization method are represented by the academic performance or grades of the students, which are impacted by the instructor's quality of instruction. As a result, educators are seen as intelligent people who work hard to help their pupils get better grades or test results. Additionally, peer-to-peer learning can benefit students and improve their academic performance.

A group of persons engaged in the learning process are involved in a technique called teaching-learning-based optimization. In this approach, learners are viewed as the population and provided with design factors to gather data from them as subjects. The final results of learners are comparable to the fitness value of the optimization problem. This method also considers the instructor, who can probably choose the student who responds best. The Learner and Teacher phases are the two separate stages of the effort required in teaching-learning-based optimization. The following is a list of the specific duties associated with each phase.

a) Teacher phase

The first step of the TLBO algorithm is a teacher guiding pupils. The instructor's goal in this phase is to get the class's mean performance up to their standard (T_A) from the arbitrary value M_1. However, doing so would be impossible and require the teacher's abilities. As a result, the class mean, M_1, may only be moved by a teacher to a number, M_2, that is higher than M_1. T_i will try to raise M_j to their level at any iteration i, where T_i stands for the teacher and M_j for the current mean. This produces a new mean, M_{new}, whose difference from the existing mean is given by a particular equation [53].

$$\text{Difference\_Mean}_i = r_i(M_{new} - T_F M_j) \tag{4}$$

The formula above shows that r_i is a random number in the range of 0 to 1, and that is the

teaching factor determining the degree of the mean's value fluctuation. As a heuristic step, T_F can have a value of 2 or 1, randomly selected with an equal chance.

$$T_F = \text{round}[1 + \text{rand}(0,1)\{2 - 1\}] \quad (5)$$

The method will generate a teaching factor at random during execution, which will be between 1 and 2. A score of one signifies that knowledge has not improved, but a score of two implies that all of the material has been transmitted. Values in the intermediate range suggest that different quantities of knowledge are transferred depending on the learner's capacities. Although the 1-2 range values were attempted to be used in this article, no difference in the results was seen. Therefore, it is advised that the teaching factor round up to 2 or 1 to simplify the process. However, any number between 1 and 2 might still be considered a potential teaching aspect.

The following equation, which considers the same variance in mean, modifies the current result. There is no difference in the number of sentences that result:

$$X_{new,i} = X_{old,i} + \text{Difference\_Mean}_i \quad (6)$$

b) Learner phase

This part deals with the second stage of the algorithm, in which students add to their knowledge by interacting with one another. To increase their knowledge, students interact randomly with other students and absorb new material if the other student knows more. This phase's learning process may be mathematically expressed as follows.

X_i and X_j are two distinct learners, where $i \neq j$, and at any given iteration i,

$$X_{new,i} = X_{old,i} + r_i(X_i - X_j) \quad \text{if } f(X_i) < f(X_j) \quad (7)$$

$$X_{new,i} = X_{old,i} + r_i(X_j - X_i) \quad \text{if } f(X_j) < f(X_i) \quad (8)$$

The admission of a new variable X_{new} in the context of teaching-learning-based optimization is contingent upon its superior function value. Six steps can be used to summarize the implementation process. First, use random generation to initialize and assess the population and design variables. Second, as a teacher, choose the top student for each topic and determine the average performance of students in each subject. Thirdly, use a teaching factor (TF) to calculate the variance between the current and optimal mean results. Fourth, use an equation (6) to update students' knowledge with that of their professors. Fifth, use equations (7) and (8) to share information with other students to keep their knowledge current. Lastly, continue from steps 2 through 5 until the termination requirements are met.

3.5. Wind-Driven Optimization (WDO)

Bayraktar, Komurcu and Werner [56] created the WDO metaheuristic method. The algorithm is built based on the flow of air parcels. Four forces—gravitational force (FG), pressure gradient force (FPG), Coriolis force (FC), and frictional force—are involved in this task's completion (FF). Air packets are believed to have no dimensions and weight to make things easier to work out. The force resulting from the pressure gradient is represented in Equation (9) when P and δV represent the air volume and pressure gradient, respectively. FF counteracts the air movement of FPG (Equation (10)). To deliver packages to the Earth's core, FG (Equation (11)) is in charge. Air parcel motion deflections can be attributed to FC (Equation (12)).

$$\overrightarrow{F_{PG}} = -\nabla P.\delta V \quad (9)$$

$$\overrightarrow{F_F} = -\rho\,\alpha\,\vec{u} \quad (10)$$

$$\overrightarrow{F_G} = \rho.\delta V.\vec{g} \quad (11)$$

$$\overrightarrow{F_C} = -2\theta \times \vec{u} \quad (12)$$

where α is a frictional coefficient, θ is the earth's rotation, $\vec{u}$ is the wind velocity vector, ρ represents the density of a short air parcel, and g is the gravitational constant.

When the forces above are included in the ideal gas equation, the following outcome is obtained [57]:

$$\overrightarrow{\nabla u} = g + \left(-\nabla P.\frac{RT}{P_{cur}}\right) + (-\alpha\vec{u}) + \left(\frac{-2\theta \times \vec{u}RT}{P_{cur}}\right) \quad (13)$$

Because air velocity is related to pressure, a rise in pressure causes the velocity to alter. Consequently, we have had to modify Equation (13). The packages are arranged according to the least pressure applied. To update location and velocity, apply the following equations, where i stands for rank:

$$\overrightarrow{U_{new}} = (1-\alpha)\vec{u}_{cur} - gx_{cur} + \left(\left|1 - \frac{1}{i}\right|.(x_{opt} - x_{cur})RT\right) + \left(\frac{C.\vec{U}_{otherdirection}}{i}\right) \quad (14)$$

$$\overrightarrow{X_{new}} = \overrightarrow{X_{old}} + \overrightarrow{U_{new}} \quad (15)$$

Where x is the air parcel location, $\vec{u}_{cur}$ and

$\overrightarrow{u_{new}}$ are allocated to the velocity of the current and upcoming iterations, and x_{opt} and x_{cur} are the optimal and current positions. Concurrently, $C = -2RT$ and $\vec{U}_{otherdirection} = \overrightarrow{F_C}$.

The updating process is finished when the predefined objective function (OF) or number of repeats is attained. The corresponding settings are applied to the air parcel with the lowest OF. Previous research has produced further information [56, 58, 59].

3.6. Whale Optimization Algorithm (WOA)

The whale optimization algorithm (WOA) was created by Australians Mirjalili and Lewis [60] as a novel use of swarm intelligence to enhance optimization. It was designed to resemble how whales hunt in the wild: they locate groups of whales, encircle them, and attack. The search procedure is optimized by prey and other parameters. The peculiar way that humpback whales hunt is what makes them so extraordinary. We call this practice "bubble-net feeding." Humpback whales prefer to hunt krill or tiny schools of fish near the surface. It has been discovered that the pattern in which these bubbles are formed resembles a circle or a "9." Before 2011, this tendency was solely assessed using surface observations. Nevertheless, using tag sensors, researchers investigated this behavior. They acquired 300 feeding encounters from nine humpback whales equipped with bubble-net tags. Two motions linked to bubbles were identified and labeled as "double loops" and "upward spirals." In the former action, humpback whales descend to a depth of around 12 meters, encircle their prey with a swirling bubble, and then rise again. The coral loop, lobtail, and catch loop are the three stages of the last movement. A great deal of information is accessible on these acts. The whale emits many bubbles near the prey and then executes a single, massive loop to capture it during the coral loop. During the lobtail, the whale sways its tail back and forth before lunging for its meal.

In terms of convergence rate and accuracy of solutions, WOA performs better. The BP neural network is utilized in this study to determine the parameters and structure of the network, and the whale algorithm is used to optimize the network. The weights of the BP neural network are then trained using the whale approach. The network immediately receives the learning result and uses it as training data. As a result, the whale algorithm's global random search power is preserved, the nonlinear and self-learning potential of the BP neural network is finally utilized, and the accuracy of credit card fraud detection is only slightly improved.

A unique heuristic optimization technique called the Whale Optimization Method (WOA) was motivated by humpback whale hunting. The following are the key algorithms:

- The surroundings of the victim

$$\begin{cases} \mathrm{D} = |\mathrm{CX}^*(\mathrm{t}) - \mathrm{X}(\mathrm{t})| \\ \mathrm{X}(\mathrm{t}+1) - \mathrm{X}^*(\mathrm{t}) - \mathrm{AD} \end{cases} \quad (16)$$

where A and C are coefficients, t is the current number of iterations, represents the whale's current location, and represents the current best fish position vector.

- The act of hunting

$$\mathrm{X}(\mathrm{t}+1) = \begin{cases} \mathrm{X}^*(\mathrm{t}) - \mathrm{AD}, \mathrm{p} < \mathrm{P_i} \\ \mathrm{X}^*(\mathrm{t}) + \mathrm{D_p e^{bl}} \cos(2\pi \mathrm{l}), \mathrm{p} \geq \mathrm{P_i} \end{cases} \quad (17)$$

The distance between the whale and the prey is represented by $D_p = |X^*(t) - X(t)|$, where b is a constant, l is a random value within (-1, 1), and P_i and $1- P_i$, respectively, pick the shrinkage bracketing mechanism and the selection spiral model for the set probability.

- Look for prey

$$\begin{cases} D = |CX_{rand} - X(t)| \\ X(t+1) = X_{rand} - AD \end{cases} \quad (18)$$

One of them, called rand X, is a random whale vector that, based on the position of the random whale, changes the positions of other whales. As a result, the whale moves away from its prey, improving its ability to find better prey, increasing the system's exploratory potential, and allowing the WOA algorithm to scan the globe.

4. Results and discussion

The MATLAB environment tests evaluates and simulates the study's model structures. Several networks with varying layers and types of neurons have been constructed to ascertain which architecture performs best. The accuracy of the approach is also affected by changes to the neuron count and the ANN layer. Many optimization methodologies begin with the initial optimization results. An increased AUC number corresponds to an increased AUC score. The subsequent subsections make use of the outcomes of these networks. More proof of the relationship between the Mean Squared Error (MSE) and the number of neurons of each type in each buried layer can be found in Figure 4. Interestingly, the ratings were based on the accuracy of the method's predictions. For instance, the model scores higher if it achieves

the lowest Mean Squared Error (MSE).

