İSTANBUL UNIVERSITY CERRAHPAŞA

ORIGINAL ARTICLE
Electrica 2024; 24(2): 436-449

Optimization of Non-Linear Problems Using Salp Swarm Algorithm and Solving the Energy Efficiency Problem of Buildings with Salp Swarm Algorithm-based Multi-Layer Perceptron Algorithm

Erdal Eker[1], Şeymanur Atar[2], Fatih Şevgin[3], İhsan Tuğal[4]

[1]Vocational School of Social Sciences, Muş Alparslan University, Muş, Türkiye
[2]Institute of Science and Technology, Muş Alparslan University, Muş, Türkiye
[3]Vocational School of Technical Sciences, Muş Alparslan University, Muş, Türkiye
[4]Department of Software Engineering, Muş Alparslan University, Muş, Türkiye

Cite this article as: E. Eker, Ş. Atar, F. Şevgin and İ. Tuğal, "Optimization of non-linear problems using salp swarm algorithm and solving the energy efficiency problem of buildings with salp swarm algorithm-based multi-layer perceptron algorithm," *Electrica*, 24(2), 436-449, 2024.

ABSTRACT

The aim of this paper is to evaluate the optimization capabilities of the salp swarm algorithm (SSA), a metaheuristic algorithm capable of addressing contemporary global challenges. The paper focuses on assessing SSA as an optimizer and observing its impact as a predictor in an example energy problem to gauge its predictive power. Salp swarm algorithm (SSA) distinguishes itself with its optimization capabilities, providing effective solutions to optimization problems. The quality, competitiveness, and efficiency of the algorithm were initially assessed using the CEC 2019 and CEC 2020 function sets. The results demonstrated that SSA is a competitive, effective, and up-to-date algorithm. This competitive nature suggests that SSA can be effectively employed across a wide range of problems. Therefore, the paper aims to evaluate its success in providing solutions to an energy prediction problem. In addressing the challenge of effective energy utilization, the accurate prediction of heat loading (HL) and cool loading (CL) factors, critical in building design, contributes significantly to the solution. In solving this problem, machine learning algorithms, specifically the multi-layer perceptron (MLP) as an artificial neural network architecture, were chosen. SSA was approached in a supervised manner, and a comparison with alternative metaheuristic algorithms was conducted. The obtained results indicate that the SSA-based MLP architecture (SSA-MLP) exhibits effective predictive capabilities in energy problems. By combining the optimization power of SSA and the learning capabilities of MLP, a robust solution with a competitive advantage in energy efficiency is presented.

***Index Terms*—**Energy efficiency problem, metaheuristic algorithm, machine learning algorithm, multi-layer perceptron, salp swarm algorithm

Corresponding author:
İhsan Tuğal

E-mail:
i.tugal@alparslan.edu.tr

Received: December 8, 2023

Revision Requested: January 7, 2024

Last Revision Received: March 6, 2024

Accepted: March 27, 2024

Publication Date: May 20, 2024

DOI: 10.5152/electrica.2024.23193

I. INTRODUCTION

In recent years, the world has witnessed significant scientific and industrial advancements. Alongside these developments, there is a growing need to address numerous complex, high-dimensional, and nonlinear optimization problems. Solving nonlinear problems entails finding the best solution among possible solutions, a process referred to as optimization [1]. Mathematically, optimization problems can be modeled as

$$min\ f(x),\ g(x) \leq 0,\ h(x) = 0,\ x \in Q \tag{1}$$

Here, $f(x)$ represents the problem to be minimized, while $g(x) \leq 0$ and $h(x) = 0$ denote functions representing the constraints of the problem [2]. The solution of optimization problems involves specific algorithmic processes, and it is important to note that not every algorithm can solve all problems. In other words, while one algorithm may be capable of solving certain problems, another algorithm may be suited for different types of problems. This underscores the absence of a universal algorithm for all problems [3]. The two fundamental methods for solving problems are the gradient method [4] and metaheuristic methods [5]. Metaheuristic algorithms are structures inspired by nature, characterized by their simple and linear formations. These algorithms are widely applied in engineering research and various fields due to their population-based randomizations. They are essentially divided into evolutionary and swarm-based algorithms [6]. Metaheuristic methods have become increasingly prevalent in recent years. This is attributed to their non-reliance on derivatives, flexible structures, and their ability to provide more optimal solutions in a shorter time [7]. Some initial works in this area include Genetic Algorithm (GA) [8], Particle Swarm Optimization (PSO) [9], Artificial Bee Colony (ABC) [10], and Ant Colony Optimization (ACO) [11]. Subsequently, numerous algorithms such as Grey Wolf Optimizer (GWO) [10], Harris Hawks Optimization (HHO) [12], Hunger Games Search (HGS) [13], and Dandelion Optimizer (DO) have been introduced. Metaheuristic algorithms are widely used in many industrial areas such as solving classical engineering problems as real-world problems, design problems, automotive industry, wind farms, and heat transfer mechanisms [14-16].

In this paper, the swarm-based Salp Swarm Algorithm's problem-solving performance was initially assessed. The Salp Swarm Algorithm (SSA), introduced in 2017, is an optimization algorithm based on the swarm mechanism of salps, offering a biology-inspired approach [17]. Salp Swarm Algorithm has been applied not only to address various optimization problems but also to solve problems mathematically modeled in diverse fields such as industry, medicine, education, and various aspects of life. Furthermore, it has found utility in different domains, including machine learning, engineering design, wireless network formation, image processing, and power energy optimization [18]. SSA, in its essence, investigates the swarming behavior of salps in the oceans. It leverages the chain formation behavior of salps as they navigate the ocean. SSA's implementation is straightforward, efficient, and capable of producing flexible results [19]. However, akin to many metaheuristic algorithms, SSA may encounter challenges such as early convergence or stagnation by getting trapped in local optima. Therefore, there is a need for further enhancement to address these issues for improved problem-solving [20-21]. Function sets, often referred to as quality functions, are employed to reveal the solution quality of metaheuristic algorithms. In this paper, the optimization power of the SSA algorithm is demonstrated using the CEC2019 [22] and CEC2020 [23] function sets, known to complicate the path to global solutions [24].

As a competitive algorithm, SSA establishes another objective of applying its capabilities to real-world problems, aiming to achieve specific results after evaluating its performance superiority. In this context, the training of a non-linear problem, the multi-layer perceptron (MLP), serves as an exemplary case [25]. MLP is utilized in supervised learning architectures for classifying data in artificial neural network training. The inclusion of a hidden layer between input and output layers in MLP training results in a higher data classification rate and a lower error rate [26]. There exists a rich literature on the use of metaheuristic algorithms in the training of MLP [27].

The objective in the training of SSA-based MLP (SSA-MLP) is to evaluate the algorithm's capability in solving energy-related problems. According to the International Energy Agency (IEA), the increase in energy consumption today, with 34% of energy consumption being energy spent in buildings, points to a significant problem [28]. The need for saving in energy consumption has led to the search for new methods. Energy used in buildings is utilized through heating, cooling, ventilation, and air conditioning (HVAC). The HVAC system is designed to create favorable air conditions inside the building by calculating the HL and CL factors of the building through heating, cooling, and ventilating the air inside the building. Estimation of HL and CL factors in buildings is important for reducing consumption according to the occupancy level, managing energy demands of the change in the performance of the building, reducing emissions of harmful gases, and reducing costs thanks to accurate estimation [29]. The required cooling and heating capacities are estimated according to basic factors such as building characteristics, utilization, and climatic conditions. Optimum design of HVAC systems plays an important role in ensuring energy saving [30]. In this context, the interior design of buildings is important in terms of human health as well as saving energy consumption, and HVAC systems are generally the preferred active systems to achieve this saving. In order to maximize high efficiency and energy saving, HVAC systems should be designed in accordance with the climatic conditions of the building [31].

Studies focused on predicting energy consumption in buildings contribute significantly to energy conservation efforts and mitigating environmentally harmful effects. In this research, the energy efficiency performance for heating load (HL) and cooling load (CL) in buildings was predicted through MLP, utilizing a dataset obtained from the UCI Machine Learning Repository [32].

The article is organized into three main sections: the introduction in the first section, the definition, operational mechanism, and application areas of the SSA algorithm in the second section, and the methodology and techniques divided into two subsections in the third section. The first subsection involves the performance evaluation of the SSA algorithm, comparing it with alternative contemporary and effective algorithms, namely Prairie Dog Optimization Algorithm (PDO) [33], Hunger Games Search Optimizer (HGS) [13], Archimed Optimization Algorithm (AOA) [34], and Harris Hawks Optimization (HHO) [12]. This comparison is carried out using performance tables consisting of statistical metrics and convergence curves through CEC 2019 and CEC 2020. The second subsection focuses on the creation of the SSA-based MLP (SSA-MLP) architecture, addressing the prediction of the energy problem dataset and generating tables with statistical measurement tools. Additionally, the interpretation of the data through median is presented using box-plot graphs. The final section, titled "Results," evaluates the paper and sheds light on future research directions.

