Time Series Anomaly Detection : Tools, Techniques & Tricks

Presenter : Dr. Dhaval Patel
pateldha@us.ibm.com

Outline

- ❑Introduction to Time Series Data
- ❑Toolkits for Time Series Anomaly Detection
- ❑Anomaly Detection Use case
- ❑API for Data Scientist
- ❑Deployment : Web based Anomaly Detection System

Topic I:
Time Series Data: A Brief Introduction

Data comes from Everywhere

Grocery Markets

E-Commerce

Stock Exchange

Hospital

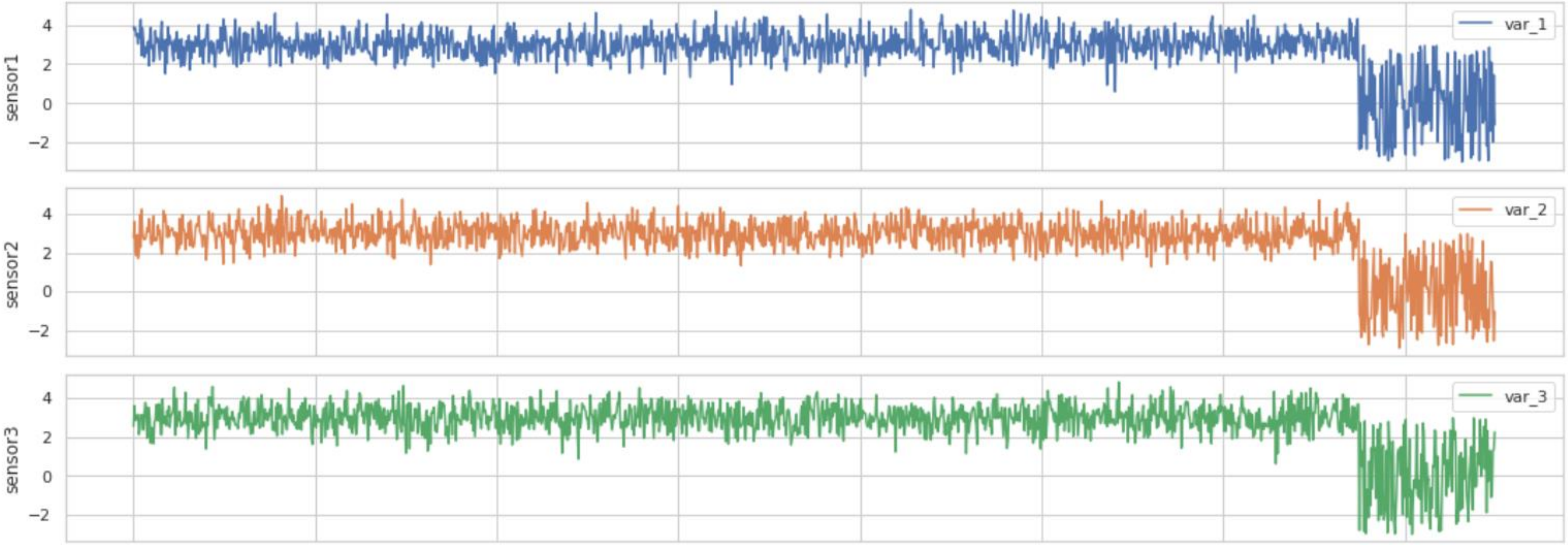
Weather Station

Social Media

Time Series Data

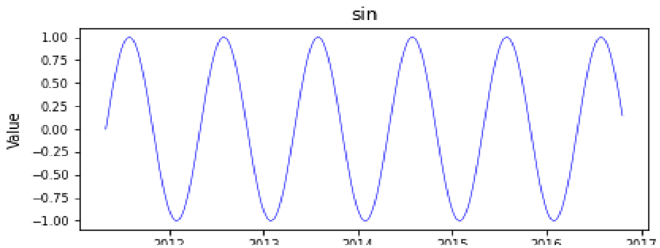
- Time series is a *sequence of data points*, measured typically at successive times, spaced at (often uniform) time intervals

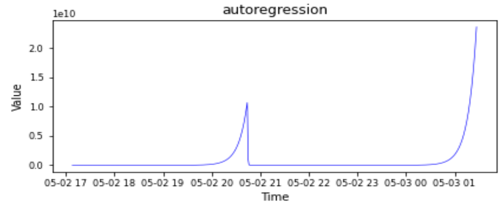
- Three classes of time series data are popular
  - Univariate time series
  - Multi-variate time series
  - Multi time series (asset centric)