The first stage of discovery will provide the groundwork for subsequent optimization tactics. As a result, the following sections use these networks' outputs. Structures with lower mean squared errors (MSEs) yield more accurate predictions. Regression and classification issues can be more precisely resolved with the estimated values of the proposed model. To predict the vulnerability of highway construction, Figure 4 shows the variation in mean squared error (MSE) as a function of repetitions for the proposed MVO-MLP, TLBO-MLP, WDO-MLP, and WOA-MLP designs. The results show that 400, 300, 500, and 300 (N_{swarm}) were chosen by the MVO-MLP, TLBO-MLP, WDO-MLP, and WOA-MLP as the best options.

Figures 4 to 8 present the performance results of various optimization algorithms (MVO, TLBO, WDO, and WOA) integrated with Multilayer Perceptron (MLP) models for occupancy detection in office environments. Figure 4 compares the best-fit models for each optimization approach, highlighting how each algorithm tunes the MLP structure to achieve optimal performance. In Figures 5 to 8, the training and testing accuracies of the models are shown separately for each algorithm. For each optimization method, the training dataset accuracy is plotted first, followed by the testing dataset accuracy, allowing for a direct comparison of how well the models perform during the training phase versus their ability to generalize to new, unseen data.

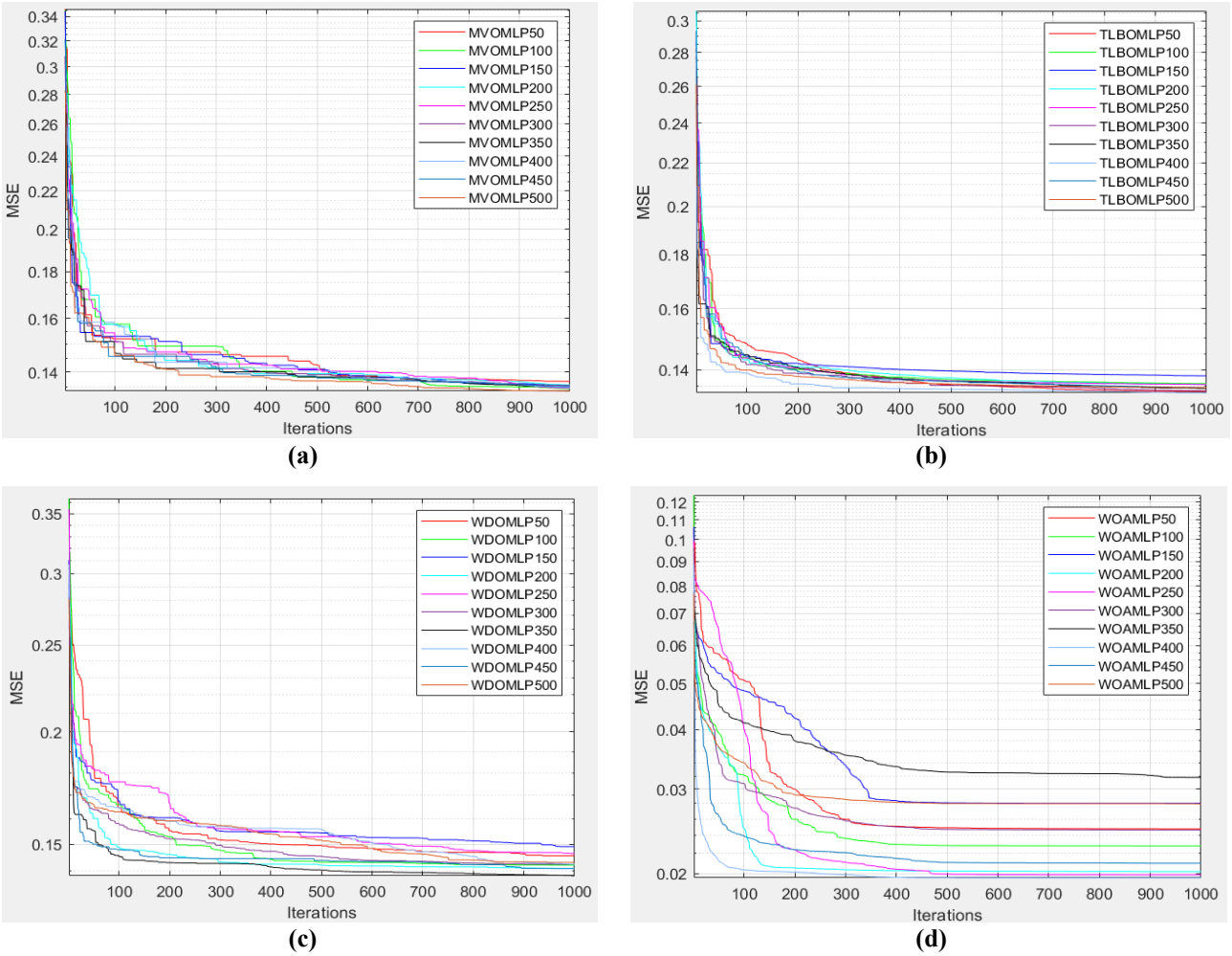
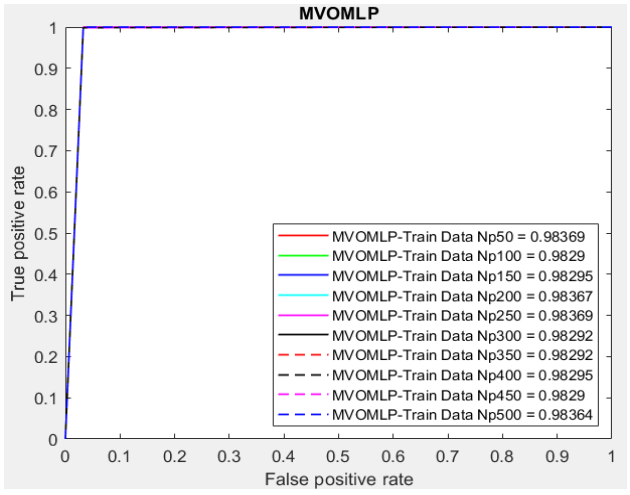


Figure 4: The model that fits the data the best (a) MVOMLP, (b) TLBOMLP, (c) WDOMLP, and (d) WOAMLP

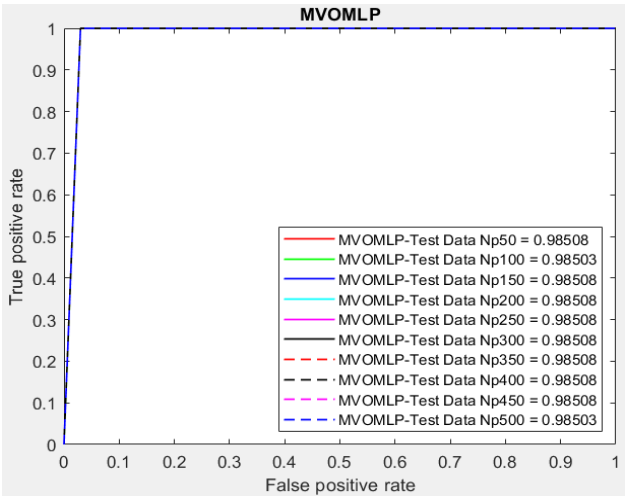
Each of the figures reveals the strengths and weaknesses of the different optimization approaches. The MVOMLP model (Figure 5) demonstrates robust training and testing accuracy, with particularly strong performance during both stages. The TLBOMLP model (Figure 6) exhibits similar trends, with the highest accuracy observed at a swarm size of 300. The WDOMLP (Figure 7) and WOAMLP (Figure 8) models also show favorable performance, with the WOAMLP achieving impressive testing accuracy across the board. These results collectively suggest that

optimization algorithms such as MVO, TLBO, WDO, and WOA can significantly enhance the performance of MLP models in occupancy detection, each offering unique strengths depending on the dataset and optimization parameters used.

Comparing the actual data with the hybrid design's projected quantities yields the second stage's results. Reader operating characteristic (ROC) curves are the most often used technique for determining the optimal hybrid design. As mentioned earlier, the graph illustrates how the diagnostic abilities of a binary classifier system vary with changes in the discriminating threshold. An AUC value of 1 indicates the best possible outcome, whereas 0 indicates no relationship. The actual zero value of an expected value is unrelated to it. The AUC assesses a classifier's ability to discriminate between many classes as a summary of ROC curves. AUC (Area Under the Curve) increases indicate how well the method can discriminate between positive and negative categories. Figures 5-8 display the ROC curves for the MVO-MLP, TLBO-MLP, WDO-MLP, and WOA-MLP techniques. The best forecasting method was developed for swarm sizes of 400, 300, 500, and 300 based on the results of the iteration phase (Table 1-4). The MVO-MLP, TLBO-MLP, WDO-MLP, and WOA-MLP methodologies are all incorporated. This result was obtained after 40,000 cycles of modeling and assessment using Mean Squared Error (MSE), Figures 5-8.

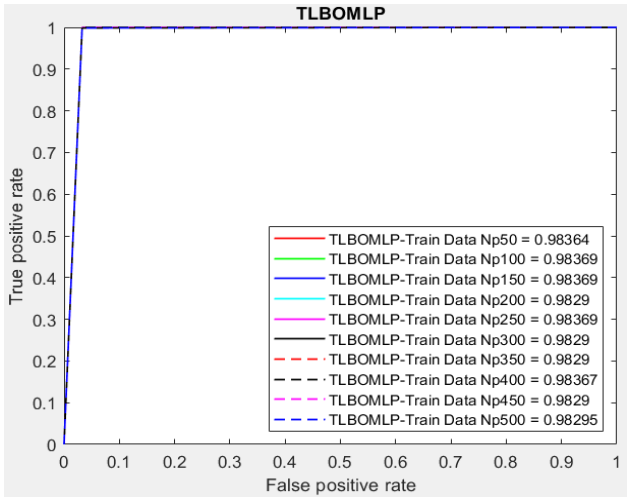


(a) MVOMLP- training datasets

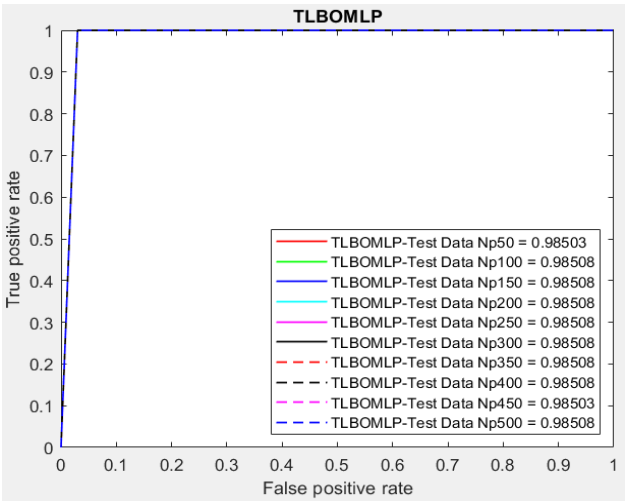


(b) MVOMLP-testing datasets

Figure 5: Accuracy results for the training dataset for several proposed MVOMLP structures.

