Evolving Neural Networks with Particle Swarm Optimization

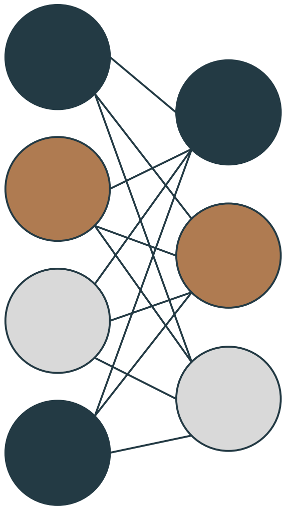
Benjamin Bandić, Anes Ćenanović, Benjamin Hadžihasanović

The Problem:
The Challenge of Hyperparamaters

Neural network performance is critically dependent on its hyperparameters (e.g., architecture, learning rate).

Manual tuning is slow, inefficient, and often misses the best solution.

Automated methods are needed to intelligently search the vast space of possible configurations.



Learning Rate?

Neurons?

Dropout?

Our Goal:
Can PSO Evolve Better Networks?

Objective → **Methodology** → **Scope**

Objective

To implement and analyze Particle Swarm Optimization (PSO) for automated hyperparameter tuning.

Methodology

- Establish a Baseline Model for each task.
- Use PSO to search for optimal hyperparameters.
- Train a Final Optimized Model.
- Compare the results.

Scope

Applied to 6 diverse datasets (tabular, audio, image) to test versatility.

The Datasets:
A Diverse Portfolio of Challenges

NASA Exoplanets: Tabular Classification (Well-structured data)

S&P 500: Tabular Regression (Time-series data)

Yu-Gi-Oh! Cards: Tabular Classification (imbalanced data)

Autolist Cars: Tabular Regression (Data with outliers)

FMA Audio: Audio Classification (Complex feature extraction)

CIFAR-100: Image Classification (Standard CV benchmark)

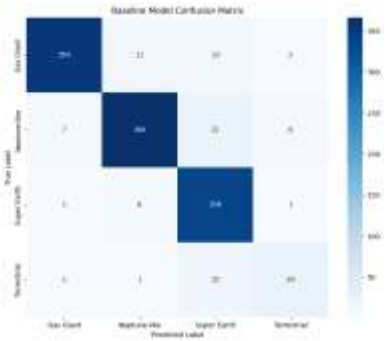
Case Study:
NASA Exoplanets - A Classic Success

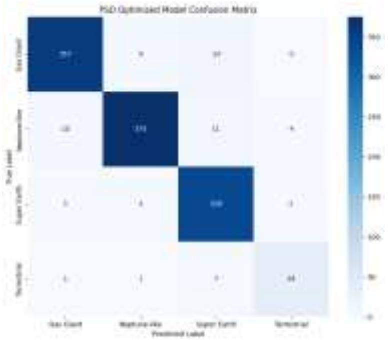
Challenge

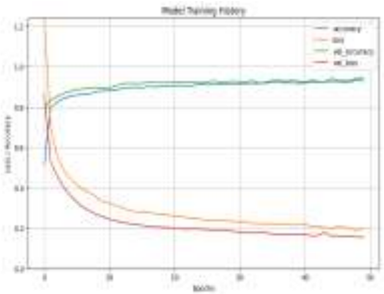
A well structured, clean tabular dataset. Can PSO improve upon an already strong baseline?

