Nature inspired computing Final Report

1. Phase 1:

1.1. Select and justify a challenging dataset

Dataset Used

The dataset was constructed by **merging two publicly available sources**:

- **Kaggle Chest X-ray dataset** → *Normal class*
- **GitHub Chest X-ray dataset** → *COVID-19 and Pneumonia classes*

Why This Dataset Is Challenging

- **Class Imbalance**: COVID-19 samples were fewer compared to Normal and Pneumonia cases.
- **High Intra-class Similarity**: COVID-19 and Pneumonia X-rays share very similar visual patterns.
- **Noise and Artifacts**: Some X-rays contained annotations, arrows, or extreme brightness.
- **Medical Image Complexity**: Chest X-rays have low contrast and subtle discriminative features.

Justification

This dataset reflects **real-world medical conditions**, making it suitable for testing **robust optimization and search techniques** rather than relying on standard deep learning alone.

1.2. Build baseline deep learning model

Models Implemented

Two deep learning models were built as **baseline classifiers**:

Model	Description
DCSNN	Deep Convolutional Neural Network for multi-class classification
SCOVNET	Specialized CNN architecture designed for COVID-19 detection

Task

- Classify X-ray images into:
  - **COVID-19**
  - **Pneumonia**
  - **Normal**

Preprocessing Applied

- Normalization & Standardization
- Gaussian Smoothing
- Histogram Equalization
- Gamma Correction
- Oversampling to handle imbalance
- Removal of annotated, over/under-exposed images

These baseline models established a **reference point** before optimization.

1.3. Feature Selection using Ant Colony Algorithm

Why Ant Colony Optimization?

- Effective for **combinatorial feature selection problems**
- Mimics ants' ability to find optimal paths using pheromone trails
- Suitable for high-dimensional medical image features

Application in the Project

- ACO was used to:
  - Select the **most informative features**
  - Reduce feature redundancy
  - Improve classification efficiency

Outcome

- Reduced feature space
- Improved learning stability
- Enhanced model generalization before applying optimization algorithms

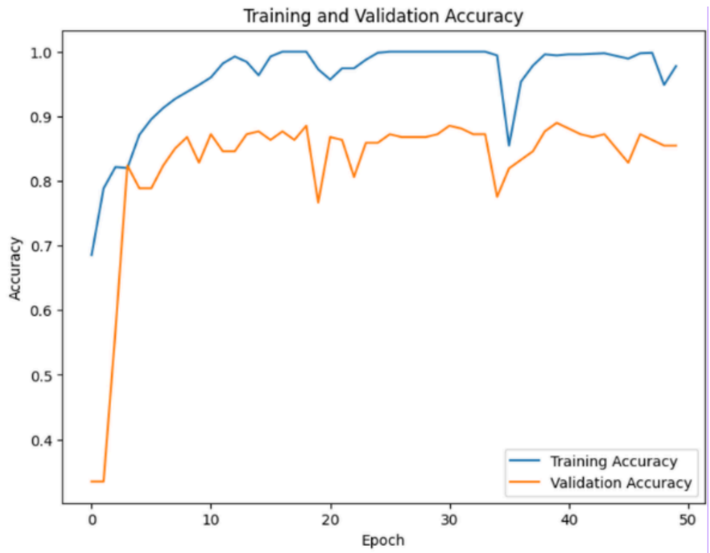
1.4. Apply 4 metaheuristic algorithms for model optimization

Four metaheuristic algorithms were applied to optimize model parameters and learning behavior:

Algorithm	Purpose

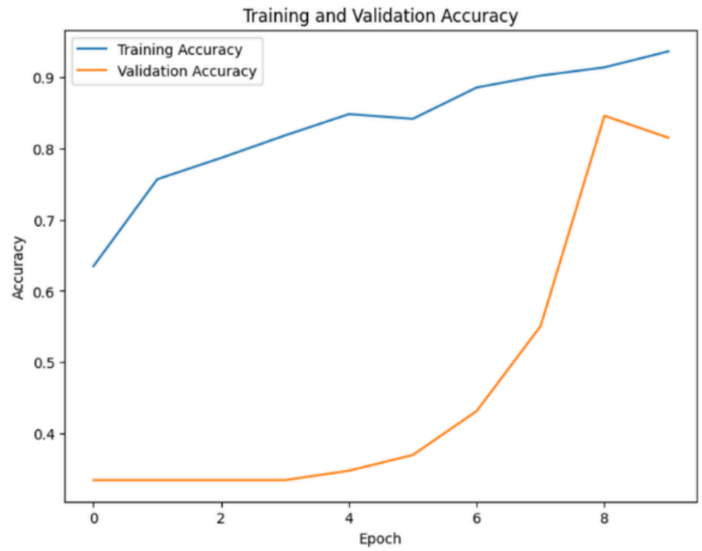
Hill Climbing	Fast local optimization
Tabu Search	Avoids local optima using memory
Simulated Annealing	Probabilistic global search
Ant Colony Optimization	Population-based exploration

