

## Optimizing Lung Cancer Classification with Extreme Learning Machine and Ant Lion Optimization for Enhanced Early Detection



Vidhya Rengasamy<sup>1\*</sup>, Mirnalinee Thanka Nadar<sup>2</sup>

<sup>1</sup> Department of Information Technology, Agni College of Technology, Chennai 600130, India

<sup>2</sup> Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering (SSN), Kalavakkam, Chennai 603110, India

Corresponding Author Email: [vidhya.it@act.edu.in](mailto:vidhya.it@act.edu.in)

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.410447>

### ABSTRACT

**Received:** 17 December 2023

**Revised:** 22 March 2024

**Accepted:** 15 June 2024

**Available online:** 31 August 2024

#### Keywords:

*hyperparameter tuning, optimization, metaheuristic algorithm, extreme learning machine, lung cancer*

Lung cancer emerges as a notable cancer affecting individuals of all genders on a global scale. Timely detection in its early stages significantly increases the chances of survival. In recent years, the advent of automatic lung cancer detection systems has played a significant role in enhancing diagnostic rates. Despite the advantages presented by machine learning models over traditional methods and their breakthroughs in various image classification tasks, accurately classifying lung cancer remains a challenge. This challenge is attributed to the complexity involved in selecting an appropriate machine learning model and fine-tuning hyperparameters. This paper aims to enhance the performance of a lung cancer classification system by optimizing hyperparameters in the Extreme Learning Machine (ELM) using metaheuristic optimization algorithms. To achieve this, Ant Lion Optimization algorithms are employed to determine optimal weight values for ELM. The novelty of this work lies in the application of ALO to enhance the performance of ELM specifically for lung cancer diagnosis, addressing a crucial gap in existing methodologies. Initially, features are extracted from Convolutional Neural Network (CNN). Subsequently, the optimal weight values and features are utilized in the ELM for the classification of Lung CT images as benign or malignant. The impact of applying hyperparameter optimization is assessed on two benchmark datasets, LIDC-IDRI and KAGGLE. The accuracy of lung cancer prediction using our method reaches 99.5% on the LIDC-IDRI dataset and 99.3% on the KAGGLE dataset. The findings of this study suggest that the proposed method outperforms existing approaches in the diagnosis of lung cancer.

## 1. INTRODUCTION

Machine learning algorithms find broad application across diverse domains and especially plays major role in medical image diagnosis. Tuning the hyperparameters of a machine learning model is essential for adapting it to different problems. The performance of a machine learning model is significantly influenced by the dataset, the chosen training algorithms and the choice of the optimal hyperparameter configuration. The selection of an appropriate training algorithm can substantially change the outcome of a model. While certain algorithms exhibit excellent performance with specific datasets, they may encounter challenges with others. Additionally, enhancing performance is achievable by fine-tuning the hyperparameters that control the training processes of an algorithm. Achieving this often demands a profound understanding of machine learning algorithms and the application of suitable hyperparameter optimization techniques. In the context of lung cancer detection using a machine learning model, initial hyperparameter values are typically selected randomly. Hyperparameters are crucial in machine learning algorithms as they govern the behavior of training algorithms and heavily influence the overall performance of machine learning models.

The values of these hyperparameters might yield satisfactory results on specific datasets, but their effectiveness can vary considerably when applied to unseen data [1]. Consequently, the development of an effective hyperparameter optimization algorithm for any machine learning method would significantly enhance the effectiveness of machine learning processes.

Many researchers commonly establish hyperparameter values before commencing the training of a machine learning model in order to construct an appropriate model, and the worth of these standards unwaveringly influences the scheme's performance [2]. The methods typically working for selecting hyperparameters include the trial and error method, forcing expert understanding, and utilizing meta-heuristic optimization techniques [3].

The trial and error approach entail testing a limited amount of hyperparameter values, training the method by apiece value, and then choosing the value that yields the optimal training for the model. Conversely, the expert practice approach entails evaluating suitable principles based on individual research exposure or embracing principles derived from prior research outcomes. In contrast, meta-heuristic search algorithms use a search model to methodically discover hyperparameter values

over a predetermined range and determine the best solution following a specified number of repetitions. It is crucial to highlight that the efficacy of meta-heuristic search algorithms depends significantly on the formulation of a fitness metric [4].

This research study approaches the hyperparameter tuning issue as a task in optimization, utilizing the Ant Lion Optimization (ALO) algorithm to tackle it. This chapter delves into the utilization of the ALO optimization algorithm, elucidating how it contributes to optimizing the weights of ELM and consequently enhancing performance. The Extreme Learning Machine (ELM) technique initializes input weights arbitrarily, which does not ensure proximity to the finest values crucial for the system to excel in specific tasks. This randomness in weights can lead ELM to memorize training data rather than generalize it to unobserved data. In the realm of medical image classification, achieving a high accuracy level holds significant importance as it can impact both diagnostic and therapeutic decisions for patients. Selecting algorithms tailored to specific medical image diagnoses becomes crucial in this context.

Further enhancing the system's performance necessitates an filtering procedure integrated into the ELM framework. Optimally adjusted weights play a pivotal role in enabling the system to excel in assigned tasks.

In practical use of ELM, there is an emphasis on considering optimal weight values to improve the model's performance and boost its capability to generalize to unseen data. To address these challenges, an introduction has been made of the Ant Lion Optimization algorithm for selecting optimal input weights, presenting a potential solution to enhance ELM's efficiency and performance.

The primary objectives of hyperparameter tuning of weights of ELM are:

- Optimal weights are tuned to improve the generalization performance of the ELM. Generalization refers to the ability of the model to perform well on unseen data. Weights generated in a random manner can result in overfitting, characterized by the model fitting the training data too precisely but encountering challenges in generalizing to novel data instances.. Optimal weights are chosen to strike a balance between fitting the training data and generalizing to unseen data.