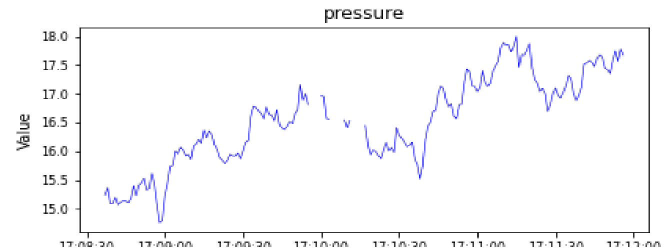


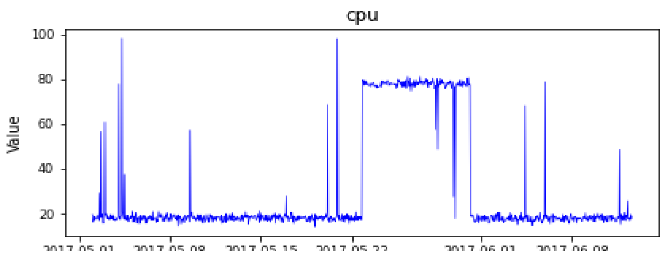
Multi-variate time series

Time Series Data

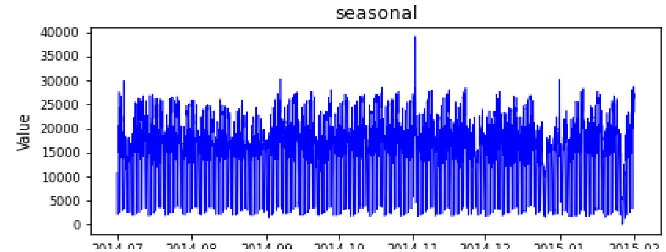


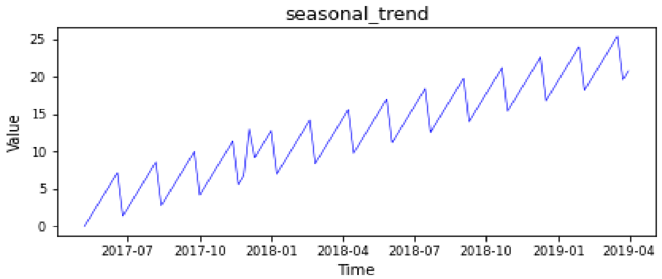


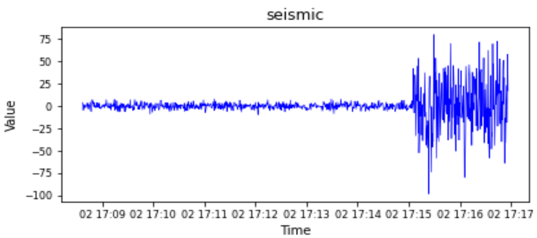


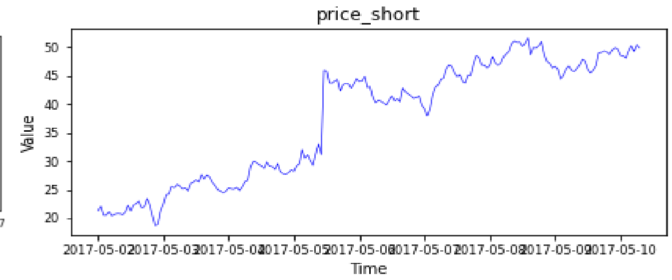


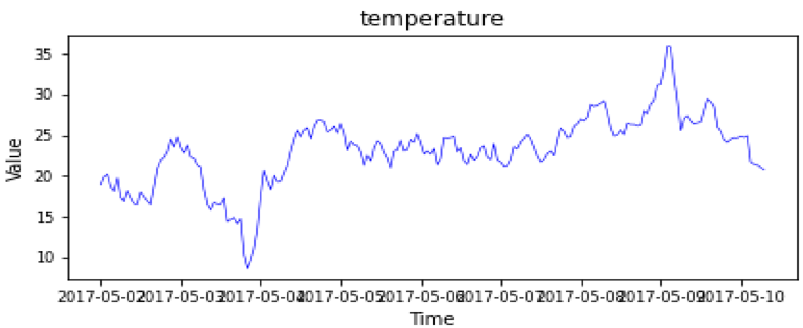
Univariate Time Series

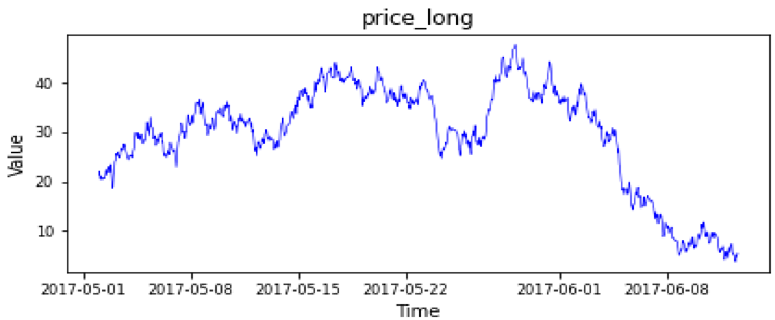




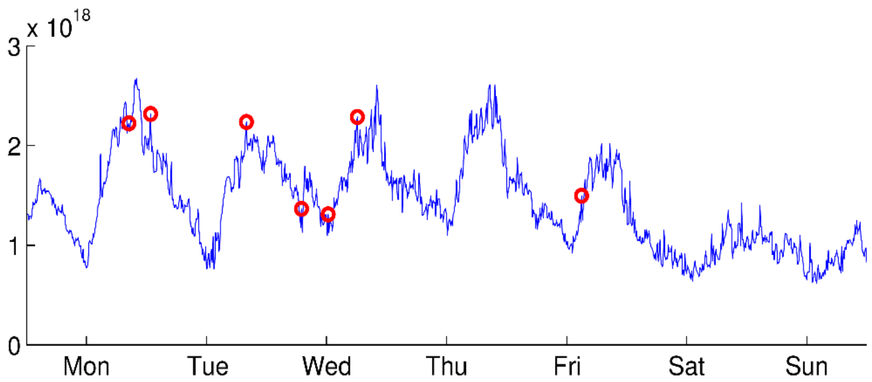




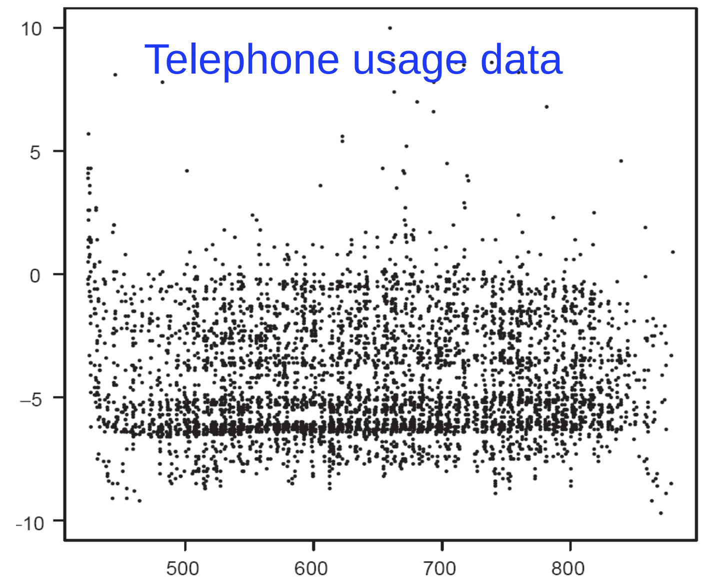




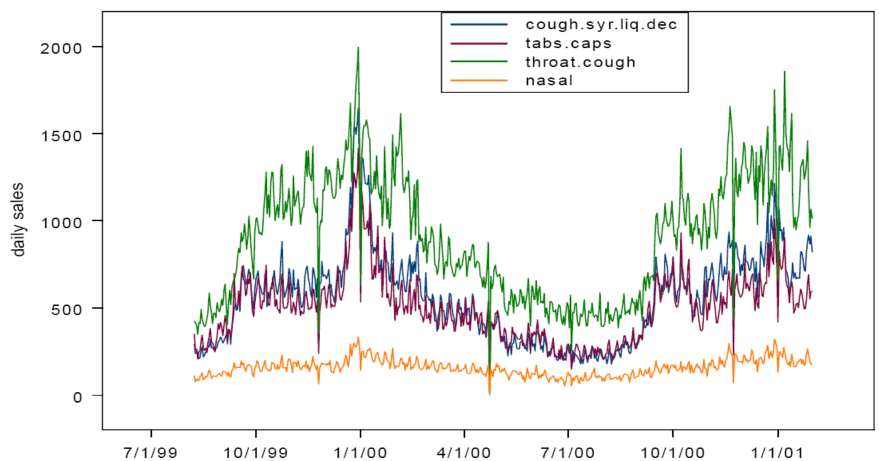
Time Series Data



Network traffic data



Telephone usage data



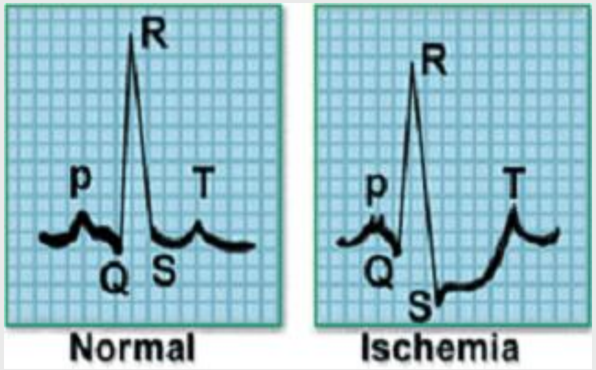
Inhalational disease related data

Matrix code

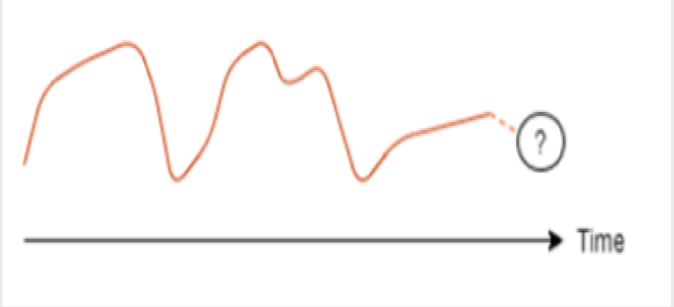
Common Time Series Analytic Tasks (ACK Prof. Eamonn)

Clustering

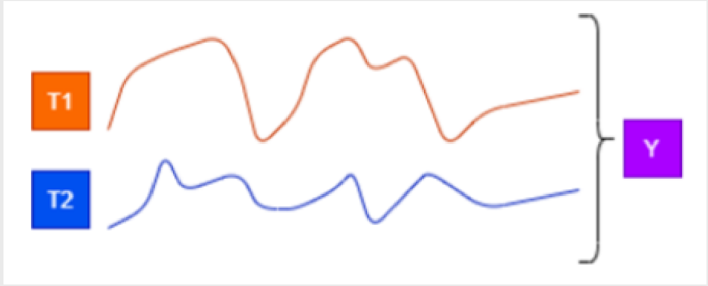
Classification



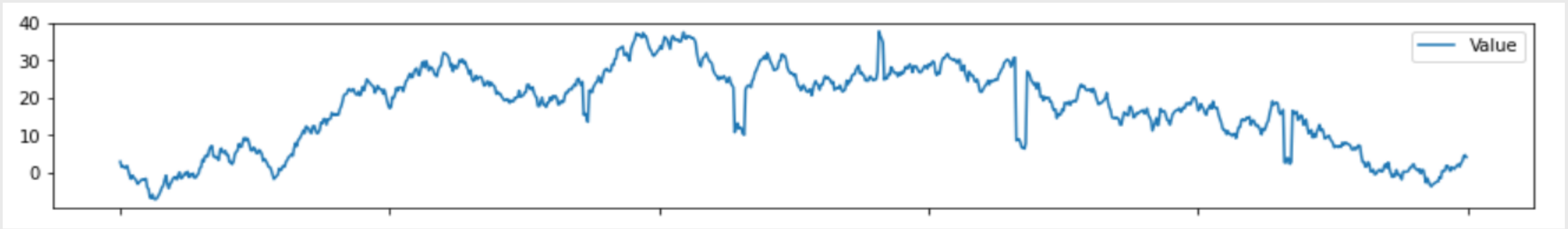
Forecasting



Regression



Anomaly Detection



Topic II:
Anomaly Detection in Time Series Data

Anomaly Detection in Time Series Data

- Anomalies in time series data are data points that ***significantly deviate*** from the ***normal pattern*** of the data sequence

- Variants of Anomaly/Outlier Detection Problems
  - Given a database D, find all the data points $\mathbf{x} \in$ D with anomaly scores greater than some threshold *t*