1.4.1 Hill Climbing + DCSNN:



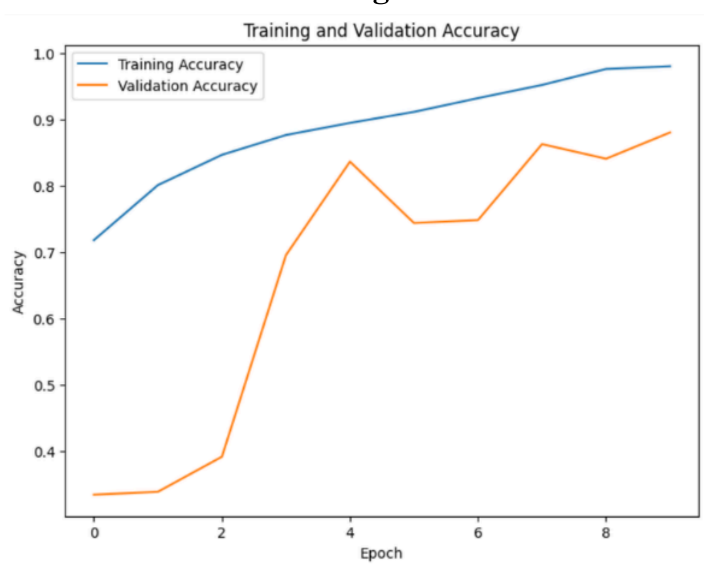
- Fast but unstable: The model reaches high accuracy very quickly but suffers from frequent, sharp performance drops (jitter).
- Overfitting: It achieves near-perfect training accuracy while leaving a consistent 10–15% gap in validation, showing it struggles to generalize

1.4.2 Tabu Search + DCSNN:



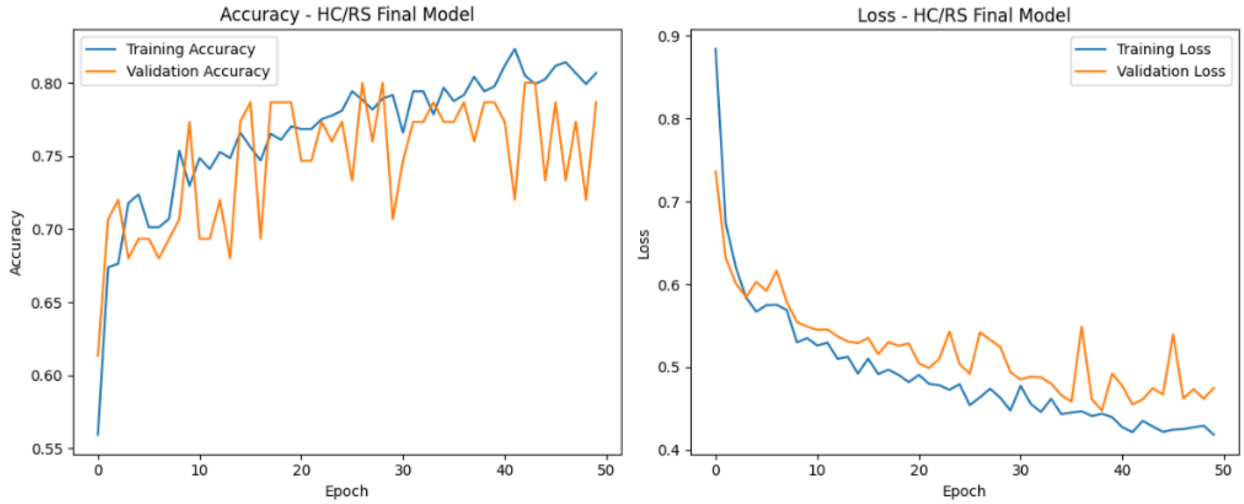
- Steady, focused growth: The training accuracy climbs consistently while the validation accuracy remains flat initially before surging after epoch 5.
- Balanced generalization: It shows a smoother learning curve than Hill Climbing, reaching a validation accuracy of 0.9163 by epoch 5 with less volatility.

