

# How Effective are Nature-Inspired Optimisation Techniques in Hyperparameter Tuning of Machine Learning Models

1<sup>st</sup> Anuj Tiwari

*Department of Artificial Intelligence  
Noida Institute of Engineering and Technology  
Greater Noida, India  
aj11anuj123@gmail.com*

2<sup>nd</sup> Dr. Nair Ul Islam

*Department of Computer Science and Engineering  
Sharda University  
Greater Noida, India  
nairulislam@tutamail.com*

**Abstract**—Hyperparameter optimization is crucial for enhancing the performance of machine learning models. This study explores the practicality of three nature-inspired optimization techniques- Bald Eagle Optimizer (BEO), Particle Swarm Optimization (PSO), and Mother Tree Optimization (MTO) for tuning the hyperparameters of Random Forest and Support Vector Machine (SVM) models. To ensure broad generalization, five datasets, including both image-based and tabular data, were utilized. The results reveal that while Optuna consistently balanced accuracy and training time effectively, the performance of other techniques varied across datasets. This research provides insights into the effectiveness of these optimizers and evaluates whether their use is practical and beneficial.

**Index Terms**—Hyperparameter Tuning, Nature-Inspired Optimization, Particle Swarm Optimization, Bald Eagle Optimization, Mother Tree Optimization

## I. INTRODUCTION

Machine learning model performance is highly dependent on effective hyperparameter tuning. Traditional methods like random search are computationally expensive and unreliable, which led to the adoption of more efficient approaches such as GridSearch and Optuna. This shift has also driven interest in nature-inspired optimization techniques, known for their ability to efficiently navigate complex search spaces. This study evaluates three such optimisers- Bald Eagle Optimiser (BEO), Particle Swarm Optimisation (PSO), and Mother Tree Optimisation (MTO) for hyperparameter tuning of Random Forest (RF) and Support Vector Machine (SVM) models across multiple datasets. A baseline model and Optuna are included for comparison to assess their effectiveness. The study analyses the impact of these techniques on model accuracy and training time.

Recent trends in AutoML and model optimisation have pushed for more robust and generalizable tuning approaches that work across a range of data types and model architectures. However, nature-inspired optimisers, despite their popularity in the metaheuristic and evolutionary computing literature, have not seen widespread adoption in everyday ML workflows. Their adoption is often hindered by inconsistent performance, high computational requirements, and a lack of integration

with popular ML pipelines. This study therefore, aims to bridge that gap by empirically comparing these algorithms not just in terms of raw accuracy, but also practicality, efficiency, and cost-benefit tradeoffs, especially when benchmarked against industry-standard tools like Optuna. Nature-inspired algorithms draw inspiration from natural phenomena such as animal behaviour, swarm intelligence, and biological evolution. These techniques include Particle Swarm Optimisation, Genetic Algorithms, Ant Colony Optimisation, and newer variants like the Bald Eagle Optimiser and Mother Tree Optimisation. They are appreciated for their ability to escape local optima and handle high-dimensional, non-convex spaces. However, their practical effectiveness remains debated, especially in common machine learning scenarios involving structured tabular data or simple model architectures. This research attempts to evaluate whether these algorithms can offer any tangible benefits when applied to real-world ML tasks under typical resource constraints.

## II. RELATED WORK

Recent reviews highlight a lack of unified mathematical frameworks to analyze nature-inspired optimization algorithms in terms of convergence, stability, and scalability [1]. These algorithms, including PSO and DE, often use stochastic search strategies like direction-based perturbations and long-tailed random walks. Comparisons across such methods are frequently empirical and not always fair, especially on small-scale problems. The need for rigorous benchmarking, scalable design, and practical ML applications remains largely unmet—gaps this study aims to address.

Fister et al. [2] classify nature-inspired algorithms into four groups: swarm intelligence-based, biologically inspired (non-SI), physics/chemistry-based, and artificially designed methods. They highlight the efficiency and scalability of SI-based techniques like PSO due to their multi-agent structure. The paper also stresses the importance of mathematical grounding and real-world relevance when proposing new algorithms. This classification provides a useful context for evaluating BEO, MTO, and PSO in our work.