  - Given a database D, find all the data points $\mathbf{x} \in$ D having the top-n largest anomaly scores $f(\mathbf{x})$
  - Given a database D, containing mostly normal (but unlabeled) data points, and a test point **x**, compute the anomaly score of **x** with respect to D

- Applications
  - IoT Asset Monitoring
  - Failure/Fault detection
  - Fraud detection (credit card, telephone)
  - Spam detection
  - Biosurveillance
    - detecting geographic hotspots
  - Computer intrusion detection
    - detecting masqueraders

Anomaly Detection Challenges

IBM

Conceptual Solution

- **Step 1.** Learn a model of normal behavior
  - Using supervised or unsupervised method

- **Step 2.** Based on this model, construct a suspicion/anomaly score
  - function of observed data
  - captures the deviation of observed data from normal model
  - raise flag if the score exceeds a threshold

Challenges

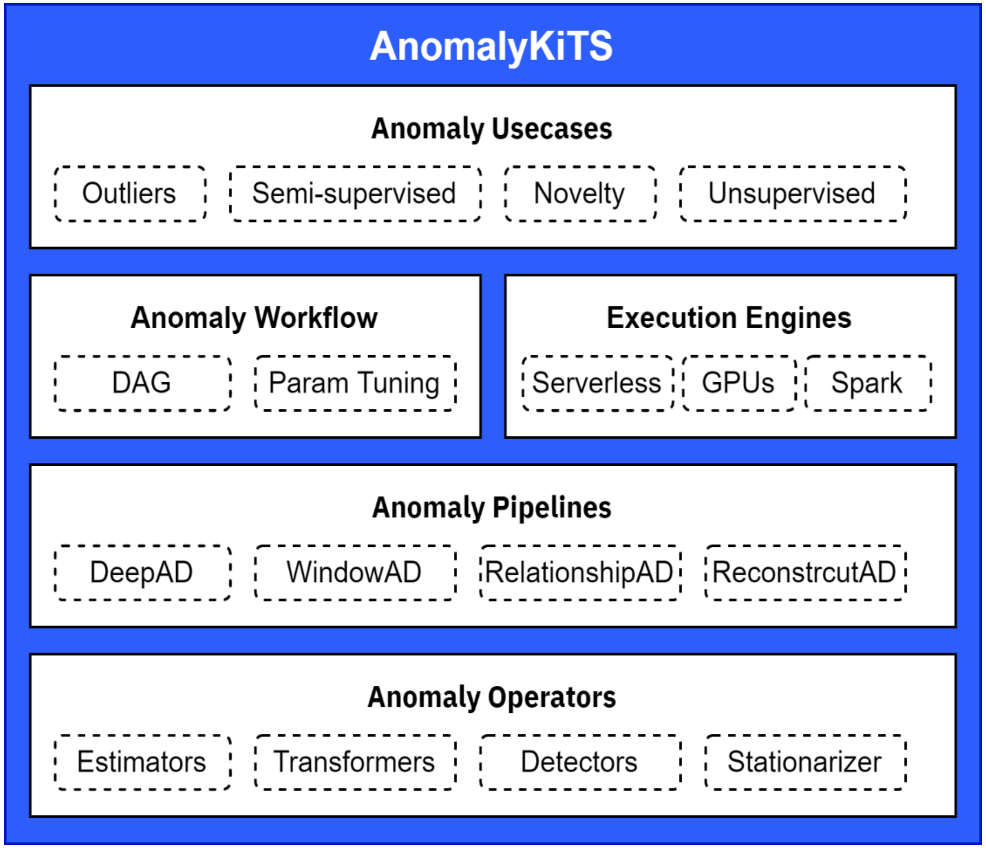
- Multiple way of building anomaly models
- User preferred unsupervised methods, avoiding making decision on what to do
  - Validation can be quite challenging (just like for clustering)
  - Parameter tuning & Comparing multiple models and obtaining a rank
- How many outliers/anomaly are there in the data?
- Anomaly Score @
  - Dataset level, Feature Level, Record Level
- Anomaly Label generation
  - Anomaly score are real value and its interpretation is difficult

