Anomaly Detection
Arrhythmia detection in ECG signal

SDS2024
IEEE SWISS CONFERENCE ON DATA SCIENCE

Aron Horvath
Antoine Madrona, Dennis Bader, Samuele Piazzetta

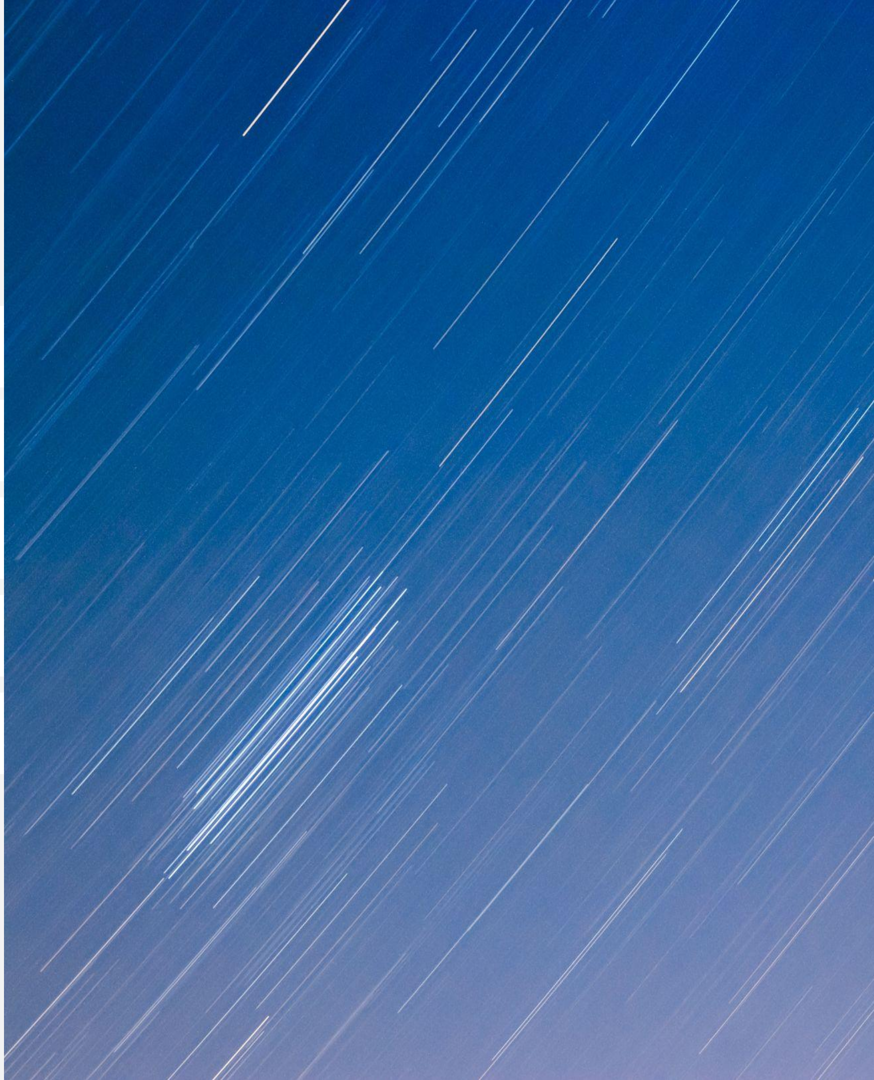
2024.05.30

Darts

Unit8™

Agenda

1. Why anomaly detection
2. Anomaly detection with Darts
3. Real-world example: Arrhythmia detection
4. Summary / concluding remarks
5. Time for the exercises



Agenda

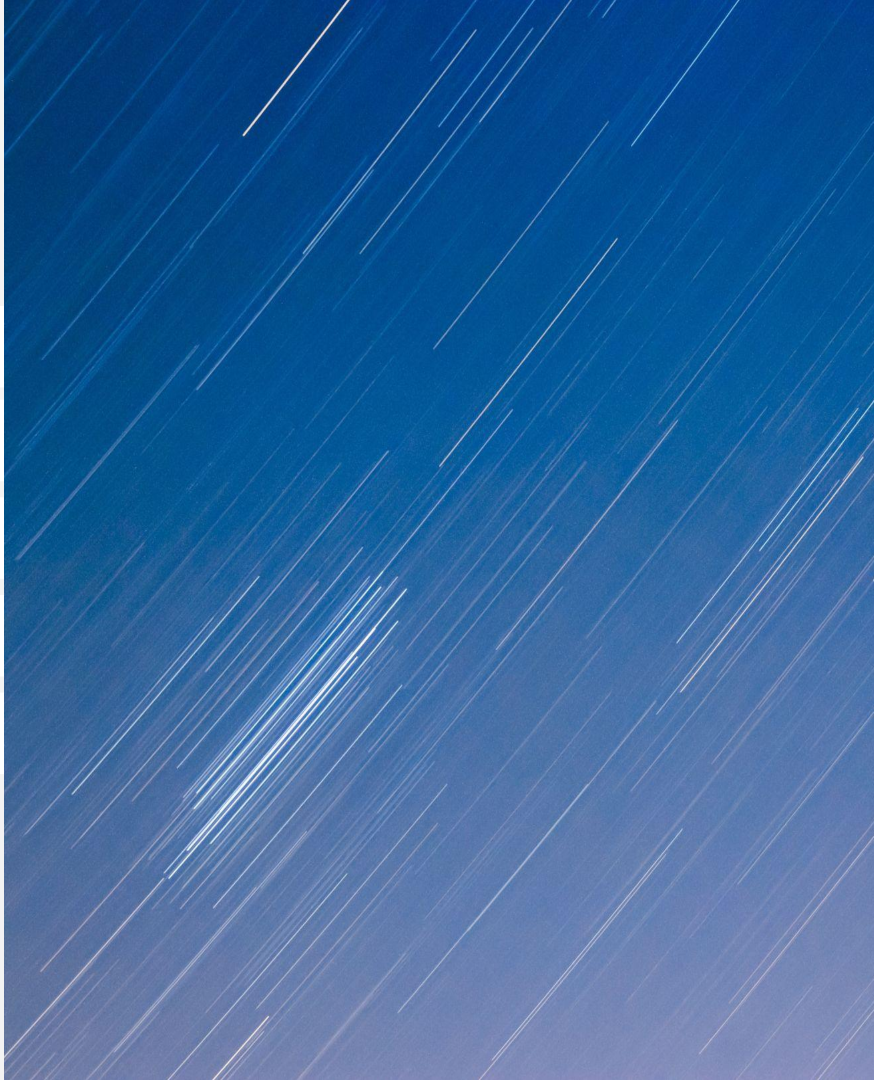
1 **Why anomaly detection**

2 Anomaly detection with Darts

3 Real-world example: Arrhythmia detection

4 Summary / concluding remarks

5 Time for the exercises



Why anomaly detection?