II. SALP SWARM ALGORITHM

A. Definition and Mathematical Expression of the SSA Algorithm

Algorithms based on swarm intelligence techniques mimic the collective behaviors exhibited by living organisms that move in groups in nature. The SSA algorithm is a swarm-based algorithm that emulates the behaviors of salp communities. Salp swarms form a specific chain while swiftly and synchronously moving in the oceans, displaying a distinctive foraging behavior. This organism propels itself by pumping water similar to a jellyfish to capture plankton in the ocean. The foraging behavior is characterized by the spontaneous formation of this chain. Individuals in front of the chain move spontaneously toward the food, while those behind the chain mimic the previous movement, resembling swarm behavior. The mathematical model of these specific behaviors of the chain constitutes the SSA algorithm. There are sources validating the efficient operation of the SSA algorithm in both small and large-scale problems.

Similar to various optimization techniques, SSA possesses some advantages (strengths) and a few disadvantages (weaknesses) [28]. These include simplicity, strong convergence acceleration, effectiveness in global search, adaptability, robust and flexible structure, as well as completing the optimization process in a reasonable time frame. Its main disadvantage is susceptibility to early convergence [19, 35-36].

The mathematical modeling of the SSA algorithm is provided in Equations 1, 2, and 3. If we refer to the mentioned chain as the salp chain, this chain can be divided into two parts: leaders (θ) and followers (α) [37].

In SSA, the population of this salp chain is evenly divided into two parts. The leader moves in real-time toward the location of the food source ($S{:}S\in\theta\, S:\forall S\in\theta$). This optimal solution, or the most suitable solution, is represented by the position of the leader in Equation 2.

$$S_j=\begin{cases}F_j+c_1\left(\left(ub_j-lb_j\right)c_2+lb_j\right) & c_3\geq 0\\ F_j-c_1\left(\left(ub_j-lb_j\right)c_2+lb_j\right) & c_3<0\end{cases} \tag{2}$$

Here, F_j represents the position of the prey, while ub_j and lb_j denote the upper and lower bounds in the array, respectively. Similarly, c_2 and c_3 are random numbers in the [0,1] range. The parameter c_1 is a control parameter that plays a crucial role in adjusting the exploration and exploitation phases during the optimization process, and it is expressed in Equation 3.

$$c_1=2e^{-\left(\frac{4t}{T_{max}}\right)^2} \tag{3}$$

Here, t and T_{max} represent the current and maximum iteration numbers, respectively. Subsequently, the position of the i-th follower, $M_j^i\left(M:\forall M \quad \alpha\right)$, is updated according to Newton's second law of motion based on the position of the follower closest to the leader. The position of the i-th follower is modeled according to Equation 4.

$$M_j^i=\frac{1}{2}\left(M_j^i+M_j^{i-1}\right) \qquad i\geq 2 \tag{4}$$

Equations 1 and 3 lead the reader to the simulation of salp chain behavior [26].

B. SSA Algorithm and Examples of Various Versions
The SSA has been applied in various domains such as automatic voltage regulator (AVR) systems [38], quality test functions [39], feature selection, training artificial neural networks [40], early detection of diabetes, gender differences in voice data, sensor networks, central models, and engineering design problems [41]. Due to its simple architecture and superior performance in the optimization process, the SSA algorithm has been successfully applied to many real-world problems. However, in some complex optimization problems, the issue of getting stuck in local optima is a challenge faced by the SSA algorithm. To address this, a new Enhanced Salp Swarm Algorithm (ESSA) has been proposed by hybridizing the SSA algorithm with three strategies, including OL, QI, and GOL, achieving improved performance [42]. Similarly, to overcome not only local optima but also the problem of entering into a deadlock in solving certain problems, an enhanced version of the SSA algorithm (DSSA) has been suggested [43].

The SSA has been rejuvenated through various modifications developed by researchers to solve comprehensive optimization problems [44]. They have produced binary solutions to enhance the exploration and exploitation capabilities of the SSA algorithm [40]. In the TCSSA algorithm, a new binary solution has been proposed to improve the leadership structure of SSA, ensuring a more successful exploitation phase. A salp swarm algorithm based on the simplex method (SMSSA) has been proposed. This unidirectional method is a random variant strategy that increases the number of search agents in the considered swarm and enhances the local searchability of the algorithm. This method not only helps achieve a more suitable balance between the exploration and exploitation stages of SSA but also makes SSA more robust and faster [45]. An SSAPSO algorithm, a hybridization of the SSA and PSO algorithms, has been developed to solve a feature selection problem [46].

In the proposed CME algorithm, the aim is to develop a new hybrid space exploration method that combines both deterministic and metaheuristic algorithms to rapidly summarize a definable map in an unknown environment. The process is optimized using the metaheuristic SSA to create a finite map with a multi-robot system. It generates random parameters that assist in determining the robot's next best positions, influencing the robot's subsequent movement. As a result, the SSA selects the robot's next movement position [47].

III. METHOD AND APPROACH

The SSA algorithm's quality and performance power were initially statistically reported. In the second stage, a real-world non-linear problem was solved. Addressing an energy problem, machine learning algorithms were employed for the classification of data in the dataset through MLP training. The SSA-MLP hybrid algorithm was utilized in a supervised manner for this purpose.

A. Superiority of SSA Algorithm Performance
Like all metaheuristic algorithms, the SSA algorithm exhibits a competitive nature. CEC 2019 and CEC 2020 function sets were used to compare the performance power of the algorithm with alternative algorithms. Thirty independent runs were conducted for each algorithm using the MATLAB program. The search agent number was set to 30 as initial parameters, and 500 iterations were performed in each run. The best results are indicated in bold. The performance of the SSA algorithm, along with alternative algorithms, was measured through statistical metrics such as mean, standard deviation, the best value, and the worst value. Additionally, the convergence trends of the SSA algorithm were observed over 500 iterations through convergence curves, reporting instances of early convergence or local minima.

To determine whether the results of the SSA algorithm obtained through the benchmark functions are different from the alternative algorithms, the P-value is performed at the 5% significance level. The results were evaluated in three different categories. winn (W) indicates that the SSA algorithm creates a completely different dataset, till (T) indicates that it does not have a very different dataset, and lost (L) indicates that there is no significant difference between SSA and the alternative algorithm [48].

The parameter settings of the SSA algorithm and alternative algorithms are given in Table I given below.

B. Measurement of Performance Superiority with CEC 2019
The CEC 2019 function set is contemporary and widely utilized in various studies. The function set includes F1–F3 functions with

TABLE I. BASIC PARAMETERS OF ALGORITHMS

Algorithms	Parameters	Value
SSA	c_1 parameter balances the stages of exploration and exploitation	$c_1 \in [0,1]$
PDO	ρ constant is the special food source alarm	$\rho = 0.1$ kHz
HGS	*l* parameter is to improve the algorithm LH parameter is the hunger threshold	*l=0.08* LH=*10, 100, 1000*
AOA	α indicates the exploit accuracy of the iterations μ parameter sets search process	$\alpha = 5$ $\mu = 0.499$
HHO	*E* parameter sets the transition of the HHO between processes	$\lvert E \rvert \geq 0.5$, soft containment occurs $\lvert E \rvert < 0.5$, hard containment occurs

multimodal characteristics and F4–F10 functions with multimodal shifted and rotated features. This function set is particularly employed in the exploration phase for revealing the universal optimum point. CEC2019 functions parameters is given by Table II, and Table III shows the experimental results.

Table IV shows that the SSA algorithm produces a different dataset than the alternative algorithms, but due to the size of some benchmark functions and the inertia of the SSA algorithm in exceeding the local optimum, it produces a similar dataset with HGS in F4, F5, F8, and F10, with AOA in F8 and with HHO in F7. These results should be considered as a successful result although the CEC2019 dataset makes it difficult to find the global optimum.