AnomalyKits :

Sklearn compliant standardized architecture, components and output schema for various AI Applications

Key differentiators

- Novel Algorithms as Estimators, Transformers, etc
- Consistent Programming API for coding various reusable AI Application problem with support of many Execution Engine to meet the diverse need
- Efficient DAG optimization method that achieve state of the art solution for various Data Science Task
- Benchmarked on public and client dataset to demonstrate core-capabilities



AAAI-2022 Conference as a Demo Paper

AnomalyKiTS: Anomaly Detection Toolkit for Time Series

Input Data Format for Anomaly Detection

- Numpy array : with time column and time series feature columns

Time	Value
0	3.000000
1	1.572558
2	1.873181
3	1.361140
4	1.408475
5	1.908858
6	0.471416
7	-0.755087
8	-1.636673
9	-0.663525

Univariate Time Series

Time	Value_0	Value_1	Value_2	Value_3
0	3.000000	0.000000	-3.000000	-2.000000
1	1.572558	0.270133	-3.320124	-1.583921
2	1.873181	0.048440	-3.154067	-1.974031
3	1.361140	-0.211421	-3.292858	-2.414144
4	1.408475	-0.559694	-3.080145	-2.242305
5	1.908858	-0.536122	-3.238631	-1.898070
6	0.471416	-0.513129	-3.185812	-1.493719
7	-0.755087	-0.605181	-3.362431	-1.373513
8	-1.636673	-0.282153	-3.708064	-1.740503
9	-0.663525	0.045805	-3.427077	-1.383114

Multi-variate Time Series

Anomaly Operators

30+

Anomaly Detection algorithms with pre-defined hyper parameter grid

Statistical

- PCA Q*
- PCA T2*
- Hotelling T2*
- Robust PCA*
- CUSUM
- Spectral Transform*
- Cost Discrepancy*
- Grubbs Test
- KS Test

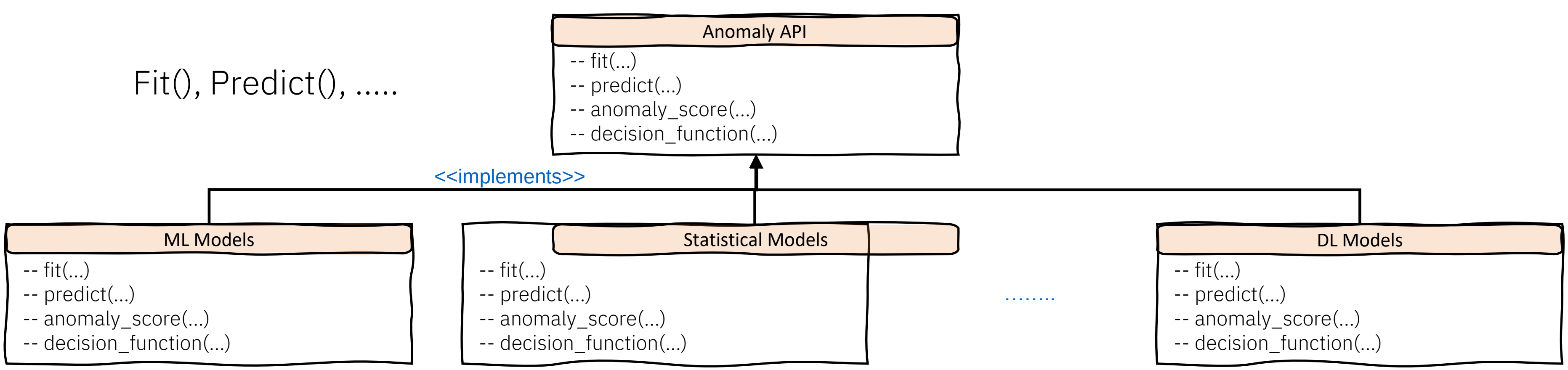
Deep Learning

- Encoder-Decoder
- Deep Negative Sampler*
- Neural Machine Translation*
- Transudative Transformer*

Machine Learning

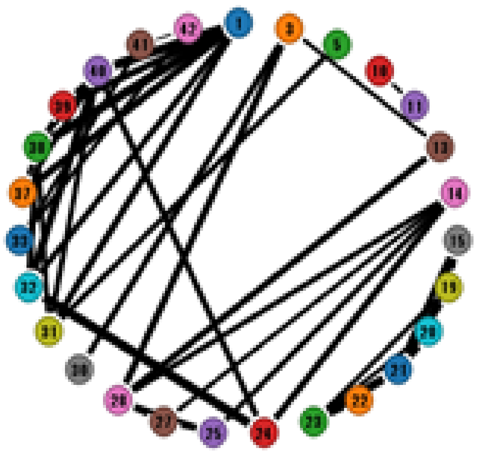
- Gaussian Graphical Models*
- Robust Gaussian Graphical Models*
- Generalized Anomaly Model*
- Ensemble Anomaly*
- Histogram Anomaly*
- Extended Histogram Anomaly*
- Negative Sampler*
- Out-of-Box Anomaly*
- Random Partition Forest*
- Gaussian Mixture*
- Bayesian Gaussian Mixture*
- Extended Isolation Forest*
- Local Outlier factor Nearest Neighbor*
- Nearest Neighbor*
- Isolation Forest
- One Class SVM
- Covariance Estimators

Fit(), Predict(), ...