- Optimal weights are chosen to enhance the learning speed of the ELM. Randomly generated weights may not be well-suited for the specific task at hand, leading to slower convergence during the training process. Optimal weights, on the other hand, are selected to facilitate faster convergence, reducing the time required for the model to learn the underlying patterns in the data.

- Optimal weights contribute to improved accuracy and overall model performance. Randomly generated weights may result in suboptimal configurations that hinder the model's ability to make accurate predictions. Optimal weights are selected to maximize the model's predictive power on the given task.

The remaining sections of this paper are structured as follows: Related work in the literature is outlined in Section 2. The proposed approach is presented in Section 3. Section 4 contains the results and discussions, and the work concluded in Section 5.

## 2. LITERATURE REVIEW

This section discusses the application of hyperparameter

tuning in machine learning algorithms, examining previous research and its relevance to hyperparameter tuning.

The authors [5] introduced the Bayesian Optimization-Support Vector Machine (BO-SVM) model for classification. This technique was employed to fine-tune hyperparameters for several machine learning models such as RF, SVM, LR, and DT, using a dataset with 23 features and 195 instances. The target feature had binary class labels (1 for PD, 0 for non-PD). Performance evaluations was done using the metrics such as accuracy, AuC, recall and error rates. The results revealed that SVM, optimized through BO, outperformed the other state-of-the-art models.

The authors [6] introduces a method for adjusting hyperparameters utilizing parameter-setting-free harmony search (PSF-HS) approach. The PSF-HS algorithm treats the hyperparameter as the harmony, generating harmony memory rationalized grounded on CNN loss. Simulations using CNN architectures demonstrate performance improvement by tuning hyperparameters, offering advancements over previous CNN architectures.

The work [7] utilized a bio-inspired optimization approach to recognize optimal hyperparameters for a CNN based approach to predict the Parkinson disease. The method iteratively minimized classification errors through backpropagation to the ACO optimizer.

The authors [8] introduces a reinforcement based efficient learning algorithm designed to autonomously adjust parameters for an optimal network conformation in a given problem. The paper demonstrates the algorithm's capability to unite on an optimal solution for the MNIST dataset using asynchronous reinforcement learning.

The authors [9] proposed a heart disease prediction system employing Hyperparameter Optimization (HPO) techniques such as Grid Search, Randomized Search, and TPOT Classifier. Their approach enhanced Random Forest and XG Boost classifier models, achieving the highest accuracy of 97.52% for the Cleveland Heart Disease Dataset. In the Z-Alizadeh Sani dataset, Random Forest with TPOT Classifier and Randomized Search yielded peak accuracies of 80.2%, 73.6%, and 76.9% for diagnosing vessel stenosis, surpassing existing studies significantly.

The authors [10] suggest employing Design of Experiments (DOE) methodology, specifically factorial designs, for hyperparameter screening, and Response Surface Methodology (RSM) for tuning machine learning algorithms. The methodology is demonstrated through a case study utilizing RF algorithm, the work has several merits such as reduced training time, and an improved parameter selection

The work [11] employed machine learning techniques for detecting fraudulent transactions, utilizing a genetic algorithm (GA) to optimize hyperparameters and comparing it with grid search (GS) methods. The chosen classifiers—random forest (RF), AdaBoost (AB), logistic regression (LR), decision tree (DT), and support vector machine (SVM)—were evaluated. Results revealed the genetic algorithm's superior performance over GS in terms of accuracy, precision, recall, and F1\_score, demonstrating efficiency within a shorter timeframe.

The authors [12] compared four bio-inspired metaheuristics-Bat Algorithm, Firefly Algorithm, Particle Swarm Optimization Algorithm, and Social Emotional Optimization Algorithm-to evaluate efficiency while maintaining effectiveness. Results from various classification problems revealed differences in efficiency, with certain bio-inspired algorithms requiring fewer SVM evaluations to find

optimal hyperparameters. The Bat Algorithm emerged as the recommended choice for SVM hyperparameter tuning based on its superior performance.

The work [13] aimed to optimize support vector machine (SVM) hyperparameters for tunnel boring machine advance rate (TBM AR) prediction using gray wolf optimization (GWO), whale optimization algorithm (WOA), and moth flame optimization (MFO). The study utilized 1,286 data samples from a Malaysian water transfer tunnel with seven input variables and one output variable. Hybrid SVM models were constructed with GWO, WOA, and MFO optimization techniques, assessing accuracy through statistical indices. Results indicated that the MFO-SVM model achieved the highest accuracy with R2 (0.9623 and 0.9724), RMSE (0.1269 and 0.1155), and VAF (96.24 and 97.34%) for training and test stages, showcasing its effectiveness in predicting TBM AR.

The authors [14] presented a method utilizing convolutional neural networks (CNNs) and a genetic algorithm (GA) for noninvasive classification of Glioma grades through magnetic resonance imaging (MRI). The CNN architecture is evolved using GA, departing from traditional trial and error or predefined structures. Bagging, an ensemble algorithm, is applied to the best GA-evolved model to reduce prediction error variance. The method achieves 91.9% accuracy in classifying three Glioma grades and 96.2% accuracy in classifying Glioma, Meningioma, and Pituitary tumor types, highlighting its effectiveness for early-stage brain tumor diagnosis via MRI.

The authors [15] introduces an algorithm that divides the solution space into subspaces, allocating search agents based on each subspace's "potential." This potential is estimated using objective values, probe solution results, and computation time. The work is associated with several ML algorithms utilizing grid search, ACO, PCA, and PSO. Simulation results, using Taipei city government data, show the proposed method's superior performance in terms of mean absolute percentage error compared to other forecasting methods in the study.