TABLE II. CEC2019 FUNCTIONS DEFINITY PARAMETERS

Functions	Dimension	Search Interval	Fitting Value
F1: Storn's Chebyshev Polynomial Fitting Problem	9	[−8192, 8192]	1
F2: Inverse Hilbert Matrix Problem	16	[−16 384, 16384]	1
F3: Lennard–Jones Minimum Energy Cluster	18	[-4.4]	1
F4: Rastrigin's Function	10	[−100, 100]	1
F5: Griewangk's Function	10	[−100, 100]	1
F6: Weierstrass Function	10	[−100, 100]	1
F7: Modified Schwefel's Function	10	[−100, 100]	1
F8: Expanded Schafer's F6 Function	10	[−100, 100]	1
F9: Happy Cat Function	10	[−100, 100]	1
F10: Ackley Function	10	[−100, 100]	1

Given the multimodal nature of all functions in the CEC 2019 function set, it serves as a metric for evaluating an algorithm's early convergence and local area traversal capabilities. In other words, it assesses the algorithm's exploratory capabilities. In Table III, SSA algorithm has demonstrated superior performance in terms of average values in functions F3, F4, F5, as well as F8 and F9 within the CEC 2019 function set. The fundamental characteristic of these functions being multimodal shifted and rotated indicates the algorithm's exceptional exploratory capabilities.

Additionally, examining the convergence curves visualized in Fig. 1 in parallel with this table reveals that functions F3, F4, F5, F8, and F9 converge regularly and steadily. It is observed that these functions do not experience early convergence and consistently approach the optimal point in each iteration, showcasing the algorithm's effectiveness across various functions.

C. Measurement of Performance Superiority with CEC 2020

The CEC 2020 function set is designed to test the algorithm with variables ranging from 5, 10, 20, 30, 50, up to 100 dimensions. This paper specifically addresses problems with 10 dimensions. Within the CEC 2020 function set whose basic parameters are given in Table V, F1 is single mode, F2--F4 is multimode, F5--F7 exhibits hybrid features and F8--F10 has combination features.This function set is a contemporary and effective tool for revealing both local and global optima, making it instrumental in evaluating algorithmic performance.

When Table VII is analyzed, it is seen that the SSA algorithm produces a different dataset than the alternative algorithms, but due to the size of some benchmark functions and the inertia of the SSA algorithm in exceeding the local optima, it produces a similar dataset with HGS in F3, F4, F5, F8, and F10, with AOA in F8 and with HHO in F7. The CEC 2020 set is a quality function set that has been successful in demonstrating the superiority of the algorithm in many studies. It has a structure that challenges the algorithm with difficult problems, especially in overcoming local optimum points and reaching the global best result. The robustness of the competitive structure of the SSA algorithm has been demonstrated experimentally.

In the CEC 2020 function set, F1 represents a unimodal structure, evaluating the algorithm's capacity to surpass local optima. This allows an analysis of the convergence performance. Upon examining the result of the SSA algorithm on the F1 function, it can be concluded that it exhibits a significantly superior performance compared to alternative algorithms, indicating a robust convergence performance. Functions F5, F6, and F7 exhibit a hybrid structure, while F8, F9, and F10 have a composition structure. These functions harbor numerous local optima, showcasing both the algorithm's ability to avoid local optima and its balance between exploration and exploitation. The SSA algorithm demonstrates a superior performance on these functions compared to alternative algorithms, implying a strong balance between exploration and exploitation.

Functions F2, F3, and F4 have a multimodal structure, assessing an algorithm's ability for early convergence and overcoming local optima. While the SSA algorithm may not exhibit the utmost performance on these functions, it still achieves competitive results. Upon inspecting Fig. 2 alongside Table VI, it is evident that the SSA

TABLE III. STATISTICAL RESULTS OF ALGORITHMS VIA CEC 2019

Function	Metric	SSA	PDO	HGS	AOA	HHO
F01	Mean	8.8064E+09	6.7995E+04	**4.5615E+04**	2.4349E+09	5.2612E+04
	Std	1.2064E+10	2.6059E+04	1.1296E+04	6.1103E+09	**5.3745E+03**
	Best	2.9787E+08	4.4845E+04	**3.9036E+04**	7.5185E+05	4.4909E+04
	Worst	4.7209E+10	1.4308E+05	9.8576E+04	2.1395E+10	**6.3537E+04**
F02	Mean	1.7348E+01	1.7806E+01	1.7343E+01	1.9314E+01	**1.7361E+01**
	Std	1.6047E−02	4.7927E−01	**1.1814E−12**	4.1514E−01	1.0900E−02
	Best	**1.7343E+01**	1.7347E+01	**1.7343E+01**	1.8391E+01	1.7345E+01
	Worst	1.7407E+01	1.8998E+01	**1.7343E+01**	1.9848E+01	1.7397E+01
F03	Mean	**1.2702E+01**	**1.2702E+01**	**1.2702E+01**	1.2703E+01	1.2702E+01
	Std	**9.0336E−15**	1.0691E−06	1.0164E−07	1.0947E−03	8.6649E−06
	Best	**1.2702E+01**	**1.2702E+01**	**1.2702E+01**	**1.2702E+01**	1.2702E+01
	Worst	**1.2702E+01**	**1.2702E+01**	**1.2702E+01**	1.2706E+01	**1.2702E+01**
F04	Mean	**3.5255E+01**	1.4406E+04	3.9087E+01	1.1587E+04	1.6897E+02
	Std	**1.6125E+01**	5.7814E+03	2.1402E+01	6.2419E+03	6.6621E+01
	Best	1.2934E+01	6.7489E+03	**9.9531E+00**	4.7120E+03	7.3361E+01
	Worst	**6.7656E+01**	2.8405E+04	8.8556E+01	3.0665E+04	3.3801E+02
F05	Mean	**1.1879E+00**	4.2123E+00	1.2027E+00	4.0258E+00	2.3724E+00
	Std	**1.2113E−01**	7.0672E−01	1.5064E−01	8.0967E−01	5.9370E−01
	Best	1.0738E+00	2.8865E+00	**1.0370E+00**	2.3112E+00	1.3620E+00
	Worst	**1.6761E+00**	5.8989E+00	1.8255E+00	5.3789E+00	3.9881E+00
F06	Mean	4.8555E+00	9.4121E+00	**3.9102E+00**	8.9709E+00	9.4357E+00
	Std	1.5940E+00	1.2391E+00	9.9071E−01	**7.9521E−01**	1.1137E+00
	Best	2.1198E+00	7.1749E+00	**1.6615E+00**	6.8619E+00	6.9697E+00
	Worst	8.0604E+00	1.1680E+01	**5.7114E+00**	1.0582E+01	1.1338E+01
F07	Mean	3.7014E+02	8.3096E+02	**2.2725E+02**	2.3495E+02	3.0854E+02
	Std	2.3619E+02	1.9259E+02	1.8923E+02	**1.2448E+02**	1.6533E+02
	Best	−2.1483E+02	4.4307E+02	−7.8153E+01	**2.5017E+01**	3.7071E+01
	Worst	8.1575E+02	1.2490E+03	6.2378E+02	**4.2756E+02**	6.7806E+02
F08	Mean	**5.1431E+00**	6.2460E+00	5.5573E+00	5.4913E+00	5.8203E+00
	Std	8.5162E−01	**4.2642E−01**	6.1332E−01	5.4100E−01	5.4000E−01
	Best	**3.2477E+00**	5.0425E+00	3.7442E+00	4.1483E+00	4.6600E+00
	Worst	6.3720E+00	6.9344E+00	**6.3490E+00**	6.5249E+00	6.6779E+00
F09	Mean	**2.6354E+00**	1.5526E+03	2.9150E+00	8.9154E+02	3.2700E+00
	Std	**1.5928E−01**	3.7459E+02	3.1843E−01	5.0173E+02	4.6170E−01
	Best	2.4177E+00	6.6172E+02	**2.4016E+00**	6.2336E+00	2.6435E+00
	Worst	**3.0975E+00**	2.4810E+03	3.8112E+00	1.7613E+03	4.3586E+00
F10	Mean	2.0048E+01	2.0304E+01	**2.0019E+01**	2.0148E+01	2.0248E+01
	Std	1.0245E−01	1.4464E−01	**2.6424E−02**	5.9806E−02	1.4820E−01
	Best	1.9979E+01	2.0087E+01	1.9999E+01	2.0063E+01	**1.9969E+01**
	Worst	2.0378E+01	2.0602E+01	**2.0099E+01**	2.0271E+01	2.0600E+01

TABLE IV. COMPARISON OF ALGORITHMS BY MEANS OF WILCOXON SIGNED-RANK TEST (CEC 2019)