Anomaly Operators : Gaussian Graphical Models

Learn sparse/dense Gaussian Graphical Model and Generate Anomaly Score at Record/Feature/Dataset Level



An Anomaly Model that captures the relationship between variables

Anomaly Models

- GraphLasso L0
- GraphLasso L1
- Empirical Covariance
- Elliptical Covariance
- Ledoit Covariance
- MinCovDet Covariance
- Oas Covariance
- Shrunk Covariance

Score Computation

- Dataset Level
- Record Level
- Feature Level

Distance Function

Temporal Window (Sliding Window > 0)

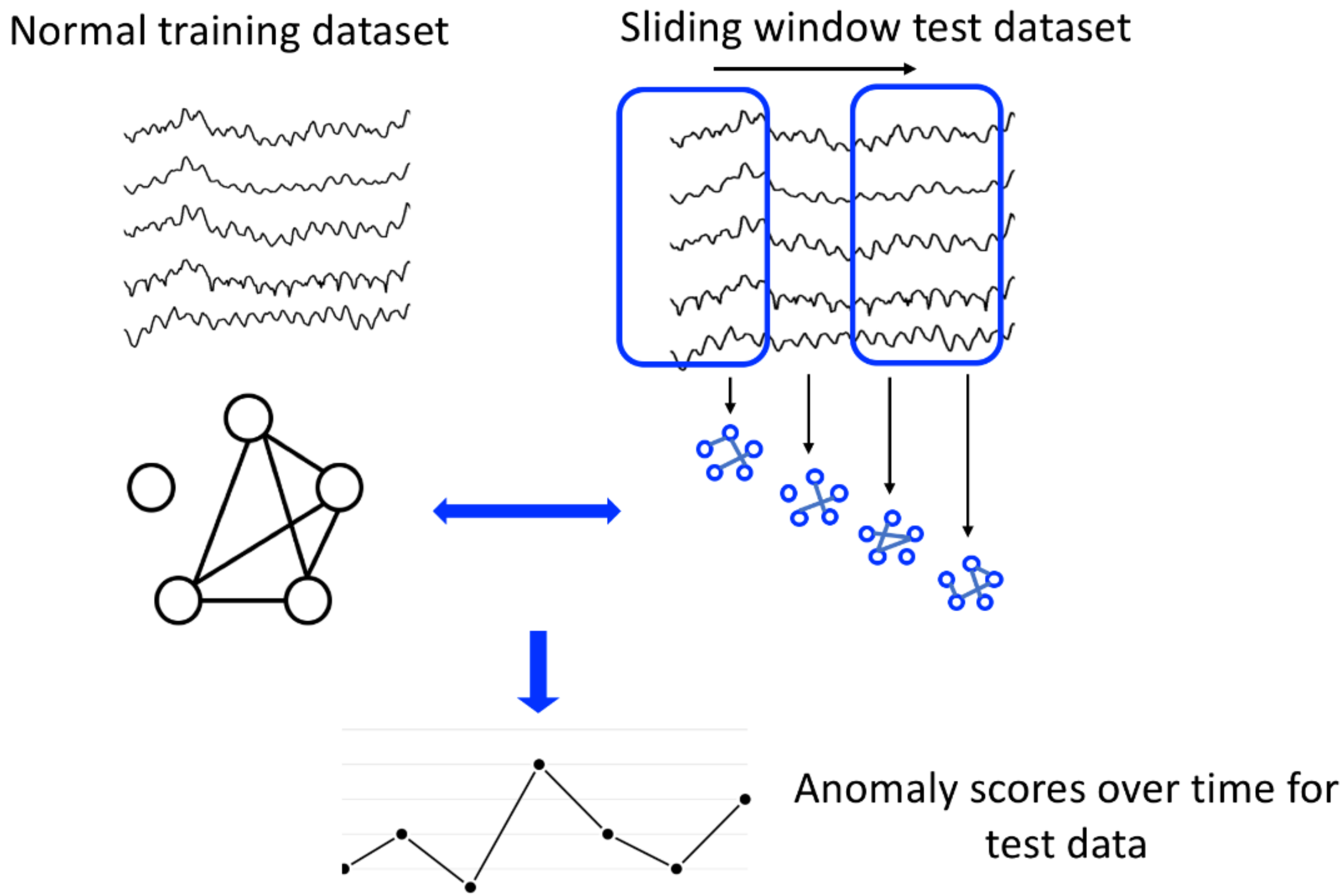
- KL Divergence Distribution
- KL Divergence Features
- Frobenius Norm
- Likelihood
- Spectral
- Mahalanobis Distance
- Sparsest k Subgraph
- Stochastic Nearest Neighbors

Individual Records as IID

Log Likelihood function

Anomaly Operators : Model Training

Learn sparse/dense Gaussian Graphical Model and Generate Anomaly Score at Record/Feature/Dataset Level



Anomaly Pipelines

Implementation of four exemplars anomaly pipelines to cover wide range of anomaly detection approaches

4

Anomaly Pipelines that logically connects anomaly operators
with support of
point and contextual time series anomaly scorings techniques

Reconstruction Based

- ReconstructAD

Prediction Based

- PredAD
- DeepAD

Relationship Based

- RelationAD

IID Windowing Based

- WindowAD

Anomaly Pipelines

Various anomaly thresholding techniques to supports anomaly label generation and alerting

2

Seemless Supports for IID and Time Series Data for generating anomaly label and subsequent Alerts

Dynamic Thresholding (Time Series)

Point Anomaly

- Q-Score
- Chi-Square Test
- Sliding-Window Threshold
- Adaptive Sliding-Window Threshold

Contextual Anomaly

- {Start, End, Severity}

Static Thresholding (IID)

Point Anomaly

- Contamination
- Adaptive Contamination
- Q-function
- Robust Q-function
- Median Absolute Deviation
- OTSU (Parameter Free)

Multiple Pipeline Evaluations for Discovering the best options for a given dataset

2

Categories of Anomaly DAGs
and its associated Hyper-Parameters

Semi-supervised Exploration

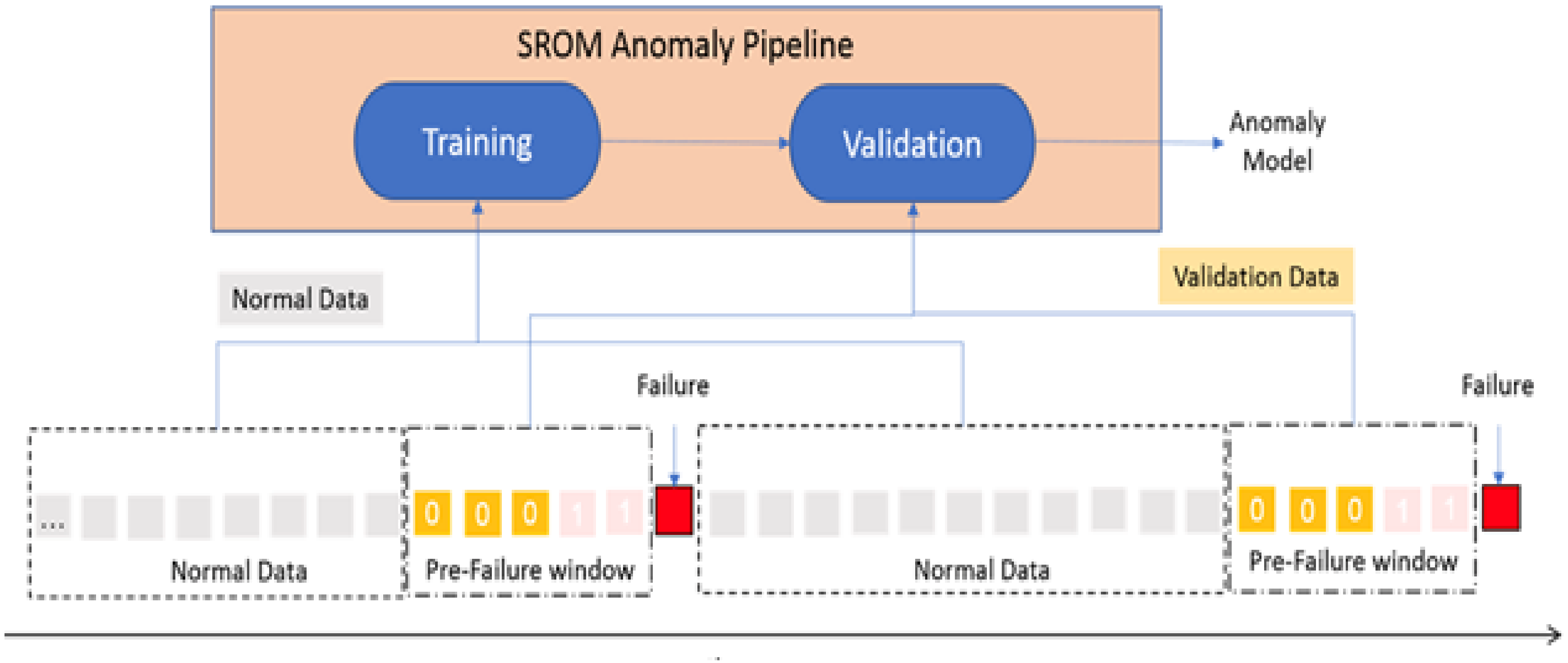
- Requires label for validation data
- Provide model selection using semi-supervised