1.4.3 Simulated Annealing + DCSNN:



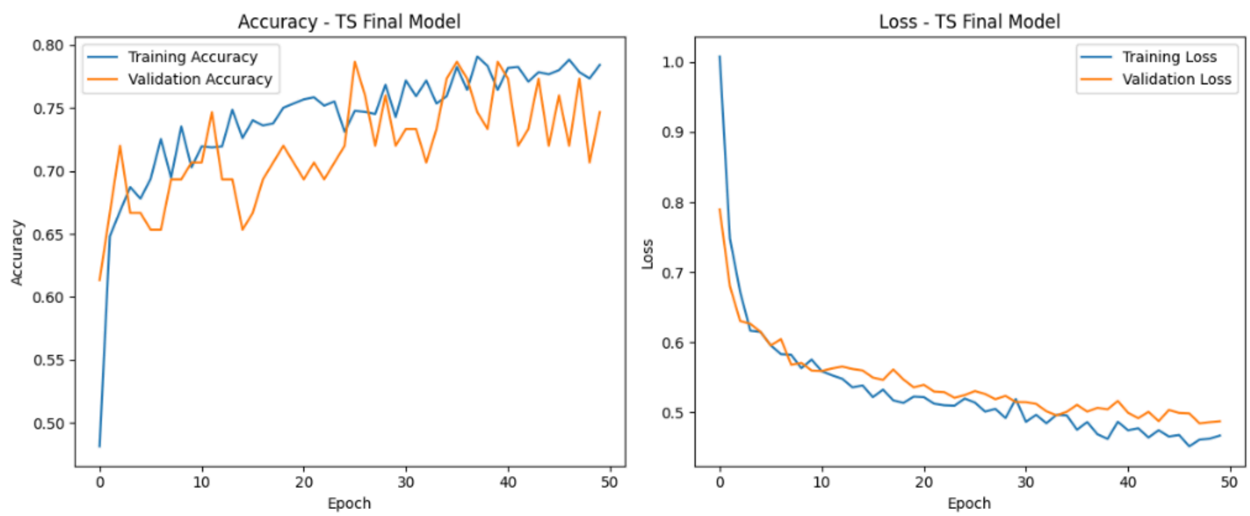
- Linear convergence: This algorithm shows a steady, almost linear improvement in training accuracy compared to others.
- Late-stage generalization: The validation accuracy remains low initially but surges significantly after epoch 2 to reach 0.8811 by epoch 5.

1.4.5 Hill Climbing + SCOVNET:



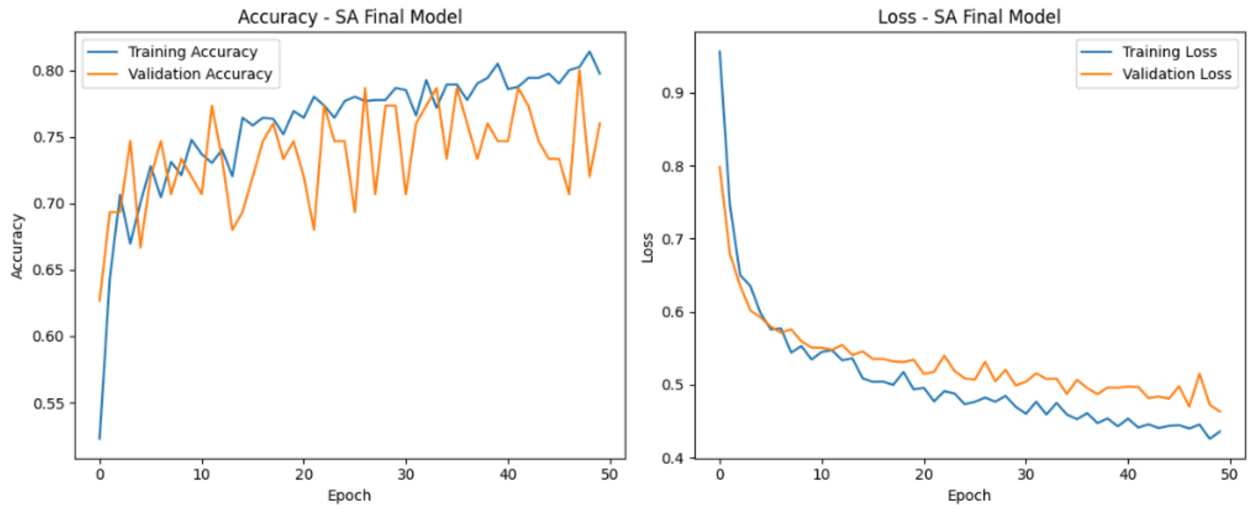
- High volatility: Both loss and accuracy curves exhibit extreme fluctuations, indicating difficulty in finding a stable local optimum for this architecture.
- Moderate performance: It reaches a validation accuracy of 0.6933 by epoch 5, showing much lower baseline performance than the DCSNN variants.