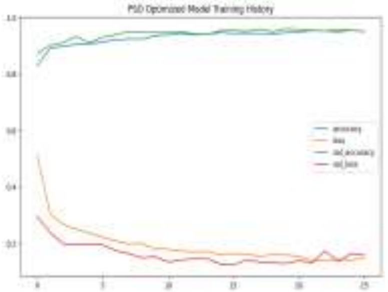
Key Takeaway

PSO acted as a powerful refiner, boosting accuracy from 92.6% to 94.4%









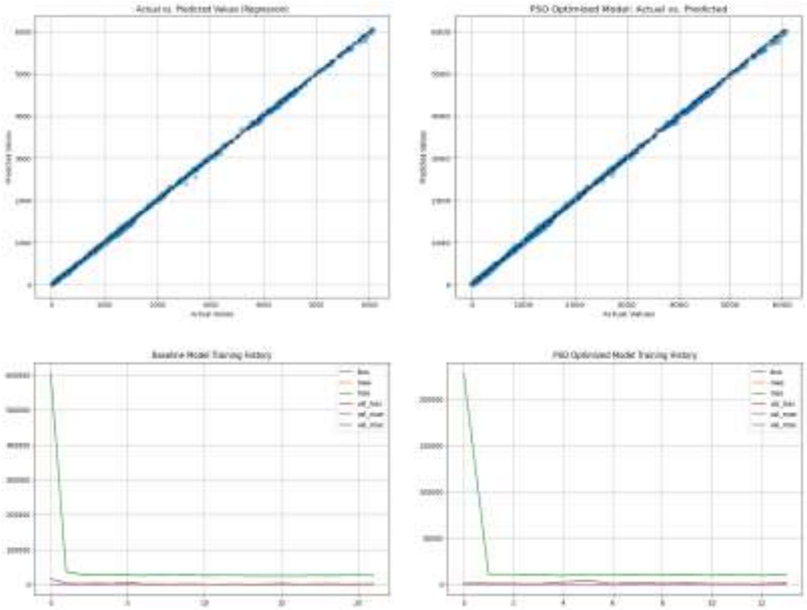
Case Study:
S&P 500 - High Accuracy on Predictable Data

Challenge

Time-series data is highly autocorrelated. How do the models perform here?

Key Takeaway

Both models achieved extremely high R-squared values (>0.999), with PSO offering a slight improvement in MAPE



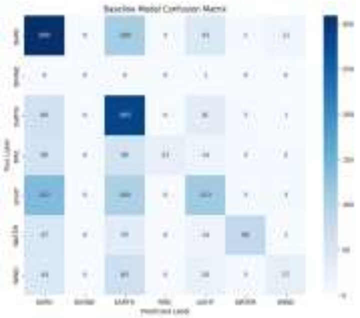
Case Study:
Yu-Gi-Oh! - The Impact of Imbalance

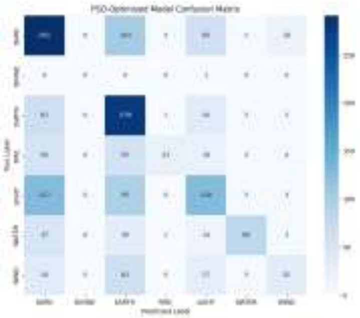
Challenge

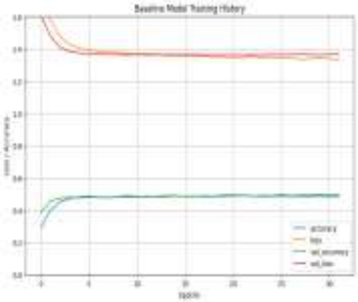
The “DIVINE” class has only 5 instances out of 8,500. Can PSO overcome this?

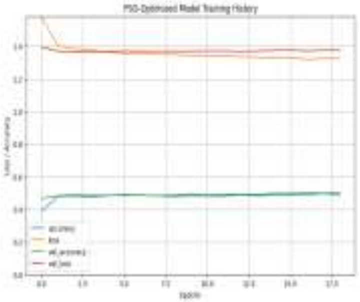
Key Takeaway

PSO provides marginal gains, but performance is ultimately limited by the imbalanced data.









