USC-C

AI in Astronomy
INAF USC-C AI 1st Workshop
Catania, 21-23 May 2025

Deep Metal: Applying Deep Learning Models to Photometric Light Curves for Metallicity Estimation

+ BONUS

Lorenzo Monti

lorenzo.monti@inaf.it

INAF
ISTITUTO NAZIONALE
DI ASTROFISICA

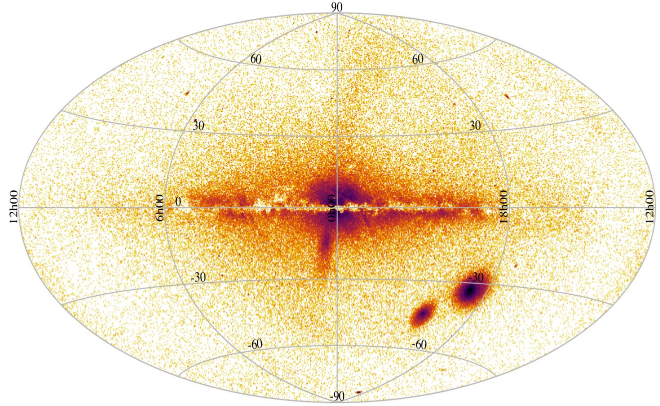
AGENDA

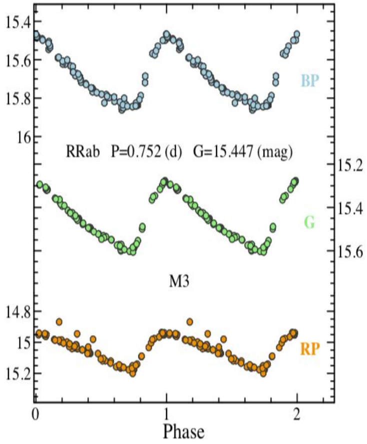
1. Our expertise
2. About the team
3. Open AI projects

- > Regression [Fe/H]
  - Transformer Regression [Fe/H]
- > AI-STARRS
- > Macchinino
- > Minos
- > AIDA

Our expertise

As an integral part of ESA's **Gaia DPAC**, our group specializes in studying **pulsating variable stars**, with a **primary focus on RR Lyrae**, by analyzing their photometric light curves from Gaia's G, BP, and RP bands.





Our expertise

> **Development of Specific Algorithms:** Design and implementation of Machine Learning and Deep Learning models optimized for:

- **Time-Series and tabular data Classification** specific to astrophysical data.
- **Regression and Parameter Estimation** from multi-modal data (e.g., CNNs, RNNs/LSTMs/GRUs, Transformers for light curves).
- **Semi-supervised clustering** for stellar archaeology.

> **Large-Scale Data Management and Processing:** experience with petabyte-scale data pipelines typical of **Gaia**.

> **Rigorous Model Validation:** Techniques to ensure the reliability and interpretability of AI results in a scientific context. We are working with **Umberto Michelucci**, Professor in Scientific Machine Learning, **HSLU**, Switzerland.

> **Integration with Astrophysical Knowledge:** Combining AI's data-driven approach with the physical understanding of stellar phenomena.

About the Team

Lorenzo Monti

Computer Scientist

Post-doc working on **Machine and Deep learning** for **time-domain astronomy**, with a focus on variable stars and photometric time series. Involved in the Gaia mission and the development of scientific data processing pipelines.

Tatiana Muraveva

Astrophysicist

Researcher specialized on **variable stars** as tracers of old stellar populations and tools for refining the cosmic distance scale, applying **Machine Learning** and **Deep Learning** to Gaia data.

About the Team

Alessia Garofalo
Astrophysicist

Researcher focuses on the **validation of RR Lyrae** pulsating variable stars observed by the **Gaia** satellite, and their scientific application as stellar tracers and standard candles as Population II stellar tracers and standard candles..

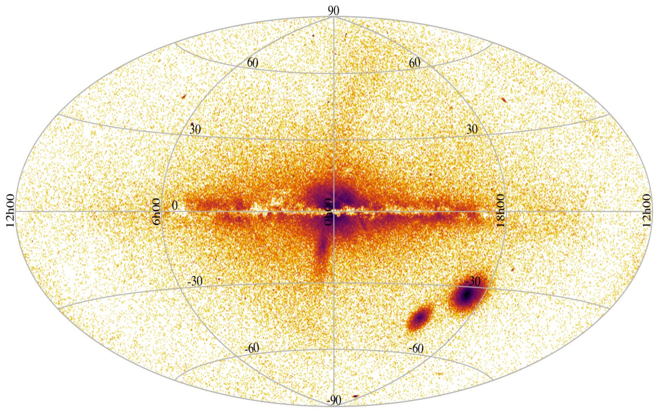
Gisella Clementini
Astrophysicist

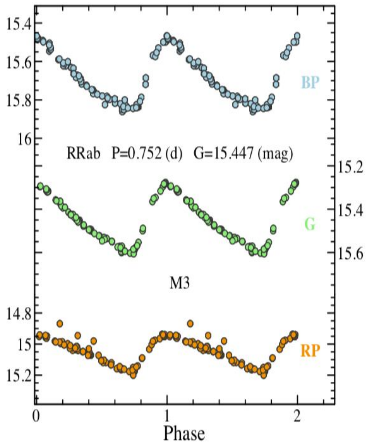
Has an internationally renowned expertise in the field of **pulsating variable stars**. She is one of the leading experts in the application of the **Gaia** mission to the determination of the properties of **RR Lyrae variable stars** and their use to establish distances and trace the ancient stellar population in galaxies.