Function	Metrics	SSA-PDO	SSA-HGS	SSA-AOA	SSA-HHO
F1	*P*	1.7344e−06	1.7344e−06	3.6000E−03	1.7344e−06
	W/T/L	**W**	**W**	**W**	**W**
F2	*P*	1.7344e−06	1.7344e−06	1.7344e−06	3.0650e−04
	W/T/L	**W**	**W**	**W**	**W**
F3	*P*	5.6061e−06	1.5600E−02	1.7344e−06	1.7344e−06
	W/T/L	**W**	**W**	**W**	**W**
F4	*P*	1.7344e−06	5.0380E−01	1.7344e−06	1.7344e−06
	W/T/L	**W**	L	**W**	**W**
F5	*P*	1.7344e−06	6.435E−01	1.7344e−06	1.7344e−06
	W/T/L	**W**	L	**W**	**W**
F6	*P*	1.7344e−06	1.9600E−02	1.7344e−06	1.9209e−06
	W/T/L	**W**	**W**	**W**	**W**
F7	*P*	1.7344e−06	5.3000E−03	2.5600E−02	1.6500E−01
	W/T/L	**W**	**W**	**W**	L
F8	*P*	3.8822e−06	5.9800E−02	1.6500E−01	1.3595e−04
	W/T/L	**W**	L	L	**W**
F9	*P*	1.7344e−06	3.3173e−04	1.7344e−06	2.3534e−06
	W/T/L	**W**	**W**	**W**	**W**
F10	*P*	8.4661e−06	4.2840E−01	9.6266e−04	1.3601e−05
	W/T/L	**W**	L	**W**	**W**

algorithm consistently and steadily converges toward the optimal point without early convergence issues for functions F1, F5, F6, F7, F8, F9, and F10. Additionally, for F2, the SSA algorithm shows stability after the 300th iteration, avoiding the stagnation observed in alternative algorithms at different iterations. Function F3 demonstrates successful convergence, showing a slower yet balanced convergence compared to the top-performing HGS algorithm. For F4, the algorithm gets stuck at certain points and shows stagnation after the 250th iteration, a phenomenon observed in other algorithms as well.

D. Prediction with SSA-MLP Hybrid Algorithm

The Artificial Neural Network (ANN) provides a solution to non-linear problems, whether unsupervised or supervised. For ANN training, if classification is the goal, a supervised learning model is selected, and in this paper, the guiding algorithm is the SSA algorithm. The fact that the MLP method has a few parameters such as the momentum coefficient, tolerable error rate, learning rate, number of hidden layer nodes and layers, and the number of iterations enables it to be used without any prior knowledge, and it is superior to traditional systems with the advantages of having adaptive flexible learning processes, optimizing non-linear problems, and hybridizing with different methods. However, the most important disadvantages are the need for a large number of repetitions, the convergence rate depends on the characteristics of the dataset, the need for multiple experiments to determine the number of nodes for the best architecture, and the black box feature of the network causes the need for different analyses to ensure causality. When classical MLP training models and metaheuristic-based MLP architecture are compared, the metaheuristic-based method does not get stuck at optimal points in the local area, is not dependent on the starting point, and can achieve more optimal results depending on the balance between exploration and exploitation processes [27].

The flowchart in Fig. 3 illustrates the supervised learning model of the SSA-MLP hybrid system. In this system, the dataset is divided into two groups: 90% training and 10% testing, and an MLP architecture is adopted as the objective function of the metaheuristic algorithm. In this architecture, the input layer and output layer are protected in accordance with the data set. While creating the hidden layer, one was added to twice the number of nodes in the input layer, that is, it consisted of $2 * 8 + 1 = 17$ nodes. This number of nodes enabled the most optimal results to be achieved. The objective in statistical learning models is to achieve the highest possible classification of a target dataset. In a classification problem, a specific example from the dataset is labeled with one of the existing classes. Models created for this purpose have adjustable parameters to perform well. In this section, the prediction of the energy problem is achieved through SSA-MLP training, and the results are reported with statistical metrics.

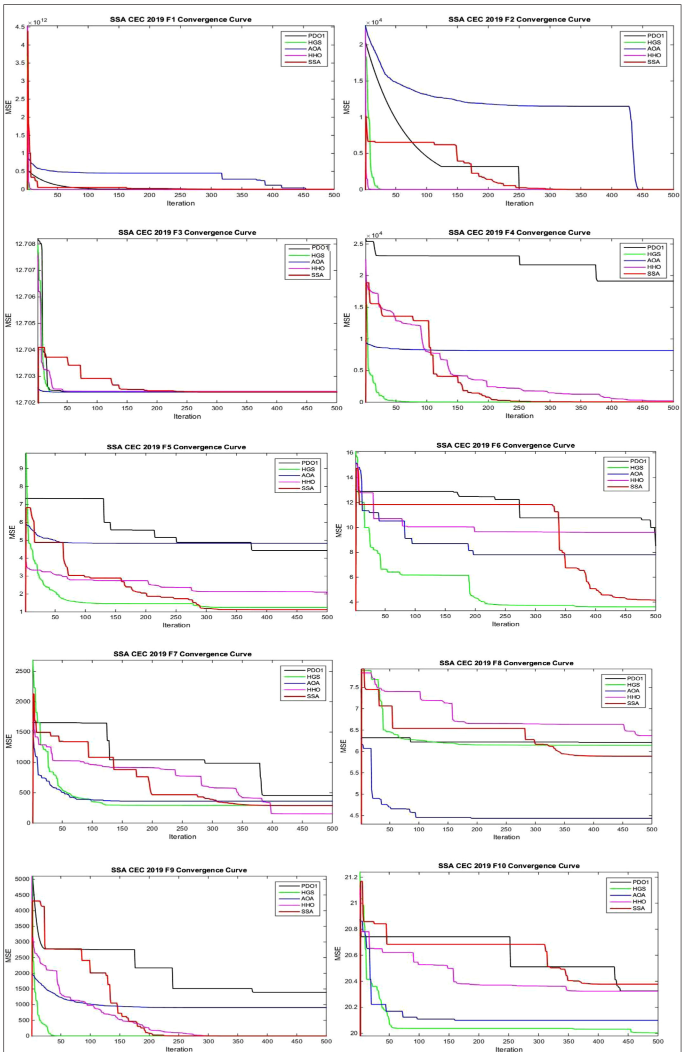

Figure 1. Converge curve of algorithms via CEC 2019.

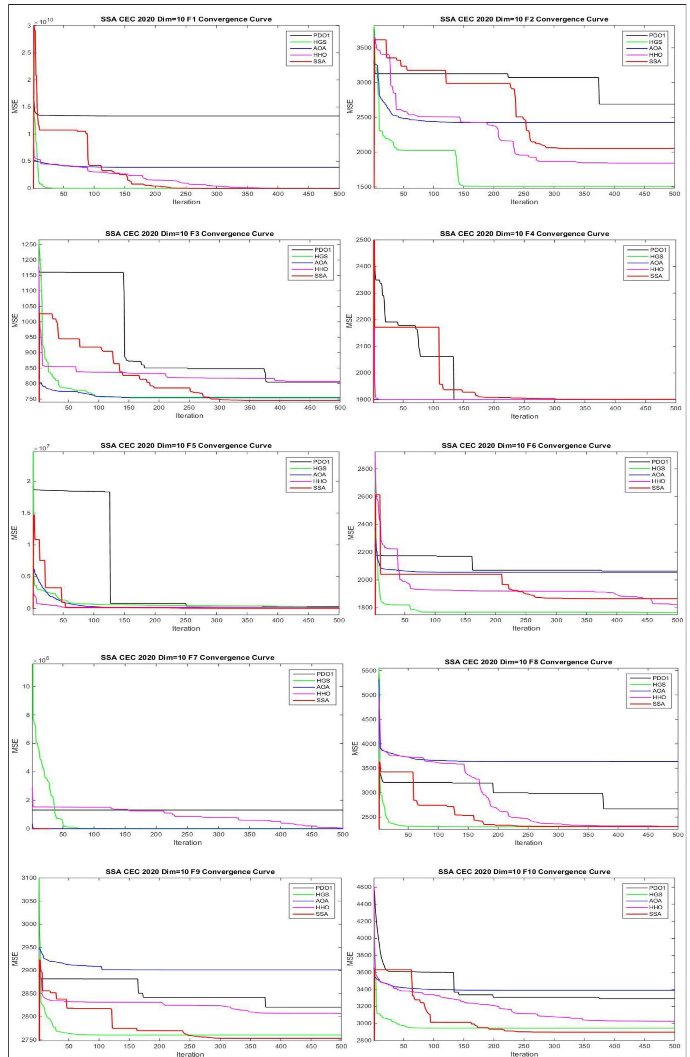

Figure 2. Converge curve of algorithms via CEC 2020.

