

# ML 2025 Project

Authors: Davide Petillo, Valerio Stancanelli

STFfanclub

Master degree (AI, Data Science)

d.petillo@studenti.unipi.it

v.stancanelli1@studenti.unipi.it

**Università di Pisa**

*Type of project: B*

Date 23/01/2026

# Objectives

## Main Objectives

- **Monk Tasks:** Achieve perfect accuracy on noiseless tasks and consistency with **noise** on Monk 3.
- **CUP Challenge:** Exploit the specific **geometry of the data** to maximize performance.

## Explored Models & Algorithms:

- **Neural Networks:** Varied architectures (shallow to deep), optimizers (Adam vs SGD), and regularization schemas.
- **Linear Basis Expansion (LBE):** Basis chosen empirically via **Fourier Analysis**.
- **Other Approaches:** K-NN, Random Forest, Ensembling, Stacking, SVR.

Studying the plots and the correlations among features, lead us to make important assumptions on the geometry of the data.

## 2. MONK: Method & Validation

**Libraries used:** PyTorch, sklearn

**Model Selection Strategy:**

- **Validation Schema:** **5-Fold Cross Validation** was used for hyperparameter tuning to ensure robustness.
- **Metric:** Model selection based on Mean Accuracy across folds.

**Hyperparameter Search Space (Grid Search):**

- **Architectures:** [5], [10], [20], [10, 10], [5, 5, 5]
- **Activations:** ReLU, Tanh
- **Learning Rate**  $\eta$ : 0.01, 0.05, 0.1, 0.2, 0.3    **Momentum**  $\alpha$ : 0.2 – 0.9
- **Regularization**  $\lambda$ :  $0, 8 \cdot 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}$
- **Epochs:** 50, 100, 150, 200, 300, 500, 600, 1000
- **Initialization:** Xavier for tanh, kaimig uniform otherwise.

This grid-search took about 2 hours to run.

### 3.1 MONK Results: Summary Table

Table 1: CUP performance summary.

Task	Best Conf. (Arch, $\eta$ , $\alpha$ , $\lambda$ , act, #ep)	MSE (TR / TS)	Acc (TR / TS) %
MONK 1	[20], 0.3, 0.95, 1e-4, ReLU, 300	0.0013/0.0002	100%/100%
MONK 2	[20], 0.2, 0.95, 8e-5, ReLU, 150	0.0003/0.0004	100%/100%
MONK 3 (Reg)	[10], 0.05, 0.9, 1e-4, ReLU, 600	0.0470/0.0460	94.3%/95.8%
MONK 3 (No-Reg)	[5, 5, 5], 0.01, 0.9, 0, tanh	0.0517/0.0501	95.1%/95.6%

Table 2: Table of results for MONKS

### 3.2 MONK 1 Results: Learning Curves

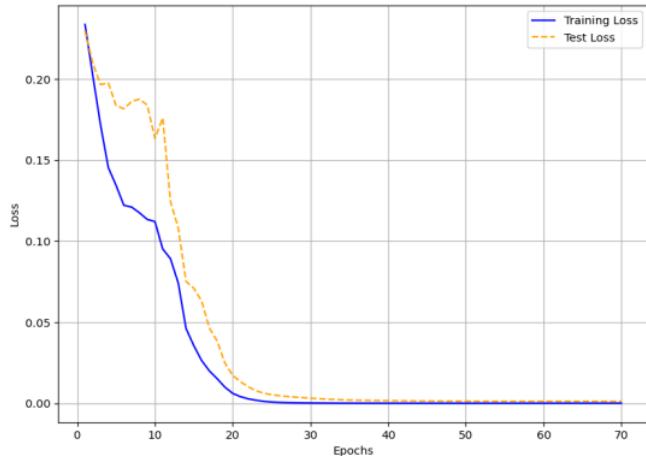


Figure 1: MONK 1 - MSE (TR vs TS)

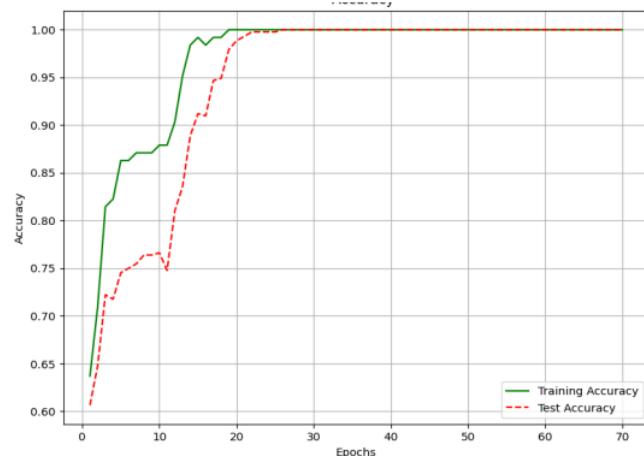


Figure 2: MONK 1 - Accuracy (TR vs TS)

