AI and Evolutionary Multi-objective Optimization Data

SIEMENS

ADITYA LAKSHMI NARAYANAN, KEVIN PATEL, YASASREE SINGAM

PURPOSE

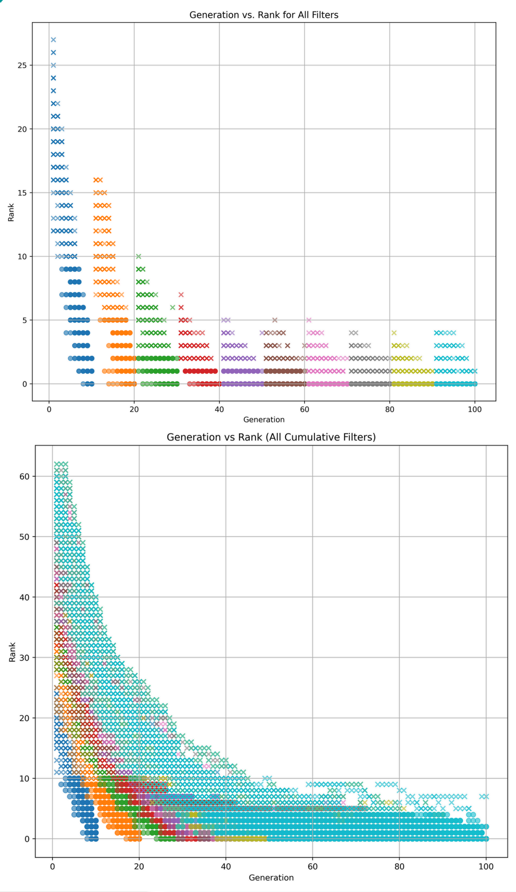
The purpose of this project is to enhance design space exploration algorithms by developing an adaptive AI that reliably identifies 'good' and 'bad' regions in the design space.

OBJECTIVE

Our project aims to develop an AI that predicts decision boundaries in a biased 10-dimensional design space, dynamically distinguishing 'good' and 'bad' classifications through iterative and cumulative filters.

Create design recommendations model using Inverse DNN and MDN for multi-objective optimized data.

FILTERING TECHNIQUES



ITERATIVE APPROACH: Non-dominated sorting across 10 filters for local rankings, iteratively refining towards Pareto efficiency.

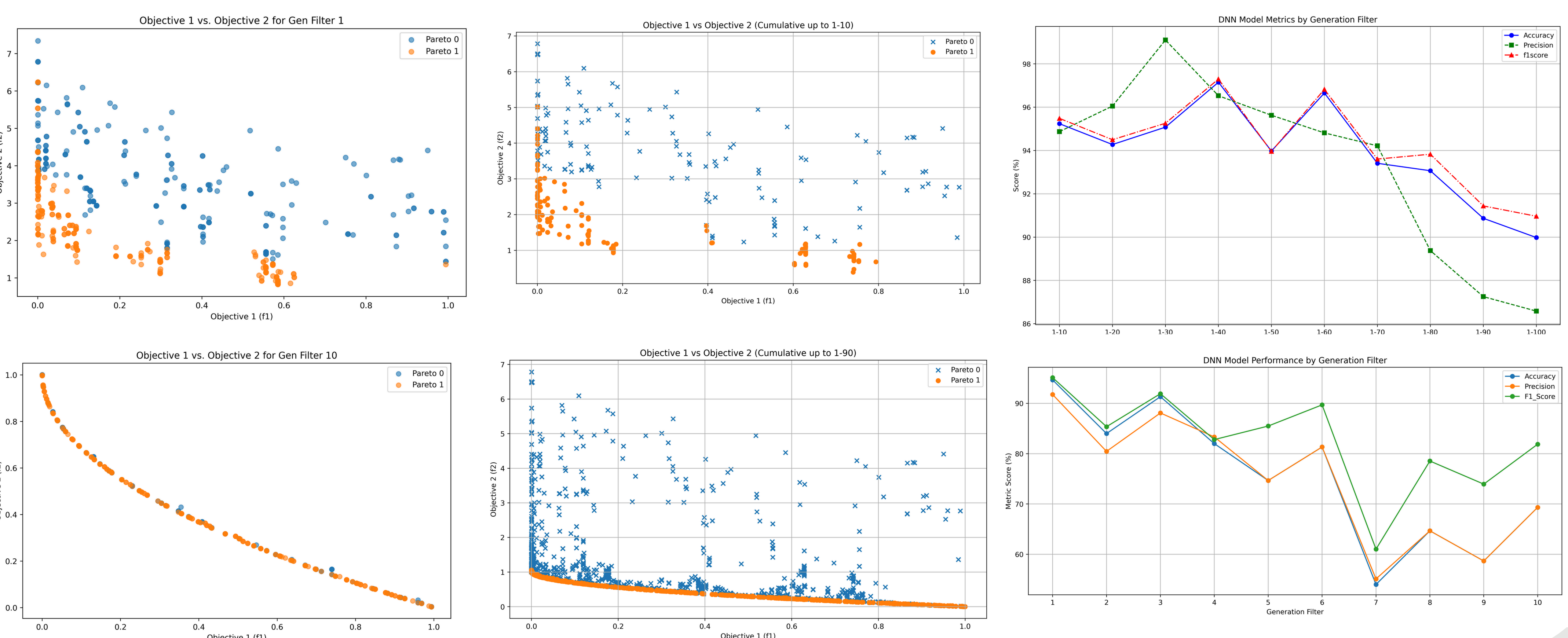
CUMULATIVE APPROACH: Non-dominated sorting on aggregated data from sequential generations, cumulatively assessing and identifying Pareto-efficient solutions.