The authors [16] introduced a novel variant of particle swarm optimization (PSO) called cPSO-CNN, specifically designed for optimizing Convolutional Neural Networks (CNNs) hyperparameters determined by architecture. This method incorporates a confidence function derived from a compound normal distribution, extracting expert knowledge to enhance traditional PSO's exploratory power. To better handle the diverse range of CNN hyperparameters, cPSO-CNN converts scalar acceleration coefficients into vectors. Additionally, a linear prediction model expedites the ranking of PSO particles, reducing the computational load for fitness function calculation. Test results underscore cPSO-CNN's competitive performance, showcasing its efficiency in CNN hyperparameter optimization compared to existing algorithms.

The authors [17] explore the relationship between machine learning model performance and hyperparameters using Gaussian processes. They formulate hyperparameter tuning as an optimization problem and employ Bayesian optimization, leveraging the Bayesian theorem. The method establishes a prior over the optimization function, updating it with information from previous samples. Experimental results demonstrate the efficacy of the approach in finding optimal hyperparameters for popular models, including random forest and neural networks, while considering time costs.

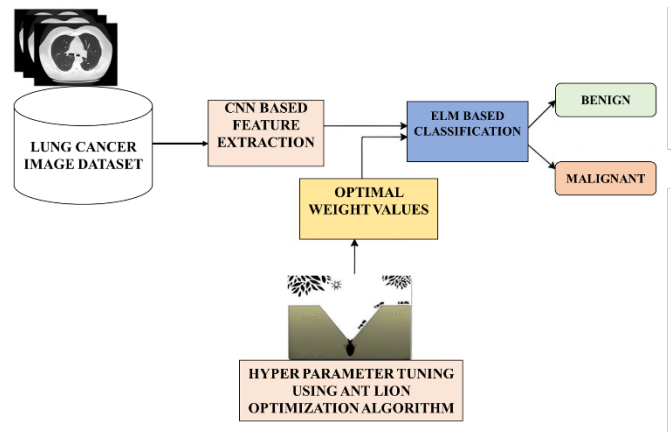
The authors [18] introduced an automatic method for optimizing hyperparameters and designing structures using

enhanced metaheuristic algorithms. The paper presents improved versions of tree growth and firefly algorithms, enhancing their original implementations. These modified metaheuristics are evaluated on standard benchmark functions, and the enhanced algorithms are then applied to hyperparameter optimization. Experiments on the MNIST image classification dataset demonstrate superior performance in classification accuracy and computational resource usage compared to other existing techniques.

The authors [19] proposed a novel hyper-parameter optimization methodology that combines the benefits of a genetic algorithm and Tabu Search. Two sets of contrast experiments are carried out to confirm the suggested algorithm's effectiveness. Good hyper-parameter values for deep convolutional neural networks are simultaneously sought after using the Tabu\_Genetic Algorithm and four other techniques. Based on experimental results, the suggested Tabu\_Genetic Algorithm finds a better model faster than Random Search and Bayesian optimization techniques.

The work [20] developed an ensemble model using pre-trained CNNs (VGG16 and VGG19) for plant disease diagnosis based on leaf images. The challenge of manually optimizing CNN hyperparameters is addressed using orthogonal learning particle swarm optimization (OLPSO). An exponentially decaying learning rate (EDLR) schema enhances training efficiency, while random oversampling and undersampling tackle dataset imbalances. Comparative experiments demonstrate the proposed model's superior accuracy over other pre-trained CNN models, showcasing its effectiveness in plant disease diagnosis.

### 3. PROPOSED APPROACH



**Figure 1.** Proposed methodology for lung cancer prediction

This section introduces the proposed method. Figure 1 depicts the process flow for predicting lung cancer, encompassing three steps: feature representation, hyperparameter tuning, and classification. To begin, a convolutional neural network is utilized for feature representation. After feature extraction, the Ant Lion Optimization algorithm is employed to identify optimal weights. The extracted features and optimal weight values are subsequently input into an extreme learning machine for classification of CT lung images as benign and Malignant. The principal aim of the proposed Ant Lion optimization-based hyperparameter tuning in lung cancer prediction is to enhance diagnostic efficiency, mitigate overfitting, and improve generalization performance.

### 3.1 Ant lion optimization (ALO) algorithm

The Ant Lion Optimization (ALO) algorithm is a metaheuristic optimization technique inspired by the hunting behavior of antlions in nature. ALO mimics the process of antlions creating traps to capture ants, with ants exploring the search space and updating their positions based on fitness evaluations. The following pseudocode outlines the main steps of the ALO algorithm for optimizing the weights in the Extreme Learning Machine (ELM) model:

#### Algorithm: Ant Lion Optimization (ALO) for Optimal Weight Values

Begin

(1). Initialize population of ants and antlions with random weights within bounds.

(2). Calculate fitness for each ant and antlion based on objective function using

$$f(v) = \sum_{i=1}^n V_i + \prod_{j=1}^n V_j$$

(3). Identify elite antlion with the highest fitness.

(4). Set iteration parameters:  $t=1$ ,  $\text{max\_iterations}=100$ .

(5). While  $t < \text{max\_iterations}$ :

a. For each ant:

i. Select a random antlion using a roulette wheel mechanism.

ii. Compute random walks around for selected antlions

iii. Update ant positions based on random walks and fitness.

b. Recalculate fitness for all ants and replace antlions based on fitness comparison.

c. Update elite antlion if a better solution is found.

d. Increment  $t$  for the next iteration.

(6). Return weights from the best elite antlion as optimal weights for ELM.

End

For the CNN used in feature extraction from lung CT images, we employ a standard architecture consisting of convolutional layers, pooling layers, and fully connected layers. The specific parameters such as kernel size, number of filters, and activation functions are chosen based on empirical studies and prior literature in medical image analysis.