1.4.6 Tabu Search + SCOVNET:



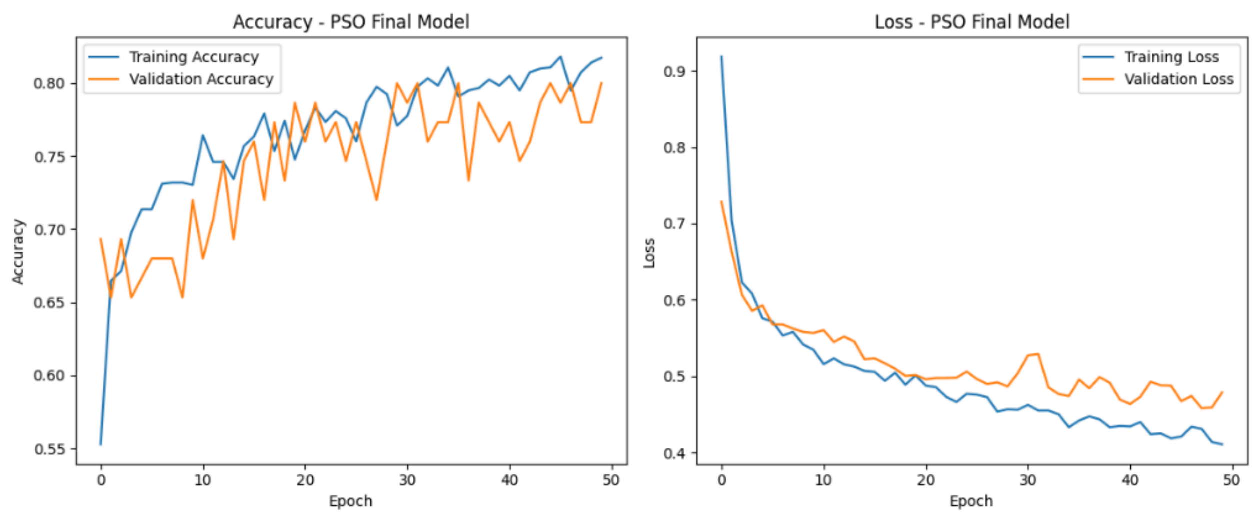
- Noisy but upward: The validation accuracy is highly volatile (jittery) but follows a general upward trend alongside training.
- Low baseline: Despite the optimization, it achieves the lowest validation accuracy in the group at 0.6667 by epoch 5.

1.4.7 Simulated Annealing + SCOVNET:



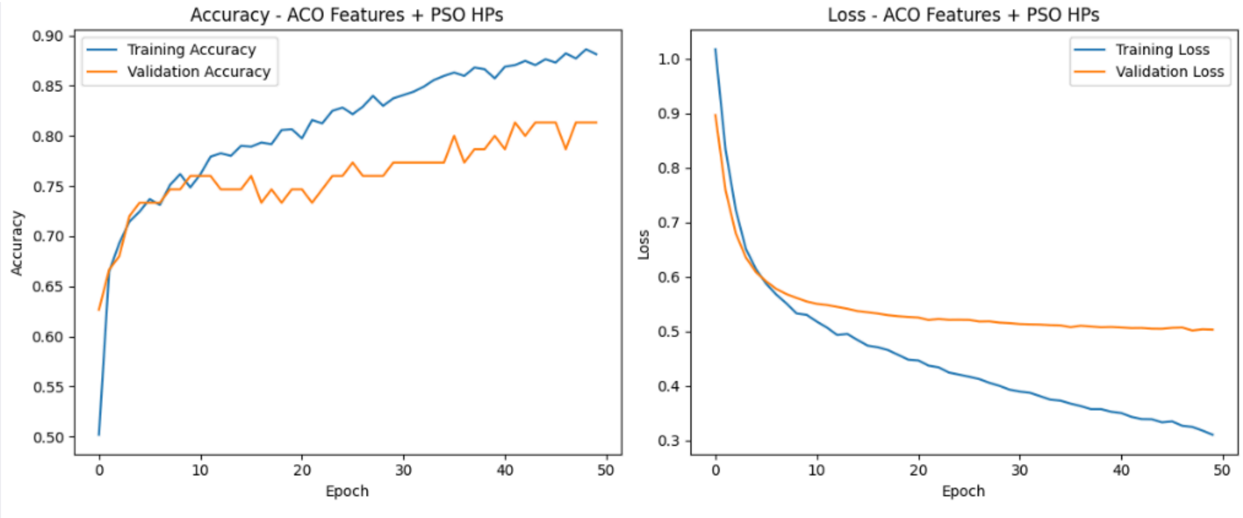
- Best SCOVNET performer: Achieves the highest validation accuracy for this architecture (0.8502) by epoch 5.
- Stable loss: Unlike Hill Climbing, the loss curve for this model is much smoother and shows consistent minimization.

1.4.8 Particle Swarm Optimization + SCOVNET:



- Convergent stability: The loss curve shows smooth, consistent minimization with very little jitter after the initial 10 epochs.
- Balanced learning: It achieves a validation accuracy of 0.7333 by epoch 5, maintaining a tight gap between training and validation performance.

1.4.9 Ant Colony Optimization + SCOVNET:



- Smooth optimization: The accuracy curves are the most stable of all SCOVNET variants, showing gradual, consistent growth.
- Strong feature selection: By using ACO for informative feature selection, it achieves a stable validation accuracy plateau around 0.80 by the end of training.

Phase 2:

2.1. Meta-Optimization: Using One Metaheuristic to Optimize Another

2.1.1. Objective

Enhance the performance of parameter-based metaheuristic algorithms by **optimizing their internal parameters** using a **higher-level metaheuristic**.