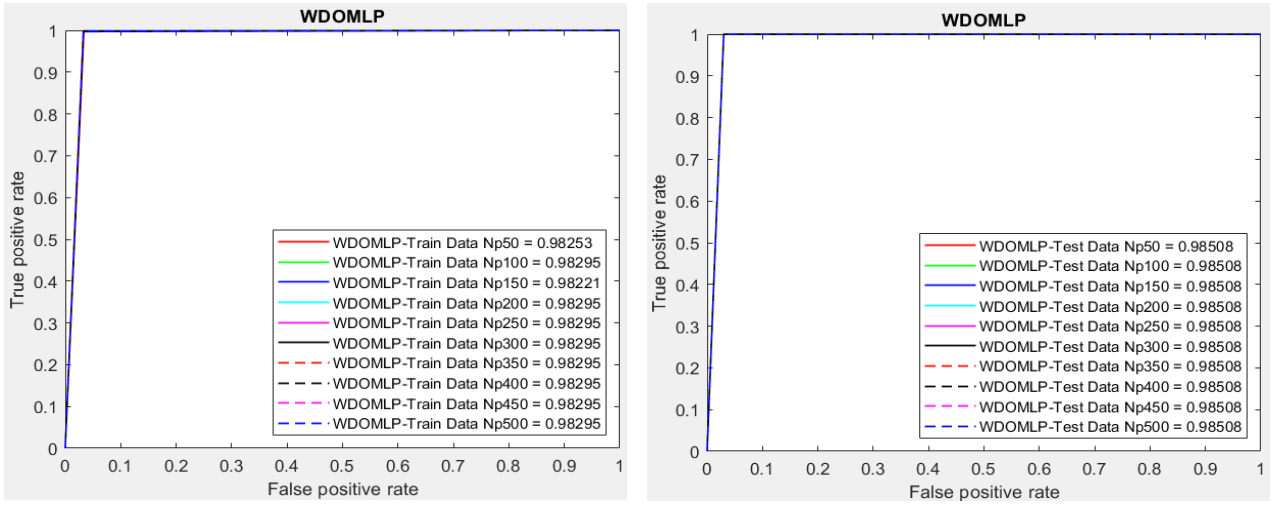


(a) TLBOMLP- training datasets



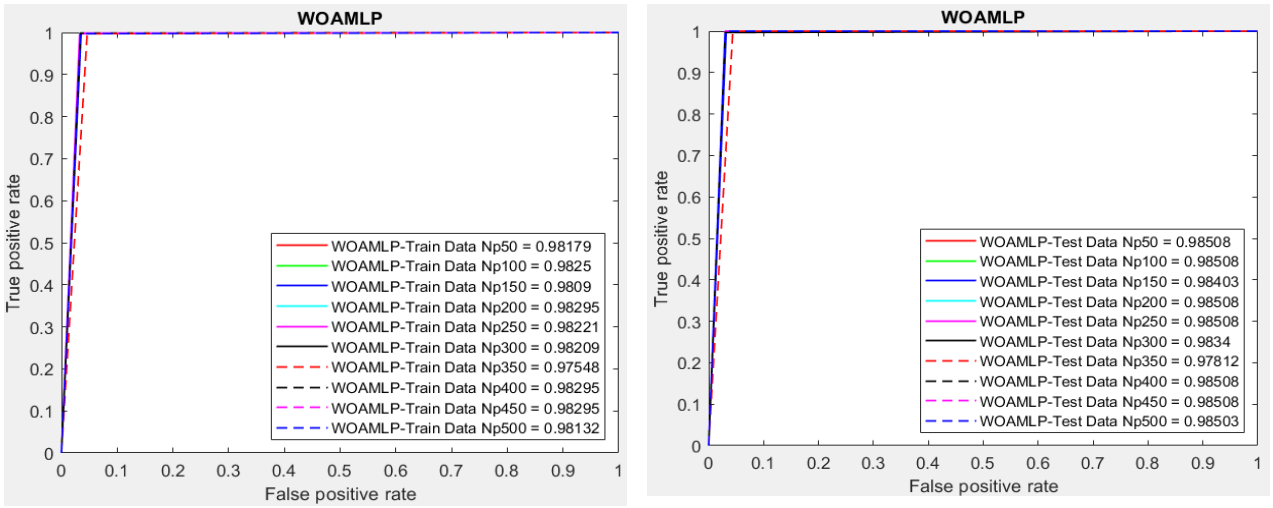
(b) TLBOMLP-testing datasets

Figure 6: Accuracy results for the training dataset for several proposed TLBOMLP structures.



(a) WDOMLP- training datasets **(b) WDOMLP-testing datasets**

Figure 7: Accuracy results for the training dataset for several proposed WDOMLP structures.



(a) WOAMLP- training datasets **(b) WOAMLP-testing datasets**

Figure 8: Accuracy results for the training dataset for several proposed WOAMLP structures

Table 1 presents the AUC statistical indices-based network findings for MVO integrated with Multilayer Perceptron (MVOMLP) across various planned swarm sizes. MVOMLP consistently achieves high AUC scores for both training and testing datasets across various swarm sizes, indicating the effectiveness of MVO in optimizing the Multilayer Perceptron (MLP) for accurate occupancy detection. Combining training and testing AUC scores, the scoring system results in diverse total scores for each configuration. Higher total scores are indicative of superior performance. Swarm size 400 achieves the highest score of 18, suggesting its effectiveness in training and testing scenarios. The model's performance exhibits sensitivity to changes in swarm size, with variations in AUC scores observed. Swarm size 400, despite achieving the highest total score, shows that larger population size in the MVO optimization process may lead to enhanced performance in terms of occupancy detection. MVOMLP demonstrates a relatively balanced performance between training and testing datasets. This balance is crucial for ensuring the model generalizes well to unseen data, minimizing the risk of overfitting. Swarm size 400 attains the top rank with the highest total score of 18, indicating superior performance. This suggests that larger population size in the MVO optimization process leads to enhanced performance in terms of occupancy detection. In summary, considering the correct criteria where higher ranks and total scores are desirable, Table 1 reaffirms the effectiveness of Multi-Verse Optimization when integrated with Multilayer Perceptron for occupancy detection. The consistently high AUC scores, the impact of

swarm size on performance, and the balanced training-testing performance collectively highlight the potential of MVO in optimizing MLPs for accurate occupancy detection in office environments, with a preference for larger population sizes.

Table 2 presents the AUC statistical indices-based network findings for TLBO with Multilayer Perceptron (TLBOMLP) across various swarm sizes. TLBOMLP consistently exhibits high AUC scores for training and testing datasets across swarm sizes. This indicates the efficacy of TLBO in optimizing the Multilayer Perceptron (MLP) for accurate occupancy detection. Considering both training and testing AUC scores, the scoring system results in diverse total scores for each configuration. Higher total scores indicate superior performance. Swarm size 300 achieves the highest total score of 13, demonstrating its effectiveness in training and testing scenarios. Swarm size 300 attains the top rank with the highest total score of 13, indicating superior performance. This suggests that larger population size in the TLBO optimization process leads to enhanced performance in terms of occupancy detection. TLBOMLP maintains a balanced performance between training and testing datasets, indicating the model's ability to generalize well to unseen data and avoid overfitting. The model's performance shows sensitivity to changes in swarm size, as reflected in the fluctuations of AUC scores. Larger swarm sizes, particularly 300 and 400, consistently demonstrate superior performance. In summary, Table 2 underscores the effectiveness of Teaching-Learning-Based Optimization when integrated with Multilayer Perceptron for occupancy detection. The consistent AUC scores, the impact of swarm size on performance, and the balanced training-testing performance collectively highlight the potential of TLBO in optimizing MLPs for accurate occupancy detection in office environments, with a preference for larger population sizes, especially 300 in this case.

Table 3 presents the AUC statistical indices-based network findings for WDO with Multilayer Perceptron (WDOMLP) across various planned swarm sizes. WDOMLP consistently achieves high AUC scores for training and testing datasets across swarm sizes. This suggests the efficacy of WDO in optimizing the Multilayer Perceptron (MLP) for accurate occupancy detection. Considering both training and testing AUC scores, the scoring system results in consistent total scores for each configuration. Higher total scores indicate superior performance. Notably, all swarm sizes achieve the same total score of 3, emphasizing the consistency of WDOMLP's performance. Swarm sizes from 100 to 500 all share the top rank with the same total score of 3, indicating that no particular swarm size outperforms the others according to this scoring system. WDOMLP demonstrates a balanced performance between training and testing datasets, indicating the model's ability to generalize well to unseen data and avoid overfitting. In this case, the model's performance does not exhibit sensitivity to changes in swarm size, as all swarm sizes achieve the same total score and rank. In summary, Table 3 underscores the consistent performance of Wind-Driven Optimization when integrated with Multilayer Perceptron for occupancy detection. The uniformity in AUC scores, the balanced training-testing performance, and the lack of sensitivity to swarm size collectively highlight the stability of WDOMLP across different population sizes for accurate occupancy detection in office environments.