Similarly, the ELM model's parameters, including the number of hidden neurons, activation function, and regularization techniques, are set based on experimental validation and best practices. The choice of these parameters aims to balance model complexity and performance while avoiding overfitting on the training data.

### 3.2 Identifying optimal weight values for elm using ant lion optimization algorithm

ALO stands as a meta-heuristic technique inspired by the predatory behavior of antlions found in natural settings. At its heart, this algorithm comprises of ants and ant lions, embodying five sequential steps reflective of the hunting process: the erratic movement of ants, creation of traps, ensnaring ants within traps, seizing the trapped ants, and rebuilding the traps.

The ALO algorithm begins by establishing the initial positions and assessing the fitness of both ants and antlions. Factors such as the boundaries of the environment and the presence of antlion traps play a crucial role in influencing the movement of ants. When an ant discovers a more favorable position, it becomes prey for an antlion, which then takes over the ant's previous location. An elite antlion is chosen based on having the highest fitness among all antlions, signifying the most optimal parameter set discovered thus far.

As the algorithm progresses, ants start exploring the parameter space around the elite antlion, updating their parameter values accordingly. A roulette wheel mechanism is used to randomly select an antlion, guiding the algorithm's decision-making process. The selected antlion acts as a focal point for the random walk behavior of surrounding ants, mimicking how ants navigate and explore the nearby parameter space. Similarly, a random walk pattern is calculated for ants around each encountered antlion.

The ALO algorithm advances through iterations, during which ants traverse the search space and adjust their pheromone levels based on the quality of solutions encountered. This iterative approach permits ants to gradually converge towards better solutions by exploring promising parameter configurations. Following the update of each ant's position, their fitness is recalculated. If an ant's fitness exceeds that of its corresponding antlion, the antlion is substituted by the ant, thereby updating the elite antlion built on improved fitness. This tracking mechanism ensures the retention of the best parameter set throughout the optimization procedure.

The mathematical expression representing an ant's random walk behavior is as follows

$$S(t) = [0, ts(2\gamma(i_1) - 1), ts(2\gamma(i_2) - 1) \dots ts(2\gamma(i_r) - 1)] \quad (1)$$

where,  $ts$  denote the total sum,  $i$  denotes current iteration number,  $r$  is the maximum number of iterations, and  $\gamma$  is a random function defined by.

$$\gamma(i) = \begin{cases} 1 & \text{if } i > 0.5 \\ 0 & \text{if } i \leq 0.5 \end{cases} \quad (2)$$

Within Eq. (2), the variable  $r$  denotes a random number that follows a uniform distribution within the range of (0, 1). The algorithm's mechanism for ant movement relies on these random numbers, which helps in exploring a broader range of solutions and prevents the algorithm from getting trapped in local optimal solutions. Additionally, to ensure that all ants' random movements stay within the defined boundaries of the search space, a normalization process is conducted using the formula provided below.

$$M_j^i = \frac{(S_i - a_i)x(ubo_j^i - lbo_j^i)}{(b_j^i - a_j^i)} + lbo_j^i \quad (3)$$

where,  $ub_j^i$  and  $lb_j^i$  represents upper and lower bounds in the plane,  $b_i$  and  $b_j$  represents the  $i^{\text{th}}$  dimension.

When ants slide towards the antlion during the process, entering the antlion's trap triggers an action to shift the ant closer to the antlion. This action entails the antlion ejecting sand till the trapped ant slowly moves nearer. Mathematically, this operation is represented with vigorously dropping the hypersphere radius for the ant's random walk, which can be

expressed as follows:

$$lbo^i = \frac{lbo^i}{1 + 10 V_j^i} \quad (4)$$

$$ubo^i = \frac{ubo^i}{1 + 10 V_j^i} \quad (5)$$

where,  $w$  is a constant value for the current iteration. The accuracy level of the iteration  $i$  is given by

$$i > \begin{cases} 0.1 i w = 2 \\ 0.5 i w = 3 \\ 0.75 i w = 4 \\ 0.9 i w = 5 \\ 0.95 i w = 6 \end{cases} \quad (6)$$

When constructing a trap, the roulette wheel is utilized to mimic the hunting prowess of the antlion, and the ALO algorithm employs the roulette wheel to select the most suitable antlion, thereby enhancing the chances of capturing ants.

During the entrapment progression of ants in traps, the random walk of the ants is influenced by the location of the antlion trap. This influence can be mathematically described using the following formula:

$$ALO_i = \frac{R_{ALO}^i + R_{ALO}^j}{2} \quad (7)$$

where,  $R_{ALO}^i$  represent the ant's random walk,

The last phase involves capturing the ants and rebuilding the trap, during which the antlion trapped the ants that descend into the trap. Afterward, the antlion must adjust its fitness to correspond by most recent weights of the captured ants, as indicated in the equation below. This adaptation improve the ability to capture other ants.

$$ALO_i = Ant + Ant > ALO \quad (8)$$

The fitness function is defined below

$$f(v) = \sum_{i=1}^n V_i + \prod_{j=1}^n V_j \quad (9)$$

where,  $v$  is the weights,  $n$  is the total number of weights. Algorithm for optimal weight value is given below:

#### Algorithm: Ant Lion Optimization (ALO) for Optimal Weight Values

##### Inputs:

- max\_iterations: Maximum number of iterations
- num\_ants: Number of ants
- num\_antlions: Number of antlions
- bounds: Search space boundaries for weights
- w\_constant: Constant value for the iteration

##### Outputs:

- Optimal\_weights: Set of optimal weight values for the ELM model

**Procedure-**ALO\_Optimize\_Weights(max\_iterations, num\_ants, num\_antlions, bounds, w\_constant):

    Initialize ants' positions randomly within bounds

    Initialize antlions' positions randomly within bounds

    Calculate fitness for each ant and antlion using the fitness function

    Set elite\_antlion = antlion with highest fitness

    for iter from 1 to max\_iterations do:

        for each ant do:

            Generate random walk using Eqs. (1) to (5)

            Apply normalization using Eq. (3) to ensure within

bounds

            Calculate fitness for the new position

            if ant's fitness > corresponding antlion's fitness then:

                Replace antlion with this ant

        for each antlion do:

            Construct traps using roulette wheel selection (Eq. (5))

            Capture ants based on antlion positions and update

fitness

        Adjust antlion fitness based on captured ants' weights (Eq. (8))

        if elite\_antlion's fitness < highest fitness among antlions then:

            Update elite\_antlion with the best antlion

        Optimal\_weights = Weights of elite\_antlion

        return Optimal\_weights

# Example Usage:

max\_iterations = 100

num\_ants = 20

num\_antlions = 5

bounds = [lower\_bound, upper\_bound] # Define lower and upper bounds for weights

w\_constant = 2 # Constant value for the iteration

Optimal\_weights

=ALO\_Optimize\_Weights(max\_iterations, num\_ants, num\_antlions, bounds, w\_constant)

### 3.3 ELM classification

ELM is a category of neural network that stands out for its simplicity and fast learning speed. In ELM, the input-to-hidden layer weights are typically initialized randomly and then optimized during training using the Moore-Penrose pseudoinverse. The main idea of this study is to improve the accuracy of lung cancer classification by optimizing the hyperparameters of the ELM model. A Convolutional Neural Network (CNN) is initially employed for feature representation. CNN in this context are used to extract relevant features from lung CT images. The ALO algorithm, a metaheuristic optimization algorithm inspired by the foraging behavior of ant lions, is employed to find optimal weight values for the ELM. ALO optimizes the weights by simulating the hunting behavior of ant lions. The extracted features from the CNN and the optimal weight values obtained through ALO are then fed into the ELM for classification. The final output represents the prediction of the ELM model for a given input. In the context of lung cancer classification, this output can indicate whether the input lung CT image is classified as benign or malignant. Figure 2 shows the ELM classifier.

The input feature vector extracted after the CT lung image undergoes multiplication through the weight vector ( $v$ ), followed by addition of an optimal bias ( $b$ ), formerly processed by the activation function ( $g$ ). In Extreme Learning Machines (ELMs), the weights connecting the input layer to the hidden layer are initially random values. This implies that the input vector itself does not influence the initial configuration of the network. Consequently, the response of

each hidden neuron to the input data is dictated by a random blend of features derived from the input data.

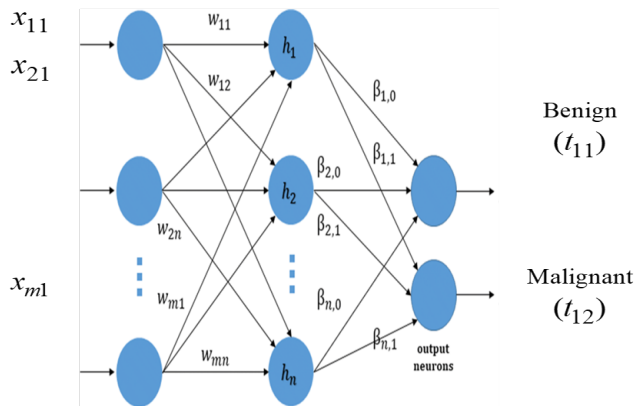


Figure 2. ELM architecture

The outcome for the features can be computed as follows

$$i_j = \begin{bmatrix} i_{1j} \\ i_{2j} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m \beta_{j,0} f(v_i x_j + b_i) \\ \sum_{i=1}^m \beta_{j,1} f(v_i x_j + b_j) \end{bmatrix} \quad (10)$$

The above Eq. (10) can be rewritten as

$$M = T \beta \quad (11)$$

where,  $M$  is the goal matrix and  $T$  is the output matrix. The weights connecting the hidden layer to the output layers are represented by the beta matrix. This matrix is computed using the Moore-Penrose pseudoinverse technique, which plays a crucial role in decreasing the least squares error among the original output and the expected output. The Moore-Penrose pseudoinverse technique calculates a generalized inverse of a matrix, even if the matrix is not square. The introduced system categorizes the CT lung images into either benign or malignant. Hence, the output layer consists of only two neurons.

Activation functions are utilized exclusively in the hidden layer of ELMs. The rectified linear unit activation function is commonly used which always returns the same value for positive integers, and always returns zero for negative integers. As a result, this function is less susceptible to the vanishing gradient issue which prevents the exponential growth in the computation required to operate a neural network.

Once the ELM has been trained using the population, predictions can be obtained by:

$$T = M \beta \quad (12)$$

Contrary to gradient-based techniques that necessitate iterative calculations for gradient determination and subsequent weight updates through training, ELM computes network parameters using the Moore-Penrose pseudoinverse method without extra iterations. This approach helps mitigate the risk of overfitting.

Optimizing the weight values in Extreme Learning Machine (ELM) contributes significantly to improving generalization, accuracy, and mitigating overfitting. Optimal weights, obtained through techniques such as Ant Lion Optimization, enable the model to generalize better to unseen data by

capturing relevant patterns without excessively tailoring the learning process to the training set. The refined weights enhance the accuracy of the model by aligning the neural network's mapping of input features to hidden layers with the underlying complexities of the data. Moreover, the process of obtaining optimal weights acts as a regularization mechanism, helping to prevent overfitting by restraining the model from fitting the training data too closely. By fine-tuning the weights based on global optimization, the ELM becomes more adept at discerning genuine patterns, striking a balance that not only maximizes accuracy on the training set but also ensures performance on diverse datasets, ultimately resulting in improved generalization and minimizing the risk of overfitting.