Regression [Fe/H]

Scientific Rational

- RR Lyrae stars are periodic (Period < 1 day), pulsating, variable stars that play a crucial role in stellar astrophysics.
- There is a correlation between RRL's light curves and their metallicities ([Fe/H]).
- Gaia Data Release 3 provides a catalogue of 270 905 RRLs along with their time-series photometry.





<u>Project Main Goal</u>: Derive metallicities of RR Lyrae stars from their time-series photometry data using Machine Learning/Deep Learning algorithms.

Overview

Time-series Extrinsic Regression TSER(1) is a *regression task* that learns the mapping from time series data to a scalar value. That *task* depend on the whole series, rather than depending more on recent than past values such as time-series forecasting (**TSF**).

As described in *Tan, et al.* (1), a **TSER model** is a function $\mathcal{T} \rightarrow \mathcal{R}$, where $\mathcal{T}$ is a class of time series and $\mathcal{R}$ a class of scalar values. **TSER** seeks to learn a regression model from a dataset $\mathcal{D} = \{(t_1, r_1), \ldots, (t_n, r_n)\}$, where t_i is a time series and r_i is a continuous scalar value.

1. Tan, Chang Wei, et al. "Time series extrinsic regression: Predicting numeric values from time series data." *Data Mining and Knowledge Discovery* 35 (2021): 1032-1060.

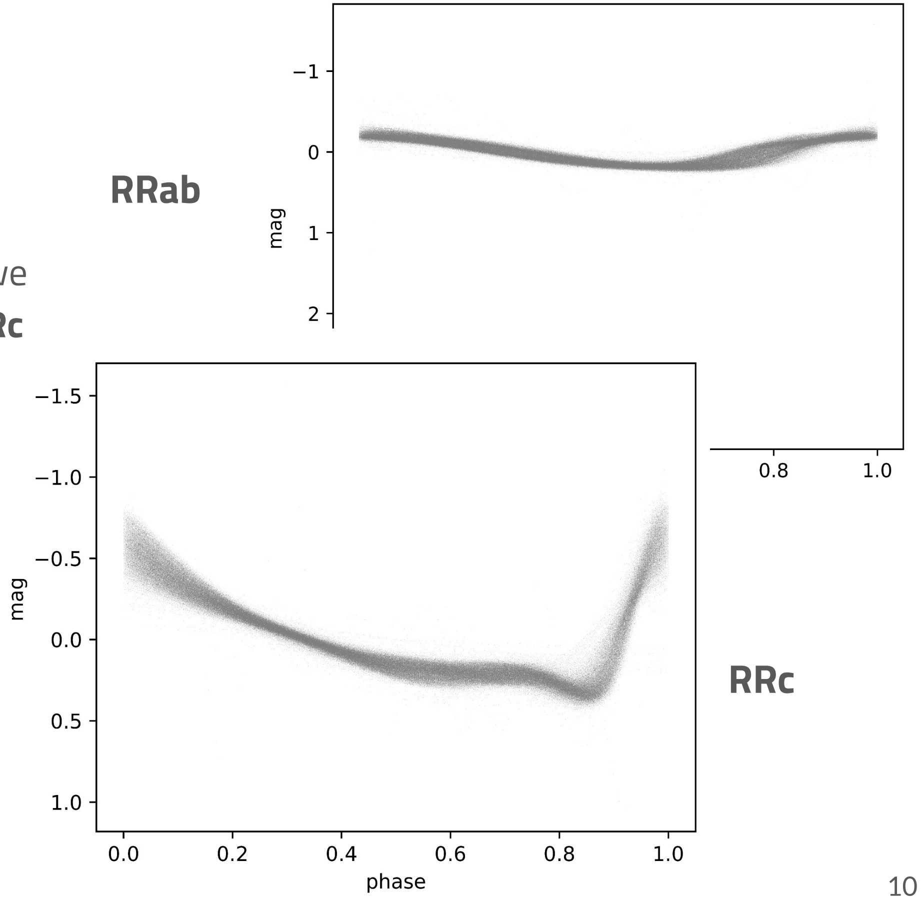
Dataset preparation

Dataset preparation

As regarding the time-series photometry dataset, we selected a set of **6002 RRab stars** and **6613 RRc stars** based on:

- err[Fe/H] < 0.4 dex
- peak-to-peak amplitude < 1.4 mag
- Number of epochs > 50
- ϕ_{31} error < 0.10