Table 4 outlines the AUC statistical indices-based network findings for the WOA with Multilayer Perceptron (WOAMLP) across various planned swarm sizes. WOAMLP exhibits varied AUC scores for training and testing datasets across swarm sizes. This suggests that the WOA has an impact on optimizing the Multilayer Perceptron (MLP) for accurate occupancy detection, with different swarm sizes influencing the model's performance. Considering both training and testing AUC scores, the scoring system results in diverse total scores for each configuration. Higher total scores are indicative of superior performance. Swarm size 300 achieves the highest total score of 19, indicating its effectiveness in training and testing scenarios. Swarm size 300 attains the top rank with the highest total score of 19, indicating superior performance. This suggests that moderate population size in the WOA optimization process leads to enhanced performance in terms of occupancy detection. WOAMLP demonstrates a relatively balanced performance between training and testing datasets, indicating the model's ability to generalize well to unseen data and avoid overfitting. The model's performance shows sensitivity to changes in swarm size, as reflected in the fluctuations of AUC scores. Swarm sizes 300 and 400 consistently demonstrate superior performance, while smaller and larger swarm sizes show variations in their effectiveness. In summary, Table 4 emphasizes the impact of the Whale Optimization Algorithm when integrated with Multilayer Perceptron for occupancy detection. The varied AUC scores, the impact of swarm size

on performance, and the balanced training-testing performance collectively highlight the potential of WOAMLP in optimizing MLPs for accurate occupancy detection in office environments, with a preference for moderate population sizes, especially 300 in this case.

Table 1: Network discoveries based on AUC statistical metrics for many intended MVOMLP swarm sizes.

Population size	Network AUC results		Scoring		Total score	RANK
	Training	Testing	Training	Testing		
50	0.9947	0.9947	1	1	2	10
100	0.9955	0.9957	7	10	17	2
150	0.9953	0.9952	6	4	10	7
200	0.9950	0.9950	2	3	5	8
250	0.9957	0.9952	10	5	15	3
300	0.9953	0.9954	5	7	12	5
350	0.9952	0.9954	4	7	11	6
400	0.9956	0.9955	9	9	18	1
450	0.9951	0.9949	3	2	5	8
500	0.9955	0.9952	8	6	14	4

Table 2: Network discoveries based on AUC statistical metrics for many intended TLBOMLP swarm sizes.

Population size	Network AUC results		Scoring		Total score	RANK
	Training	Testing	Training	Testing		
50	0.9836	0.9850	5	1	6	6
100	0.9837	0.9851	6	3	9	2
150	0.9837	0.9851	6	3	9	2
200	0.9829	0.9851	1	3	4	7
250	0.9837	0.9851	6	3	9	2
300	0.9929	0.9851	10	3	13	1
350	0.9829	0.9851	1	3	4	7
400	0.9837	0.9851	6	3	9	2
450	0.9829	0.9850	1	1	2	10
500	0.9829	0.9851	1	3	4	7

Table 3: Network discoveries based on AUC statistical metrics for many intended WDOMLP swarm sizes.

Population size	Network AUC results		Scoring		Total score	RANK
	Training	Testing	Training	Testing		
50	0.9825	0.9851	1	1	2	10
100	0.9829	0.9851	2	1	3	1
150	0.9829	0.9851	2	1	3	1
200	0.9829	0.9851	2	1	3	1
250	0.9829	0.9851	2	1	3	1
300	0.9829	0.9851	2	1	3	1
350	0.9829	0.9851	2	1	3	1
400	0.9829	0.9851	2	1	3	1
450	0.9829	0.9851	2	1	3	1

500	0.9829	0.9851	2	1	3	1

Table 4: Network discoveries based on AUC statistical metrics for many intended WOAMLP swarm sizes.

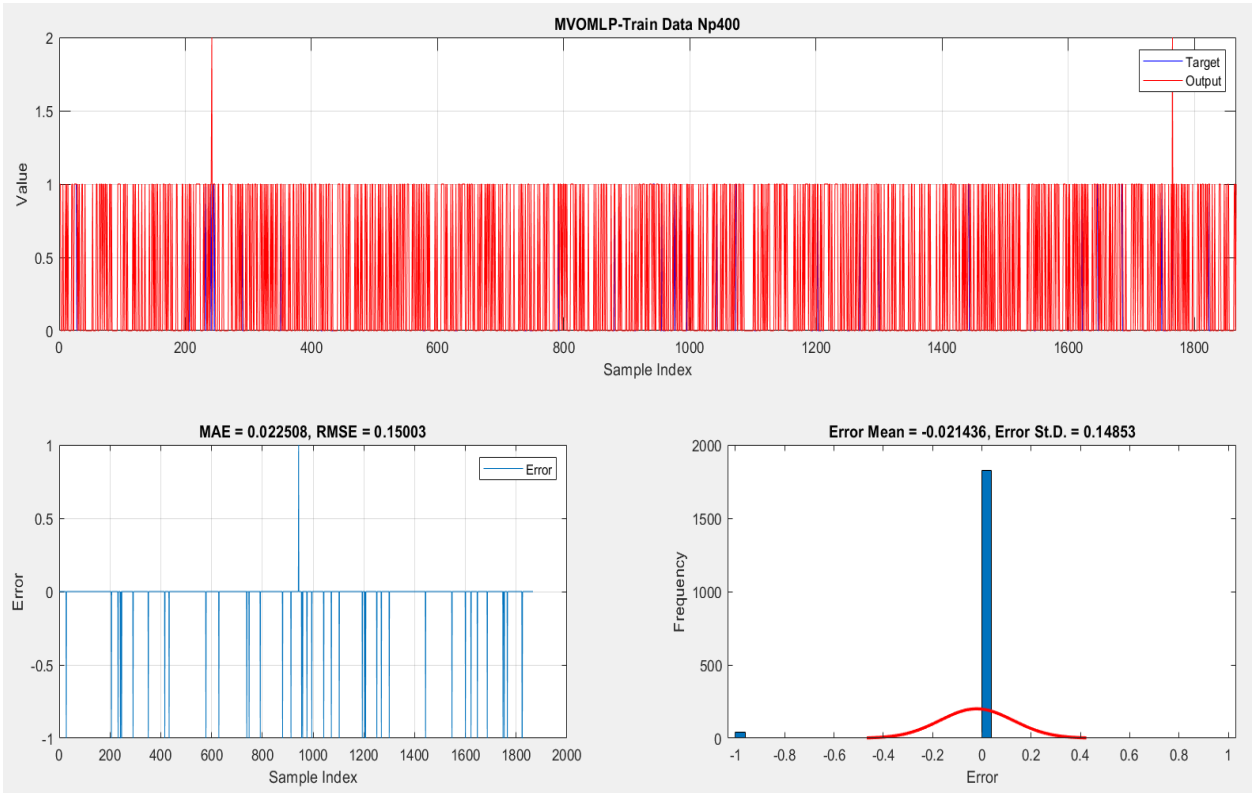
Population size	Network AUC results		Scoring		Total score	RANK
	Training	Testing	Training	Testing		
50	0.9908	0.9920	1	1	2	10
100	0.9930	0.9938	5	6	11	5
150	0.9912	0.9926	3	3	6	8
200	0.9939	0.9949	8	9	17	3
250	0.9938	0.9938	7	5	12	4
300	0.9942	0.9950	9	10	19	1
350	0.9911	0.9921	2	2	4	9
400	0.9948	0.9948	10	8	18	2
450	0.9932	0.9936	6	4	10	7
500	0.9926	0.9942	4	7	11	5

An essential component of assessing the performance of the MVO-MLP, TLBO-MLP, WDO-MLP, and WOA-MLP networks is examining the prediction errors during the validation and test phases, as shown in Figure 9-12. Using the error histogram model, the highest and smallest prediction errors are displayed. The data in the figures below show that the trained network may not be reliable in predicting how sensitive occupancy detection is to certain circumstances.

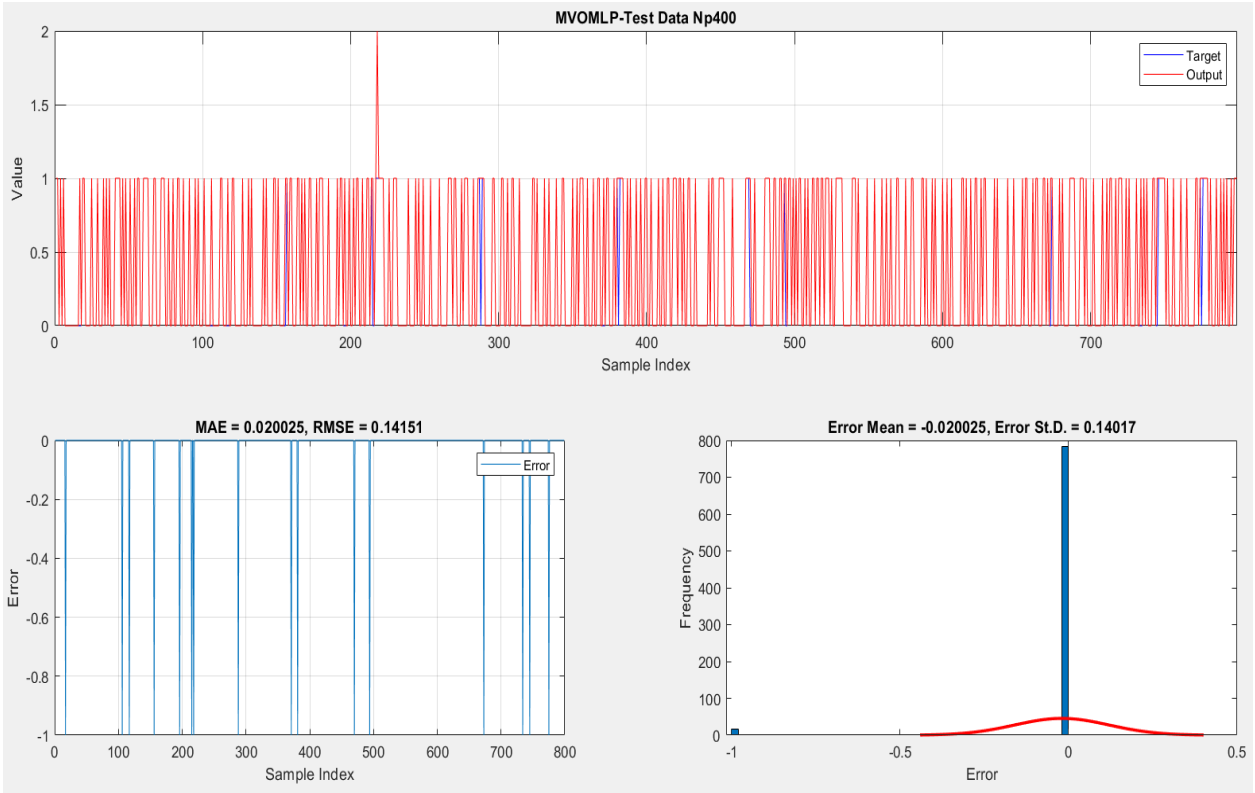
The provided information has been utilized to accurately assess the recommended methodologies. This study was based on an analysis and inspection of the provided figures, which depict the early testing and training stages of the MVO-MLP, TLBO-MLP, WDO-MLP, and WOA-MLP networks. 400, 300, 500, and 300 employees were managing these networks. The quality of training affects the architecture's resilience, the networks' precision, and the kind of data used for initial validation and testing. Consequently, the final models may evaluate and forecast both recognized and indeterminate data.