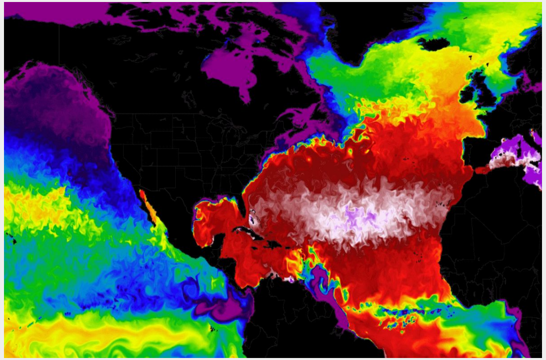
Finance

Unusual fluctuations in financial data, transactions or market behavior can signal potential fraud, revenue leakage

Cybersecurity

Anomalies in network traffic can signify potential cybersecurity breaches or attacks

Environment / weather



Natural anomalies impacting agriculture, food production & safety.

Why anomaly detection?

Production / operation

Manufacturing errors of products, detecting inefficiencies or bottlenecks in processes

Diagnostic

Professionals to make informed decisions for improved patient outcomes, predictions, and patterns identification

Predictive Maintenance

Detection of anomalous device behavior before serious damage occurs

Agenda

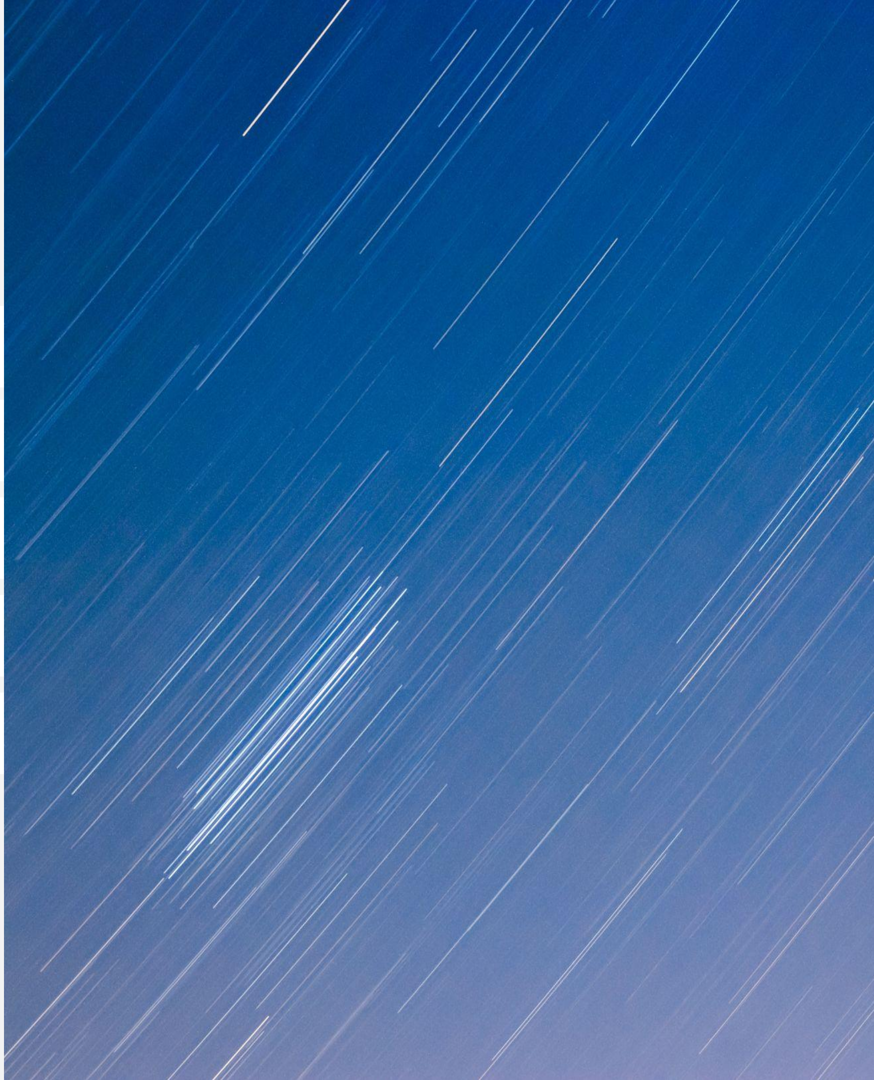
1 Why anomaly detection

2 Anomaly detection with Darts

3 Real-world example: Arrhythmia detection

4 Summary / concluding remarks

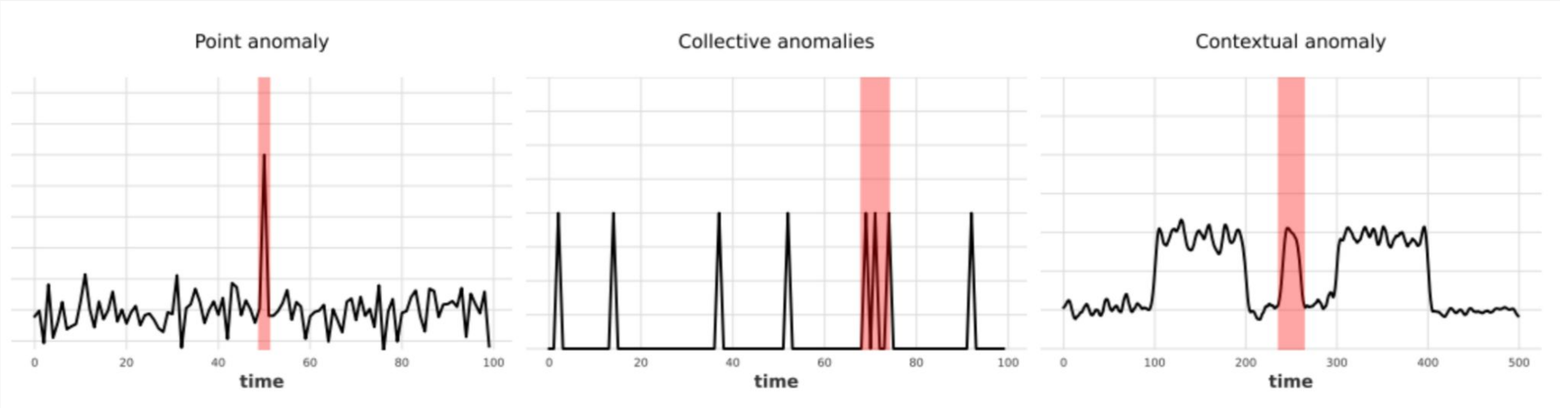
5 Time for the exercises



Anomaly detection

Anomalies can be of different lengths and types, such as an irregular shift in:

- Amplitude
- Frequency
- Mean
- Trend
- Variance or a change in the pattern.