### 3.3 MONK 2 Results: Learning Curves

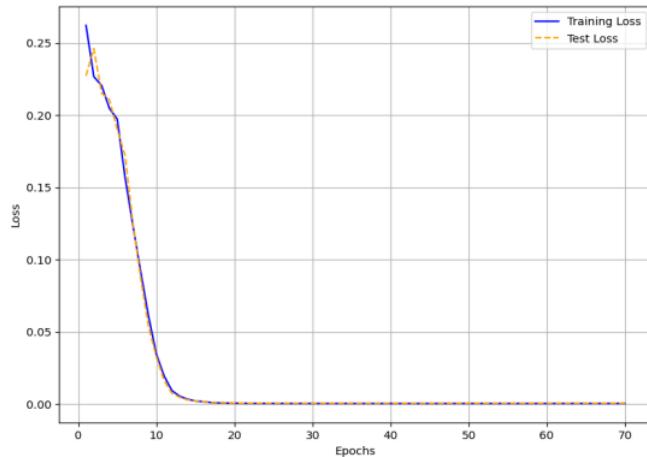


Figure 3: MONK 2 - MSE (TR vs TS)

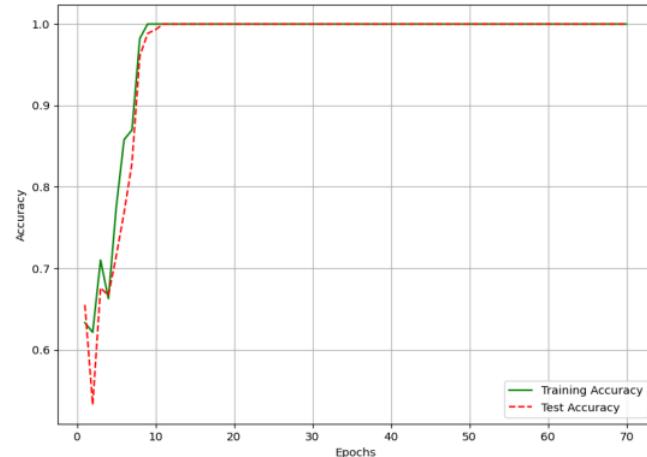


Figure 4: MONK 2 - Accuracy (TR vs TS)