TABLE V. CEC2020 FUNCTIONS DEFINITY PARAMETERS

Functions	Dimension	Search Interval	Fitting Value
F1: Shifted and Rotated Bent Cigar Function	10	[−100,100]	100
F2: Shifted and Rotated Schwefel's Function	10	[−100,100]	1100
F3: Shifted and Rotated Lunacek bi-Rastrigin Function	10	[−100,100]	700
F4: Expanded Rosenbrok's plus Griewank's Function	10	[−100,100]	1900
F5: Hybrid Function 1 (N=3)	10	[−100,100]	1700
F6: Hybrid Function 2 (N=4)	10	[−100,100]	1600
F7: Hybrid Function 3 (N=5)	10	[−100,100]	2100
F8: Composition Function 1 (N=3)	10	[−100,100]	2200
F9: Composition Function 2 (N=4)	10	[−100,100]	2400
F10 : Composition Function 3 (N=5)	10	[−100,100]	2500

E. Energy Efficiency Problem

The main objective of the proposed method and the studied dataset is to efficiently estimate heating load (HL) and cooling load (CL) to assist engineers in the construction of optimal energy-efficient buildings. The calculation of HL and CL in building designs holds great importance for the proper selection of indoor climate control equipment. Reducing these loads is critical for energy savings [49]. An effective forecasting strategy is needed to optimize energy consumption. Different studies have been conducted using the UCI dataset [32] for HL and CL predictions.

Irfan et al. employed machine learning algorithms, specifically artificial neural networks and deep neural networks, to address the energy problem. They demonstrated the superiority of this strategy by comparing it with 16 different algorithms [30]. Tien et al. conducted an extensive review of the literature regarding the problem-solving capabilities of machine learning algorithms [50]. Nilashi et al. trained the data by applying Expectation Maximization (EM) clustering and Adaptive Neuro-Fuzzy Inference System (ANFIS) methods through the dataset specified in this study. They gave the importance of HL and CL load predictions

TABLE VI. STATISTICAL RESULTS OF ALGORITHMS VIA CEC 2020

Function	Metric	SSA	PDO	HGS	AOA	HHO
F1	Mean	**3.6623E+03**	8.5352E+09	7.8432E+03	1.0478E+10	3.4794E+06
	Std	**3.2699E+03**	4.1008E+09	4.2935E+03	4.6193E+09	9.4448E+06
	Best	**1.0627E+02**	2.9881E+09	2.1263E+02	2.8185E+09	2.9156E+05
	Worst	**1.2733E+04**	1.9938E+10	1.2742E+04	2.0059E+10	5.0434E+07
F2	Mean	1.9993E+03	2.6486E+03	**1.6903E+03**	2.2388E+03	2.1276E+03
	Std	**1.7937E+02**	2.9189E+02	2.1321E+02	3.1185E+02	2.7888E+02
	Best	1.3872E+03	2.0348E+03	**1.2337E+03**	1.6337E+03	1.5941E+03
	Worst	2.2151E+03	3.0570E+03	**2.0309E+03**	3.0150E+03	2.6069E+03
F3	Mean	7.4130E+02	8.1483E+02	**7.4075E+02**	8.0237E+02	7.8766E+02
	Std	**1.3581E+01**	2.8797E+01	1.8342E+01	1.7566E+01	2.0224E+01
	Best	7.2164E+02	7.7724E+02	**7.0597E+02**	7.5386E+02	7.4963E+02
	Worst	**7.7724E+02**	9.0590E+02	7.9616E+02	8.4126E+02	8.3316E+02
F4	Mean	1.9017E+03	**1.9000E+03**	**1.9000E+03**	**1.9000E+03**	**1.9000E+03**
	Std	**5.9752E-01**	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Best	1.9008E+03	**1.9000E+03**	**1.9000E+03**	**1.9000E+03**	**1.9000E+03**
	Worst	1.9034E+03	**1.9000E+03**	**1.9000E+03**	**1.9000E+03**	**1.9000E+03**
F5	Mean	**1.9623E+04**	4.1855E+05	8.9957E+04	3.6849E+05	6.0535E+04
	Std	**2.7224E+04**	2.3041E+05	1.4403E+05	1.8964E+05	5.2291E+04
	Best	**3.1198E+03**	2.8989E+04	3.2650E+03	1.3920E+04	5.2928E+03
	Worst	**1.1208E+05**	8.7327E+05	7.0653E+05	7.4014E+05	1.8151E+05
F6	Mean	**1.7353E+03**	2.0749E+03	1.7804E+03	2.0895E+03	1.8828E+03
	Std	**9.0788E+01**	1.3240E+02	9.3024E+01	2.0559E+02	1.1178E+02
	Best	**1.6084E+03**	1.8394E+03	1.6129E+03	1.7630E+03	1.6220E+03
	Worst	**1.8777E+03**	2.2988E+03	2.0342E+03	2.8252E+03	2.1521E+03
F7	Mean	**4.4954E+03**	5.4502E+05	1.2243E+04	8.3203E+05	5.4682E+04
	Std	**1.8042E+03**	8.1288E+05	1.1550E+04	1.4718E+06	1.2584E+05
	Best	2.7256E+03	1.2492E+04	**2.4422E+03**	2.9293E+03	3.1130E+03
	Worst	**9.5660E+03**	3.2751E+06	3.2394E+04	6.8501E+06	6.6883E+05
F8	Mean	2.2956E+03	2.9446E+03	2.3372E+03	**3.2013E+03**	2.3461E+03
	Std	**2.2508E+01**	3.7750E+02	1.7930E+02	3.8606E+02	1.9947E+02
	Best	**2.2192E+03**	2.4980E+03	2.3003E+03	2.6428E+03	2.2462E+03
	Worst	**2.3050E+03**	4.1352E+03	3.2864E+03	3.9394E+03	3.3984E+03
F9	Mean	**2.7415E+03**	2.8402E+03	2.7613E+03	2.8666E+03	2.7959E+03
	Std	4.7187E+01	**2.1015E+01**	5.0827E+01	7.2937E+01	1.1836E+02
	Best	2.5000E+03	2.7939E+03	2.5000E+03	2.7112E+03	**2.4246E+03**
	Worst	**2.7839E+03**	2.8772E+03	2.7977E+03	3.0175E+03	2.9633E+03
F10	Mean	2.9356E+03	3.3282E+03	2.9397E+03	3.4088E+03	**2.9322E+03**
	Std	2.8658E+01	2.1243E+02	**2.8222E+01**	2.8598E+02	5.5981E+01
	Best	2.8977E+03	3.0283E+03	2.8979E+03	3.1332E+03	**2.7563E+03**
	Worst	**3.0242E+03**	4.0540E+03	3.0243E+03	4.1498E+03	3.0285E+03

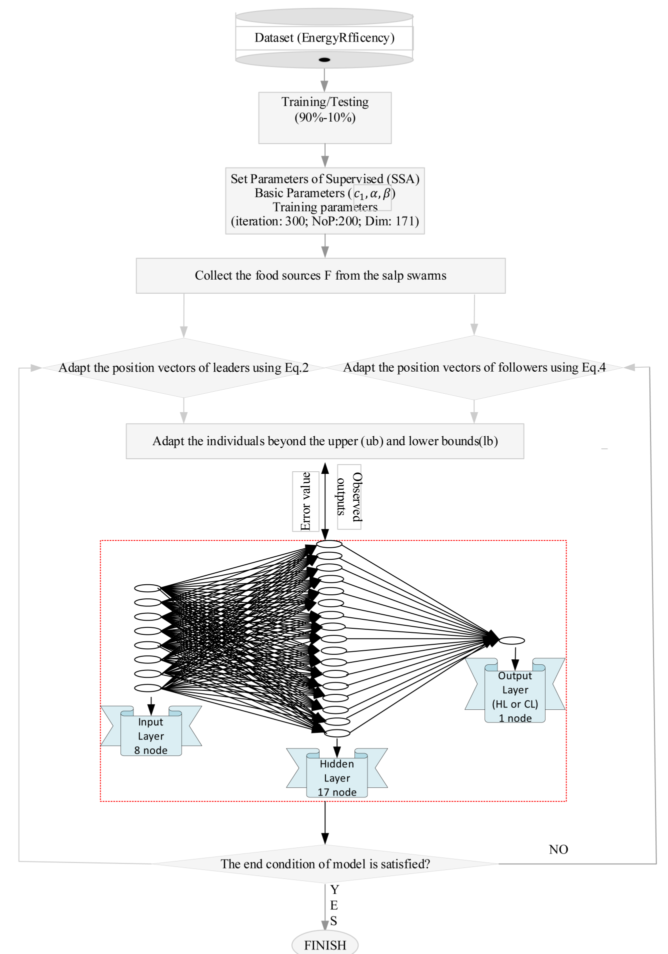

Figure 3. Flowchart of SSA-MLP training of supervised learning diagram.

in smart buildings by comparing many methods [51]. Roy et al., in their study, performed HL and CL factor predictions by comparing deep neural network, Gaussian process regression, minimax probability machine regression methods, and drew attention to the importance of these predictions in smart buildings [29]. Wang et al. proposed a hybrid extreme learning machine-fuzzied hunger games search model (ELM-Fuzz-HGS) for the measurement of indoor individual thermal comfort in order to create an ecologically beneficial and comfortable space, compared it with ELM training of different metaheuristic algorithms and reported that the prediction power increased [52]. Another example of metaheuristic-based MLP training is the model proposed by Kosarirad et al. for noise removal of big data sonar database and the classification rates are quite high [53]. In order to increase comfort and minimize energy cost in ecological smart buildings, Wang et al. developed a metaheuristic-based Green building energy optimization system model and stated that chimp optimization algorithm is the best algorithm to optimize energy saving [54]. He et al. proposed a metaheuristic-based algorithm model to enhance energy efficiency and extend the network lifetime. They also introduced a hybrid hierarchical chimp optimization algorithm for efficient clustering and multi-hop routing procedures [55].