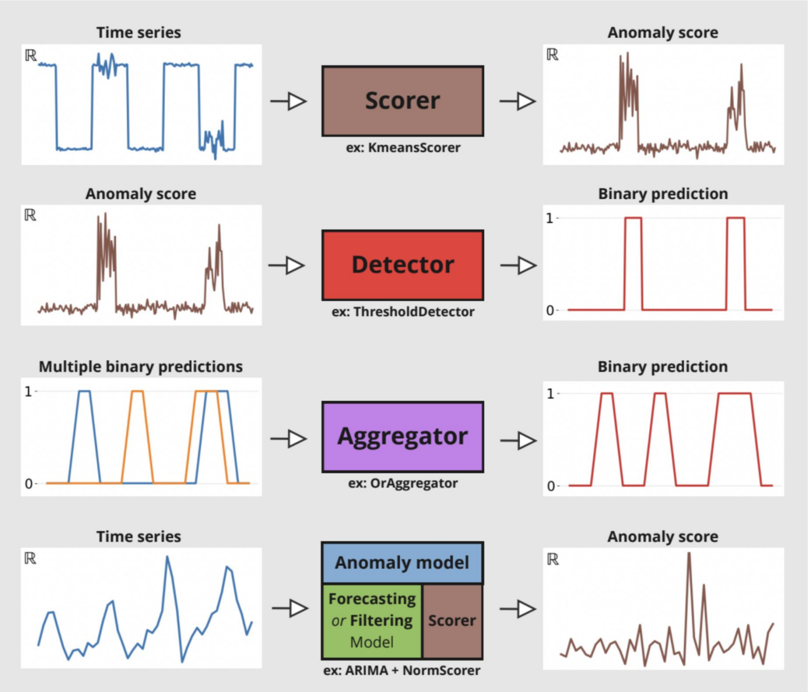
Anomaly detection

	Supervised	Semi-supervised	Un-supervised
Training data	Both non-anomalous and anomalous values with labels indicating where are the anomalies	Only non-anomalous samples	Both non-anomalous and anomalous values without labels
Model / detection	Learns to distinguish the two categories, and the location of the anomalies	Learn to characterize non-anomalous behavior and identify anomalies which are far from expected behavior	Assumes that anomalous subsequence are rare and thus can be separated from the non-anomalous one
Effort	Labeling often done manually - can be prohibitively expensive.	Minimal effort ensuring anomaly free training dataset	-
Remarks	Algorithms are restricted in their ability to detect unseen anomalies	Requiring little to no effort, these two learning types are the most common for anomaly detection models.	

Darts Anomaly Detection Module

Darts anomaly detection module composed of 4 entities:

- Scorers:
  - 12 own implemented + >30 PyOD through a dedicated wrapper
- Detectors:
  - 1 trainable, 1 non-trainable
- Aggregator:
  - 7 trainable, 2 non-trainable
- Anomaly Model:
  - 32 Forecasting + 3 Filtering model