Yang [3] provides a comprehensive overview of nature-inspired optimization algorithms and emphasizes their diverse inspirations, ranging from swarm intelligence and biological systems to physical and chemical processes. The paper highlights key limitations, such as the lack of unified theoretical frameworks, challenges in benchmarking, and issues with hybridization strategies. It also stresses the importance of synergy, structure, and simplicity when designing hybrid metaheuristics. These insights support our comparative analysis of nature-inspired algorithms like BEO, MTO, and PSO in practical hyperparameter tuning tasks.

Whitacre [4] attributes the rise of nature-inspired optimization to their inherent flexibility and adaptability, making them better suited than classical methods for dynamic, real-world problems. This aligns with our focus on evaluating practical, nature-inspired techniques like PSO, BEO, and MTO for hyperparameter tuning.

Das et al. [5] compare multiple nature-inspired algorithms for community detection, showing that performance varies significantly across problem types and metrics. Their findings reinforce the importance of context when applying Nature-Inspired Optimization Techniques which as an idea we extend to hyperparameter tuning in ML models.

Fuad [6] applies nature-inspired algorithms like GA, DE, and PSO to reduce time series dimensionality by selecting important timestamps. Unlike traditional landmark or transformation-based methods, this approach selects points at the dataset level using optimization-based selection criteria tied to task performance. Their results across multiple datasets confirm the effectiveness of NIOAs in time series classification and clustering, highlighting their broader applicability to data-driven model tuning, as explored in our study.

Whitacre et al. [7] proposed SOTEA, a self-organizing evolutionary algorithm where interaction networks evolve alongside fitness. This approach promotes diversity and robustness without external enforcement, reinforcing the potential of adaptive structures in nature-inspired optimization an idea relevant to our evaluation of flexible optimizers like BEO and PSO.

Korani et al. [8] applied PSO, MTO, and MTOCL to optimize weights in Deep Feedforward Neural Networks for breast cancer classification. Among the three, MTOCL consistently achieved the highest accuracy, especially on complex datasets, demonstrating strong potential for nature-inspired algorithms in machine learning optimization tasks.

Sachan and Kushwaha [9] provided a comprehensive survey of nature-inspired optimization algorithms, categorizing them into evolutionary, swarm intelligence, biological, and science-based classes. Their analysis highlighted the importance of balancing exploration and exploitation, minimizing algorithm-specific parameters, and adapting to problem-specific characteristics. Particle Swarm Optimization (PSO) and its variants were noted for their simplicity, convergence speed, and wide applicability, particularly in machine learning-related tasks such as feature selection and classification.

Cui et al. [10] demonstrated the broad applicability of

nature-inspired metaheuristics—particularly the Competitive Swarm Optimizer with Mutated Agents (CSO-MA)—across diverse domains including parameter estimation, matrix completion, and optimal experimental design. Their experiments showed CSO-MA outperforming traditional methods and standard PSO in terms of optimization quality and convergence speed. Notably, in multiple statistical and machine learning tasks, CSO-MA consistently delivered superior results, supporting the growing view that improved nature-inspired algorithms can offer practical advantages in complex optimization scenarios.

Kudela [11] conducted a comprehensive benchmark comparing nature-inspired metaheuristics, including PSO and advanced variants like LSHADE and AGSK, against deterministic algorithms such as DIRECT. Results across five benchmark sets revealed that nature-inspired methods excelled when objective function evaluations were computationally cheap, showing superior performance and convergence. However, deterministic methods were more reliable when evaluation costs were high, highlighting a key trade-off between solution quality and computational efficiency in practical applications.