#### 4. PERFORMANCE METRICS

The evaluation of the proposed system's ability to classify CT lung cancer images into benign or malignant relies on classification accuracy, sensitivity, and specificity. Performance metrics, including True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN), are computed to track falsely predicted and exactly predicted data. Table 1 displays the presentation metrics utilized in the work.

Table 1. Performance metrics used for lung cancer prediction

Performance Metrics	Sensitivity	Accuracy	Specificity
Formulae	$\frac{TP}{TP + FN}$	$\frac{TP + TN}{TP + FN + Tn + FP}$	$\frac{TN}{TN + FP}$

#### 5. DISCUSSION OF RESULTS

The evaluation of the proposed ant Lion optimization based hyperparameter tuning for lung cancer prediction is assessed utilizing the evaluation parameters.

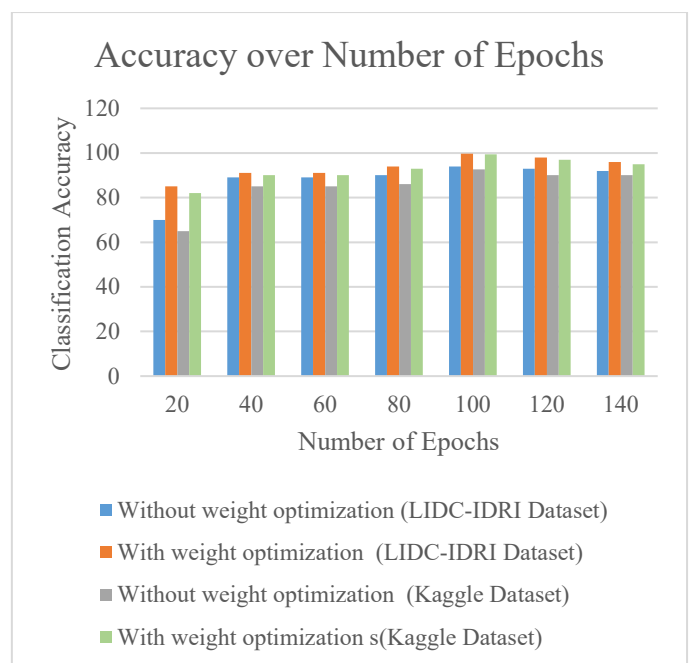


Figure 3. Classification accuracy over number of epochs

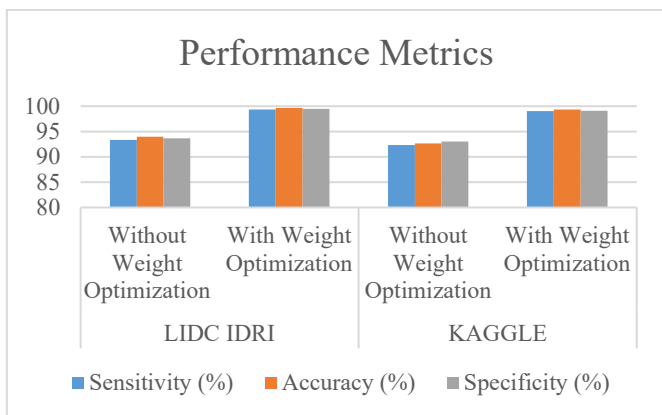


Figure 4. Performance comparison

Using the LIDC-IDRI and KAGGLE benchmark datasets, the accuracy of the proposed Ant lion optimization based hyperparameter tuning approach is measured. Both the optimal and non-optimal weight values for the classification accuracy are projected in the results. The graph shows that, for both datasets, Ant Lion optimization-based hyperparameter tuning has outperformed random weights. The graph plotting the number of epochs against classification accuracy is displayed in Figure 3. The graph shows that, for both datasets, accuracy increases gradually over time at regular intervals of epoch, peaking at epoch 100. Nevertheless, when both datasets have optimal weight values, the accuracy is constant between epochs 100 and 120 before slightly declining at epoch 140. After 100 epochs, the ant-lion optimization algorithm converges. However, the accuracy levels off after epoch 100. In summary, accuracy at 100 epochs yielded the finest results, overall., the construal indicates that models that have undergone hyperparameter tuning (i.e., optimal weight values) typically exhibit less accuracy fluctuation and are more stable than models that use random weights. Figure 4 shows the performance comparison of the proposed method with other current methods.

### 5.1 Results comparison

The remarkable accuracy of the proposed methodology was highlighted. For instance, an accuracy rate of 88% was achieved by Lima et al. [21] using Tree-of-Parzen-estimators, which assumed conditional independence among optimized variables. A 96% accuracy rate was demonstrated in the study of Lv et al. [22] utilizing the Minibatch Stochastic Gradient Descent (MB-SGD) method, although it was prone to overfitting. Meanwhile, an accuracy rate of 99.33% was attained by Alamgeer et al. [23] using the Moth Swarm Optimization (MSO) system to enhance hyperparameters such as learning rate, epoch count, and batch size in the LSTM model. However, this method incurred computational expenses as the number of hyperparameters increased.

Table 2. Comparative analysis of proposed method

Author	Architecture	Accuracy
Lima et al. [21]	Tree-of-Parzen-estimators	87.65
Lv et al. [22]	MB-SGD	95.45
Alamgeer et al. [23]	MSO	98.93
Proposed system	Ant Lion Optimization	99.67

In contrast, optimal values for the weights of the ELM were identified by the proposed system, leading to an exceptional accuracy rate of 99.50% as shown in Table 2.