Agenda

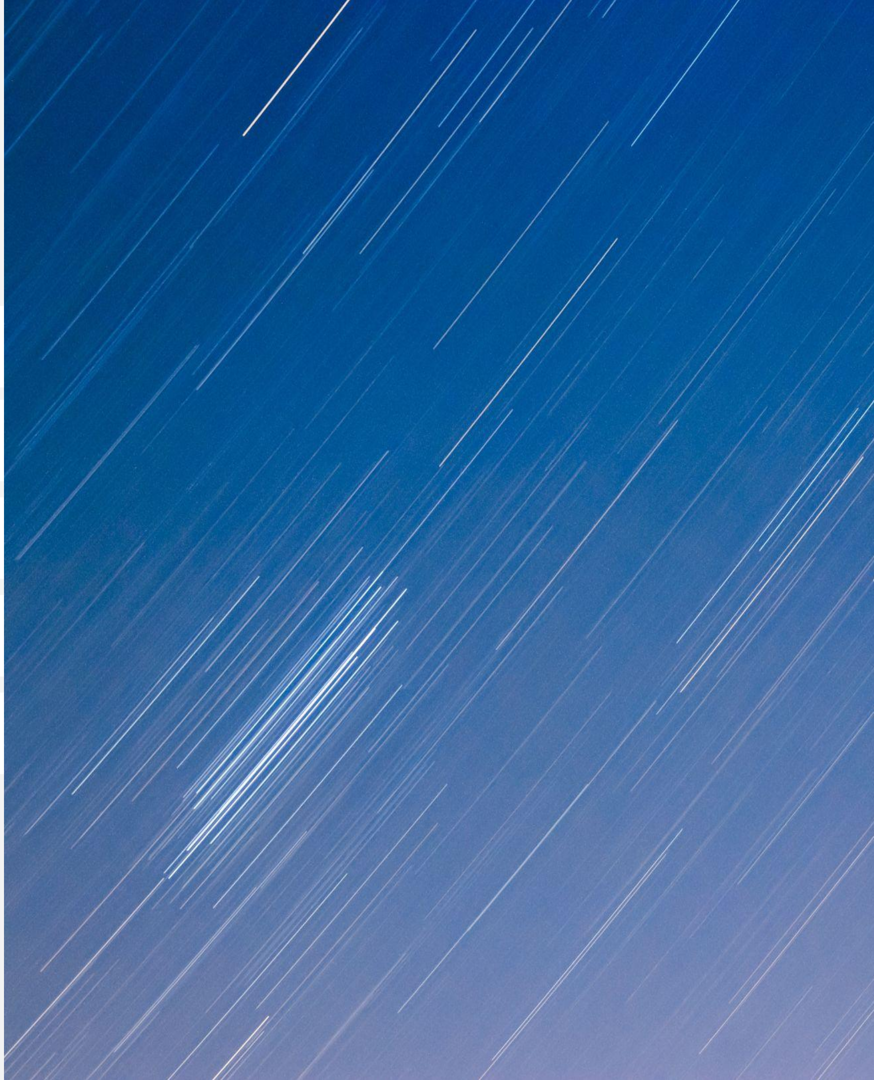
1 Why anomaly detection

2 Anomaly detection with Darts

3 Real-world example: Arrhythmia detection

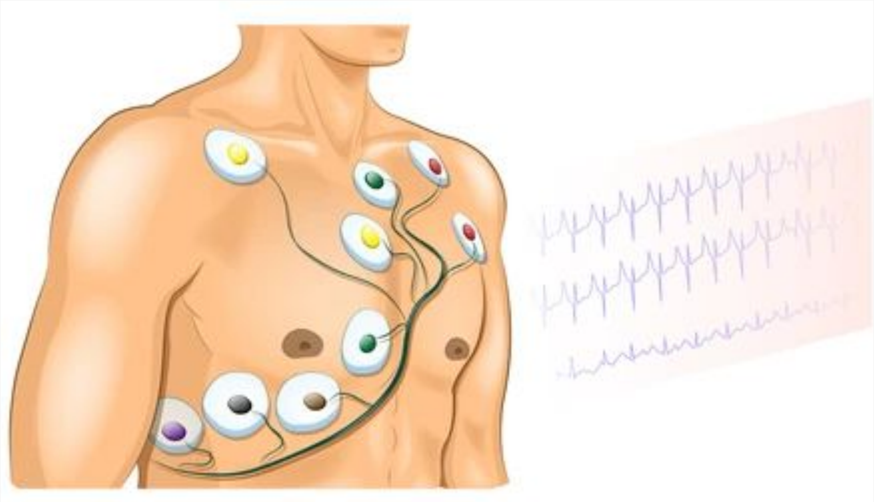
4 Summary / concluding remarks

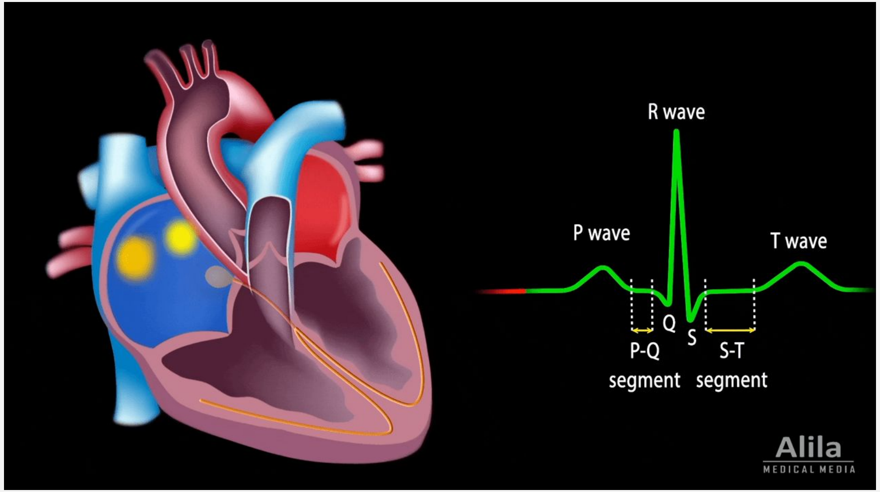
5 Time for the exercises



ECG - arrhythmia

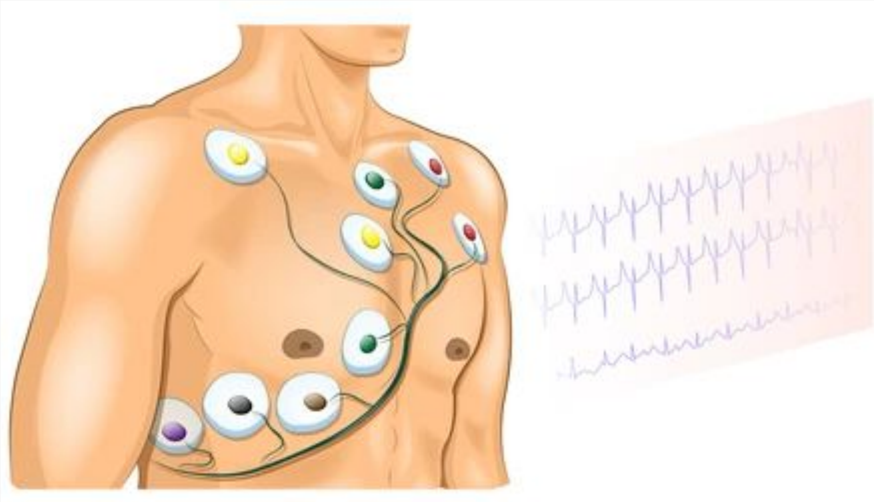
ECG - Electrocardiogram is the simplest and fastest tests used to evaluate the heart.

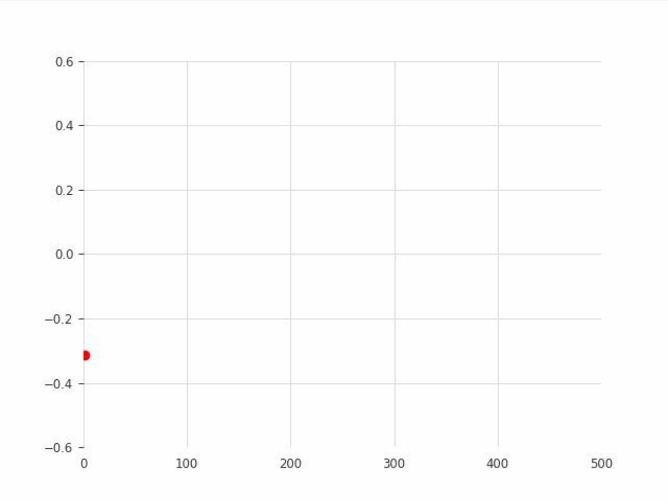




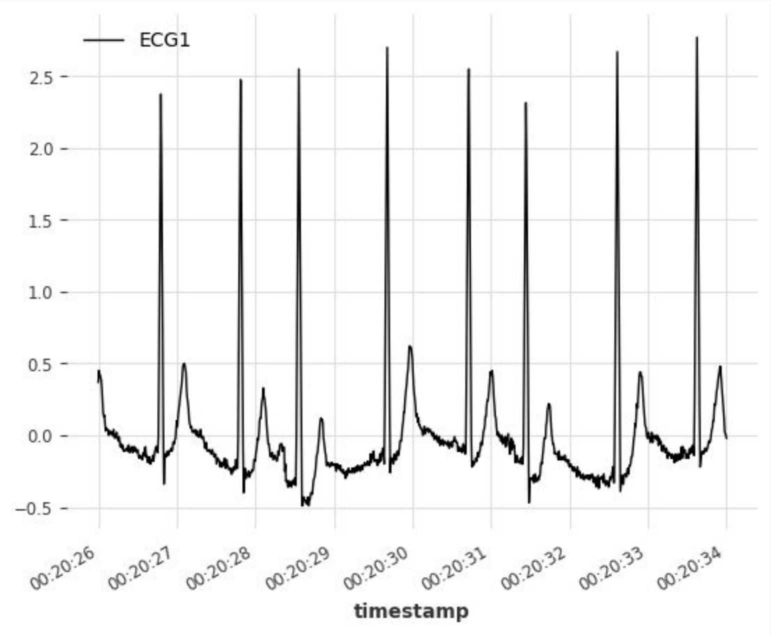
ECG - arrhythmia

Arrhythmia - irregular heartbeat (rate or rhythm)