Unsupervised Exploration

- No Label
- Unsupervised Model Ranking for IID data (Experimental)
  - EM Score
  - AL Score
  - MV Score
- Dynamic Ensembles for Time Series Data

Anomaly Workflow : Semi-supervised Analysis

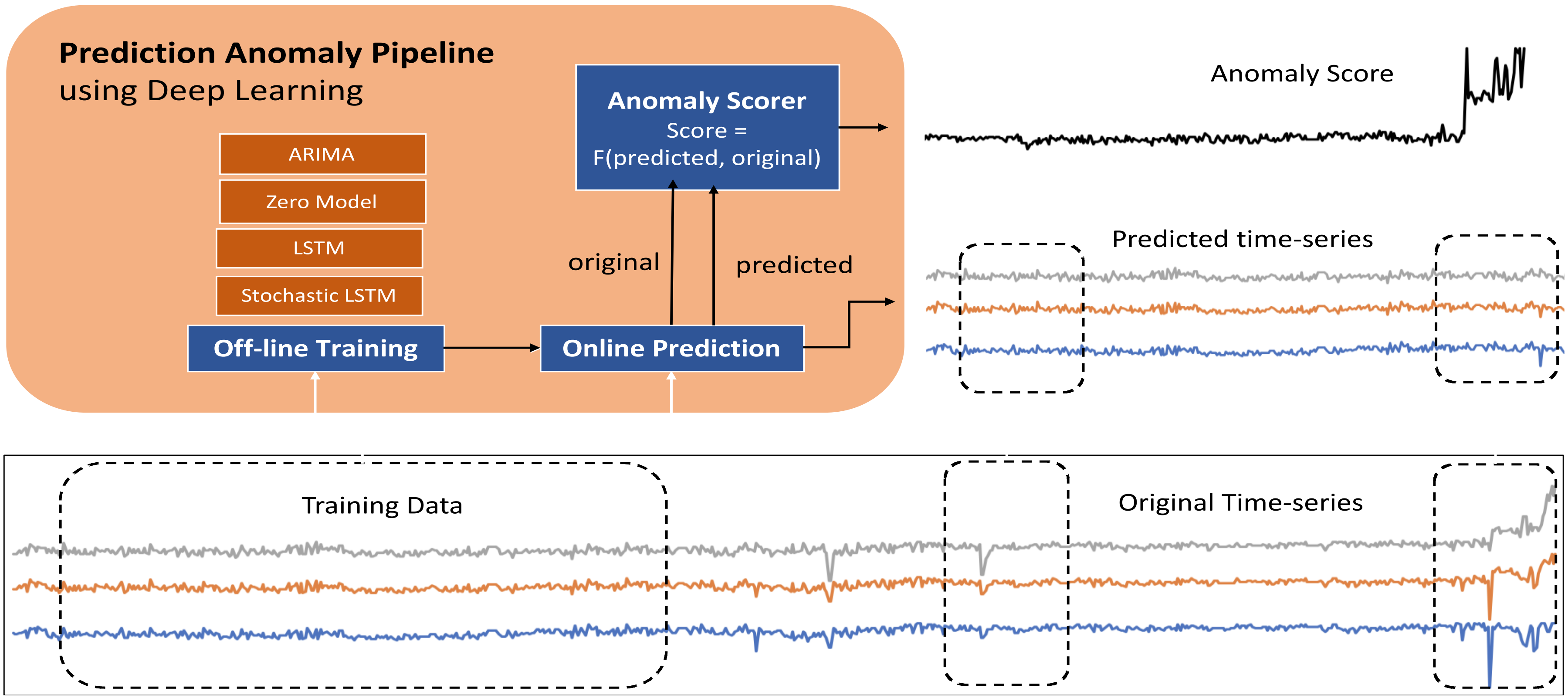
Multiple Pipeline Evaluations for Discovering the best options for a given dataset



Semi-Supervised Anomaly Analysis

Anomaly Workflow : Unsupervised Analysis

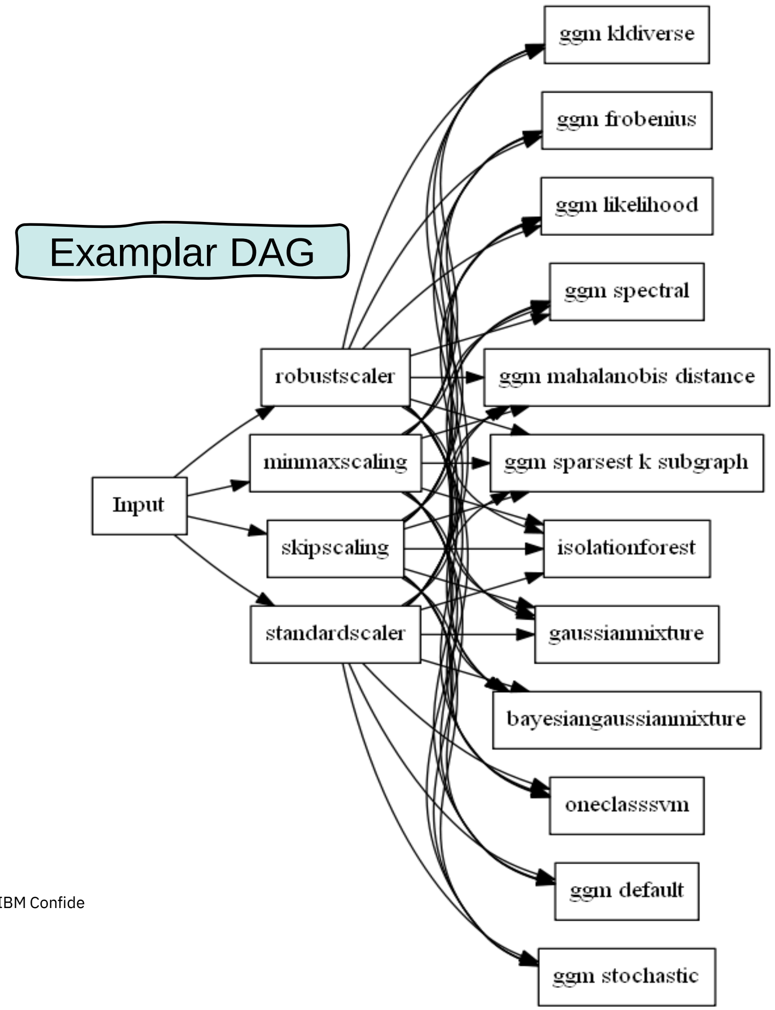
Multiple Pipeline Evaluations for Discovering the best options for a given dataset