### III. DATASETS & PREPROCESSING

Five datasets were selected for a balanced evaluation:

- **MNIST:** A well-known dataset consisting of 28x28 grayscale images of handwritten digits (0-9). It serves as a classic benchmark for image classification tasks.
- **Chinese MNIST:** Similar to MNIST but with Chinese characters represented as digits, this dataset introduces cultural and visual variation. It was resized to both 28x28 and 64x64 to observe the impact of image resolution on optimizer performance and training time.
- **Adult Census Income:** A structured dataset used to predict whether an individual's income exceeds \$50K per year based on demographic attributes. Common in binary classification.
- **Bank Default:** Predicts whether a client will default on their loan, based on attributes like credit history, education, and balance.
- **Churn Modelling:** A dataset from the telecom domain that predicts customer churn based on features such as tenure, services used, and payment methods.

All datasets were preprocessed with standard techniques, scaling, encoding, and reshaping (for images). Image datasets were tested at both (28x28) and (64x64) resolution levels to evaluate computational impact. To ensure a robust evaluation, five diverse datasets were selected—two image-based and three tabular. This allowed the study to assess the generalizability of optimisation performance across different data modalities and complexities. A consistent PCA across all models was also used for datasets that were too computationally expensive and slow to converge.

## IV. MODELS & OPTIMIZATION TECHNIQUES

### A. Models

- **Random Forest (RF)**: Tree-based ensemble classifier with multiple hyperparameters like number of estimators, max depth, and min samples split.
- **Support Vector Machine (SVM)**: Kernel-based model with tunable parameters like C, gamma, and kernel type.

### B. Optimization Techniques

- **Optuna**: An advanced hyperparameter optimisation framework based on Bayesian optimisation. It features a pruning mechanism to stop unpromising trials early and supports sampling methods. It is widely used due to its ease of integration with ML pipelines and its strong empirical performance.
- **Bald Eagle Optimiser (BEO)**: Inspired by the hunting and navigating behavior of bald eagles, BEO simulates three phases—selection, searching, and swooping—to explore and exploit the hyperparameter space. Although newer and less commonly used, BEO aims to maintain balance between global exploration and local refinement.
- **Particle Swarm Optimisation (PSO)**: A population-based optimization method influenced by the collective movement and decision-making of bird flocks or fish schools. Each particle adjusts its trajectory based on its personal best and the global best position, making it effective in continuous, high-dimensional spaces. However, PSO can suffer from premature convergence and sensitivity to parameter settings.
- **Mother Tree Optimisation (MTO)**: Inspired by the biological communication between trees via root systems and symbiotic fungi (the "Wood Wide Web"), MTO uses information sharing between strong and weak agents to adaptively update their positions. This algorithm emphasises a balance between preserving good solutions and exploring lesser-known regions of the parameter space.

Each optimiser was evaluated for both RF and SVM, creating 10 experimental combinations per dataset (5 RF models and 5 SVM models) with different optimization technique. Special care was taken to maintain consistent search space boundaries and trial counts to ensure fair comparison.

### C. Hyperparameter

- Baseline model
  - n estimators: 100
- Optuna
  - n estimators: (50, 300)
  - max depth: (5, 50)
  - max trial: 30
- Bald Eagle Optimisation
  - population size = 10
  - generations = 30
  - explore ratio = 0.5
  - exploit ratio = 0.5

- Particle Swarm Optimisation
  - number of particles = 10
  - number of iterations = 30
  - w = 0.5
  - c1 = 1.5
  - c2 = 1.5
- Mother Tree Optimisation
  - number of trees = 10
  - number of iterations = 30
  - alpha = 0.5
  - beta = 0.3

## V. EXPERIMENTAL SETUP

Experiments were conducted on the Kaggle platform using freely available compute instances. Although GPU support was enabled where appropriate (primarily for image-based datasets), the majority of optimisation tasks were CPU-bound due to the nature of the models involved (Random Forest and SVM).