FORWARD DNN AND RESULTS

DNN MODEL ARCHITECTURE:

- Input Layer: 10 Dimensional Feature vector
- Layers: 2 hidden layers with 64 neurons each, ReLU Activation.
- Output: Single Nueron, Sigmoid Activation
- Binary Threshold: > 0.5 for classification, 1: Pareto, 0- Non Pareto
- Loss Function: Binary Cross Entropy
- Optimizer: Adam, Learning rate of 0.001

CONCLUSION: The cumulative approach outperformed the Iterative model, maintaining over 90% accuracy across all the filters in distinguishing between Pareto and non-Pareto solutions.



INVERSE DNN

Aim: Create Inverse DNN to recommend design feature

Inverse DNN: an inverse version of Deep Neural Network that learns the distribution property of data to generate features for targeted class

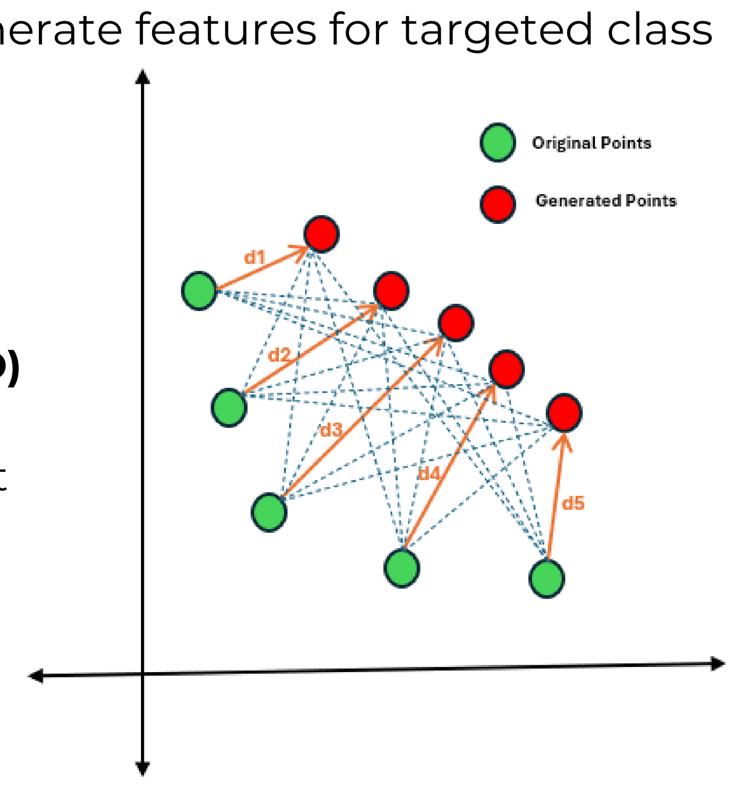
Architecture :
Input: Random Variable
Output: features set (Multi-objective Optimised Data)