4.1. Measurment of occupancy detection using MLP

Table 5 provides a summary of the outcomes for all suggested methods, incorporating Multi-Verse Optimization (MVO), Teaching-Learning-Based Optimization (TLBO), Wind-Driven Optimization (WDO), and Whale Optimization Algorithm (WOA) when integrated with Multilayer Perceptron (MLP) for occupancy detection. Swarm size 400 achieves the highest AUC scores for training and testing datasets, resulting in a balanced and competitive total score of 8. This performance leads to MVO-MLP securing the top rank. TLBO-MLP, with a swarm size of 300, demonstrates consistently high AUC scores and achieves the second-highest total score of 3. The model exhibits balanced performance, earning the third rank. WDO-MLP, with a swarm size of 500, achieves moderate AUC scores. While the model attains the highest total score of 4, indicating balanced performance, it secures the fourth rank due to lower AUC scores than MVO-MLP and TLBO-MLP. WOA-MLP, with a swarm size of 300, showcases strong AUC scores, resulting in a competitive total score of 6. The model secures the second rank, emphasizing its effectiveness in occupancy detection. In summary, Table 5 provides a concise overview of the comparative performance of different optimization algorithms when integrated with MLP for occupancy detection. MVO-MLP stands out with the top rank, demonstrating superior performance, while TLBO-MLP and WOA-MLP exhibit competitive results. WDO-MLP, while achieving a balanced total score, falls slightly behind in AUC scores and overall ranking. These findings can guide the selection of optimization algorithms based on specific performance criteria and preferences for swarm size in occupancy detection applications.

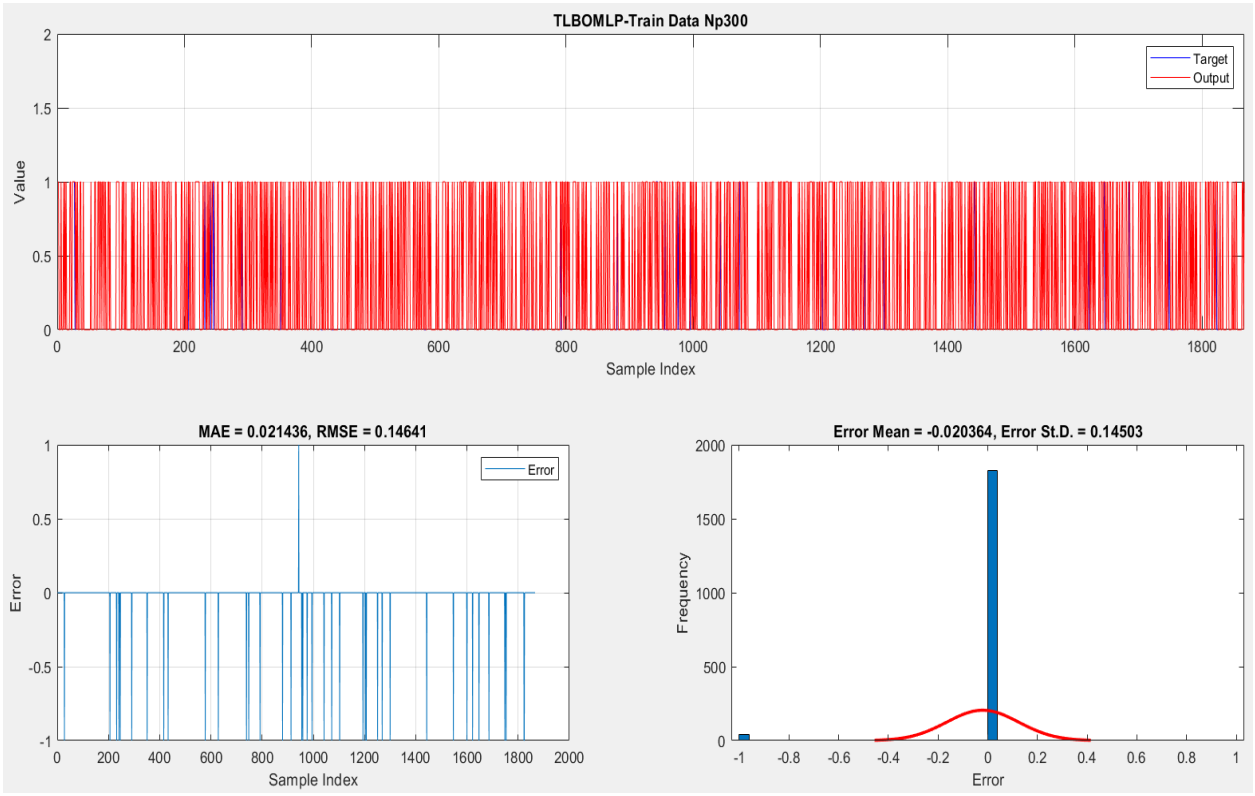


a) MVO 400-training

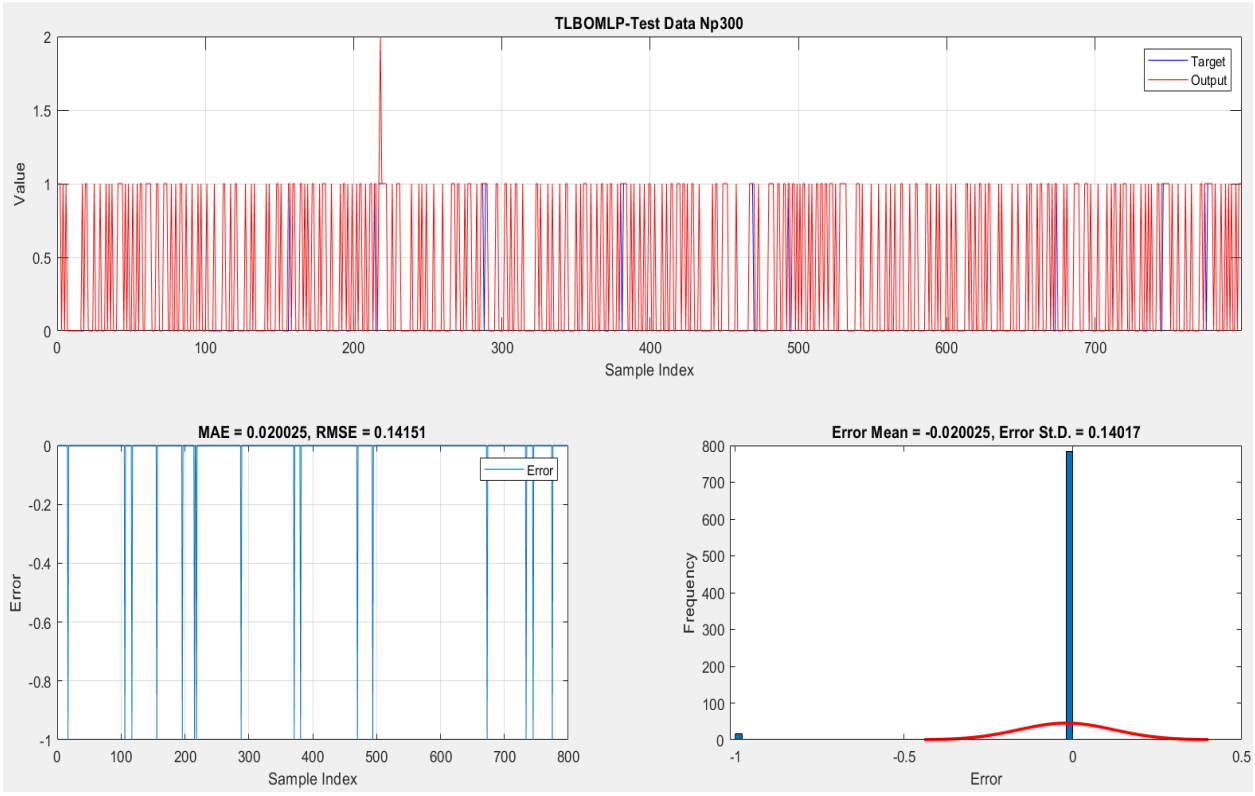


b) MVO 400-Testing

Figure 9: The MVOMLP-recommended ideal match model, together with its error and MAE frequency