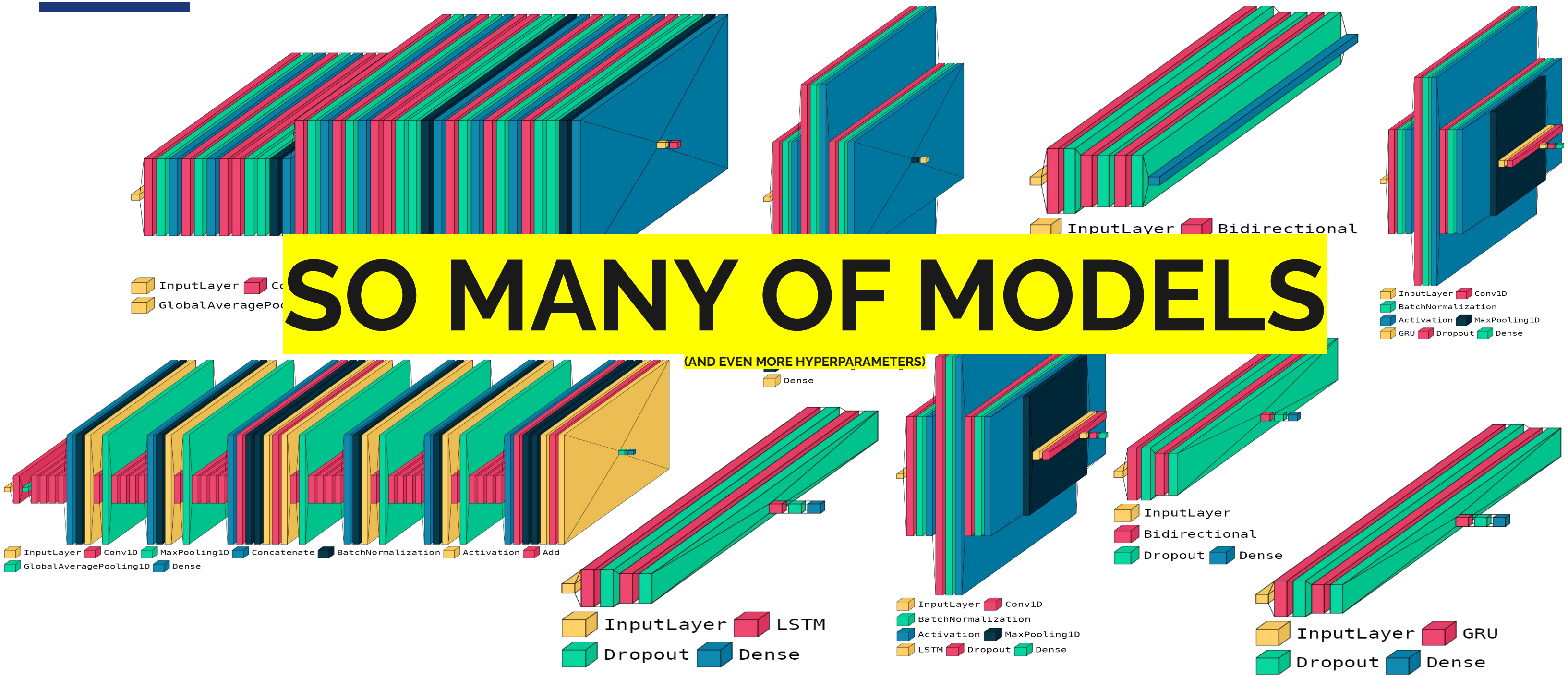
RRab



RRc

Dataset preprocessing

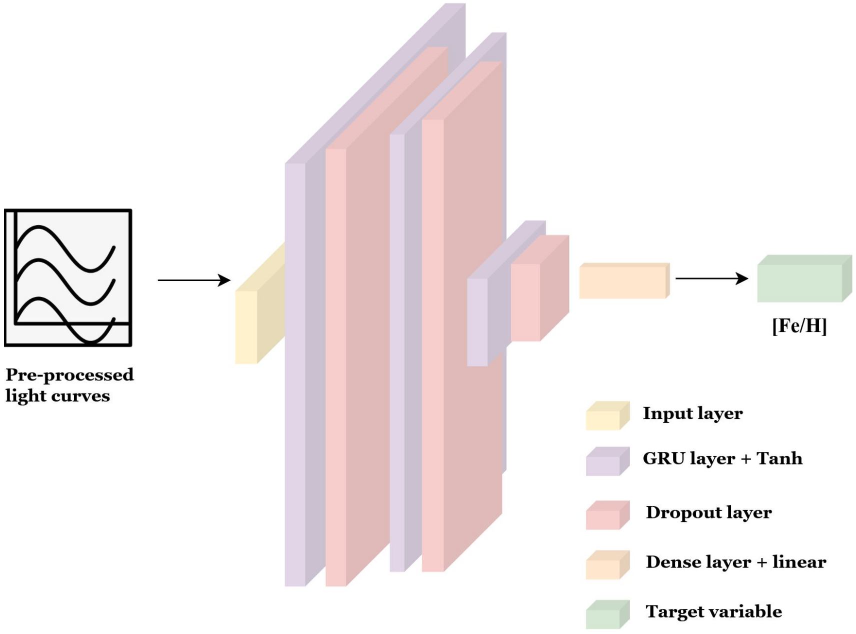
Methods	Description	Notes
two-dimensional sequences	$X^{<t>} = \begin{cases} m^{<t>} - <m> \\ Ph * P \end{cases} \quad t = 1, ..., N_{ep}$	where $m^{\langle t \rangle}$ is the magnitude of the light curve, *<m>* is the mean magnitude, *Ph* is the phase and *P* the period. N_{ep} is the number of epochs.
Smoothing spline	The method is applied to **minimize fluctuations, noise, outliers** and obtain the **same number** of points for each light curve (264 and 265).	final input tensor have the shape of: [**6002, 264, 2**] for *RRab* and [**6002, 265, 2**] for RRc → **[** batch size, time steps, features**]**.
Sample weights for metallicity distribution	we computed **Gaussian kernel density** estimates of the [Fe/H] distributions.	Evaluated them for every object in the datasets, and assigned a density weight w_d to each data point by taking the inverse of the estimated normalized density.



SO MANY OF MODELS

(AND EVEN MORE HYPERPARAMETERS)

Our architecture: GRU



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 264, 2)]	0
gru (GRU)	(None, 264, 20)	1440
dropout (Dropout)	(None, 264, 20)	0
gru_1 (GRU)	(None, 264, 16)	1824
dropout_1 (Dropout)	(None, 264, 16)	0
gru_2 (GRU)	(None, 8)	624
dropout_2 (Dropout)	(None, 8)	0
dense (Dense)	(None, 1)	9

Technical details

Training Phase:

- Used mean squared error (**MSE**) with **sample weights** as the cost function.
- Prevented overfitting with methods such as **kernel regularization** (L1 and L2) and **dropout** layers.