In addition to quantitative metrics, it is crucial to discuss the qualitative aspects and pros and cons of each method:

- **Tree-of-Parzen-estimators:** While effective, this method relies on the assumption of conditional independence among optimized variables, which may limit its performance in capturing complex relationships in medical imaging data.
- **MB-SGD:** This technique shows good accuracy results but is susceptible to overfitting, especially with complex datasets like medical images, where generalization is crucial.
- **MSO applied to LSTM:** MSO optimization yields high accuracy, but it comes with increased computational expenses as the number of hyperparameters in LSTM models grows.
- **Proposed ELM with ALO optimization:** The ALO optimization method provides highly accurate weight optimization for ELM, enhancing generalization and mitigating overfitting. It strikes a balance between accuracy and computational efficiency, making it suitable for practical applications in medical image analysis.

### 5.2 Statistical analysis

In this section, we conduct a statistical analysis to determine the significance of the accuracy results obtained using the proposed Ant Lion Optimization (ALO) based hyperparameter tuning for lung cancer prediction.

#### Confidence Intervals

Confidence Intervals (CI) provide valuable insights into the precision of our accuracy estimates. We calculate confidence intervals around the reported accuracies to quantify the range within which the true accuracy of the model is likely to fall.

For example, with a confidence level of 95%, the confidence interval can be calculated as:

$$CI = Accuracy \pm Z * \sqrt{\frac{Accuracy * (1 - Accuracy)}{Total\ Samples}}$$

where,  $Z$  is the critical value from the standard normal distribution corresponding to the chosen confidence level.

#### P-values for Statistical Significance

To determine if the improvements in accuracy are statistically significant, we calculate p-values using appropriate statistical tests such as the t-test or ANOVA. The null hypothesis  $H_0$  assumes no significant difference between the accuracies of the proposed method and existing methods, while the alternative hypothesis  $H_1$  suggests a significant difference.

A low p-value (typically  $<0.05$ ) indicates that the observed improvements in accuracy are unlikely to occur by random chance and are thus statistically significant.

### 5.3 Generalization and clinical implications

Generalization refers to the ability of a machine learning model to perform well not only on the training dataset but also

on unseen data from similar distributions. In the context of lung cancer classification using the proposed Ant Lion Optimization (ALO) based hyperparameter tuning for Extreme Learning Machine (ELM), generalization is crucial for practical clinical deployment.

### Model Robustness and Generalization

We evaluate the robustness and generalization capability of our model through cross-validation on independent datasets. By training our model on one dataset and testing it on separate, unseen datasets, we assess its ability to generalize across different data distributions. This process helps validate the model's performance in real-world scenarios beyond the training data.

Furthermore, we employ techniques such as data augmentation and transfer learning to enhance the model's generalization. Data augmentation involves generating additional training data by applying transformations such as rotation, flipping, and scaling to the original images. Transfer learning leverages knowledge learned from pre-trained models on large datasets to improve performance on smaller, domain-specific datasets.

### Clinical Implications

The successful deployment of machine learning models in clinical settings requires careful consideration of various factors beyond accuracy metrics. These factors include model interpretability, explainability of predictions, regulatory compliance, integration with existing clinical workflows, and ethical considerations regarding patient data privacy and security.

Our study acknowledges these challenges and emphasizes the importance of model explainability in medical decision-making. By visualizing important features learned by the model, such as key image regions indicative of malignancy, we aim to enhance trust and understanding among healthcare professionals using the system.

## 6. CONCLUSION

This study explores the use of Ant Lion Optimization (ALO) algorithm for optimizing weights in diagnosing lung cancer through the Extreme Learning Machines (ELM) architecture. The aim of employing the ALO algorithm is to fine-tune the weights within the ELM framework to achieve optimal model performance. In the ELM model, the initial input weights are randomly assigned, without any assurance that they are close to the optimal values necessary for effective task handling. Optimized weights are crucial as they enable the model to effectively process the complexities inherent in medical images. Thus, the importance of determining optimal weight values cannot be overstated, as it significantly contributes to enhancing the network's performance and its ability to generalize well to unseen data. This consideration is pivotal when practically applying ELM in real-world scenarios. To overcome these challenges, the study introduces the Ant Lion Optimization algorithm, which selects the most suitable input weights. This approach offers a promising solution to boost the efficiency and overall performance of the ELM model. The evaluation of the application of hyperparameter optimization is conducted on two benchmark datasets, namely LIDC-IDRI and KAGGLE. Our method achieves an accuracy of 99.5% for lung cancer prediction on the LIDC-IDRI dataset and 99.3%

on the KAGGLE dataset. The outcome of this work shows that the proposed approach surpasses current methods in the diagnosis of lung cancer.

### Future Directions

Future research directions include collaborative efforts with healthcare providers to conduct prospective clinical studies validating the model's performance in real-time clinical environments. Integration with Picture Archiving and Communication Systems (PACS) and Electronic Health Records (EHRs) streamlines the deployment process and ensures seamless interaction with clinical workflows.

Moreover, ongoing model monitoring, validation, and updating protocols are essential to adapt to evolving data distributions, patient demographics, and medical practices. Continuous feedback loops between data scientists, clinicians, and patients contribute to iterative model improvement and increased confidence in AI-driven decision support systems.