F. SSA-MLP Training

In this paper, the problem encompassing HL and CL obtained from UCI [32] will be optimized and predicted using the MLP method. Within the scope of prediction, two primary outcomes will be sought. The first is the classification rate, as given in (5) (Class_Rate), and the other is the mean square error (MSE) rate, as provided in (6). Among competitive algorithms, the algorithm with the highest Class_Rate and the lowest MSE rate will be considered successful.

$$Class\_Rate = 100\left(\frac{CR}{NoS}\right) \tag{5}$$

where CR is the current rate resulting from a run and NoS refers to the number of samples in the dataset.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(P_i - O_i\right)^2 \tag{6}$$

where P_i is predicted values, O_i is the observed values, and N is the number of observations.

The basic features of the dataset are given in Table VIII below.

When Table X is examined, the SSA algorithm can produce different datasets from the other three algorithms except HGS and is superior. In other words, the SSA algorithm can serve as a unique algorithm in MLP training.

When examining Table IX, the results of effective HVAC algorithms with specific error rates are provided. The successful use of meta-heuristic algorithms in MLP training has been observed in this paper within the energy sector. In this table, as with the lowest MSE rate observed in the SSA algorithm, it is also seen that the SSA algorithm has the highest classification rate. The prominence of the SSA algorithm in addressing buildings with different features and its superiority in evaluating all these features together for output loads indicate the suitability of using this algorithm in calculating HL and CL in building designs. Fig. 4 is the convergence curve of the algorithms in MLP training. Convergence curves provide the opportunity to visually examine whether an algorithm is subject to early convergence in the process of optimizing the MLP architecture depending on the iterations and the early convergence status. When the Fig. 4 is examined, it will be observed that the SSA algorithm optimizes both HL and CL values better. In this way, it can be observed that the SSA algorithm is not exposed to early convergence and at the same time does not get stuck in local optima. Fig. 5 presents box-plot graph results. Box-plot graphs are useful measurement tools in statistics, displaying the median, first and third quartiles, lower and upper whiskers, and outliers. Observations reveal that SSA algorithm values are very close to each other for both CL and HL values, serving as two separate output values. Additionally, no outliers are observed in the HL values, and only one outlier is noticed in the CL values. Therefore, parallel values are provided, as indicated in Table VI.

IV. CONCLUSION

The aim of this paper is to investigate the problem-solving quality of the SSA, its ability to compete with alternative algorithms, and its ability to classify an artificial neural network with MLP architecture as a trainer.

TABLE VII. COMPARISON OF ALGORITHMS BY MEANS OF WILCOXON SIGNED-RANK TEST (CEC 2020)

Function	Metrics	SSA-PDO	SSA-HGS	SSA-AOA	SSA-HHO
F1	*P*	1.7344e−06	**5.7064e−04**	**1.7344e−06**	1.7344e−06
	W/T/L	W	**W**	**W**	W
F2	*P*	2.3534e−06	**1.6394e−05**	**2.6000E−03**	2.7000E−02
	W/T/L	W	**W**	**W**	W
F3	*P*	1.7344e−06	0.8130E+00	**1.7344e−06**	2.3534e−06
	W/T/L	W	L	**W**	W
F4	*P*	1.7344e−06	5.0380E−01	**1.7344e−06**	1.7344e−06
	W/T/L	W	L	**W**	W
F5	*P*	1.7344e−06	6.4350E−01	**1.7344e−06**	1.7344e−06
	W/T/L	W	L	**W**	W
F6	*P*	1.7344e−06	**1.9600E−02**	**1.7344e−06**	1.9209e−06
	W/T/L	W	**W**	**W**	W
F7	*P*	1.7344e−06	**5.3000E−03**	**2.5600E−02**	1.650E−01
	W/T/L	W	**W**	**W**	L
F8	*P*	3.8822e−06	5.9800E−02	1.6500E−01	1.3595e−04
	W/T/L	W	L	L	W
F9	*P*	1.7344e−06	**3.3173e−04**	**1.7344e−06**	2.3534e−06
	W/T/L	W	**W**	**W**	W
F10	*P*	8.4661e−06	4.2840E−01	**9.6266e−04**	1.3601e−05
	W/T/L	W	L	**W**	W

Salp swarm algorithm and alternative algorithms were evaluated using CEC 2019 and CEC 2020 function sets and as a result of these evaluations, it was observed that SSA was more advantageous in both test sets. Through the Wilcoxon signed-rank test, it was shown that the SSA algorithm has a robust structure in generating unique data sets by comparing it with alternative algorithms. This shows that the SSA algorithm can be used to solve problems of any size, but it is also observed that the algorithm is suitable for development and hybridization with different strategies due to its flexible structure and can offer a variety of solutions for different problems when different studies are carried out.

TABLE VIII. FEATURES OF ENERGY EFFICIENCY DATASET

Input/Output Variable	Variable Name	Definition of Variable
Input	I1	Relative compactness
	I2	Surface area
	I3	Wall area
	I4	Roof area
	I5	Overall height
	I6	Orientation
	I7	Glazing area
	I8	Glazing area distribution
Output	O1	Heating load
	O2	Cooling load

One way to solve the energy problem that our planet faces today is to ensure efficient energy use. In this context, minimizing HL and CL demand in energy efficient smart buildings will contribute to energy saving. It can be predicted that building designers can achieve the required efficiency in HVAC systems with the model produced in this study. A successful problem optimization was achieved with 100% prediction ability and MSE values of approximately 0.20. With this model, in the context of addressing the global energy crisis, the importance of selecting the right environmental control equipment, especially in the design of buildings aiming to improve the quality of climate control, is also emphasized. In this context, it is experimentally observed that metaheuristic algorithms, especially the SSA algorithm, can play an effective role in the calculation of HL and CL factors. In addition, the competitive nature of metaheuristic algorithms is emphasized and the opportunity to find and use the best algorithm within their framework is provided. The SSA-based SSA-MLP algorithm has shown successful results confirming its potential

TABLE IX. STATISTICAL RESULTS OF MLP TRAINING OF DATASET

Function	Metric	PDO	HGS	AOA	HHO	SSA
Output class heating load	Mean	2.0840E−01	2.0450E−01	2.1840E−01	2.1170E−01	**2.0380E−01**
	Std	4.9000E−03	8.2000E−03	7.7000E−03	**1.7000E−03**	3.3000E−03
	Best	2.0400E−01	**1.9760E−01**	2.0770E−01	2.0960E−01	**2.0030E−01**
	Worst	2.2140E−01	2.2520E−01	2.2860E−01	2.1410E−01	**2.1090E−01**
	Class. rate	%99.8667	%99.8560	%99.8660	%99.8702	**%100**
	Rank	3	2	5	4	1
Output class cooling load	Mean	2.8790E−01	2.8462E−01	2.9630E−01	2.8900E−01	**2.8460E−01**
	Std	**2.0000E−03**	8.2000E−03	5.2000E−03	3.7000E−03	2.7000E−03
	Best	2.8520E−01	**2.7890E−01**	2.9010E−01	2.8310E−01	2.8160E−01
	Worst	2.9210E−01	3.0210E−01	3.0610E−01	2.9410E−01	**2.9130E−01**
	Class. rate	%99.8667	%99.8560	%99.8660	%99.8698	**%100**
	Rank	3	2	5	4	**1**

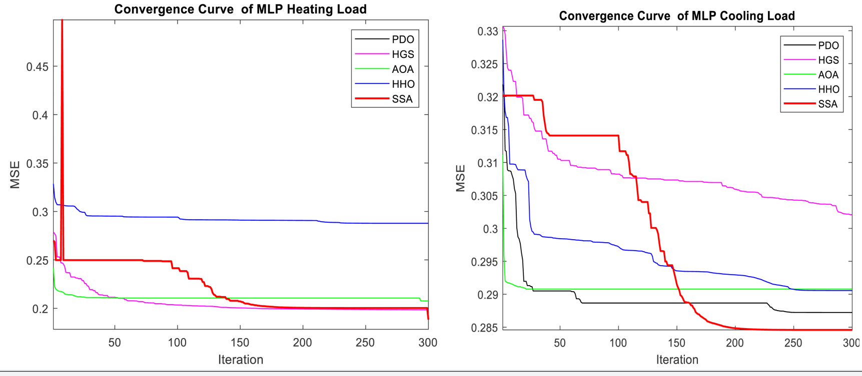

Figure 4. Converge curves of algorithms via MLP training.