Hyperparameter Optimization:

- Optimization performed via **hyperband tuner**.
- Evaluated **dropout rates** [0.1, 0.2, 0.4, 0.6], **learning rates** [0.001, 0.01, 0.1], and **batch sizes** [32, 64, 128, 256, 512].

Training Details:

- Utilized the **Adam** optimization algorithm with a **learning rate** of 0.01.
- **Mini-batch** size set to 256
- Determined optimal **early stopping epochs** based on network type to prevent overfitting.
- Repeated stratified K-fold cross-validation (5 folds)

Performance Metrics:

- Root mean squared error (RMSE)
- Mean absolute error (MAE)
- Weighted RMSE (wRMSE)
- Weighted MAE (wMAE)
- Coefficient of determination (R^2 score)

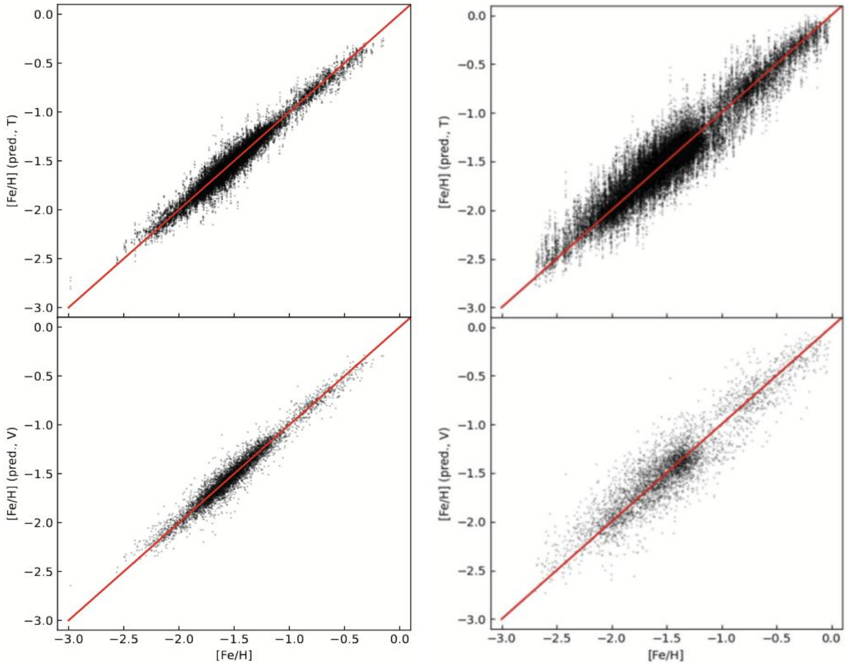
Results

We compared our results with the best result in scientific literature.

Our	(RRab) Fundamental-model		(RRc) First-overtone	
Metrics	Train	Val	Train	Val
R^2	0.9447	0.9401	0.9668	0.9625
wRMSE	0.0733	0.0763	0.0679	0.0722
wMAE	0.0547	0.0563	0.0490	0.0504
RMSE	0.0735	0.0765	0.0681	0.0720
MAE	0.0549	0.0565	0.0492	0.0505

Dekany	BiLSTM training	BiLSTM validation
r2	0,96	0,93
wRMSE	0,1	0,13
wMAE	0,07	0,1
RMSE	0,15	0,18
MAE	0,12	0,13

The plot on the left shows **our RRab metallicity prediction** results while the one on the right shows **Dekany's metallicity prediction** results(1).



1. Dékány, István, and Eva K. Grebel. "Photometric Metallicity Prediction of Fundamental-mode RR Lyrae Stars in the Gaia Optical and K s Infrared Wave Bands by Deep Learning." *The Astrophysical Journal Supplement Series* 261.2 (2022).

Results

Monti, Lorenzo, et al. "**Leveraging Deep Learning for Time-Series Extrinsic Regression in Predicting the Photometric Metallicity of Fundamental-Mode RR Lyrae Stars.**" *Sensors* 24.16 (2024): 5203.

Muraveva, Tatiana, et al. "**Metallicity of RR Lyrae stars from the Gaia Data Release 3 catalogue computed with Machine Learning algorithms.**" *Monthly Notices of the Royal Astronomical Society* 536.3 (2025): 2749-2769.

Monti, Lorenzo, et al. "**Unified Deep Learning Approach for Photometric Metallicity Estimation of Fundamental-mode and First-overtone RR Lyrae Stars Using Gaia Light Curves.**" [*under review in Astronomy and Astrophysics]*

Transformer Regression [Fe/H]