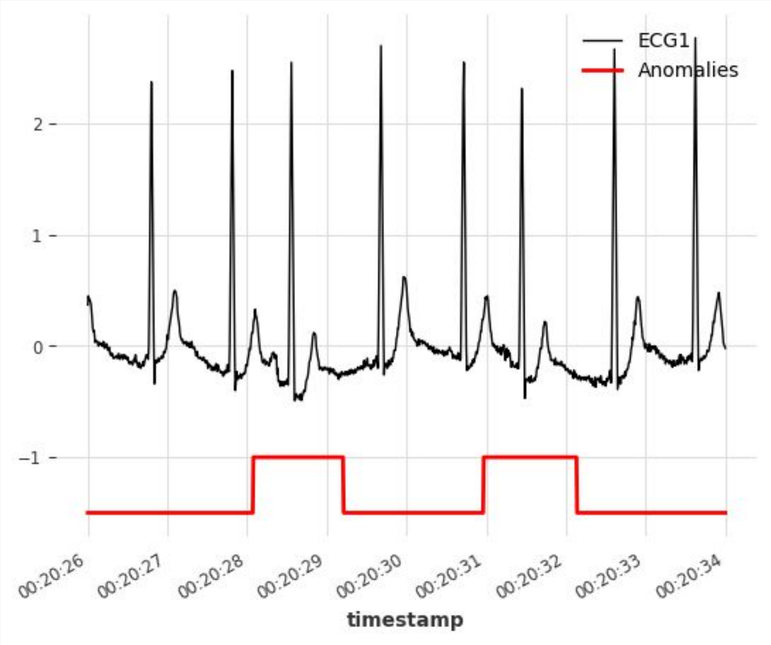




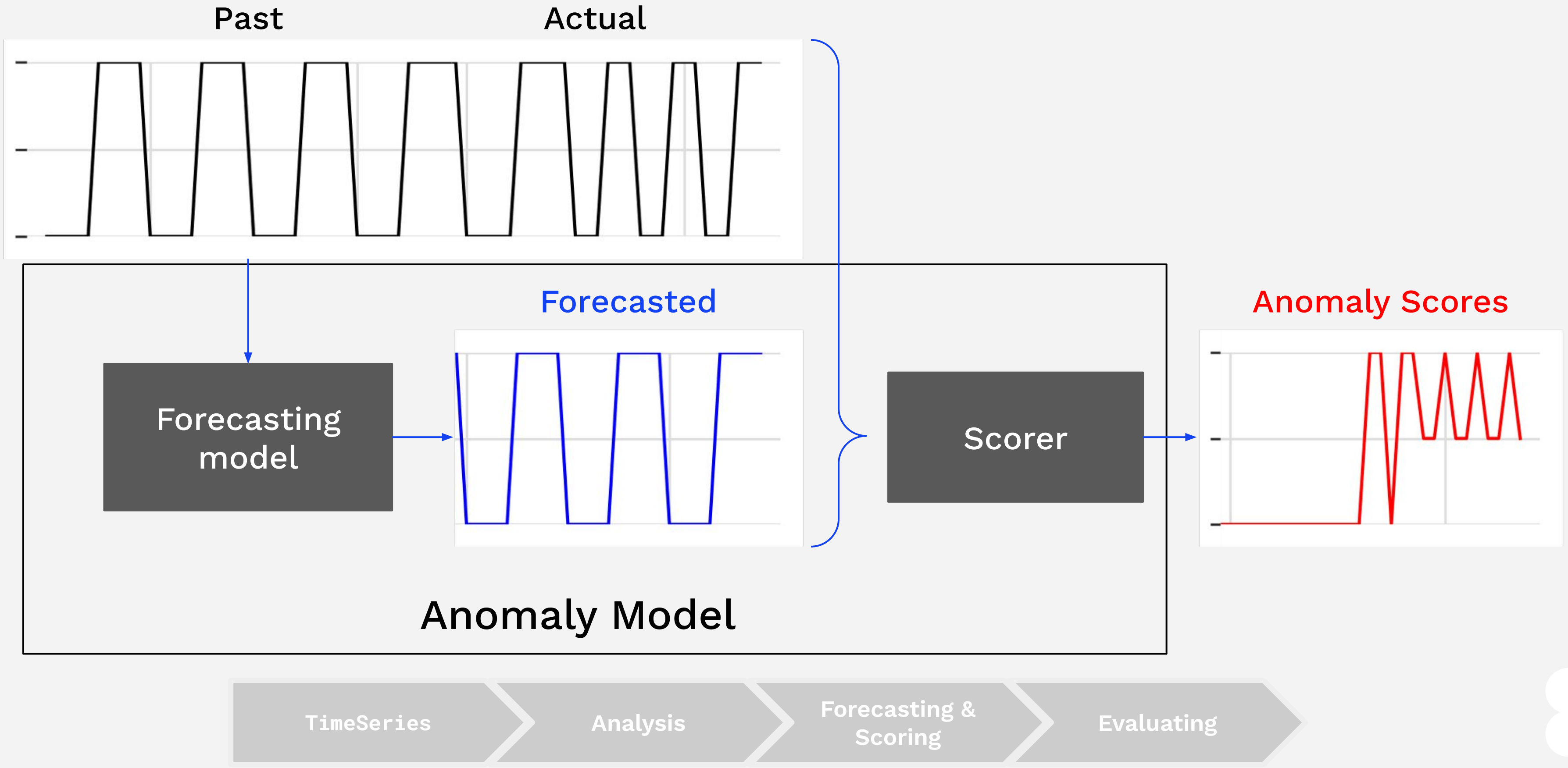
Goal



Anomaly detection →

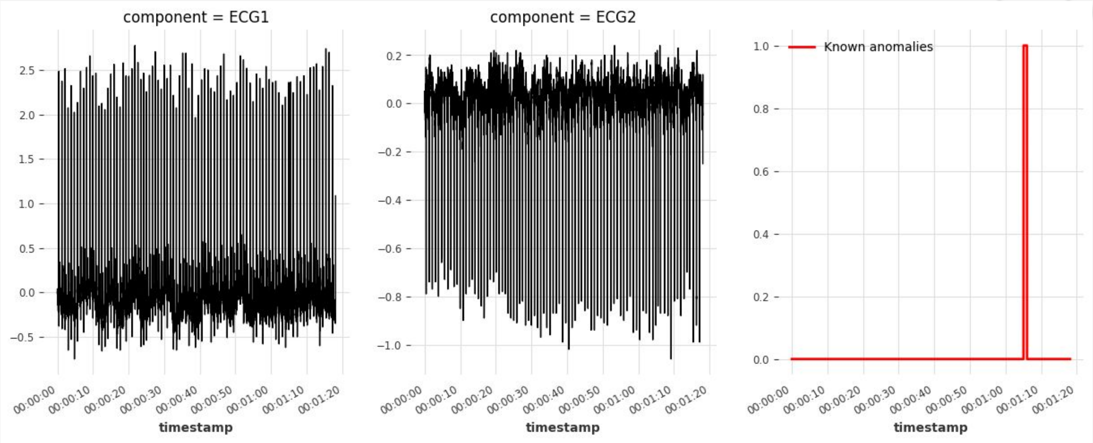


Approach


Past
Actual
Forecasted
Anomaly Scores
Forecasting model
Scorer
Anomaly Model
TimeSeries
Analysis
Forecasting & Scoring
Evaluating