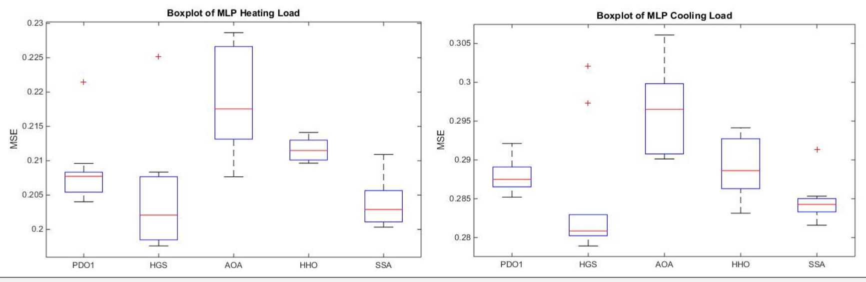

Figure 5. Box plots of algorithms via MLP training.

TABLE X. COMPARISON OF ALGORITHMS BY MEANS OF WILCOXON SIGNED-RANK TEST

Function	Metrics	SSA-PDO	SSA-HGS	SSA-AOA	SSA-HHO
Heating load	*P*-value	**9.8100E-03**	**9.2190E-01**	**2.0250E-03**	**2.0217E-03**
	W/T/L	**W**	**L**	**W**	**W**
Cooling load	*P*-value	**1.3700E-02**	**4.1360E-01**	**2.0001E-03**	**4.8800E-02**
	W/T/L	**W**	**L**	**W**	**W**

in this area. Future work can lead to new applications for solving energy problems through hybridization with both ANN and various machine learning algorithms, taking advantage of different reliable data sets and the flexibility of the SSA algorithm. It is believed that this paper will be a guiding reference in this field.

Peer-review: Externally peer-reviewed.

Author Contributions: Concept – E.E.; Design – İ.T.; Supervision – E.E.; Resources – F.Ş.; Materials – Ş.A.; Data Collection and/or Processing – E.E., İ.T.; Analysis and/or Interpretation – E.E., Ş.A.; Literature Search – İ.T, F.Ş.; Writing – E.E., Ş.A.; Critical Review – E.E.

Declaration of Interests: The authors have no conflict of interest to declare.

Funding: The authors declared that this study has received no financial support.

REFERENCES

1. A. R. Yıldız, *Tasarım Optimizasyon Problemlerinin Çözümü Için Çok Amaçlı Arama Tabanlı Bir Sistemin Geliştirilmesi*. Doctoral dissertation, Bursa Uludag University (Turkey), 2006.
2. Z. Cui, Y. Xu, and J. Zeng, "Social emotional optimization algorithm with random emotional selection strategy". *Theory and New Applications of Swarm, Intelligence*, pp. 33–50, 2012.
3. X. S. Yang, "Swarm-based metaheuristic algorithms and no-free-lunch theorems," *Theory and New Applications of Swarm Intelligence*, pp. 1–16, 2012.
4. L. S. Lasdon, R. L. Fox, and M. W. Ratner, "Nonlinear optimization using the generalized reduced gradient method,". "Revue française d'automatique, informatique, recherche opérationnelle," *RAIRO Recherche opérationnelle*, vol. 8, no. V3, pp. 73–103, 1974. [CrossRef]
5. E. V. Altay, and O. Altay, "Comparison of current metaheuristic optimization algorithms with CEC2020 test functions," *Dicle Univ. Eng. J.*, vol. 12, no. 5, pp. 729–741, 2021.
6. R. Salgotra, U. Singh, S. Singh, G. Singh, and N. Mittal, "Self-adaptive salp swarm algorithm for engineering optimization problems," *Appl. Math. Modell.*, vol. 89, pp. 188–207, 2021. [CrossRef]
7. Y. Çelik, İ. Yıldız, and A. T. Karadeniz, "A brief review of metaheuristic algorithms developed in the last three years," *Eur. J. Sci. Technol.*, pp. 463–477, 2019.
8. S. Mirjalili, and S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks: Theory and Applications*, 2019, pp. 43–55. [CrossRef]
9. J. Kennedy, and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International conference on neural networks*, Vol. 4. IEEE Publications, 1995. [CrossRef]
10. M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comp. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, 2006.
11. S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014. [CrossRef]
12. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019. [CrossRef]
13. Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Syst. Appl.*, vol. 177, p. 114864, 2021. [CrossRef]
14. N. Panagant, N. Pholdee, S. Bureerat, A. R. Yildiz, and S. Mirjalili, "A comparative study of recent multi-objective metaheuristics for solving constrained truss optimisation problems," *Arch. Comp. Methods Eng.*, vol. 28, no. 5, pp. 4031–4047, 2021. [CrossRef]
15. B. Aslan, and A. R. Yıldız, "Optimum design of automobile components using lattice structures for additive manufacturing," *Mater. Test.*, vol. 62, no. 6, pp. 633–639, 2020. [CrossRef]
16. T. Kunakote *et al.*, "Comparative performance of twelve metaheuristics for wind farm layout optimisation," *Arch. Comp. Methods Eng.*, vol. 29, no. 1, pp. 717–730, 2022. [CrossRef]
17. A. E. Hegazy, M. A. Makhlouf, and G. S. El-Tawel, "Improved salp swarm algorithm for feature selection," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 32, no. 3, pp. 335–344, 2020. [CrossRef]
18. L. Abualigah, M. Shehab, M. Alshinwan, and H. Alabool, "Salp swarm algorithm: A comprehensive survey," *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11195–11215, 2020. [CrossRef]
19. M. Masdari, M. Tahani, M. H. Naderi, and N. Babayan, "Optimization of airfoil Based Savonius wind turbine using coupled discrete vortex method and salp swarm *algorithm*," *J. Cleaner Prod.*, vol. 222, pp. 47–56, 2019. [CrossRef]
20. M. A. Tawhid, and A. M. Ibrahim, "Improved salp swarm algorithm combined with chaos," *Math. Comput. Simul.*, vol. 202, pp. 113–148, 2022. [CrossRef]
21. L. Abualigah, A. Diabat, and R. A. Zitar, "Orthogonal learning Rosenbrock's direct rotation with the gazelle optimization algorithm for global optimization," *Mathematics*, vol. 10, no. 23, p. 4509, 2022. [CrossRef]
22. I. Naruei, and F. Keynia, "Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems," *Eng. Comput.*, vol. 38, no. S4(Suppl 4), pp. 3025–3056, 2022. [CrossRef]
23. S. Alsamia, H. Albedran, and K. Jármai, "Comparative study of different metaheuristics on CEC 2020 benchmarks," In *Vehicle and Automotive Engineering*. Cham: Springer International Publishing, 2023, pp. 709–719. [CrossRef]
24. O. Altay, and E. V. Altay, "Investigation of slime mould algorithm and hybrid slime mould algorithms' performance in global optimization problems," *Dicle Univ. Eng. J.*, vol. 13, no. 4, pp. 661–671, 2023.
25. S. Haykin, and N. Network, "A comprehensive foundation," *Neural Netw.*, vol. 2, p. 41, 2004.
26. S. Mirjalili, "How effective is the Grey Wolf optimizer in training multi-layer perceptrons," *Appl. Intell.*, vol. 43, no. 1, pp. 150–161, 2015. [CrossRef]
27. E. Eker, M. Kayri, S. Ekinci, and D. İzci, "Comparison of swarm-based metaheuristic and gradient descent-based algorithms in artificial neural network training," *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J.*, vol. 12, no. 1, p. e29969, 2023. [CrossRef]
28. R. Cichowicz, and T. Jerominko, "Comparison of calculation and consumption methods for determining Energy Performance Certificates (EPC) in the case of multi-family residential buildings in Poland (Central-Eastern Europe)," *Energy*, vol. 282, p. 128393, 2023. [CrossRef]
29. S. S. Roy, P. Samui, I. Nagtode, H. Jain, V. Shivaramakrishnan, and B. Mohammadi-ivatloo, "Forecasting heating and cooling loads of buildings: A comparative performance analysis," *J. Ambient Intell. Human. Comput.*, vol. 11, no. 3, pp. 1253–1264, 2020. [CrossRef]
30. M. Irfan, R. Faizir, W. Widianto, and A. Faruq, "Prediction of residential building energy efficiency performance using deep neural network," *IAENG Int. J. Comput. Sci.*, vol. 48, no. 3, pp. 731–737, 2021.
31. A. Çam, *Bina Enerji Modellemesi yöntemiyle konut yapısı enerji analizi: Ankara Yeşiltepe Yapı Kooperatifi örneği. (Yayımlanmamış yüksek lisans tezi)*," Konya: Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü, Mimarlık Anabilim Dalı, 2023.
32. A. Tsanas, and A. Xifara, *Energy Efficiency*. UCI Machine Learning Repository, 2012. [CrossRef]
33. A. E. Ezugwu, J. O. Agushaka, L. Abualigah, S. Mirjalili, and A. H. Gandomi, "Prairie dog optimization algorithm," *Neural Comput. Appl.*, vol. 34, no. 22, pp. 20017–20065, 2022. [CrossRef]
34. F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems" *Appl. Intell.*, vol. 51, no. 3, pp. 1531–1551, 2021. [CrossRef]
35. S. Dhawan, R. Gupta, A. Rana, and S. Sharma, "Various swarm optimization algorithms: Review, challenges, and opportunities," *Soft Computing for Intelligent Systems: Proceedings of ICSCIS 2020*, pp. 291–301, 2021.
36. T. A. A. Ali, Z. Xiao, J. Sun, S. Mirjalili, V. Havyarimana, and H. Jiang, "Optimal design of IIR wideband digital differentiators and integrators using salp swarm algorithm," *Knowl. Based Syst.*, vol. 182, p. 104834, 2019. [CrossRef]
37. C. Wang *et al.*, "An efficient salp swarm algorithm based on scale-free informed followers with self-adaption weight," *Appl. Intell.*, vol. 53, no. 2, pp. 1759–1791, 2023. [CrossRef]
38. S. Ekinci, B. Hekimoğlu, and S. Kaya, "Tuning of PID controller for AVR system using salp swarm algorithm," in international conference on artificial intelligence and data processing (IDAP). IEEE Publications, 2018, pp. 1–6. [CrossRef]
39. H. Bingö, and M. Yıldırım, "Salp swarm algorithm, a swarm-based approach for global optimization," *Fırat Univ. J. Sci.*, vol. 33, no. 1, pp. 51–59, 2021.