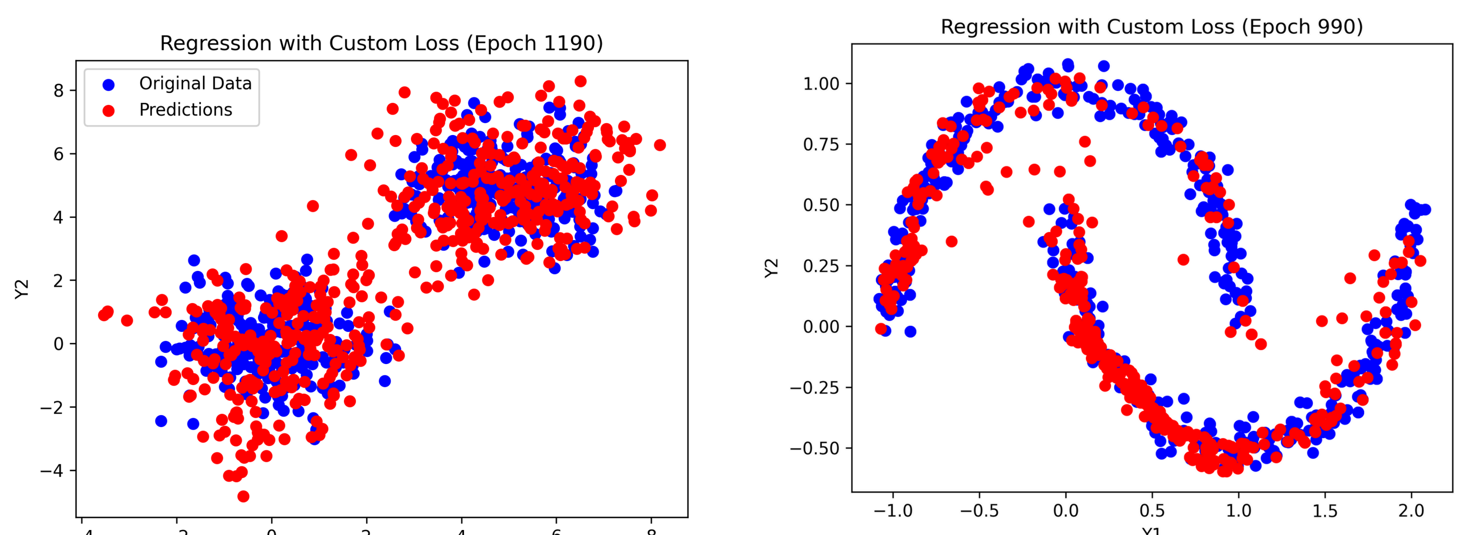
loss: Inverted Generational Distance (IGD)

- Average of the distance of closest predicted points for each original point

$$\mathrm{IGD}(A) = \frac{1}{|Z|}\left(\sum_{i=1}^{|Z|} \hat{d}_i^{\,p}\right)^{1/p}$$

Aim : reduce Avg(d1,d2,d3,d4,d5)





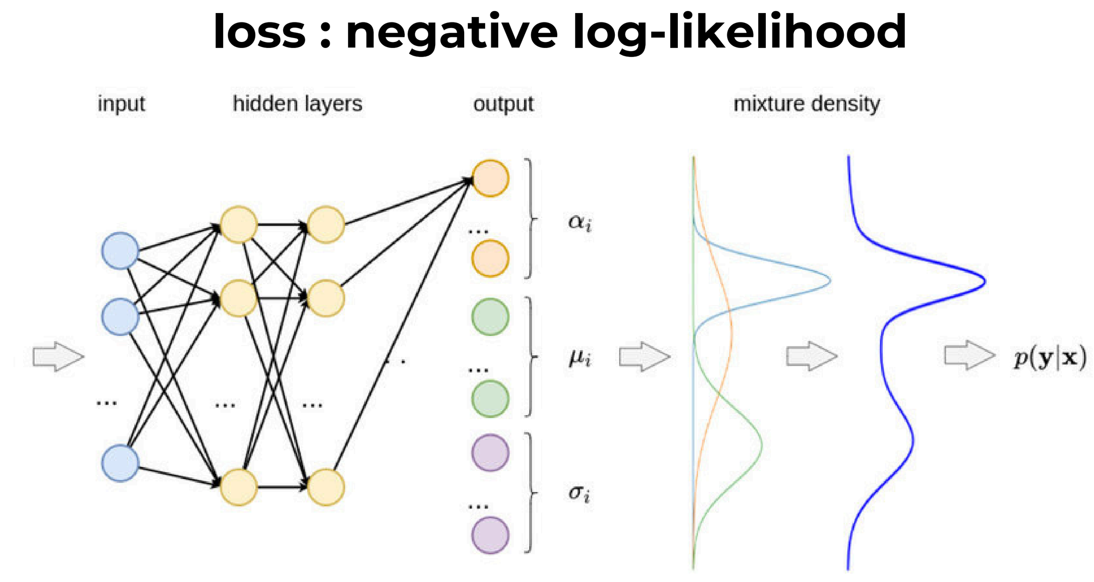
Observation and Conclusion :