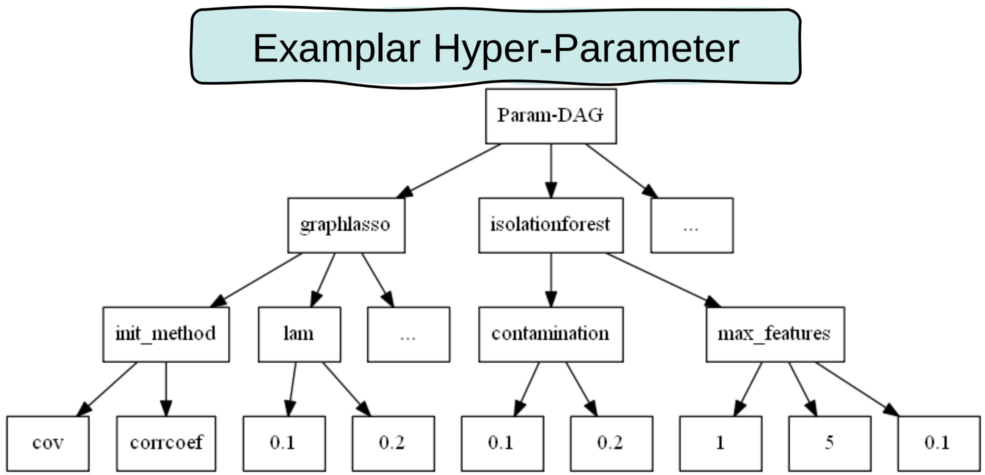


Unsupervised Anomaly Analysis

Anomaly Workflow

Example: DAG for Semi-supervised Exploration

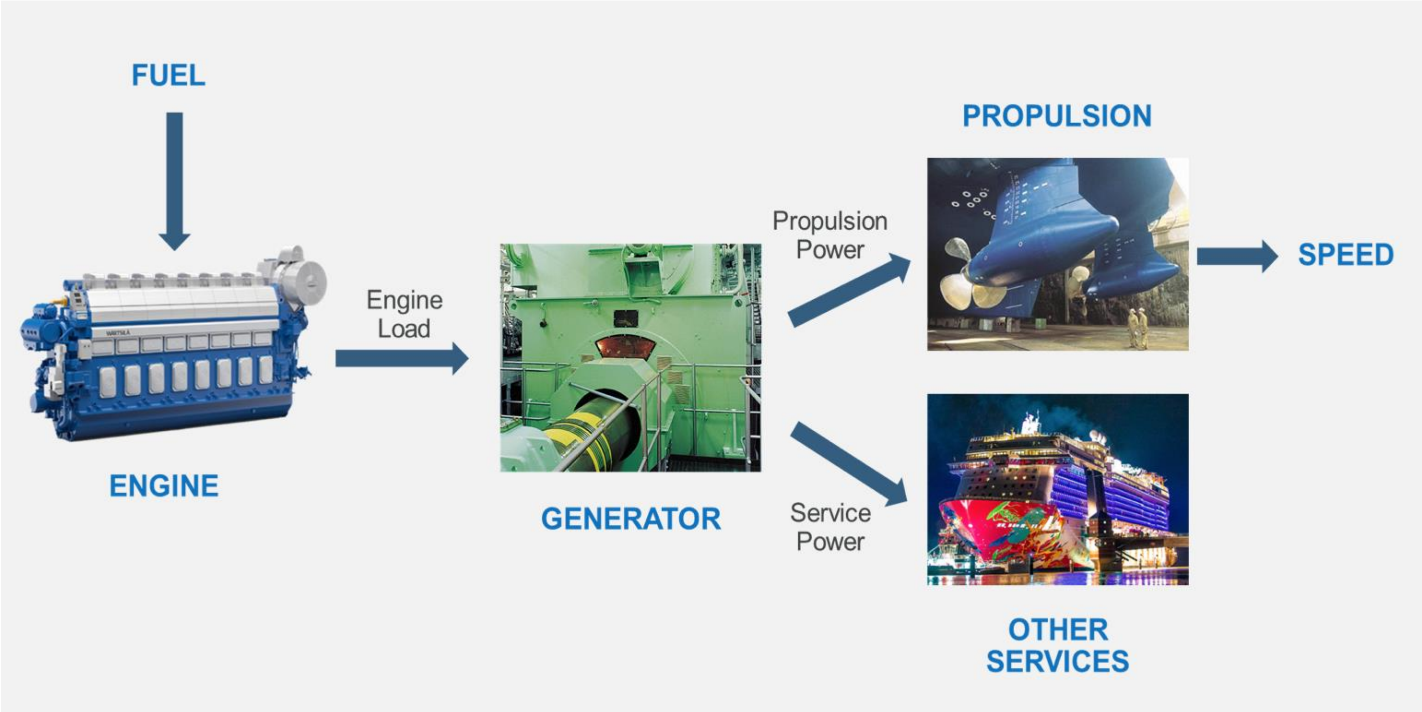




Topic III:
Anomaly Detection Use case

Use case 1: Anomaly Detection in Engine Dataset

- Total 2GB of data, including different temperatures and pressures measured from engines
- 170 different variables per engine
- Data is collected when exceeding the change threshold and at 1s intervals
- Total 93 different trips ranging from one day to seven-day duration



Use case 1: Dependency structure in engine measurements

- Data is segregated using LEG information into different Trips
  - Total 93 different trips ranging from one day to seven-day duration
  - Each trip include an observation of 89 variables sampled at every 30 seconds
- We study the *cross-correlation* between pair of 89 variables for each trips
- Figure show the plot of number of pairs having absolute cross-correlation higher than 0.7 for each trip
- We can see, trips between **40 to 80 has** less number of correlated pairs of variables as compared to other trips