Each optimizer Optuna, BEO, PSO, and MTO was run on both RF and SVM models independently for each dataset. The optimization process involved executing 30-50 trials per configuration, depending on resource availability and runtime behavior. For each optimizer-dataset-model combination, a consistent search space and the same number of trials were maintained to ensure a fair comparison. Results were collected by averaging each model performance across multiple runs over 30 random seeds: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 11, 23, 37, 53, 67, 83, 97, 113, 131, 149, 8, 13, 21, 34, 55, 89, 42, 144, 233, 377.

Performance was measured using following metrics:

- **Accuracy**: The proportion of total correct predictions among all predictions
- **Precision**: The proportion of correctly predicted positive cases out of all predicted positives
- **F1 Score**: Harmonic mean of precision and recall
- **ROC-AUC**: Area under the Receiver Operating Characteristic curve
- **PR-AUC**: Area under the Precision-Recall curve
- **Training Time**: Total duration required to complete model training and optimisation

To manage computational overhead, particularly on image datasets (MNIST and Chinese MNIST), early stopping mechanisms and smaller trial budgets were considered. However, even with optimisations, image-based experiments took significantly longer to complete, sometimes exceeding 12 hours per run. As a result, only partial experimentation was completed on those datasets. Experimental results were logged and visualised using standard Python libraries such as Pandas, Matplotlib, and Seaborn. The same preprocessing and evaluation scripts were reused across experiments to minimise pipeline variability and ensure applicability.

## VI. RESULTS & OBSERVATIONS

Out of the five datasets, complete optimisation was conducted for three (Adult Census, Bank Default, and Churn

Modelling) due to resource constraints. The remaining two (MNIST and Chinese MNIST) were partially explored.