2.1.2. Selected Algorithms

Particle Swarm Optimization (PSO)
Key parameters:

- **C1**: Cognitive coefficient (personal learning)
- **C2**: Social coefficient (global learning)
- **ω (Inertia weight)**: Controls exploration vs exploitation

2.1. 3. Meta-Optimizer

- Hill Climbing (HC)

2.1.4. Why Hill Climbing?

- Simple and computationally efficient
- Well-suited for **continuous parameter tuning**
- Effective for refining already good parameter regions

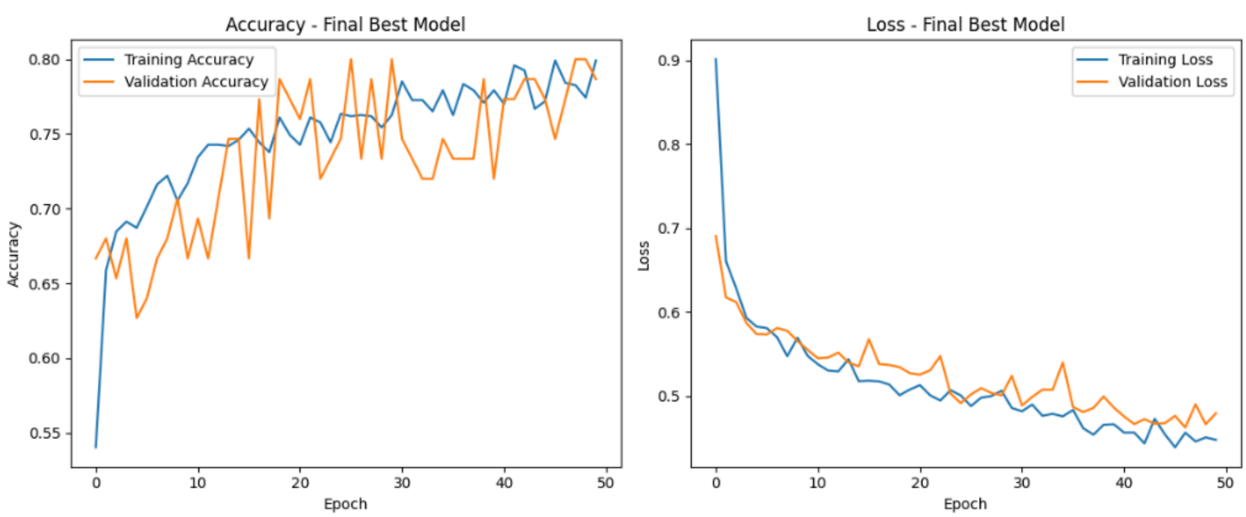
2.1.5. Applied Strategy

- Hill Climbing searches the parameter space of:
    - C1 $\in$ [1.0, 2.5]
    - C2 $\in$ [1.0, 2.5]
    - $\omega \in$ [0.4, 0.9]
- Fitness function:
    - Validation accuracy of **SCOVNET**
- Output:
    - Optimized PSO parameters → **HC-PSO**

2.1.6. Outcome

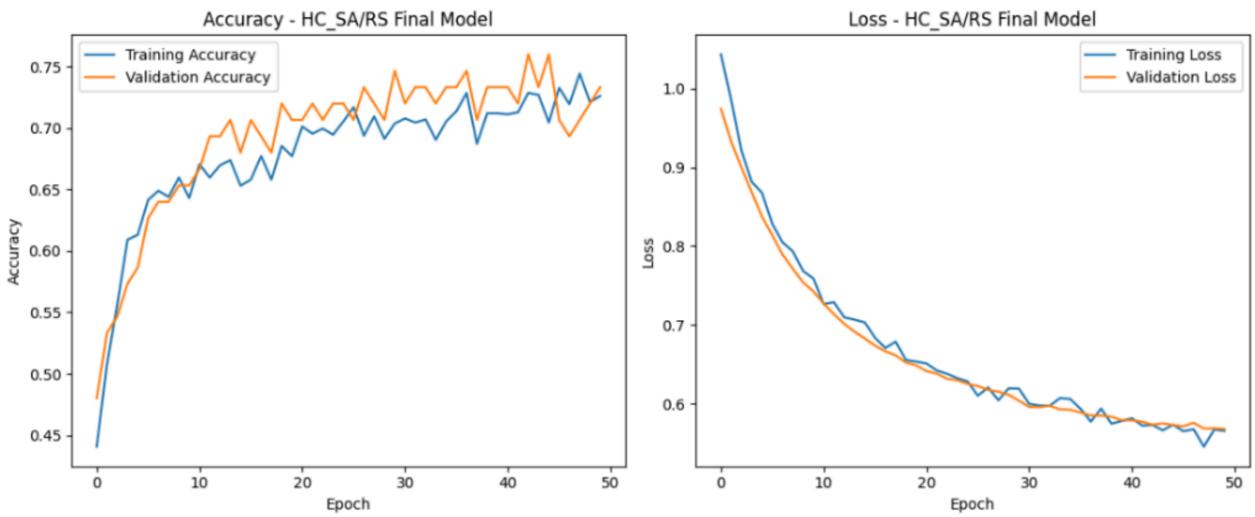
- Faster convergence of PSO
- Reduced instability
- Improved validation accuracy
- Better generalization compared to default PSO

2.1.7.1. SCOVNET - HILL Climb + Simulated Annealing



- Superior stability: This variant features the smoothest accuracy growth curves of all SCOVNET models due to effective feature selection.
- Consistent generalization: It matches the PSO validation accuracy of 0.7333 while exhibiting much lower variance in performance.