## REFERENCES

- [1] Li, W., Cao, P., Zhao, D., Wang, J. (2016). Pulmonary nodule classification with deep convolutional neural networks on computed tomography images. *Computational and Mathematical Methods in Medicine*, 2016(1): 6215085. <https://doi.org/10.1155/2016/6215085>
- [2] Sarhan, B.B., Altwaijry, N. (2022). Insider threat detection using machine learning approach. *Applied Sciences*, 13(1): 259. <https://doi.org/10.3390/app13010259>
- [3] Akay, H., Kim, S.G. (2021). Reading functional requirements using machine learning-based language processing. *CIRP Annals*, 70(1): 139-142. <https://doi.org/10.1016/j.cirp.2021.04.021>
- [4] Yang, L., Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415: 295-316. <https://doi.org/10.1016/j.neucom.2020.07.061>
- [5] Elshewey, A.M., Shams, M.Y., El-Rashidy, N., Elhady, A.M., Shohieb, S.M., Tarek, Z. (2023). Bayesian optimization with support vector machine model for parkinson disease classification. *Sensors*, 23(4): 2085. <https://doi.org/10.3390/s23042085>
- [6] Lee, W.Y., Park, S.M., Sim, K.B. (2018). Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm. *Optik*, 172: 359-367. <https://doi.org/10.1016/j.ijleo.2018.07.044>
- [7] Singh, S., Janghel, R.R. (2022). Early diagnosis of Alzheimer's disease using ACO optimized deep CNN classifier. In *Ubiquitous Intelligent Systems: Proceedings of ICUIS 2021*, Springer Singapore, pp. 15-31. [https://doi.org/10.1007/978-981-16-3675-2\\_2](https://doi.org/10.1007/978-981-16-3675-2_2)
- [8] Neary, P. (2018). Automatic hyperparameter tuning in deep convolutional neural networks using asynchronous reinforcement learning. In *2018 IEEE International Conference on Cognitive Computing (ICCC)*, San Francisco, CA, USA, pp. 73-77. <https://doi.org/10.1109/ICCC.2018.00017>
- [9] Valarmathi, R., Sheela, T. (2021). Heart disease prediction using hyper parameter optimization (HPO) tuning. *Biomedical Signal Processing and Control*, 70:



103033. <https://doi.org/10.1016/j.bspc.2021.103033>
- [10] Lujan-Moreno, G.A., Howard, P.R., Rojas, O.G., Montgomery, D.C. (2018). Design of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study. *Expert Systems with Applications*, 109: 195-205. <https://doi.org/10.1016/j.eswa.2018.05.024>
- [11] Tayebi, M., El Kafhali, S. (2021). Hyperparameter optimization using genetic algorithms to detect frauds transactions. In *The International Conference on Artificial Intelligence and Computer Vision*. Cham: Springer International Publishing, pp. 288-297. [https://doi.org/10.1007/978-3-030-76346-6\\_27](https://doi.org/10.1007/978-3-030-76346-6_27)
- [12] Godínez-Bautista, A., Padierna, L.C., Rojas-Domínguez, A., Puga, H., Carpio, M. (2018). Bio-inspired metaheuristics for hyper-parameter tuning of support vector machine classifiers. *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*, 115-130. [https://doi.org/10.1007/978-3-319-71008-2\\_10](https://doi.org/10.1007/978-3-319-71008-2_10)
- [13] Zhou, J., Qiu, Y., Zhu, S., Armaghani, D.J., Li, C., Nguyen, H., Yagiz, S. (2021). Optimization of support vector machine through the use of metaheuristic algorithms in forecasting TBM advance rate. *Engineering Applications of Artificial Intelligence*, 97: 104015. <https://doi.org/10.1016/j.engappai.2020.104015>
- [14] Anaraki, A.K., Ayati, M., Kazemi, F. (2019). Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. *Biocybernetics And Biomedical Engineering*, 39(1): 63-74. <https://doi.org/10.1016/j.bbe.2018.10.004>
- [15] Tsai, C.W., Fang, Z.Y. (2021). An effective hyperparameter optimization algorithm for DNN to predict passengers at a metro station. *ACM Transactions on Internet Technology (TOIT)*, 21(2): 1-24. <https://doi.org/10.1145/3410156>
- [16] Wang, Y., Zhang, H., Zhang, G. (2019). cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyperparameters of convolutional neural networks. *Swarm and Evolutionary Computation*, 49: 114-123. <https://doi.org/10.1016/j.swevo.2019.06.002>
- [17] Wu, J., Chen, X.Y., Zhang, H., Xiong, L.D., Lei, H., Deng, S.H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, 17(1): 26-40. <https://doi.org/10.11989/JEST.1674-862X.80904120>
- [18] Bacanin, N., Bezdán, T., Tuba, E., Strumberger, I., Tuba, M. (2020). Optimizing convolutional neural network hyperparameters by enhanced swarm intelligence metaheuristics. *Algorithms*, 13(3): 67. <https://doi.org/10.3390/a13030067>
- [19] Guo, B., Hu, J., Wu, W., Peng, Q., Wu, F. (2019). The Tabu\_Genetic algorithm: A novel method for hyperparameter optimization of learning algorithms. *Electronics*, 8(5): 579. <https://doi.org/10.3390/electronics8050579>
- [20] Darwish, A., Ezzat, D., Hassanien, A.E. (2020). An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis. *Swarm and Evolutionary Computation*, 52: 100616. <https://doi.org/10.1016/j.swevo.2019.100616>
- [21] Lima, L.L., Ferreira Junior, J.R., Oliveira, M.C. (2021). Toward classifying small lung nodules with hyperparameter optimization of convolutional neural networks. *Computational Intelligence*, 37(4): 1599-1618. <https://doi.org/10.1111/coin.12350>
- [22] Lv, E., Liu, W., Wen, P., Kang, X. (2021). Classification of benign and malignant lung nodules based on deep convolutional network feature extraction. *Journal of Healthcare Engineering*, 2021(1): 8769652. <https://doi.org/10.1155/2021/8769652>
- [23] Alamgeer, M., Mengash, H.A., Marzouk, R., Nour, M.K., Hilal, A.M., Motwakel, A., Zamani, A.S., Rizwanullah, M. (2022). Deep learning enabled computer aided diagnosis model for lung cancer using biomedical CT images. *Computers, Materials & Continua*, 73(1): 1437-1448. <https://doi.org/10.32604/cmc.2022.027896>