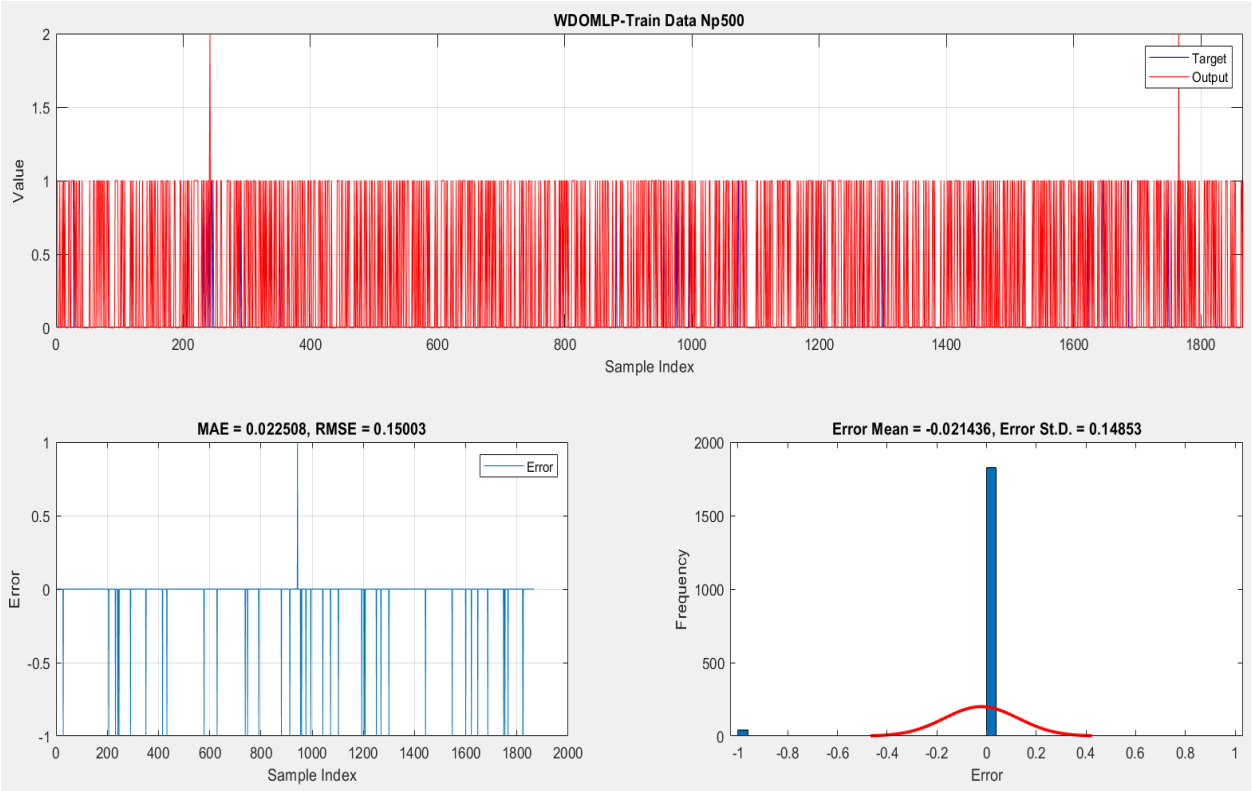


a) TLBOMLP 300-training

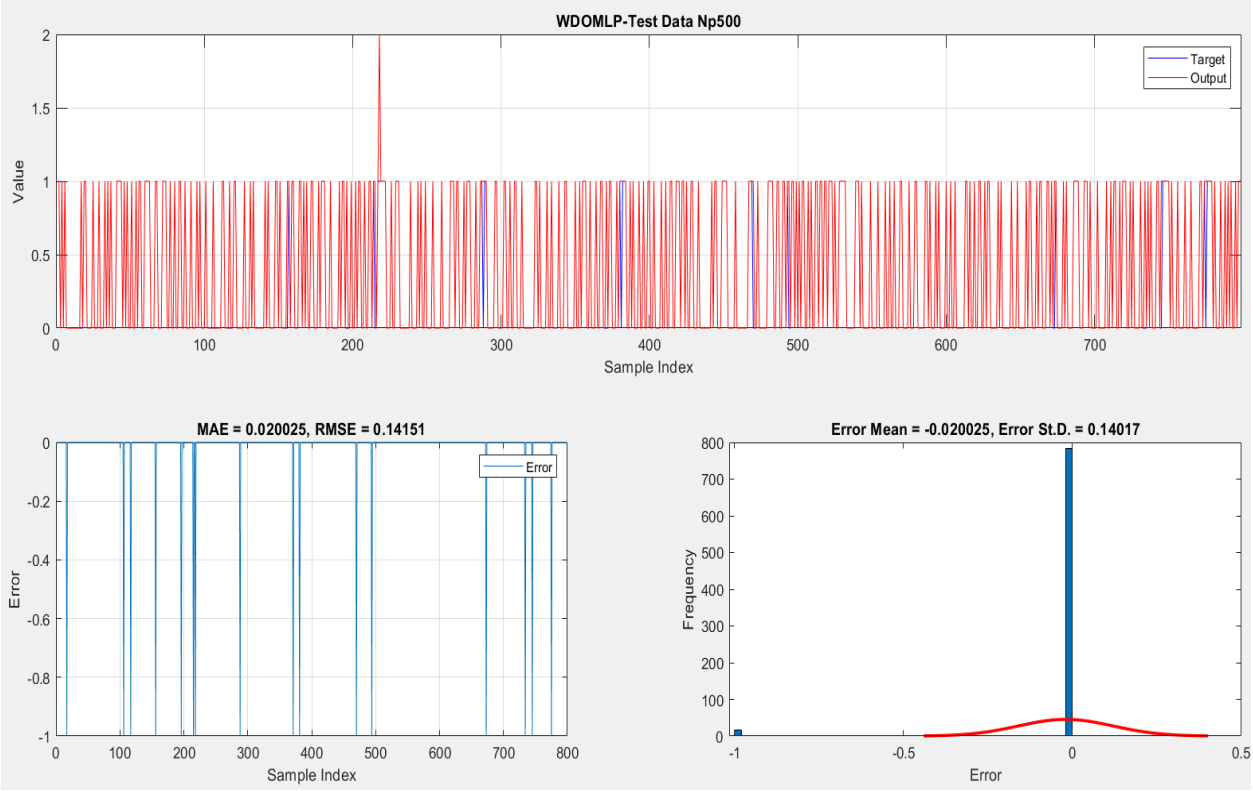


b) TLBOMLP 300-Testing

Figure 10: The TLBOMLP-recommended ideal match model, together with its error and MAE frequency

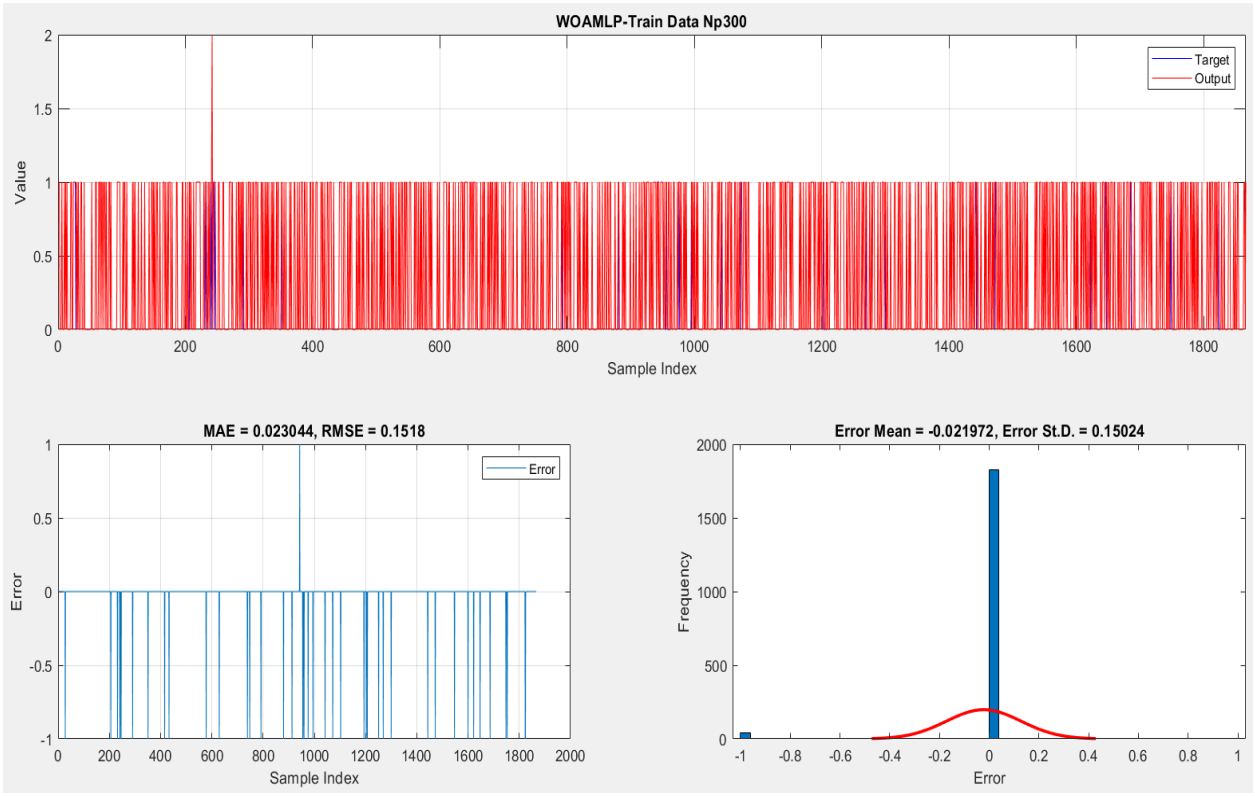


a) WDOMLP 500-training

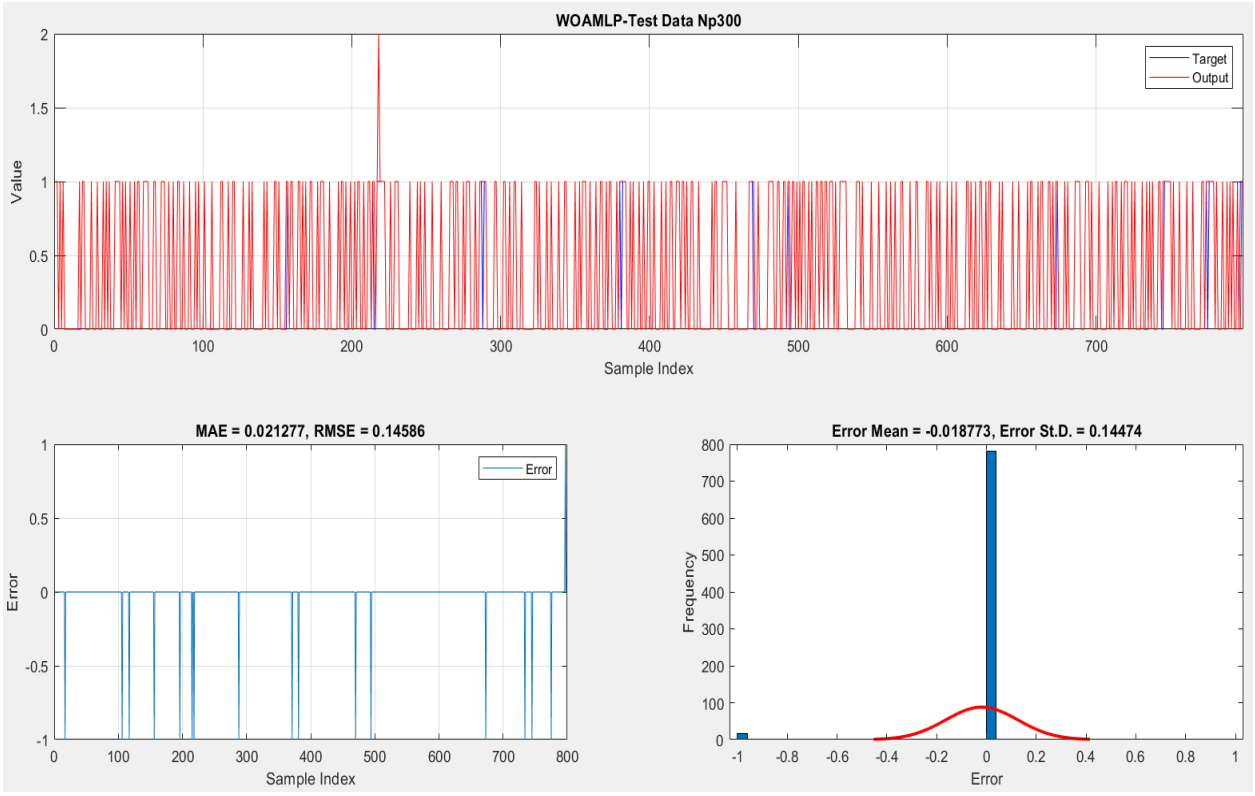


b) WDOMLP 500-Testing

Figure 11: The WDOMLP-recommended ideal match model, together with its error and MAE frequency