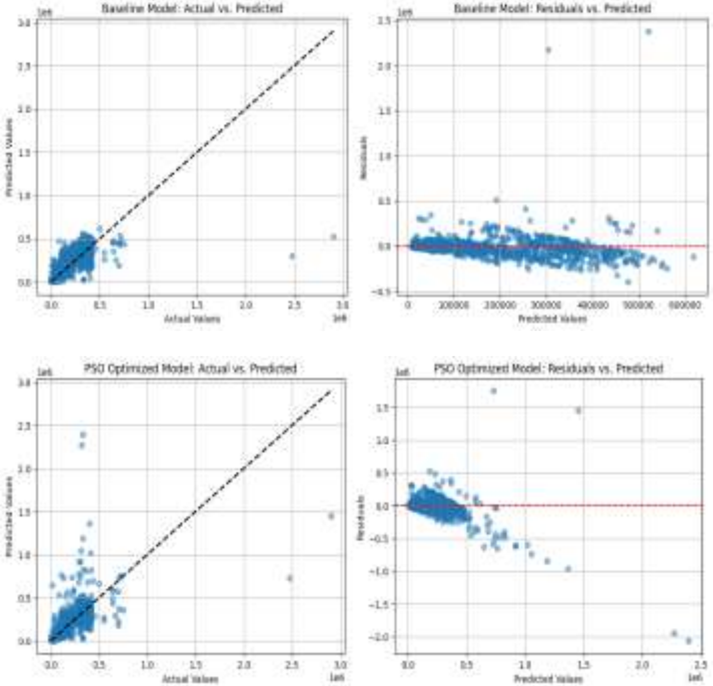
Case Study:
Autolist - Data with Outliers

Challenge

The dataset contains significant price outliers. How does this affect the model?

Key Takeaway

PSO improved the “typical” error (MAE/MedAE) at the cost of outlier-sensitive error (RMSE).

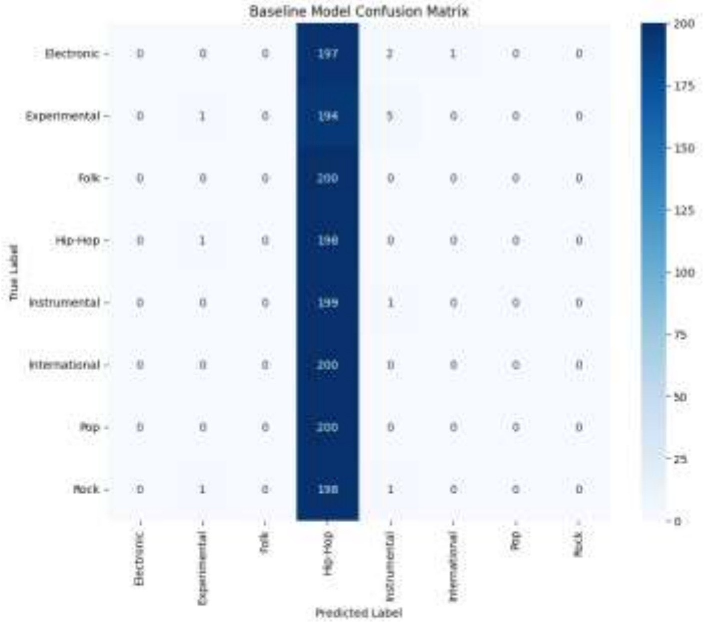


Case Study:
FMA Audio - From Failure to Function

A difficult genre classification task.

Made harder by using only 10-second audio clips.

Result: The Baseline Model completely failed (12.5% accuracy)