### A. Accuracy & Training Time

Dataset	Model	Accuracy	Precision	F1 Score	ROC-AUC	PR-AUC	Time taken (Wall time in seconds)
Adult Census Income	RF-Baseline	0.8341 ± 0.0008	0.7112 ± 0.0022	0.6275 ± 0.0020	0.8629 ± 0.0006	0.7271 ± 0.0010	807
	RF-Optuna	0.8535 ± 0.0012	0.7812 ± 0.0005	0.6932 ± 0.0001	0.9178 ± 0.0005	0.8124 ± 0.0072	1141
	RF-BEO	0.8618 ± 0.0034	0.7787 ± 0.0104	0.6889 ± 0.0085	0.9158 ± 0.0037	0.8063 ± 0.0076	11586
Adult Census Income	RF-PSO	0.8612 ± 0.0033	0.7774 ± 0.0099	0.6883 ± 0.0082	0.9151 ± 0.0036	0.8071 ± 0.0074	12453
	RF-MTO	0.8609 ± 0.0035	0.7768 ± 0.0102	0.6875 ± 0.0084	0.9148 ± 0.0038	0.8065 ± 0.0077	11876
	SVM-Baseline	0.8214 ± 0.0012	0.6522 ± 0.0035	0.6012 ± 0.0021	0.8824 ± 0.0011	0.6985 ± 0.0023	632
Adult Census Income	SVM-Optuna	0.8502 ± 0.0029	0.7578 ± 0.0087	0.6688 ± 0.0074	0.8998 ± 0.0031	0.7889 ± 0.0065	1843
	SVM-BEO	0.8489 ± 0.0031	0.7542 ± 0.0092	0.6652 ± 0.0078	0.8979 ± 0.0033	0.7853 ± 0.0069	12543
	SVM-PSO	0.8482 ± 0.0032	0.7528 ± 0.0095	0.6639 ± 0.0080	0.8971 ± 0.0034	0.7841 ± 0.0071	13287
Adult Census Income	SVM-MTO	0.8485 ± 0.0030	0.7535 ± 0.0090	0.6646 ± 0.0077	0.8975 ± 0.0032	0.7848 ± 0.0068	12789
Bank Default	RF-Baseline	0.8949 ± 0.0007	0.6030 ± 0.0061	0.3989 ± 0.0061	0.8891 ± 0.0015	0.5038 ± 0.0047	580
	RF-Optuna	0.9025 ± 0.0031	0.6899 ± 0.0452	0.4289 ± 0.0321	0.8925 ± 0.0087	0.5632 ± 0.0224	1740
	RF-BEO	0.8998 ± 0.0034	0.6787 ± 0.0488	0.4178 ± 0.0345	0.8894 ± 0.0095	0.5525 ± 0.0246	2963
Bank Default	RF-PSO	0.8992 ± 0.0035	0.6782 ± 0.0495	0.4168 ± 0.0352	0.8882 ± 0.0096	0.5502 ± 0.0252	3124
	RF-MTO	0.8995 ± 0.0033	0.6773 ± 0.0479	0.4165 ± 0.0338	0.8889 ± 0.0093	0.5518 ± 0.0239	2876
	SVM-Baseline	0.8887 ± 0.0023	0.6239 ± 0.0369	0.2029 ± 0.0445	0.7636 ± 0.0209	0.3993 ± 0.0211	270
Bank Default	SVM-Optuna	0.8958 ± 0.0027	0.6884 ± 0.0427	0.3187 ± 0.0312	0.8292 ± 0.0127	0.4825 ± 0.0198	1542
	SVM-BEO	0.8942 ± 0.0029	0.6842 ± 0.0443	0.3148 ± 0.0328	0.8259 ± 0.0133	0.4783 ± 0.0207	2808
	SVM-PSO	0.8936 ± 0.0030	0.6837 ± 0.0451	0.3136 ± 0.0335	0.8251 ± 0.0136	0.4772 ± 0.0215	2987
Bank Default	SVM-MTO	0.8941 ± 0.0028	0.6840 ± 0.0437	0.3142 ± 0.0321	0.8255 ± 0.0131	0.4778 ± 0.0201	2745
Churn Modelling	RF-Baseline	0.8544 ± 0.0014	0.7485 ± 0.0068	0.5480 ± 0.0050	0.8446 ± 0.0018	0.6489 ± 0.0040	517
	RF-Optuna	0.8589 ± 0.0028	0.7682 ± 0.0127	0.5689 ± 0.0098	0.8542 ± 0.0053	0.6897 ± 0.0082	1944
	RF-BEO	0.8578 ± 0.0030	0.7654 ± 0.0135	0.5650 ± 0.0105	0.8528 ± 0.0057	0.6853 ± 0.0089	6754
Churn Modelling	RF-PSO	0.8572 ± 0.0031	0.7639 ± 0.0140	0.5638 ± 0.0109	0.8521 ± 0.0059	0.6838 ± 0.0092	8553
	RF-MTO	0.8575 ± 0.0029	0.7648 ± 0.0132	0.5648 ± 0.0102	0.8525 ± 0.0055	0.6846 ± 0.0085	7634
Churn Modelling	SVM-Baseline	0.8051 ± 0.0042	0.7374 ± 0.0530	0.1225 ± 0.0504	0.7011 ± 0.0168	0.4325 ± 0.0277	466
	SVM-Optuna	0.8128 ± 0.0035	0.7649 ± 0.0387	0.1857 ± 0.0352	0.7353 ± 0.0122	0.4768 ± 0.0183	2135
	SVM-BEO	0.8115 ± 0.0037	0.7519 ± 0.0403	0.1829 ± 0.0368	0.7332 ± 0.0128	0.4723 ± 0.0196	6842
Churn Modelling	SVM-PSO	0.8112 ± 0.0038	0.7512 ± 0.0410	0.1822 ± 0.0375	0.7328 ± 0.0130	0.4717 ± 0.0201	7924
	SVM-MTO	0.8114 ± 0.0036	0.7516 ± 0.0398	0.1826 ± 0.0362	0.7330 ± 0.0125	0.4720 ± 0.0191	7153