40. F. D. Atlı, *Improving the Salp Swarm Optimization Algorithm for Training Artificial Neural Networks*. Doctoral dissertation. Necmettin Erbakan University (Turkey), 2022.
41. A. R. Yildiz, and P. Mehta, "Manta ray foraging optimization algorithm and hybrid Taguchi salp swarm-Nelder–Mead algorithm for the structural design of engineering components," *Mater. Test.*, vol. 64, no. 5, pp. 706–713, 2022. [CrossRef]
42. H. Zhang *et al.*, "A multi-strategy enhanced salp swarm algorithm for global optimization," *Eng. Comput.*, vol. 38, no. 2, pp. 1177–1203, 2022. [CrossRef]
43. M. Tubishat, S. Ja'afar, M. Alswaitti, S. Mirjalili, N. Idris, M. A. Ismail and M. S. Omar, 2021. "Dynamic salp swarm algorithm for feature selection," *Expert Syst. Appl.*, vol. 164, p. 113873.
44. R. M. Rizk-Allah, A. E. Hassanien, M. Elhoseny, and M. Gunasekaran, "A new binary salp swarm algorithm: Development and application for optimization tasks," *Neural Comput. Appl.*, vol. 31, no. 5, pp. 1641–1663, 2019. [CrossRef]
45. D. Wang, Y. Zhou, S. Jiang, and X. Liu, "A simplex method-based salp swarm algorithm for numerical and engineering optimization," in Intelligent Information Processing IX: 10th IFIP TC 12 International Conference, IIP 2018, Nanning, China, October 19–22, 2018, Proceedings 10. Springer International Publishing, 2018, pp. 150–159.
46. R. A. Ibrahim, A. A. Ewees, D. Oliva, M. Abd Elaziz, and S. Lu, "Improved salp swarm algorithm based on particle swarm optimization for feature selection," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 8, pp. 3155–3169, 2019. [CrossRef]
47. A. E. Romeh, and S. Mirjalili, "Multi-robot exploration of unknown space using combined meta-heuristic salp swarm algorithm and deterministic coordinated multi-robot exploration," *Sensors (Basel)*, vol. 23, no. 4, p. 2156, 2023. [CrossRef]
48. W. Qiao, M. Khishe, and S. Ravakhah, "Underwater targets classification using local wavelet acoustic pattern and multi-layer Perceptron neural network optimized by modified Whale Optimization Algorithm," *Ocean Eng.*, vol. 219, p. 108415, 2021. [CrossRef]
49. A. Zeng, H. Ho, and Y. Yu, "Prediction of building electricity usage using Gaussian Process Regression," *J. Build. Eng.*, vol. 28, p. 101054, 2020. [CrossRef]
50. P. W. Tien, S. Wei, J. Darkwa, C. Wood, and J. K. Calautit, "Machine learning and deep learning methods for enhancing building energy efficiency and indoor environmental quality–a review," *Energy Ai*, vol. 10, p. 100198, 2022. [CrossRef]
51. M. Nilashi, M. Dalvi-Esfahani, O. Ibrahim, K. Bagherifard, A. Mardani, and N. Zakuan, "A soft computing method for the prediction of energy performance of residential buildings," *Measurement*, vol. 109, pp. 268–280, 2017. [CrossRef]
52. L. Wang, M. Khishe, M. Mohammadi, and A. Mahmoodzadeh, "Extreme learning machine evolved by fuzzified hunger games search for energy and individual thermal comfort optimization," *J. Build. Eng.*, vol. 60, p. 105187, 2022. [CrossRef]
53. H. Kosarirad, M. Ghasempour Nejati, A. Saffari, M. Khishe, and M. Mohammadi, "Feature selection and training multilayer Perceptron neural networks using grasshopper optimization algorithm for design optimal classifier of big data sonar," *J. Sens.*, vol. 2022, 1–14, 2022. [CrossRef]
54. Q. Wang, G. Chen, M. Khishe, B. F. Ibrahim, and S. Rashidi, "Multi-objective optimization of IoT-based green building energy system using binary metaheuristic algorithms," *J. Build. Eng.*, vol. 68, p. 106031, 2023. [CrossRef]
55. S. He, Q. Li, M. Khishe, A. Salih Mohammed, H. Mohammadi, and M. Mohammadi, "The optimization of nodes clustering and multi-hop routing protocol using hierarchical chimp optimization for sustainable energy efficient underwater wireless sensor networks," *Wirel. Netw.*, vol. 30, no. 1, pp. 233–252, 2024. [CrossRef]

Erdal Eker was graduated from Van Yüzüncü Yıl University, in Mathematics. He received his Msc from Ataturk University, Türkiye, in Applied Mathematics and received his PhD from Yüzüncü Yıl University in Statistics. He is currently Asst. Prof. at Muş Alparslan University and working on Artificial intelligence, the applications of metaheuristic optimization, and statistic.

Şeyma Nur Atar was graduated from Atatürk University in Department of Civil Engineering. She is doing her master's degree in Nuclear Energy and Energy Systems at Muş Alparslan University.

Fatih Şevgin was born in Van in 1982. He was graduated from Dicle University, Faculty of Engineering and Architecture, Department of Civil Engineering. He received his master's degree from Fırat University, Faculty of Engineering, Department of Civil Engineering, Hydraulics and received his PhD from from Dicle University Faculty of Engineering, Department of Civil Engineering, Department of Hydraulics, in 2021. He is currently Asst. Prof. at Muş Alparslan University.

İhsan Tuğal was graduated from Fırat University, in Computer Engineering. He received his Msc from Fırat University, Türkiye, in Computer Engineering and received his PhD from İnönü University in Computer Engineering. He is currently Asst. Prof. at Muş Alparslan University and working on machine learning and graphs.