a) WOAMLP 300-training



b) WOAMLP 300-Testing

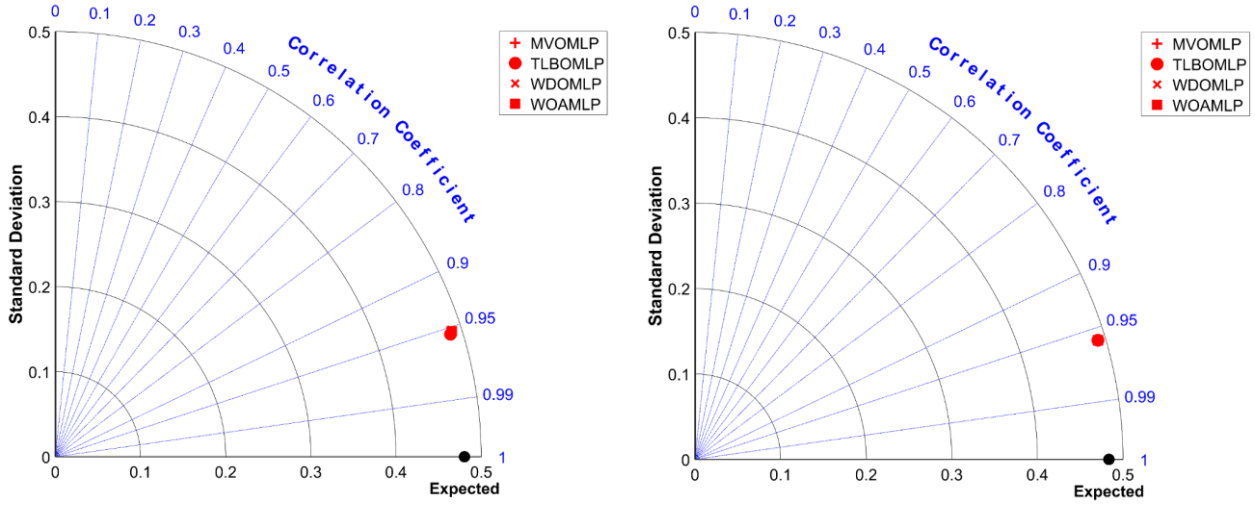
Figure 12: The WOAMLP-recommended ideal match model, together with its error and MAE frequency

Table 5: The outcomes of networks for all suggested methods.

Methods	Swarm size	AUC		Scoring		Total Score	Rank
		Training	Testing	Training	Testing		
MVO-MLP	400	0.9956	0.9955	4	4	8	1
TLBO-MLP	300	0.9929	0.9851	2	1	3	3
WDO-MLP	500	0.9829	0.9851	1	1	2	4
WOA-MLP	300	0.9942	0.9950	3	3	6	2

4.2. Taylor diagrams

In meteorology and climate research, the Taylor diagram—named for Karl E. Taylor—is a graphical tool used to assess how well many datasets align with a reference dataset. It is frequently used to evaluate the performance of model outputs on observational data, such as numerical simulations or climate models. Researchers may identify the most sophisticated datasets with this graphic, which provides a complete perspective of model performance across several domains. For evaluating and comparing models, Taylor diagrams are useful tools. They could also help create the model by drawing attention to places that need work. They conduct a detailed analysis of the model's performance regarding variability, correlation, and overall agreement with observational data. It was first introduced by Taylor [61] and provides a graphical depiction of the degree of similarity between a pattern or set of patterns and data. The standard deviations, the centered root-mean-square difference, and the similarity score between the two patterns are obtained from the correlation. These photos are very helpful for analyzing complicated models with several components or assessing how well different models work, as the IPCC has shown [62]. The Taylor diagram in Figure 13 compares the model's capacity to the accuracy with which it can identify occupancy in the present datasets. On four models with labels, we performed a statistical analysis. The position of each label on the map shows how well the predicted precipitation pattern of the model matches the observed data. The pattern correlation coefficients for the MVO-MLP, TLBO-MLP, WDO-MLP, and WOA-MLP are about 0.95.



a) Training **b) Testing**

Figure 13. The occupancy detection Taylor diagram

4.3. Discussion

The findings presented in this study shed light on the efficacy of various optimization algorithms when integrated with Multilayer Perceptron (MLP) for occupancy detection in office environments. The comparative analysis of optimization algorithms, including MVO, TLBO, WDO, and WOA, reveals varying degrees of effectiveness in

optimizing MLPs for accurate occupancy detection. MVO emerges as a standout performer, securing the top rank with the highest total score. The algorithm's ability to explore diverse solution spaces contributes to its effectiveness in optimizing MLPs for occupancy detection, as reflected in superior AUC scores. TLBO demonstrates consistent and competitive performance, earning the third rank. The algorithm's teaching and learning phases contribute to its ability to converge toward optimal solutions, as balanced AUC scores indicate. WOA also exhibits strong performance, securing the second rank. The algorithm's mimicry of the hunting behavior of whales proves effective in optimizing MLPs for accurate occupancy detection, leading to competitive AUC scores and a balanced total score. WDO achieves moderate performance, securing the fourth rank. While the algorithm demonstrates balanced AUC and total scores, it falls slightly behind MVO-MLP, TLBO-MLP, and WOA-MLP regarding overall effectiveness. The optimal swarm size varies for each algorithm. MVO-MLP achieves peak performance with a swarm size of 400, while TLBO-MLP and WOA-MLP excel with swarm sizes 300. WDO-MLP, with a swarm size of 500, achieves moderate performance. All algorithms exhibit a commendable ability to generalize well to unseen data, avoiding overfitting. Sensitivity to changes in swarm size is evident, emphasizing the importance of selecting an appropriate population size based on the algorithm's characteristics. The study provides valuable insights for practitioners seeking optimal solutions for occupancy detection in office environments. MVO, TLBO, WDO, and WOA present viable options with unique strengths and considerations. This paper highlights the nuanced performance of optimization algorithms in the context of MLP-based occupancy detection. The findings offer guidance for selecting an algorithm based on specific performance criteria. MVO is a robust choice for achieving accurate and reliable occupancy detection in real-world office scenarios.

5. Conclusion

Optimizing energy usage and guaranteeing occupant comfort need building management systems integrating comprehensive occupancy data. The accuracy and granularity of occupancy data are improved by using high-precision technology, such as thermal and optical cameras, as well as environmental sensors like carbon dioxide (CO_2) and passive infrared (PIR). Because of this accuracy, building systems can be more precisely controlled and adapted, significantly reducing energy use and improving occupant comfort. As technology develops, intelligent and energy-efficient future buildings will be greatly influenced by the smooth integration of many sensors and advanced data processing techniques. In the last article, we attempted to use various techniques inspired by machine learning and artificial intelligence to obtain favorable occupancy detection results in office spaces. The following are the outcomes:

- The comparative analysis revealed varying degrees of performance among the optimization algorithms. MVO demonstrated exceptional effectiveness, securing the top rank with superior AUC scores. TLBO and WOA also exhibited competitive performance, while WDO demonstrated moderate effectiveness.
- The optimal swarm size for each algorithm was identified, emphasizing the significance of selecting an appropriate population size. MVO-MLP achieved peak performance with a swarm size of 400, while TLBO-MLP and WOA-MLP excelled with swarm sizes 300. WDO-MLP demonstrated moderate performance with a swarm size of 500.
- All algorithms demonstrated a commendable ability to generalize well to unseen data, highlighting their suitability for real-world occupancy detection applications. Sensitivity to changes in swarm size underscored the importance of carefully selecting the population size based on algorithm characteristics.
- The study provides practical insights for practitioners seeking optimal solutions for occupancy detection in office environments. MVO, TLBO, WDO, and WOA offer viable options with unique strengths and considerations, allowing for informed algorithm selection based on specific application requirements.

Future research endeavors may explore the application of these optimization algorithms in diverse contexts, consider additional algorithms, or investigate hybrid approaches for further improving the accuracy and efficiency of occupancy detection systems. Additionally, exploring the robustness of these algorithms in the presence of noisy or dynamic environments could enhance their practical utility. This study contributes valuable insights into the selection and

application of optimization algorithms for MLP-based occupancy detection. The findings offer a foundation for enhancing the development of intelligent occupancy detection systems, with MVO standing out as a particularly promising algorithm. The versatility of these algorithms positions them as valuable tools for advancing smart building technologies and energy-efficient management systems.

6. References

[1] Nguyen, T.A. and M. Aiello, Energy intelligent buildings based on user activity: A survey. Energy and Buildings, 2013. 56: p. 244-257.

[2] Zhang, Y., et al., Rethinking the role of occupant behavior in building energy performance: A review. Energy and Buildings, 2018. 172: p. 279-294.

[3] Agarwal, Y., et al., Occupancy-driven energy management for smart building automation, in Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building. 2010, Association for Computing Machinery: Zurich, Switzerland. p. 1–6.

[4] Jin, M., et al., Occupancy Detection via Environmental Sensing. IEEE Transactions on Automation Science and Engineering, 2018. 15(2): p. 443-455.