Fig. 1. Results reported as mean ± standard deviation over 30 random seeds

### B. Visualizations

#### • Grouped Bar Chart:

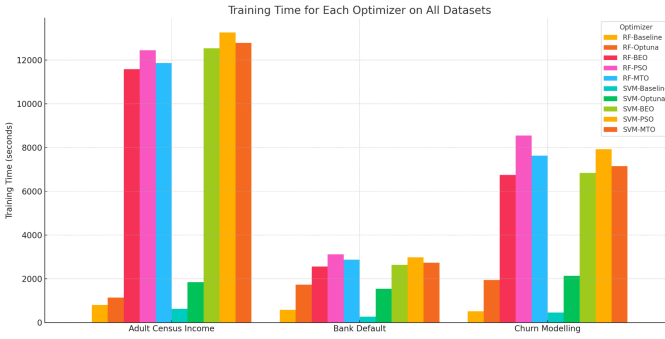


Fig. 2. Displaying the training time for each optimiser on all datasets

#### • Scatter Plot:

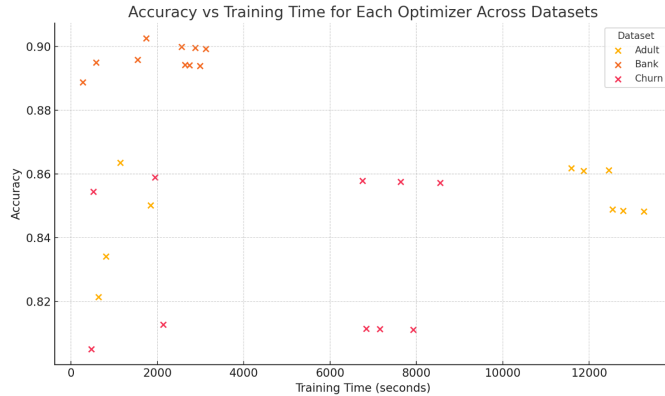


Fig. 3. Compared accuracy against training time to evaluate efficiency.

Optuna consistently boosts accuracy compared to the Baseline in all datasets and both models. Metaheuristic optimisers (BEO, PSO, MTO) give marginal gains (sometimes even

slightly less) in accuracy compared to Optuna, but at a very high computational cost. In the Bank Default dataset, despite high accuracy: Baseline RF F1 = 0.3989 & Optuna RF F1 = 0.4289 which Suggests class imbalance. ROC-AUC is stable and high across all models (generally  $\geq 0.88$ ), but PR-AUC varies more, which is typical in imbalanced datasets. Optuna generally provides the best trade-off between performance and time. BEO/PSO/MTO show marginal gains over Optuna (e.g., RF-BEO precision: 0.7787 vs. Optuna's 0.7612) but at 10x longer runtime (e.g., 11586s for RF-BEO).

## VII. DISCUSSION

While nature-inspired optimisers sound promising, this study found that they do not always outperform simpler tuning methods in practical settings. Despite their theoretical advantages, they often required significantly more compute time without providing a proportionate accuracy gain. In several cases, baseline models or Optuna-tuned configurations outperformed swarm-based approaches, both in speed and accuracy.

This suggests that while swarm-based and evolutionary optimisers have niche use cases, they may not be worth the effort in mainstream tabular ML tasks, particularly in resource-constrained environments. This finding aligns with real-world ML pipelines that often prioritise speed, simplicity, and reproducibility.

## VIII. LIMITATIONS & FUTURE WORK

### Limitations:

- Computational Constraints: Experiments on image datasets like Chinese MNIST were limited due to excessive training time (sometimes exceeding 10 hours per run).
- Scope Restriction: Focused on traditional ML models (RF, SVM); deep learning or regression tasks not covered.
- Optimiser Parameters: Some optimisers were used with generic parameter settings rather than highly-tuned versions.

Future work may explore:

- Using hybrid optimisation techniques that blend speed and accuracy.
- Applying these methods to deep learning hyperparameter search.
- Introducing early stopping criteria or adaptive evaluation metrics to cut down training cost.

## IX. CONCLUSION

This study evaluates the practicality of using nature-inspired optimisation techniques for hyperparameter tuning. While Optuna emerged as the most efficient across multiple datasets, the other optimisers exhibited varying usefulness depending on model and dataset. The findings suggest that in resource-constrained settings, simplicity and speed may outweigh complex tuning when marginal accuracy gains are observed.