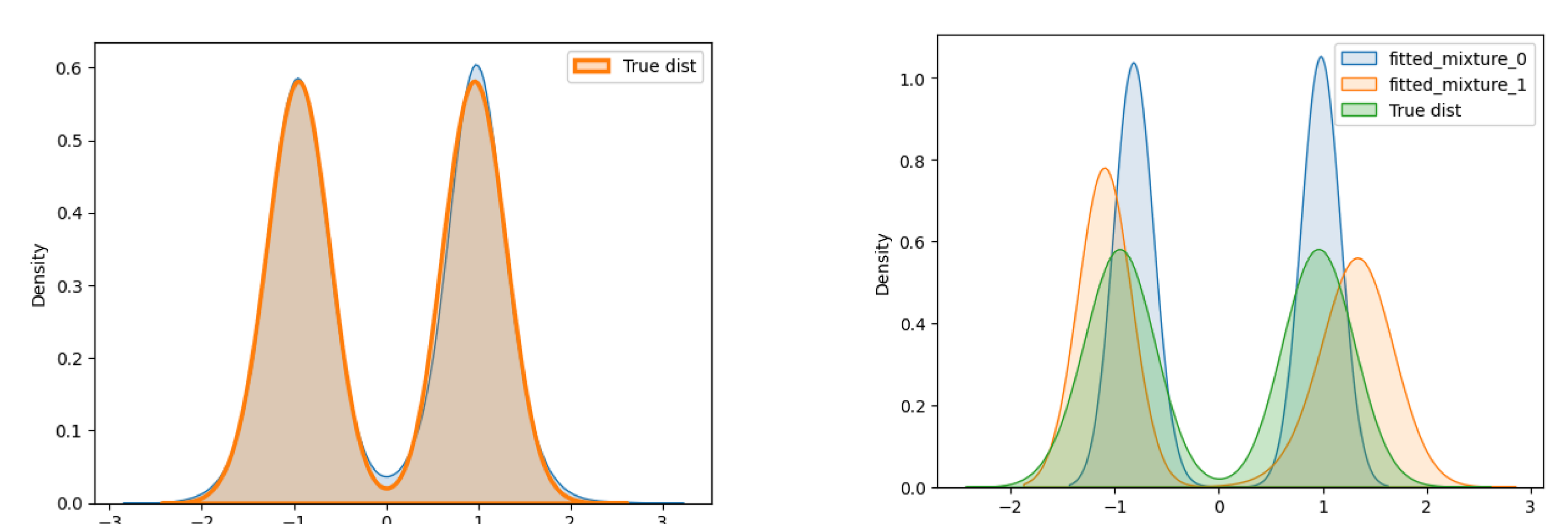
- If we use 1 random variable, then irrespective of our original distribution and model architecture, we will able to generate a distribution that will be a continuous line
- We cannot generate the provided distribution with a Meshgrid distribution of random points.
- Based on experiments we have analyzed that the higher number of random variables introduces some unnecessary noise to generation.
- Hypothesis: As per the experiment we felt the 2-inverted funnel architecture of DNN will be the most suitable architecture for generative purposes.

MIXTURE DENSITY NETWORKS (MDN)

MDNs are a neural network model designed to predict the probability density function (PDF) for a multimodal regression problem. The underlying multimodal model is a Gaussian mixture model, which is characterized by mixing proportions, means, and covariances of the mixture components. The deep feedforward network learns to output these properties to enable the calculation of the PDF

loss : negative log-likelihood





Observation and Conclusion :

- When working on the MDN, we notice that the mixtures' fittings (cumulative sum of the Gaussians) achieve almost perfect replication of the y variable.

MICHIGAN STATE UNIVERSITY