The TimeSeries object

```python
from darts import TimeSeries

# Load data into darts TimeSeries object
timeseries = TimeSeries.from_csv("./multivariate/SVDB/827.test.csv", time_col='timestamp')
ts_ecg = timeseries[['ECG1','ECG2']]
ts_anomaly = timeseries['is_anomaly']
```

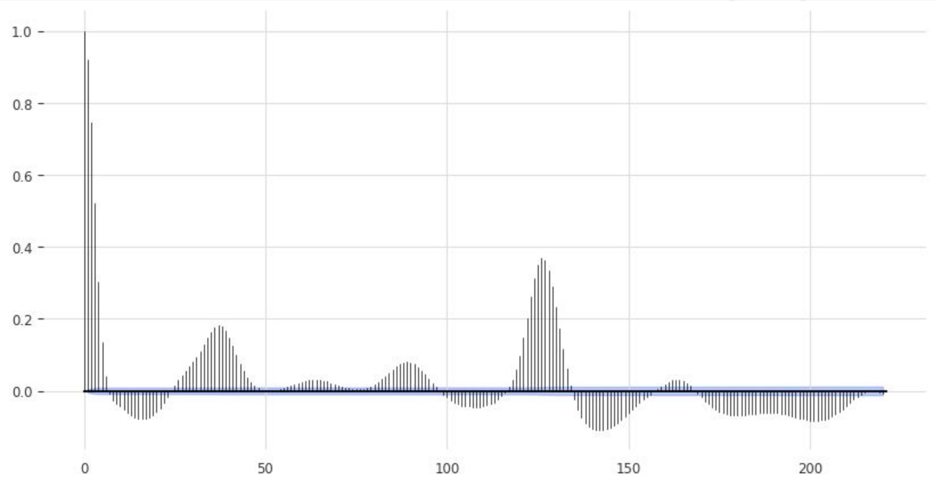


The MIT-BIH Supraventricular Arrhythmia Database (SVDB) contains 2 channels, and 78 half-hour ECG recordings obtained from 47 objects between 1975-1979.

TimeSeries → Analysis → Forecasting & Scoring → Evaluating

TimeSeries analysis

```python
from darts.utils.statistics import plot_acf

# Visualise signal auto correlation to identify most common periodicity
plot_acf(ts=ts_ecg['ECG1'], max_lag=220)
plt.show()
```



Most common periodicity:

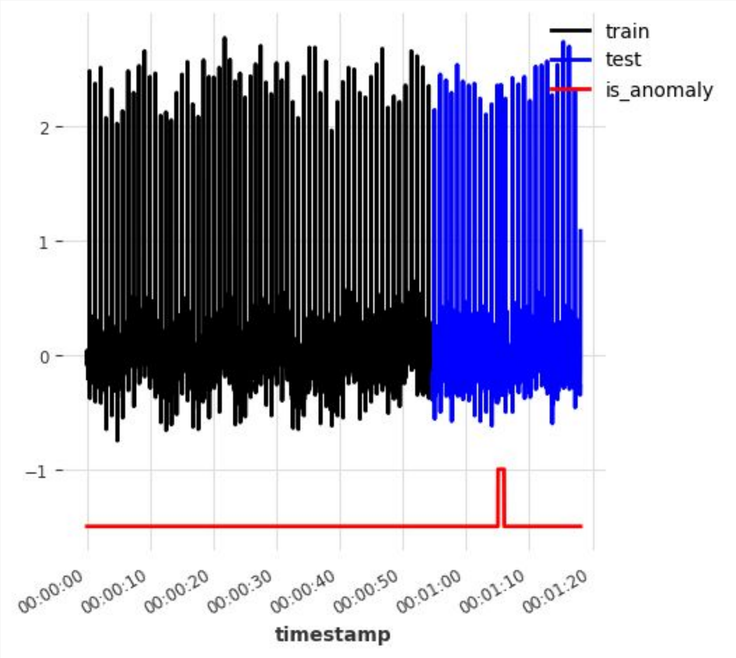
- ~130 measured point
- Pulse of ~98

TimeSeries → **Analysis** → Forecasting & Scoring → Evaluating

Training/Validation split

```python
# Create train and test dataset for demonstration
ts_ecg_train = ts_ecg[:7000]
ts_ecg_test = ts_ecg[7000:10000]
ts_anomaly_test = ts_anomaly[7000:10000]
```

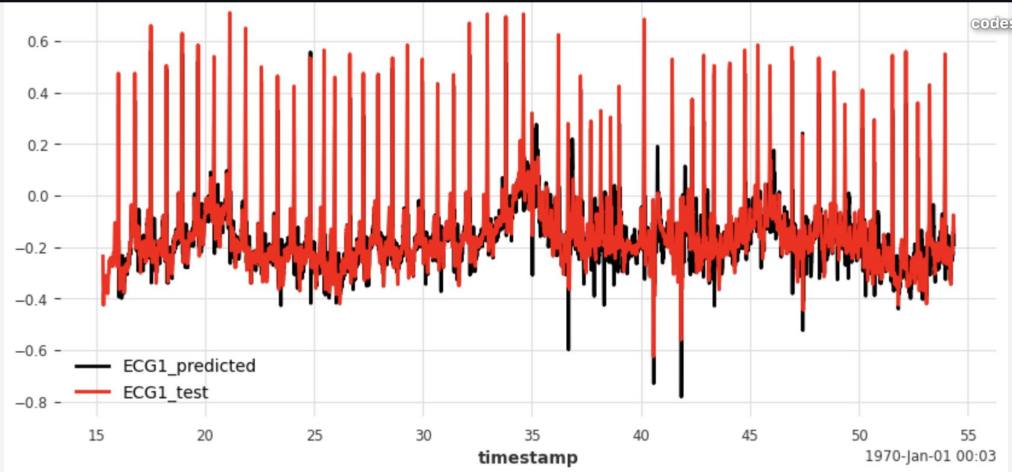
codesnap.dev



Forecasting & Scoring

```python
from darts.models import LinearRegressionModel

# Instatiate of a forecasting model - e.g. RegressionModel with a defined lag
forecasting_model = LinearRegressionModel(lags=period)

# Train the forecasting model on the training dataset
forecasting_model.fit(ts_ecg_train)

# Historical predictions
ts_ecg_test_predicted = forecasting_model.historical_forecasts(ts_ecg_test, retrain=False)
```