[5] Jin, M., et al., Sensing by proxy: Occupancy detection based on indoor CO2 concentration. UBICOMM 2015, 2015. 14.

[6] Leephakpreeda, T., Adaptive Occupancy-based Lighting Control via Grey Prediction. Building and Environment, 2005. 40(7): p. 881-886.

[7] Scott, J., et al., PreHeat: controlling home heating using occupancy prediction, in Proceedings of the 13th international conference on Ubiquitous computing. 2011, Association for Computing Machinery: Beijing, China. p. 281–290.

[8] Yokoishi, T., et al. Room occupancy determination with particle filtering of networked pyroelectric infrared (PIR) sensor data. in SENSORS, 2012 IEEE. 2012.

[9] Peng, Y., et al., Occupancy learning-based demand-driven cooling control for office spaces. Building and Environment, 2017. 122: p. 145-160.

[10] Guzmán, C., K. Agbossou, and A. Cardenas, Real-Time Emulation of Residential Buildings by Hardware Solution of Multi-Layer Model. IEEE Transactions on Smart Grid, 2019. 10(4): p. 4037-4047.

[11] Rafsanjani, H.N., C.R. Ahn, and M. Alahmad, A Review of Approaches for Sensing, Understanding, and Improving Occupancy-Related Energy-Use Behaviors in Commercial Buildings. Energies, 2015. 8(10): p. 10996-11029.

[12] Conti, J., et al., International energy outlook 2016 with projections to 2040. 2016, USDOE Energy Information Administration (EIA), Washington, DC (United States ….

[13] Yoshino, H., T. Hong, and N. Nord, IEA EBC annex 53: Total energy use in buildings—Analysis and evaluation methods. Energy and Buildings, 2017. 152: p. 124-136.

[14] Jia, M., R.S. Srinivasan, and A.A. Raheem, From occupancy to occupant behavior: An analytical survey of data acquisition technologies, modeling methodologies and simulation coupling mechanisms for building energy efficiency. Renewable and Sustainable Energy Reviews, 2017. 68: p. 525-540.

[15] Melfi, R., et al. Measuring building occupancy using existing network infrastructure. in 2011 International Green Computing Conference and Workshops. 2011.

[16] Kjærgaard, M.B. and F.C. Sangogboye, Categorization framework and survey of occupancy sensing systems. Pervasive and Mobile Computing, 2017. 38: p. 1-13.

[17] Balvedi, B.F., E. Ghisi, and R. Lamberts, A review of occupant behaviour in residential buildings. Energy and Buildings, 2018. 174: p. 495-505.

[18] Yang, J., M. Santamouris, and S.E. Lee, Review of occupancy sensing systems and occupancy modeling methodologies for the application in institutional buildings. Energy and Buildings, 2016. 121: p. 344-349.

[19] Mane, P.K. and K. Narasimha Rao. Review of Research Progress, Trends and Gap in Occupancy Sensing for Sophisticated Sensory Operation. in Cybernetics and Algorithms in Intelligent Systems. 2019. Cham: Springer International Publishing.

[20] Labeodan, T., et al., Occupancy measurement in commercial office buildings for demand-driven control applications—A survey and detection system evaluation. Energy and Buildings, 2015. 93: p. 303-314.

[21] Saha, H., et al., Occupancy sensing in buildings: A review of data analytics approaches. Energy and Buildings, 2019. 188-189: p. 278-285.

[22] Chen, Z., C. Jiang, and L. Xie, Building occupancy estimation and detection: A review. Energy and Buildings, 2018. 169: p. 260-270.

[23] Sun, K., Q. Zhao, and J. Zou, A review of building occupancy measurement systems. Energy and Buildings, 2020. 216: p. 109965.

[24] Zuraimi, M.S., et al., Predicting occupancy counts using physical and statistical Co2-based modeling methodologies. Building and Environment, 2017. 123: p. 517-528.

[25] Lam, K.P., et al., Occupancy detection through an extensive environmental sensor network in an open-plan office building. 2009.

[26] Dong, B., et al., An information technology enabled sustainability test-bed (ITEST) for occupancy detection through an environmental sensing network. Energy and Buildings, 2010. 42(7): p. 1038-1046.

[27] Kraipeerapun, P. and S. Amornsamankul, Room Occupancy Detection using Modified Stacking, in Proceedings of the 9th International Conference on Machine Learning and Computing. 2017, Association for Computing Machinery: Singapore, Singapore. p. 162–166.

[28] Yang, Z., et al. A non-intrusive occupancy monitoring system for demand driven HVAC operations. in Construction Research Congress 2012: Construction Challenges in a Flat World. 2012.

[29] Villariba, K., A neural network approach to detecting the occupancy state of a home from electricity usage. 2014, Mills College.

[30] Hobson, B.W., et al., Opportunistic occupancy-count estimation using sensor fusion: A case study. Building and environment, 2019. 159: p. 106154.

[31] Masood, M.K., Y.C. Soh, and V.W.-C. Chang. Real-time occupancy estimation using environmental parameters. in 2015 international joint conference on neural networks (IJCNN). 2015. IEEE.

[32] Masood, M.K., Y.C. Soh, and C. Jiang, Occupancy estimation from environmental parameters using wrapper and hybrid feature selection. Applied Soft Computing, 2017. 60: p. 482-494.

[33] Masood, M.K., C. Jiang, and Y.C. Soh, A novel feature selection framework with Hybrid Feature-Scaled Extreme Learning Machine (HFS-ELM) for indoor occupancy estimation. Energy and Buildings, 2018. 158: p. 1139-1151.

[34] Zou, H., et al., A Fast and Precise Indoor Localization Algorithm Based on an Online Sequential Extreme Learning Machine. Sensors, 2015. 15(1): p. 1804-1824.

[35] Zou, H., et al., Non-intrusive occupancy sensing in commercial buildings. Energy and Buildings, 2017. 154: p. 633-643.

[36] Ertuğrul, Ö.F., Y. Kaya, and M. EminTağluk, Detecting occupancy of an office room by recurrent extreme learning machines. trees, 2016. 2(4): p. 17.

[37] Candanedo, L.M. and V. Feldheim, Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Energy and Buildings, 2016. 112: p. 28-39.

[38] Jiang, C., et al., Indoor occupancy estimation from carbon dioxide concentration. Energy and Buildings, 2016. 131: p. 132-141.

[39] Ryden, K., Environmental systems research institute mapping. The American Cartographer, 1987. 14(3): p. 261-263.

[40] Yilmaz, I., Comparison of landslide susceptibility mapping methodologies for Koyulhisar, Turkey: conditional probability, logistic regression, artificial neural networks, and support vector machine. Environmental Earth Sciences, 2010. 61: p. 821-836.

[41] Huang, F., et al., Landslide susceptibility mapping based on self-organizing-map network and extreme learning machine. Engineering Geology, 2017. 223: p. 11-22.

[42] Dou, J., et al., A comparative study of the Binary Logistic Regression (BLR) and Artificial Neural Network (ANN) models for GIS-based spatial predicting landslides at a regional scale. Landslide Dynamics: ISDR-ICL Landslide Interactive Teaching Tools, 2018. 1: p. 139-151.

[43] Xing, K., et al., Deadlock-free genetic scheduling algorithm for automated manufacturing systems based on deadlock control policy. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2011. 42(3): p. 603-615.

[44] Kang, Q., et al., Optimal load scheduling of plug-in hybrid electric vehicles via weight-aggregation multi-objective evolutionary algorithms. IEEE Transactions on Intelligent Transportation Systems, 2017. 18(9): p. 2557-2568.

[45] Meng, X., et al., Population-based incremental learning algorithm for a serial colored traveling salesman problem. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2016. 48(2): p. 277-288.

[46] Tian, G., et al., Fuzzy grey choquet integral for evaluation of multicriteria decision making problems with interactive and qualitative indices. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019. 51(3): p. 1855-1868.

[47] Fu, Y., et al., Artificial-molecule-based chemical reaction optimization for flow shop scheduling problem with deteriorating and learning effects. Ieee Access, 2019. 7: p. 53429-53440.

[48] Zhao, Z., et al., Decomposition method for new single-machine scheduling problems from steel production systems. IEEE Transactions on Automation Science and Engineering, 2019. 17(3): p. 1376-1387.

[49] Zhao, J., et al., Modified cuckoo search algorithm to solve economic power dispatch optimization problems. IEEE/CAA Journal of Automatica Sinica, 2018. 5(4): p. 794-806.

[50] Mirjalili, S., S.M. Mirjalili, and A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications, 2016. 27(2): p. 495-513.

[51] Rao, R.V., V.J. Savsani, and D. Vakharia, Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Information sciences, 2012. 183(1): p. 1-15.

[52] Rao, R.V., V. Savsani, and J. Balic, Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. Engineering Optimization, 2012. 44(12): p. 1447-1462.

[53] Rao, R.V., V.J. Savsani, and D. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-aided design, 2011. 43(3): p. 303-315.

[54] Rao, R.V., V.J. Savsani, and D.P. Vakharia, Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems. Information Sciences, 2012. 183(1): p. 1-15.

[55] Rao, R.V., V.J. Savsani, and D.P. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer-Aided Design, 2011. 43(3): p. 303-315.

[56] Bayraktar, Z., M. Komurcu, and D.H. Werner. Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics. in 2010 IEEE antennas and propagation society international symposium. 2010. IEEE.

[57] Derick, M., et al., An improved optimization technique for estimation of solar photovoltaic parameters. Solar Energy, 2017. 157: p. 116-124.

[58] Bayraktar, Z., et al., The wind driven optimization technique and its application in electromagnetics. IEEE transactions on antennas and propagation, 2013. 61(5): p. 2745-2757.

[59] Bhandari, A.K., et al., Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. Expert Systems with Applications, 2014. 41(7): p. 3538-3560.

[60] Mirjalili, S. and A. Lewis, The whale optimization algorithm. Advances in engineering software, 2016. 95: p. 51-67.

[61] Taylor, K.E., Summarizing multiple aspects of model performance in a single diagram. Journal of geophysical research: atmospheres, 2001. 106(D7): p. 7183-7192.

[62] Change, I.C., The Physical Science Ba-sis. Contribution of Working Group 1 to the Fourth Assessment Report ofthe Intergovernmental Panel on Climate Change. 2007, Cambridge. UK.