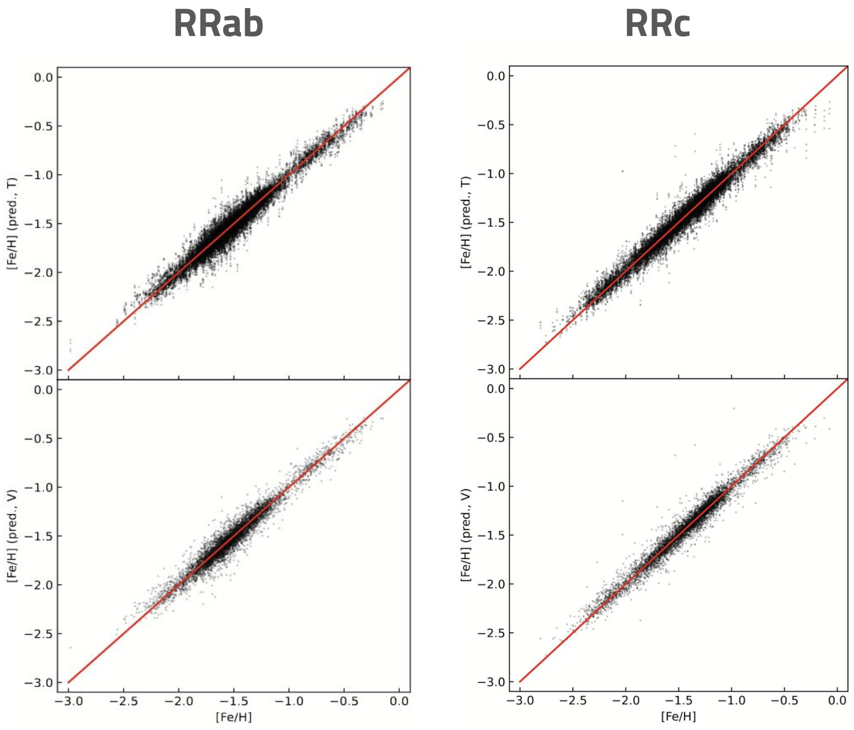
Overview

The application of the same pipeline (preprocessing and models) to **first-overtone RR Lyrae (type c)**, get results better than those obtained with RR Lyrae ab stars.

This led us to push the boundaries further by experimenting with new models: **Transformer** and **Informer**.

Only the encoder section of the classic Transformer encoder-decoder architecture is used.

From a computational perspective, the training phase is very demanding: we exploited a **Leonardo node at Cineca** for this purpose. The training was carried out using **4 Nvidia A100 GPUs**.



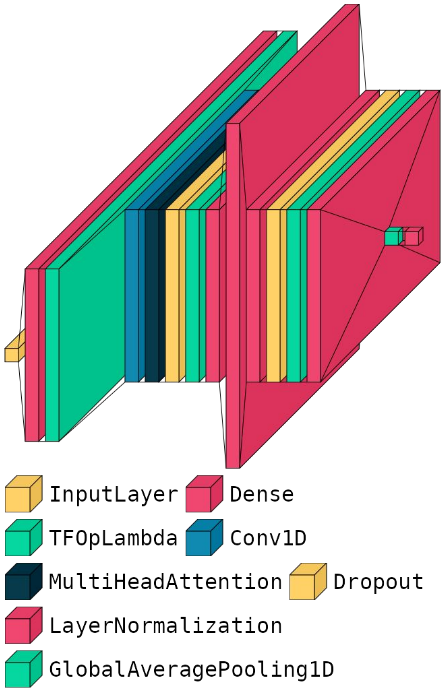
Differences between Transformer and Informer

Aspect	Transformer Encoder	Informer Encoder
Self-Attention	**Full attention**. Quadratic complexity O(n^2)	**ProbSparse** attention. Complexity: O(n log n).
Sequence Length Handling	General-purpose	Optimized for long sequences.
Efficiency	High memory and computational cost.	Lower memory and computational cost.
Positional Encoding	**Fixed or learnable**	Enhanced for periodicity (**timestamp**)
Distillation	**None**	**Sequence compression**

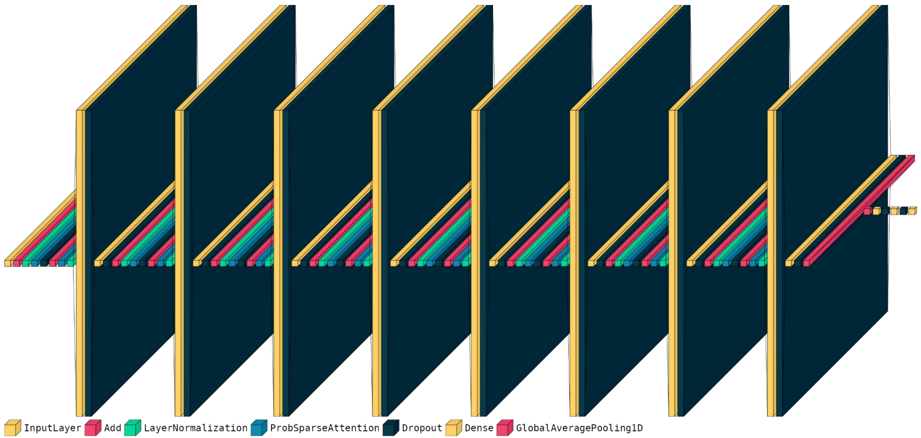
In summary, the Informer encoder is a specialized variant **optimized for long sequences**, achieving greater **efficiency through self-attention** (sparsity) and **distillation** while retaining much of the Transformer's representational power.

Our architecture

Architecture based on Transformer



Architecture based on Informer



Results

Metrics comparison between GRU results and Transformer results.

RR Lyrae ab

	GRU	
	training	validation
r2	0,947	0,9449
wRMSE	0,0723	0,0736
wMAE	0,0545	0,0551
RMSE	0,0727	0,074
MAE	0,0548	0,0554

	Transformer	
Metrics	training	validation
r2	0.9434	0.9390
w rmse	0.0742	0.0770
w mae	0.0543	0.0557
rmse	0.0743	0.0771
mae	0.0545	0.0560

The Transformer architecture **performs slightly worse** because the lengths of the light curves (264 and 265) are not very long. **We are waiting for Gaia DR4**, where the number of epochs will, on average, double.