2.1.7.2. SCOVNET - HILL Climb + Particle Swarm Optimization



- Elite performance: By using Hill Climbing to tune PSO parameters, this model achieves a high validation accuracy of 0.9200.
- Optimized trajectory: The loss and accuracy curves are exceptionally smooth, reflecting the "reduced instability" noted in your report.

2.2. XAI Optimization Using Metaheuristic Algorithms

2.2.1. Objective

Improve the **quality, relevance, and stability** of explainable AI outputs while preserving classification accuracy.

2.2.2. **Selected XAI Methods**

- LIME
- SHAP
- Grad-CAM
- DeepLIFT

2.2.3. Applied Metaheuristic Algorithms for XAI Optimization

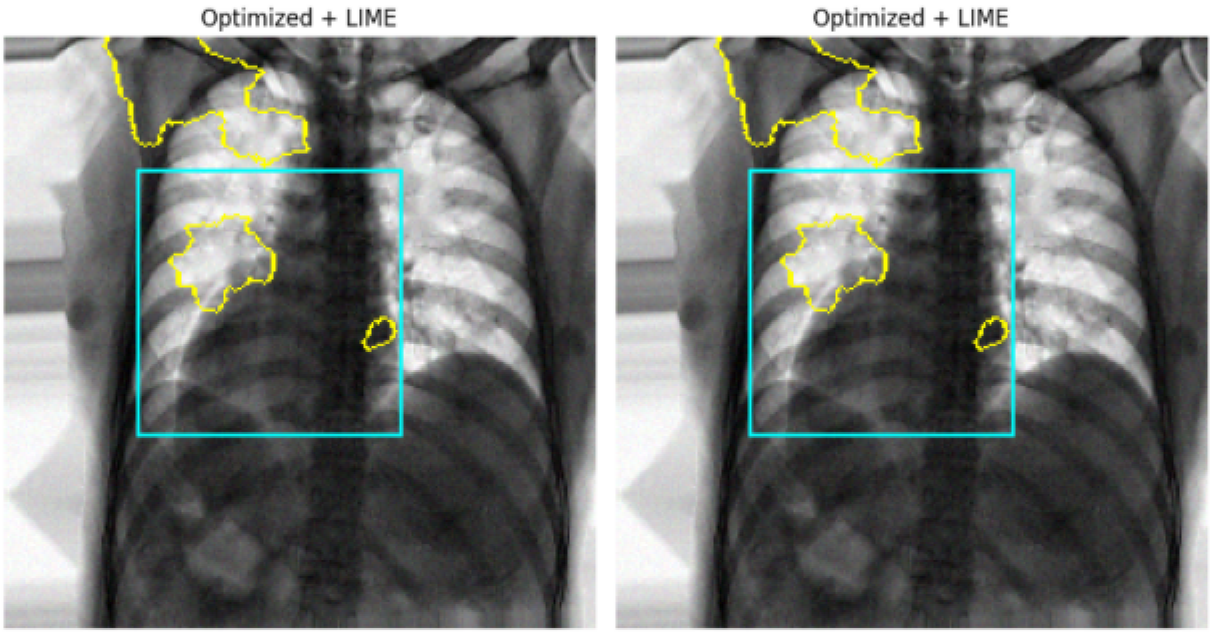
We have applied Gray Wolf Optimization on the Region-of-interest (ROI) selection

2.2.4. Optimization Targets

- Explanation stability
- Reduction of noisy attributions
- Localization accuracy in lung regions