## 3.4 MONK 3 Results: Learning Curves

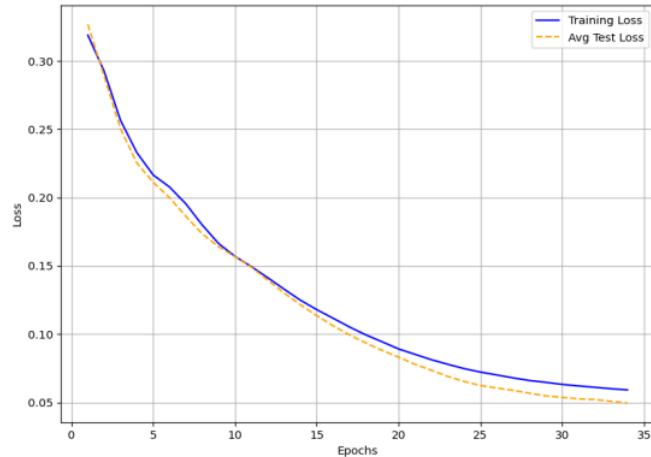


Figure 5: MONK 3 (Reg) - MSE

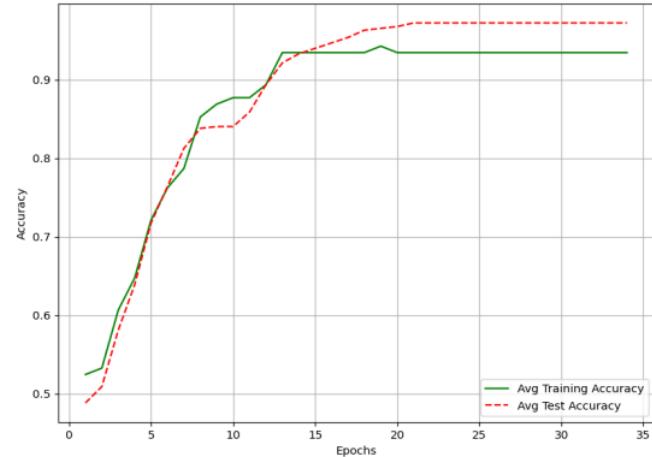


Figure 6: MONK 3 (Reg) - Accuracy

Note: *TS Accuracy (Blue) surpasses TR Accuracy (Red), indicating successful noise handling.*

## 4. CUP: From Naive to Geometric

### Initial Approaches (Failed):

- Standard FFNN (Input → Output), Chaining, Error-correction NNs.
- Result: High error, inability to capture the underlying manifold.

**Data Analysis Breakthrough:** We found that  $z = y_3 - y_4$  could determine all the other targets exactly. After some plotting, and some reconnaissance of functions, and after having fit them on the data, we ended up with the following exact equations:

$$\begin{cases} z = y_3 - y_4 \\ y_1 = 0.5463 \cdot z \cdot \cos(1.1395 \cdot z) \\ y_2 = 0.5463 \cdot z \cdot \sin(1.1395 \cdot z) \\ y_3 + y_4 = -z \cdot \cos(2z) \end{cases}$$

## 4.1 CUP: new discoveries

We also found a strong linear correlation between  $z$  and inputs, especially if we took the first principal component. From this point on, we focused on predicting  $z$ .

- Linear Regression on  $z$  using inputs failed ( $\text{MAE} \approx 1$ ).
- using NN, Random Forest, SVR failed
- k-NN did not fail ( $\text{MAE} \approx 0.8$ , still not enough but better)

So we tried k-NN in a lot of variants, using PCA, using products of the inputs, products of the PC, but nothing improved.

But after some time the pseudo-success of k-NN gave us an idea: "Maybe the manifold has some local structure which is predictable, but that can not be seen if looking at the whole picture".

## 4.2 CUP: Other geometric discoveries

That idea lead us to zoom on the plot of  $z$  against other variables, and what we found was very good.

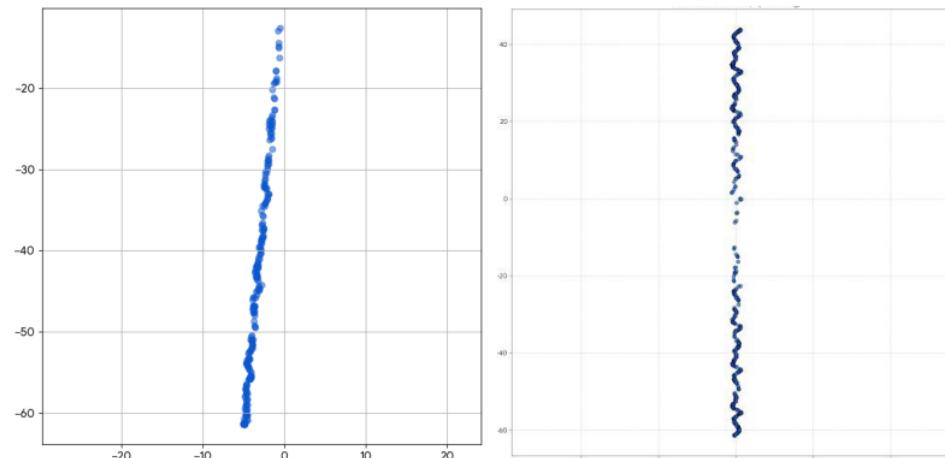


Figure 7: Plot of  $z$  against PC1 (left) and PC2 (right)

There were oscillations! And they seemed to be periodic.

## 4.3 CUP: The inverse idea

Noticing that there seemed to be a relationship of the kind  $\text{pc2} = f(z)$ , made us think about reversing the process of prediction. We train a model to predict a certain representation of the input and then we try to invert the process to get the correct  $z$ . NN and other common

models didn't work, but one did. **Random Forest (Inverse Model):**

- Idea: Train  $RF(z) \rightarrow$  First K Principal Components of  $x$ .
- Result: MEE dropped to  $\approx 12$  with  $K = 6$  and 1000 estimators.
- *Limitation:* Worked well only in dense data regions. Needed a continuous function representation.