RR Lyrae c

	GRU	
Metrics	training	validation
r2	0.9668	0.9625
w rmse	0.0679	0.0722
w mae	0.0490	0.0504
rmse	0.0681	0.0720
mae	0.0492	0.0505

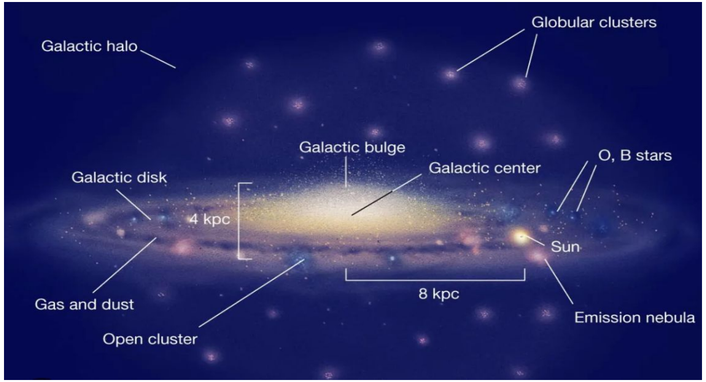
	Transformer	
Metrics	training	validation
r2	0.9546	0.9504
w rmse	0.0794	0.0830
w mae	0.0557	0.0565
rmse	0.0795	0.0832
mae	0.0560	0.0567

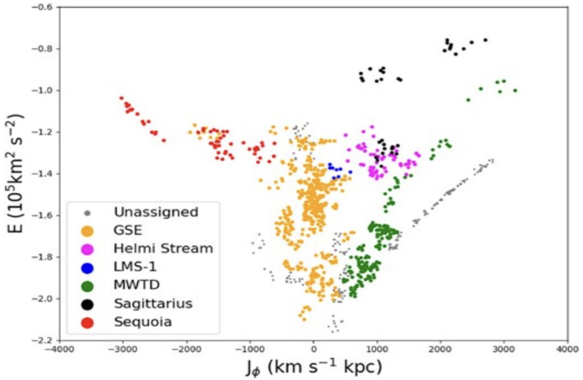
+ BONUS

AI-STARRS

Scientific Rational

RR Lyrae stars, being old stellar populations, are very useful for studying the **dynamical structures of the Milky Way**. By analysing **integrals of motion** of RR Lyrae stars (such as energy and angular momentum), it is possible to identify coherent stellar streams, remnants of disrupted satellites, and trace the assembly history of the Galaxy's halo.





Project Main Goal: derive sub-structured traced with RRLs from the integral of motion tabular data using clustering algorithms.

Overview and dataset preparation

Overview

This is a **clustering** task in which we have tabular data from **Gaia DR3** as input. In this sense, we derive the **integral of motion** clustering into the Milky Way.

Tests have been conducted with a **dataset**:

- [4042 rows, 7 features] → [*source_id, Jr, Jz, Energy, Lz, Lperp, FeH*]

Dataset preparation

> **Cluster tendency**: refers to the degree to which a dataset **inherently contains meaningful clusters** exploiting **Hopkins statistic.**

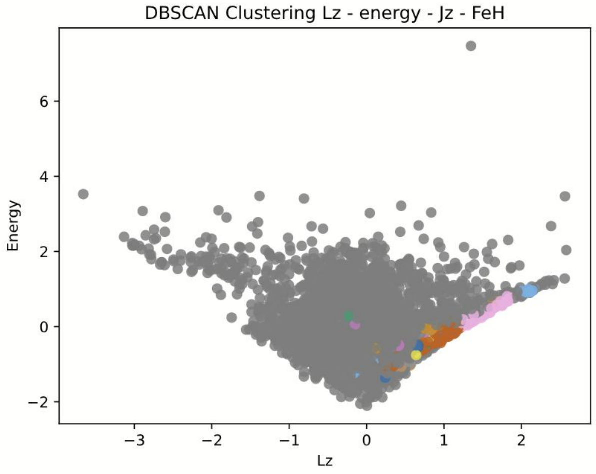
> We have therefore preprocessed the **energy** (integral of motion value) and **scaled the features.**

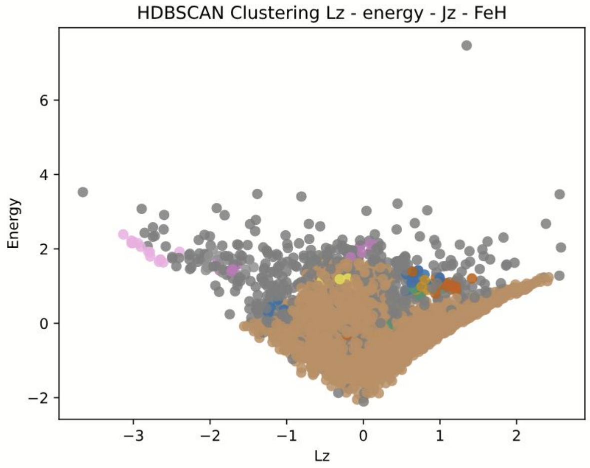
Preliminary model selection

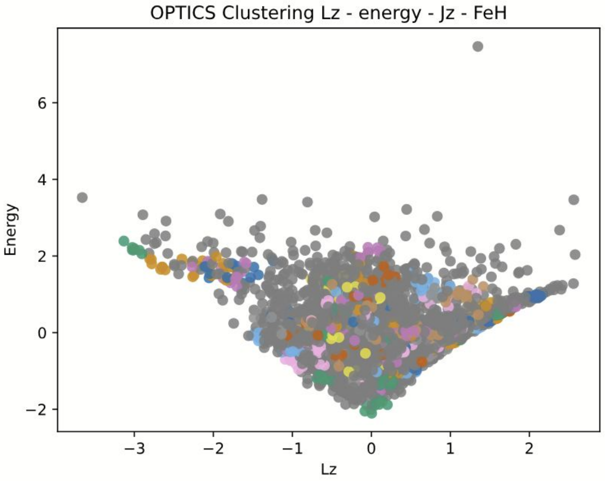
We started by testing various **density clustering algorithms**:

- **DBSCAN**
- **HDBSCAN**
- **Optics**

results → **not effective**







Our model: CLiMB

We implement a new **Multiphase Semi-supervised Clustering Framework** from scratch. **CLiMB (CLustering In Multiphase Boundaries)** is a versatile two-phase clustering algorithm designed to handle datasets with both known and exploratory components.

How it works

CLiMB operates in two phases:

1. **Constrained Phase (KBound)**: A modified K-means that:
    - Uses **seed points** and **centroids** to guide initial clustering.
    - Applies **density** and **distance** constraints.
    - Prevents centroids from drifting too far using **radial thresholds.**
    - Supports customizable distance metrics through the distance_metric and metric_params parameters.
    - Handles advanced seed points via a dictionary structure for more controlled initialization.
2. **Exploratory Phase**: Uses **density-based clustering** methods to discover patterns in points not assigned during the first phase. Supports **multiple clustering algorithms** (DBSCAN, HDBSCAN, OPTICS) through a strategy pattern.

Our model: CLiMB

We implement a new multiphase semi-supervised clustering algorithm from scratch. **CLiMB (CLustering In Multiphase Boundaries)** is a versatile two-phase clustering algorithm designed to handle datasets with both known and exploratory components.

Key Features

- **Two-Phase Clustering:** Integrates prior knowledge through constrained clustering and discovers new patterns via exploratory clustering.
- **Density-Aware Filtering:** Utilizes local density estimation to intelligently filter and assign data points.
- **Distance Filtering:** Applies distance-based criteria to refine cluster assignments, enhancing the separation of closely situated astronomical objects.
- **Flexible Exploratory Phase:** Supports multiple clustering algorithms (DBSCAN, HDBSCAN, OPTICS) through a strategy pattern.
- **Visualization Tools:** Includes built-in 2D and 3D visualization capabilities for cluster analysis.

- **Parameter Tuning**: Builder pattern for flexible parameter adjustment.
- **Customizable Distance Metrics**: supports various distance metrics such as Euclidean, Mahalanobis, and custom metrics, offering greater flexibility in distance calculation.
- **Advanced Seed Points**: Ability to initialize clustering with known seed points provided in a dictionary structure, allowing for more precise control over centroid initialization.

CLiMB: an example

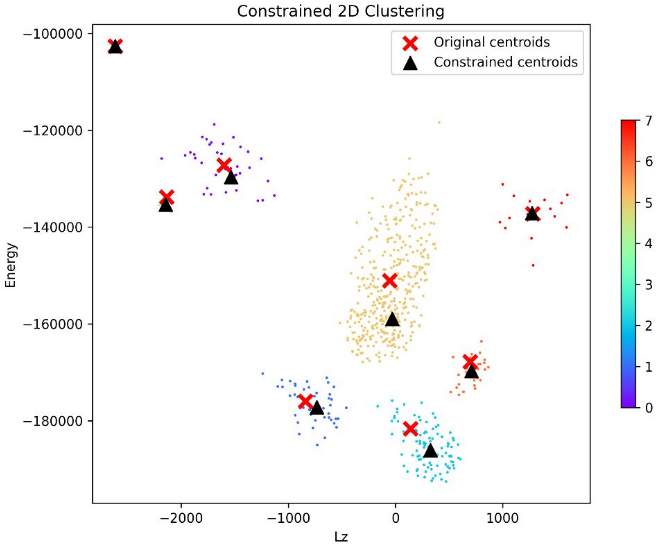
```
// put your coimport numpy as np
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler
from CLiMB.core.CLiMB import CLiMB
from CLiMB.exploratory.DBSCANExploratory import DBSCANExploratory

# The number of centers to generate
centers = 4

# Generate synthetic data with 5 dimensions
X, y = make_blobs(n_samples=500, centers=centers, n_features=5, random_state=42)

# Scale the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Create seed points (optional)
seed_points = np.array([
    X[y == i].mean(axis=0) for i in range(centers)
])
seed_points_scaled = scaler.transform(seed_points)

# Example of seed points as a dictionary for more precise control
seed_dict_scaled = {
    tuple(seed_points_scaled[0]): [tuple(X_scaled[y == 0][0]), tuple(X_scaled[y == 0][1])], # Centroid 1 and associated seed points
    tuple(seed_points_scaled[1]): [tuple(X_scaled[y == 1][0])],                             # Centroid 2 and associated seed points
    tuple(seed_points_scaled[2]): [],                                                         # Centroid 3 with no specific seed points
    tuple(seed_points_scaled[3]): [tuple(X_scaled[y == 3][0]), tuple(X_scaled[y == 3][1]), tuple(X_scaled[y == 3][2])] # Centroid 4 and seed points
}
```