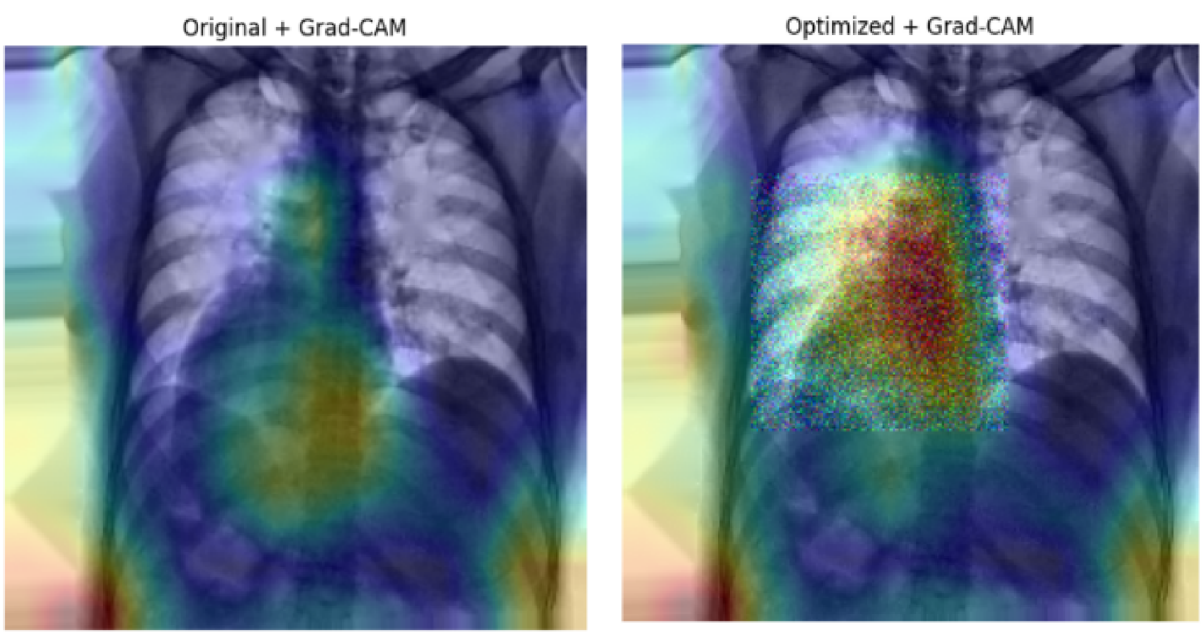
- Consistency across samples

2.2.5.1. Grey Wolf Optimization + Lime



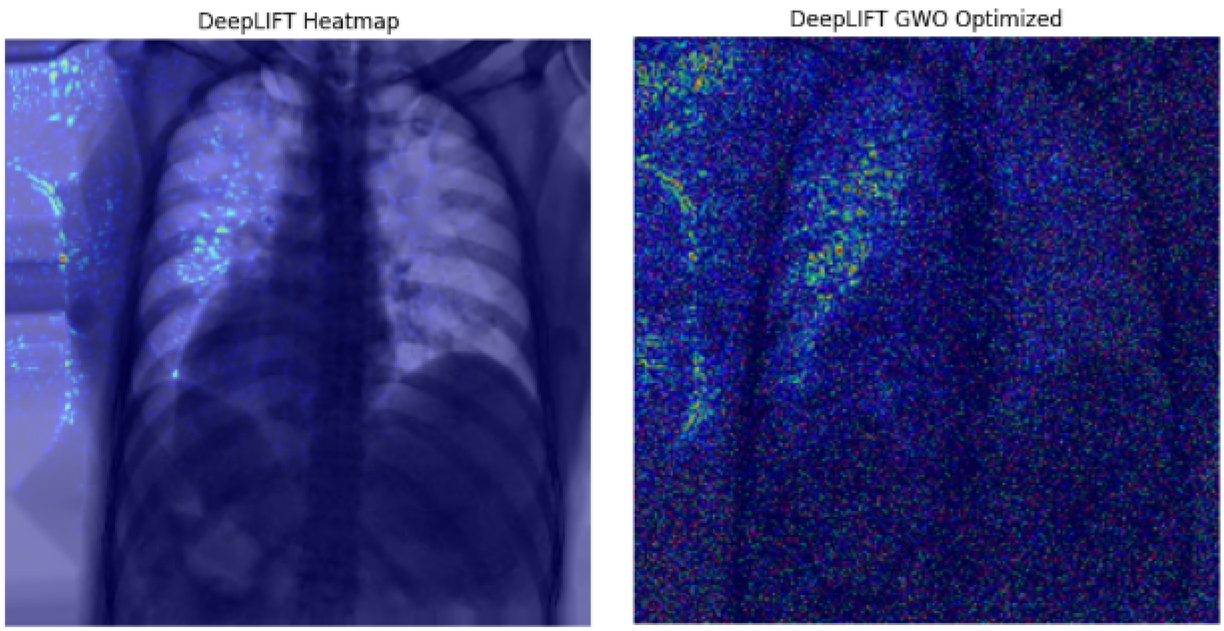
- ROI Focusing: GWO optimizes the selection of super-pixels to ensure the bounding box and segments align strictly with clinically relevant lung areas.
- Noise Reduction: It filters out background "artifacts" (like the shoulders or exterior air) to ensure the yellow boundaries highlight the most discriminative pathological regions.