codesnap.dev

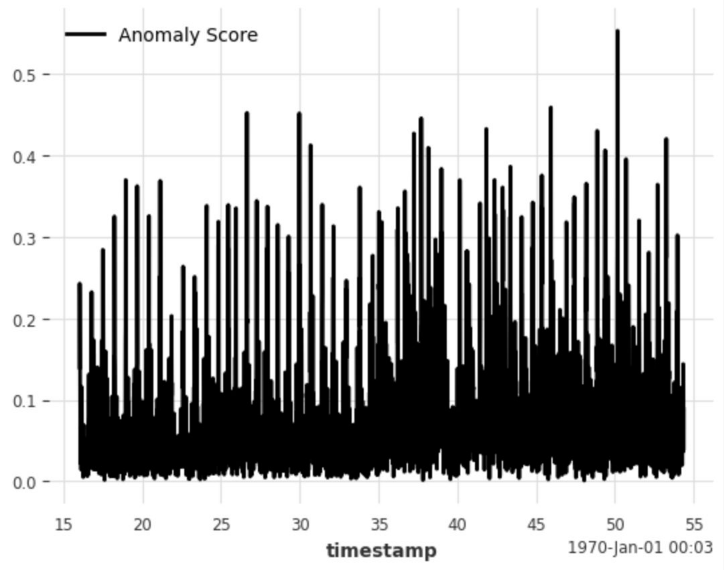


TimeSeries | Analysis | Forecasting & Scoring | Evaluating

Forecasting & Scoring

```
from darts.ad.scorers import NormScorer
scorer = NormScorer(ord=1, component_wise=False)
scores =  scorer.score_from_prediction(
                series=ts_ecg_test,
                pred_series=ts_ecg_test_predicted
          )
scores.plot(label='Anomaly Score')
```

codesnap.dev

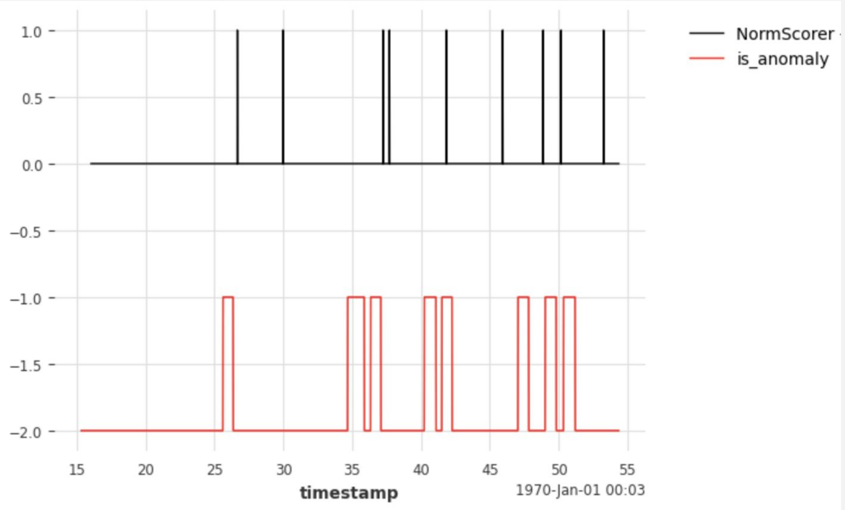


TimeSeries | Analysis | Forecasting & Scoring | Evaluating

Evaluate Anomaly Model

```python
# Evaluate the calculated anomaly score using utility methods in darts
from darts.ad.utils import eval_metric_from_scores
eval_metric_from_scores(
    pred_scores=scores,
    anomalies=ts_anomaly_test,
    window=1,
    metric='AUC_ROC'
)
```

AUC-ROC: ~0.63



TimeSeries → Analysis → Forecasting & Scoring → **Evaluating**

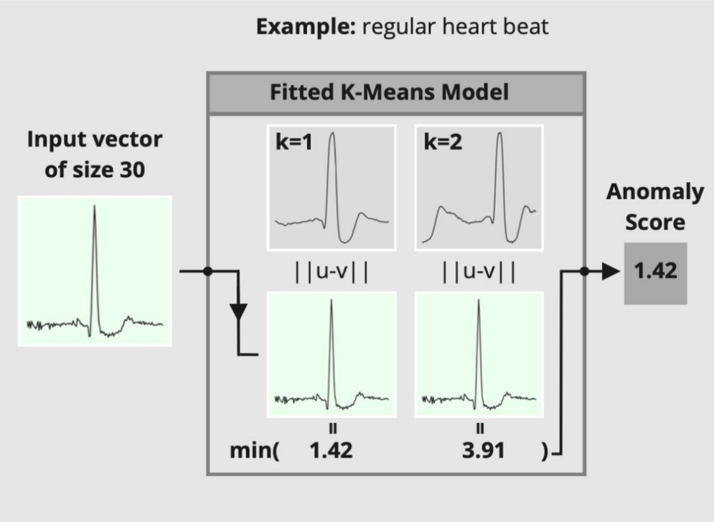
Anomaly Detection model

```
from darts.models import LinearRegressionModel
from darts.ad.scorers import NormScorer, KMeansScorer
from darts.ad.anomaly_model.forecasting_am import ForecastingAnomalyModel

# Forecasting model
forecasting_model = LinearRegressionModel(lags=period)

# Anomaly model with: one forecasting model, and one or more scorers
anomaly_model = ForecastingAnomalyModel(
    model=forecasting_model,
    scorer=[
        NormScorer(ord=1),
        KMeansScorer(k=50, window=2*period, component_wise=False)
    ],
)
```