## 4.4 CUP: The residual analysis

After some time spent trying to improve the models above, we noticed that other principal components seemed to be oscillating, so our intuition was: "Maybe every input has an oscillating component apart from the linear one...".

**Residual Analysis:** We analyzed the residuals of the inputs  $x_i$  when approximated linearly with respect to  $z$ .

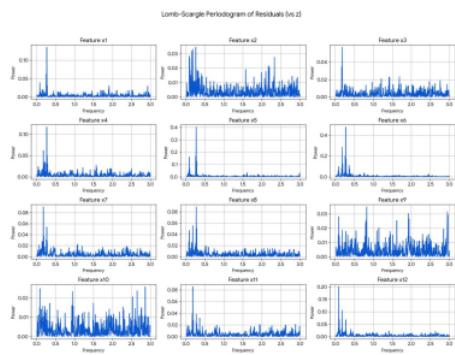


Figure 8: Fourier Analysis of the residuals of the input  $x_i$  approximated linearly with  $z$ .

- Fourier analysis revealed a clear **Fundamental Frequency**  $\omega = 0.57$ .
- Inputs are not just noisy; they contain systematic harmonic components dependent on  $z$ .
- This justified the use of a basis expansion model.

## 4.4 CUP: LBE Inverse Solver

We model each input  $x_i$  as a function of  $z$  using **Linear Basis Expansion (LBE)**:

$$\hat{x}_i(z) = c_{i,0}z + \sum_{k=1}^K (a_{i,k} \sin(k\omega z) + b_{i,k} \cos(k\omega z))$$

### Components:

- Linear trend ( $z$ ).
- $K$  Harmonics of the fundamental frequency  $\omega = 0.57$ .
- Weights  $w_i = 1/MSE_i$  calculated during training to penalize noisy inputs.

## 4.5 CUP: Inference and Refinement

To predict  $z_{pred}$  given a new input vector  $x_{new}$ :

- ➊ **Coarse Grid Search:** Evaluate the weighted error function  $E(z)$  over  $z \in [-70, 50]$  (discretized):

$$z_{init} = \arg \min_z \sum_{i=1}^{12} w_i (\hat{x}_i(z) - x_{new,i})^2$$

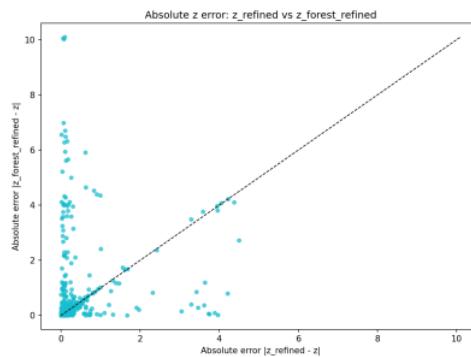
- ➋ **Fine Refinement:** Apply a Newton-based optimization starting from  $z_{init}$  to find the local minimum.
- ➌ **Target Reconstruction:** Compute  $y_1, y_2, y_3, y_4$  using the geometric equations (Slide 8) with the refined  $z$ .

**pc2 refinement:** estimate  $pc2$  via PCA and minimize  $(f(z) - pc2)^2 + \lambda(z - z_{init})^2$  to obtain  $z_{refined}$ .

## 4.6 CUP: Ensemble $z_{ens}$

We combine two estimates when they agree:

- if  $|z_{refined} - z_{forest\_refined}| < \alpha$ , use their weighted average
- otherwise keep  $z_{refined}$



$z_{refined}$  is the value given by the LBE inverse solver, and  $z_{forest\_refined}$  is the value given by a random forest model. This last model has higher MEE, so we give it low weight, and we use it only when we estimate it's accurate enough (i.e., close to  $z_{refined}$ ).

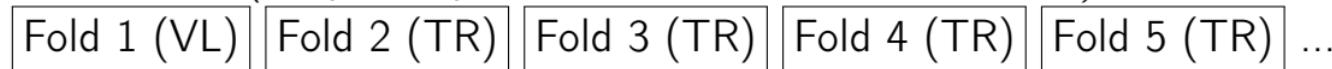
Figure 9: Error comparison:  $z_{refined}$  vs  $z_{forest\_refined}$ .

## 5.1 CUP: Validation Schema

### Data Splitting:

- **Internal Test Set:** 10% of the full training set is held out.
- **Cross Validation:** 5-fold CV is run on the remaining 90% for model selection.
- **Final Evaluation:** selected hyperparameters are evaluated once on the internal test set.