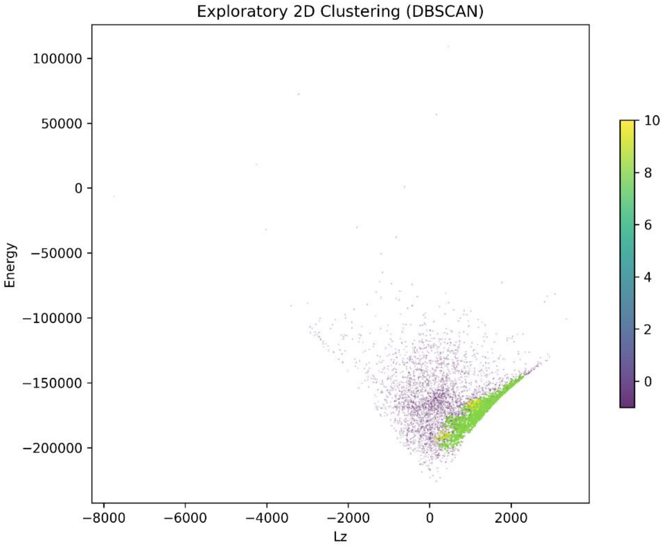
CLiMB: an example

```
# Initialize and fit CLiMB with Mahalanobis metric and dictionary seed points
climb = CLiMB(
    constrained_clusters=4,
    seed_points=seed_dict_scaled, # Use the dictionary of seed points
    density_threshold=0.15,
    distance_threshold=2.5,
    radial_threshold=1.2,
    convergence_tolerance=0.05,
    distance_metric='euclidean',
    metric_params=None,
    exploratory_algorithm=DBSCANExploratory(0.5)
)
climb.fit(X_scaled)

# Get cluster labels
labels = climb.get_labels()

# Visualize results (only possible in lower dimensions)
climb.inverse_transform(scaler)
fig = climb.plot_comprehensive_3d(save_path="./3d")
fig2 = climb.plot_comprehensive_2d(save_path="./2d")de here
```

Results

the plot shows **Constrained Phase (KBound)** on the left, and finds all the **known clusterings**, while in the plot on the right we use **DBSCAN** for the **Exploratory Phase**.

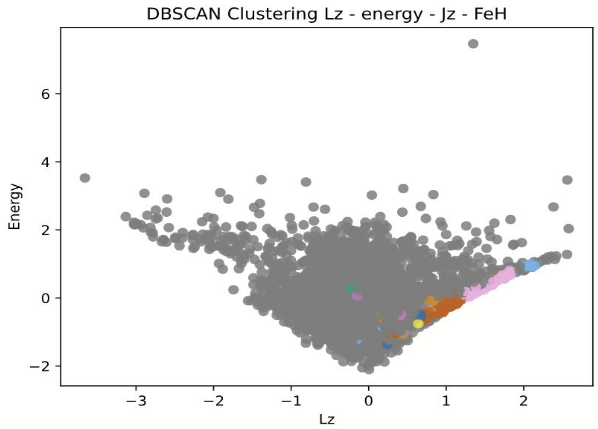


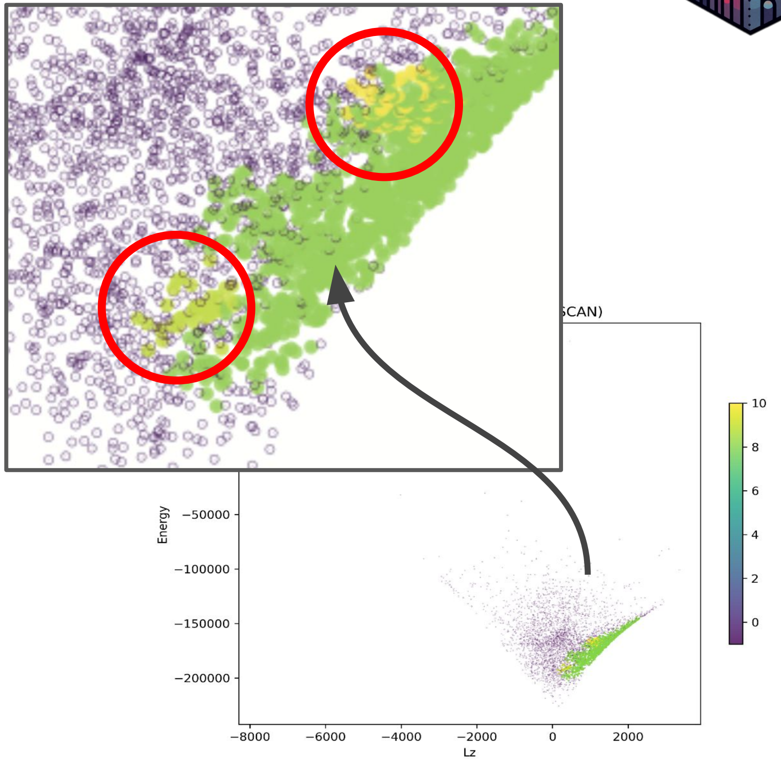


Results

Our framework automatically finds the (already known) **disk** and two new clusters that refer to **Shakti** and **Shiva**, 2 presumed proto-galactic fragments in the inner Milky Way[1]. We are **conducting further scientific tests** to verify the reliability of the obtained results.

BEFORE





AFTER

1. Malhan, Khyati, and Hans-Walter Rix. "Shiva and Shakti: Presumed proto-galactic fragments in the inner Milky Way." *The Astrophysical Journal* 964.2 (2024): 104.

Results

Monti, Lorenzo, et al. "**CLiMB: Charting the Milky Way's Past with Multiphase Semi-Supervised Clustering Framework.**" *[in preparation in Journal of Machine Learning Research]*


THANK YOU!
LYRAE