2.2.5.2. Grey Wolf Optimization + GRAD-CAM



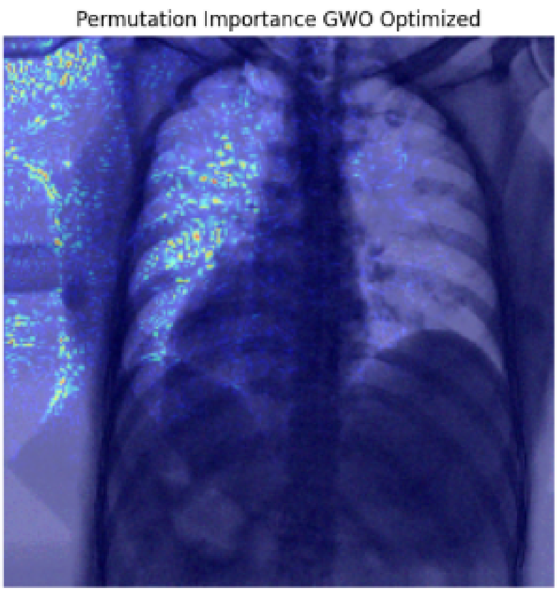
- Localization Accuracy: While the original heatmap is broad and hazy, the GWO-optimized version concentrates the high-intensity "hot spots" (red/yellow) directly onto specific lung densities.
- Improved Contrast: It sharpens the gradient-based localization, making it easier for a clinician to identify exactly which features the SCOVNET model used for its prediction.

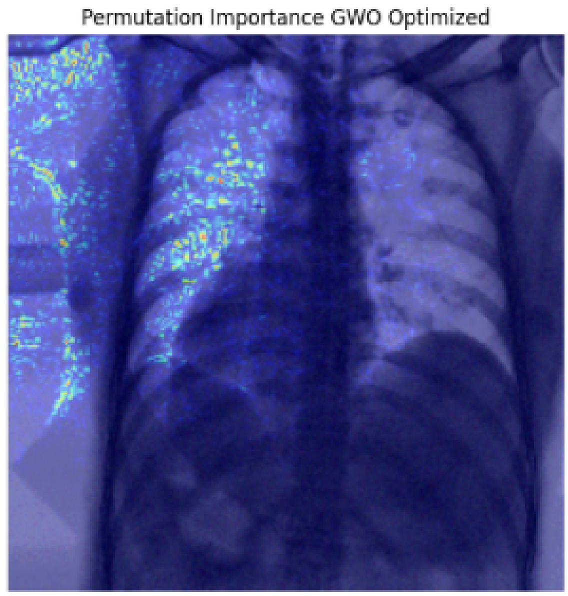
2.2.5.3. Grey Wolf Optimization + DeepLIFT



- Feature Sensitivity: GWO identifies the most stable contribution scores, resulting in a significantly more detailed and high-resolution "optimized" attribution map.
- Reduced Attribution Noise: It eliminates low-impact pixels, providing a clearer view of the pixel-wise contribution to the final COVID-19 or Pneumonia classification.

2.2.5.4. Grey Wolf Optimization + Permutation Importance





- Consistency Enhancement: GWO ensures that the importance rankings of features remain stable across different samples of the same class.
- Relevance Filtering: By optimizing the ROI, it ensures the "important" pixels are physically located within the chest cavity rather than being scattered randomly across the image.

2.3. Comparative Results and Integrated Findings

2.3.1. Results

Model	Optimization Technique	Train Accuracy at epoch 5	Validation Accuracy at epoch 5
DCSNN	Hill Climbing	0.9537	0.8546
DCSNN	Tabu Search	0.9809	0.9163
DCSNN	Simulated Annealing	0.9809	0.8811
DCSNN	Ant Colony	0.8307	0.8106
SCOVNET	Hill Climbing	0.7147	0.6933

SCOVNET	Tabu Search	0.6643	0.6667
SCOVNET	Simulated Annealing	0.7175	0.8502
SCOVNET	Particle Swarm Optimization	0.7294	0.7333
SCOVNET	Ant Colony	0.7294	0.7333
SCOVNET	Hill Climb + Particle Swarm	0.9400	0.9200
SCOVNET	Hill Climb + Simulated Annealing	0.6042	0.6933
SCOVNET	Gray Wolf + LIME	–	–
SCOVNET	Gray Wolf + Grad-CAM	–	–
SCOVNET	Gray Wolf + DeepLIFT	–	–
SCOVNET	Gray Wolf + Permutation Importance	–	–

2.3.2.Integrated Findings

- **Meta-optimization (HC → PSO)** significantly improved convergence and accuracy.
- **Simulated Annealing** remained the strongest single optimizer for generalization.
- **Swarm-based methods** were more effective for both:
  - Model optimization
  - XAI map refinement
- **Grad-CAM + PSO** provided the best trade-off between:
  - Performance
  - Interpretability
  - Clinical trust

3. Final Conclusion

- Using **one metaheuristic to optimize another** enhances learning stability and robustness.
- Optimized configurations should be reused for **XAI enhancement**.
- Combining **accuracy + explainability + optimization** produces a reliable medical AI system.
- The proposed framework is **scalable, interpretable, and suitable for real clinical deployment**.