*(Graphic representation of 5-Fold CV on 90%)*



## 5.2 CUP: Model Selection

### Hyperparameters Tuned (Grid Search):

- **Refinement**  $\lambda$ : regularization for  $z$  refinement via  $pc2$ .
- **Ensemble weight**  $\gamma$ : mixing coefficient between  $z_{refined}$  and  $z_{forest\_refined}$ .
- **Agreement threshold**  $\alpha$ : decides when the ensemble average is used.

### Selection Criterion:

- pick the combination with the lowest average MEE across CV folds.

### Grid Search:

- $\lambda \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 1.0, 2.0, 5.0\}$
- $\gamma \in \{0.05, 0.1, 0.15, 0.2, 0.3\}$
- $\alpha \in \{0.3, 0.6, 1.0, 1.5, 2.0, 3.0\}$

The best hyperparameters found (with lowest MEE on the validation set) are the following:  
 $\lambda = 1.0$ ,  $\gamma = 0.1$ ,  $\alpha = 1.5$ . This model clearly outperforms the other models we have tried.

## 5.2.1 CUP: Grid Search Summary

Total combinations: 300

Rank	$\lambda$	$\gamma$	$\alpha$	Train (avg $\pm$ std)	Val (avg $\pm$ std)	Test (avg)
1	1.00	0.10	1.50	$6.62 \pm 0.33$	$8.48 \pm 0.97$	7.84
2	1.00	0.10	3.00	$6.63 \pm 0.33$	$8.49 \pm 0.99$	7.87
5	1.00	0.15	1.00	$6.55 \pm 0.32$	$8.53 \pm 1.06$	7.82
20	2.00	0.10	1.00	$6.40 \pm 0.29$	$8.56 \pm 0.96$	7.73
60	5.00	0.05	1.50	$6.43 \pm 0.31$	$8.65 \pm 0.91$	7.72
300	0.02	0.30	3.00	$7.79 \pm 0.40$	$9.36 \pm 1.21$	8.39

## 5.3 CUP: Final Results

**Model:** Geometric Inverse Solver (LBE + Newton Refinement)

Dataset Partition	MEE (Mean)	Std. Dev
Training	6.620	±0.328
Validation	8.482	±0.965
Internal Test	7.843	—

Table 3: CUP performance summary.

**Computing Time:** approximately one minute in total. **Hardware:** Standard CPU (No GPU required).

## 6. Conclusions

### Summary:

- **MONK**: Confirmed that procedural complexity handling implies success on simpler tasks.
- **CUP**: Demonstrated that domain analysis (geometry) can vastly outperform black-box optimization.
- Achieved a reasonable performance with MEE  $\approx 8.5$ .

### Blind Test Results:

- Filename: STFfanclub\_ML-CUP25-TS.csv
- Nickname: **STFfanclub**

# Appendix A: Additional Plots

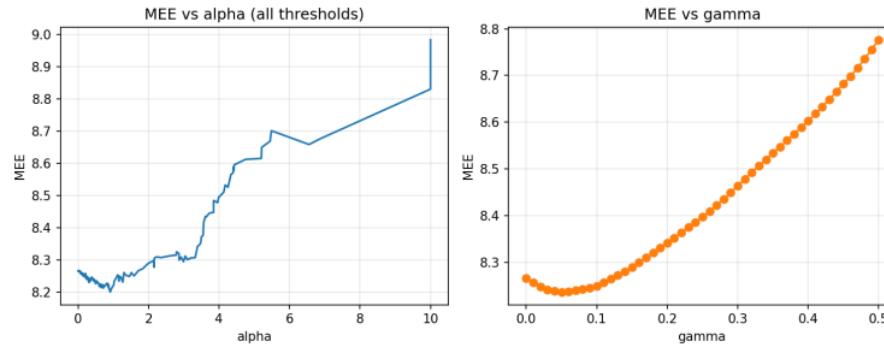


Figure 10: MEE vs  $\alpha$  for a fixed  $\gamma$ , and MEE vs  $\gamma$  for a fixed  $\alpha$ .

$z_{\text{forest\_refined}}$  has a relatively high MEE, and when it is far from  $z_{\text{refined}}$  it is most likely wrong. However, we can still improve the final MEE by giving it a low weight and using it only when the two values of  $z$  are close to each other.