codesnap.dev

TimeSeries | Analysis | **Forecasting & Scoring** | Evaluating

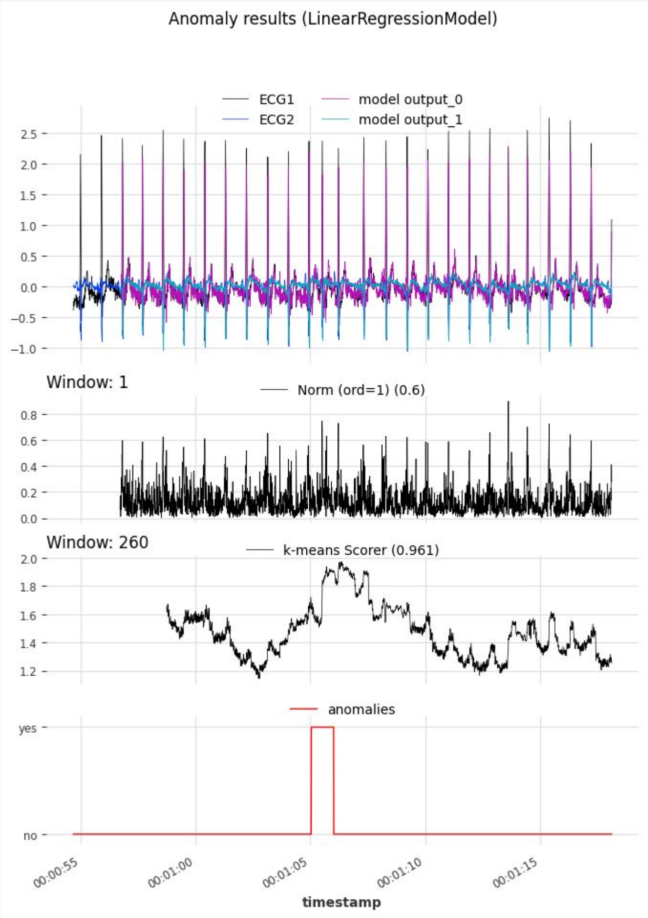
KMeanScorer



TimeSeries
Analysis
Forecasting & Scoring
Evaluating

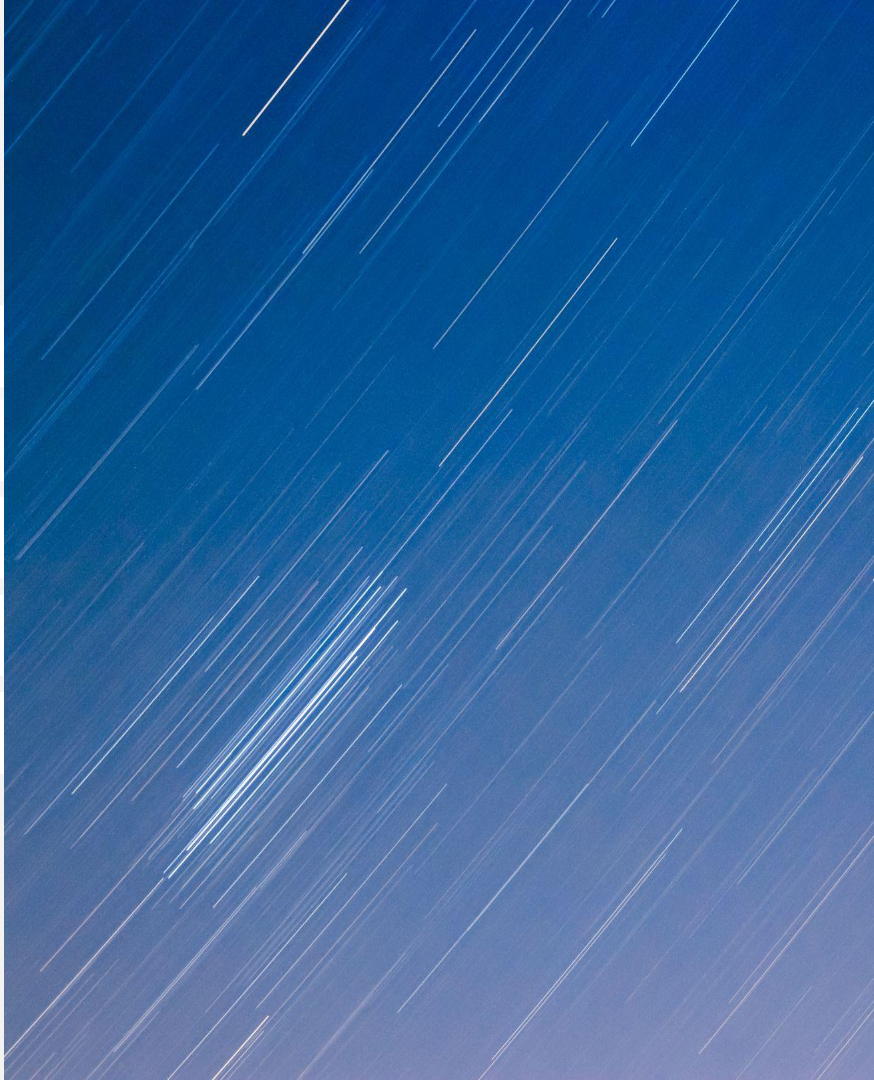
Evaluate Anomaly Model

```python
# Visualize and evaluate detection of anomalies
anomaly_model.show_anomalies(
    series=ts_ecg_test,
    anomalies=ts_anomaly_test,
    metric="AUC_ROC",
)
```



TimeSeries → Analysis → Forecasting & Scoring → **Evaluating**

Agenda

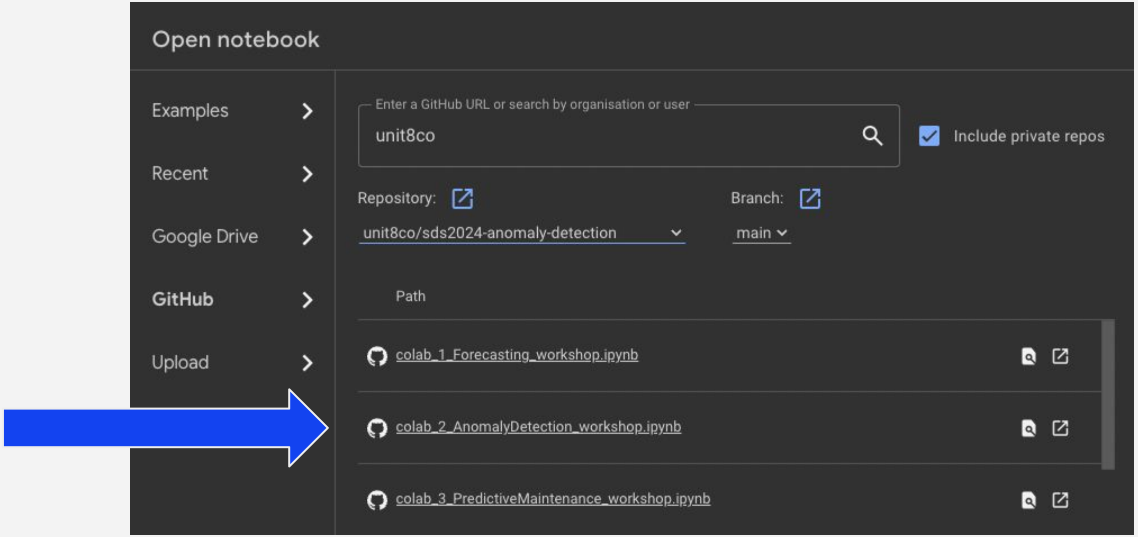


Concluding remarks

- Darts is a tool that simplifies time series manipulation, forecasting and anomaly detection
- In just a few lines of code, we can use and compare different models using in-built visualization
- The unified API allows to use neural network based models in the same way as simpler models

Time To Work

- Open: https://colab.google/
- Click on “Open Colab”
- On the left sidebar click GitHub
    - Enter GitHub URL: “unit8co”
    - Select repository: “unit8co/sds2024-anomaly-detection”
    - Select notebook: “colab_2_AnomalyDetection_workshop.ipynb”



unit8.co

thank you