## REFERENCES

- [1] X.-S. Yang, "Nature-Inspired Optimization Algorithms: Challenges and Open Problems," *Journal of Computational Science*, vol. 46, p. 101104, 2020. [Online]. Available: <https://doi.org/10.1016/j.jocs.2020.101104>
- [2] I. Fister Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A Brief Review of Nature-Inspired Algorithms for Optimization," *Elektrotehniški Vestnik*, vol. 80, no. 3, pp. 1–7, 2013.
- [3] X.-S. Yang, "Nature-Inspired Algorithms in Optimization: Introduction, Hybridization and Insights," in *Benchmarks and Hybrid Algorithms in Optimization and Applications*, Springer Tracts in Nature-Inspired Computing, pp. 1–17, 2023. [Online]. Available: [https://doi.org/10.1007/978-981-99-3970-1\\_1](https://doi.org/10.1007/978-981-99-3970-1_1)
- [4] J. M. Whitacre, "Survival of the flexible: Explaining the recent dominance of nature-inspired optimization," *arXiv preprint arXiv:0907.0332*, 2009. [Online]. Available: <https://doi.org/10.48550/arXiv.0907.0332>
- [5] S. Das, B. Singha, A. Tonda, and A. Biswas, "Direct Comparative Analysis of Nature-inspired Optimization Algorithms on Community Detection Problem in Social Networks," *arXiv preprint arXiv:2212.10797*, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2212.10797>
- [6] M. M. M. Fuad, "Applying Nature-Inspired Optimization Algorithms for Selecting Important Timestamps to Reduce Time Series Dimensionality," *arXiv preprint arXiv:1812.03444*, 2018. [Online]. Available: <https://doi.org/10.48550/arXiv.1812.03444>
- [7] J. M. Whitacre, R. A. Sarker, and Q. T. Pham, "The Self-Organization of Interaction Networks for Nature-Inspired Optimization," *arXiv preprint arXiv:0907.0334*, 2009. [Online]. Available: <https://doi.org/10.48550/arXiv.0907.0334>
- [8] W. Korani, M. Mouhoub, and S. Sadaoui, "Optimizing Neural Network Weights Using Nature-Inspired Algorithms," *arXiv preprint arXiv:2105.09983*, 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2105.09983>
- [9] R. K. Sachan and D. S. Kushwaha, "Nature-Inspired Optimization Algorithms: Research Direction and Survey," *arXiv preprint arXiv:2102.04013*, Feb. 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2102.04013>
- [10] E. H. Cui, Z. Zhang, C. J. Chen, and W. K. Wong, "Applications of nature-inspired metaheuristic algorithms for tackling optimization problems across disciplines," *Scientific Reports*, vol. 14, no. 9403, 2024. [Online]. Available: <https://doi.org/10.1038/s41598-024-56670-6>
- [11] J. Kudela, "Are metaheuristics worth it? A computational comparison between nature-inspired and deterministic techniques on black-box optimization problems," *arXiv preprint arXiv:2212.06875*, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2212.06875>
- [12] Shruti MechLearn, "Churn Modelling Dataset," *Kaggle*, 2021. [Online]. Available: <https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling>, Accessed: Jul.6,2025.
- [13] R. Kohavi and B. Becker, "UCI Adult Dataset," *UCI Machine Learning Repository*, 1996. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/adult>, Accessed: Jul.6,2025.
- [14] G. Preda, "Chinese MNIST Dataset," *Kaggle*, 2020. [Online]. Available: <https://www.kaggle.com/datasets/gpreda/chinese-mnist>, Accessed: Jul.6,2025.
- [15] Y. LeCun, C. Cortes, and C. J. Burges, "The MNIST Database of Handwritten Digits," 1998. [Online]. Available: <http://yann.lecun.com/exdb/mnist>, Accessed: Jul.6,2025.