FMA:
The PSO Difference

Comparison

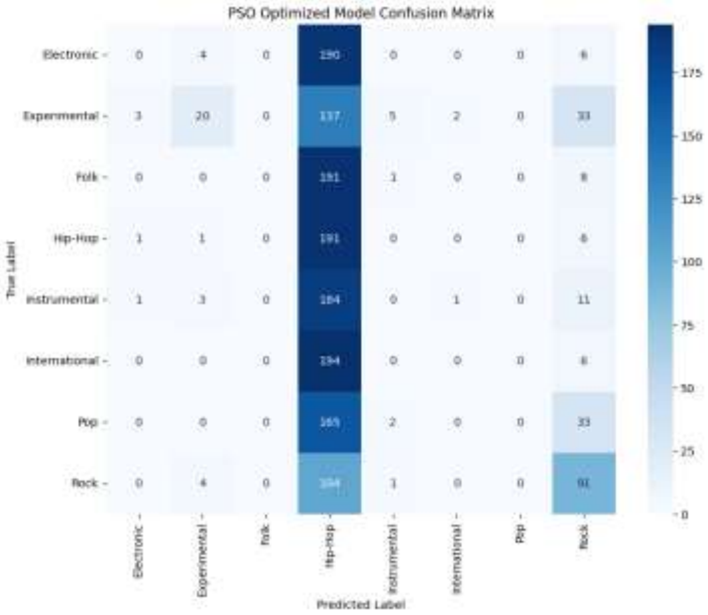
- Baseline Accuracy: 12.51% (Random Guessing)
- PSO-Optimized Accuracy: 25.45% (+103% improvement)

Challenge

PSO found that a much smaller learning rate was the key to unlocking the model's ability to learn.

Key Takeaway

PSO acted as a powerful problem-solver.



Case Study:
CIFAR-100 - Polishing a Strong Model

Challenge

Our baseline was already strong for this problem (~59% accuracy). Can PSO still find improvements?

Key Takeaway

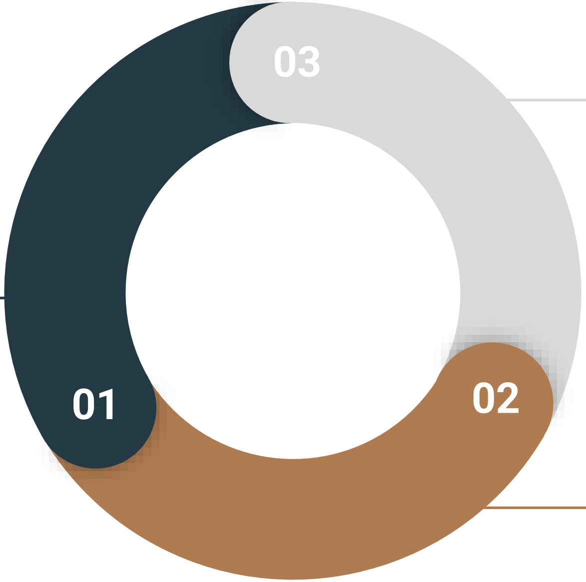
PSO provided a modest but consistent improvement across all metrics, acting as a fine-tuner.

Metric	Baseline	PSO	Change
Accuracy	0.5897	0.5975	+0.78%
Balanced Accuracy	0.5897	0.5975	+0.78%
Macro F1-score	0.5868	0.5954	+0.86%
MCC	0.5857	0.5935	+0.78%
Cohen’s Kappa	0.5856	0.5934	+0.78%

The Versatile Roles of Particle Swarm Optimization

The Problme-Solver

- On difficult tasks where the baseline fails.
- Finds fundamentally better configurations.
- Example: FMA Audio



The Revealer of Trade-offs

- On complex data with outliers or imbalance.
- Highlights that the “best” model depends on the goal.
- Example: Autolist Cars, Yu-Gi-Oh!

The Refiner

- On well-posed problems with a strong baseline.
- Polishes hyperparamaters for small, consistent gains.
- Example: CIFAR-100, NASA Exoplanets

Conclusion & Future Work

Conclusion

- This project successfully validated PSO as a powerful and flexible method for evolving neural networks.
- Its true value is its adaptability–its role and impact depend heavily on the problem context.
- The success of any model is an interplay between architecture, hyperparameter optimization, and the data itself

Future Work

- Implement advanced outlier handling (e.g., log transforms).
- Perform more exhaustive PSO searches on GPU.
- Use data augmentation for image/audio tasks.

Thank You

GitHub Link:
https://github.com/bbandic1/Evolving-Neural-Networks-with-PSO