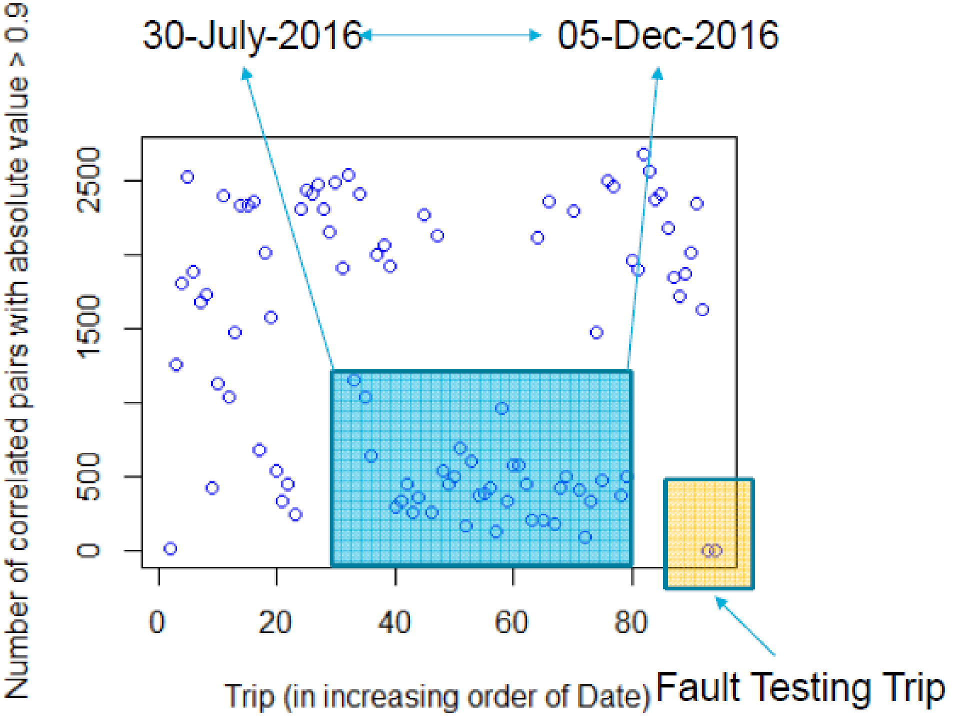
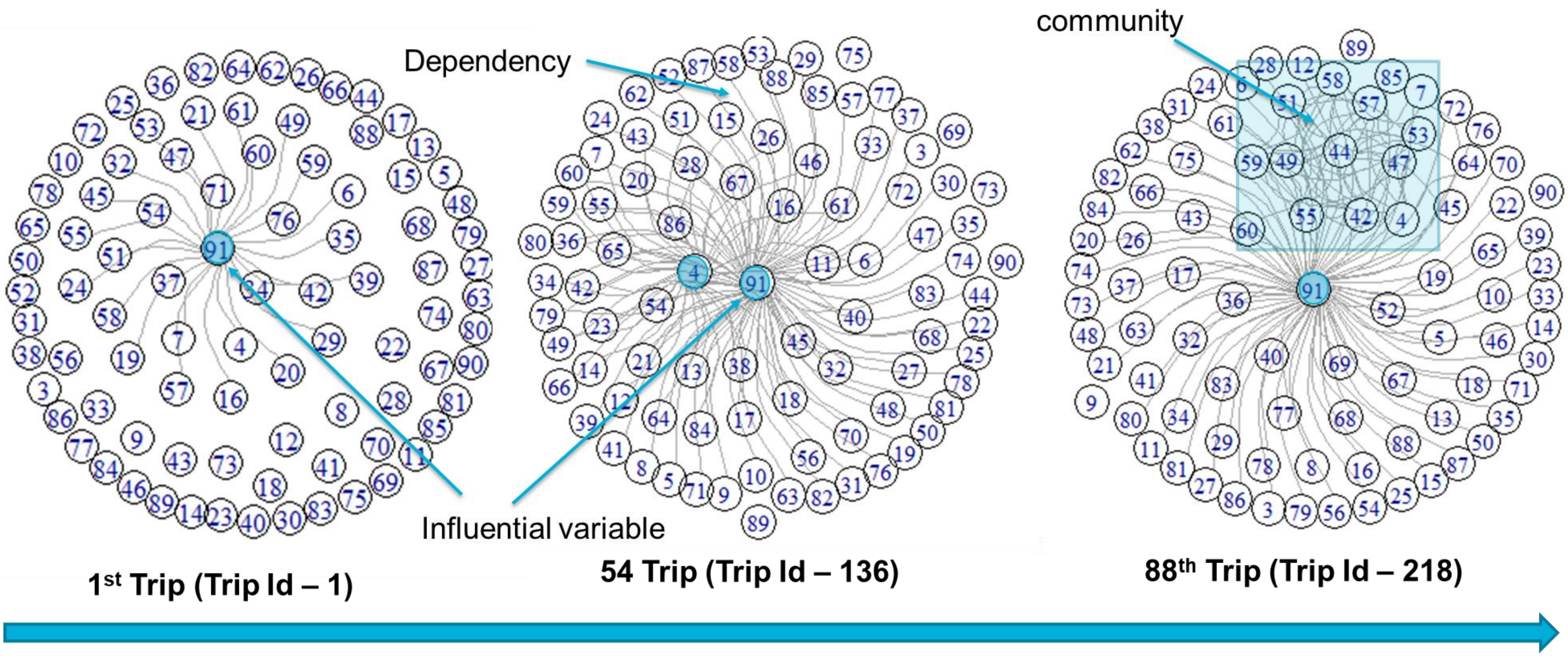


Figure 1

Use case 1: Dependency structure in engine measurements

- Data is segregated using LEG information into different Trips
  - Total 93 different trips ranging from one day to seven-day duration
  - Each trip include an observation of 89 variables sampled at every 30 seconds



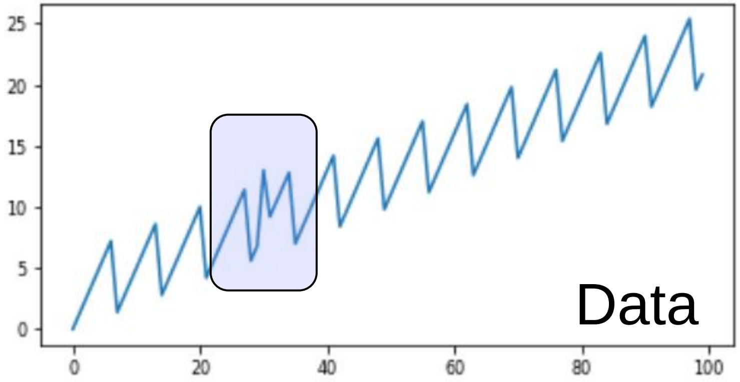
Each vertex is a one variable. Eg. Vertex 91 is “ENGINE3_TURBO.SPEED.SE518”. The graph captures relationship across variables

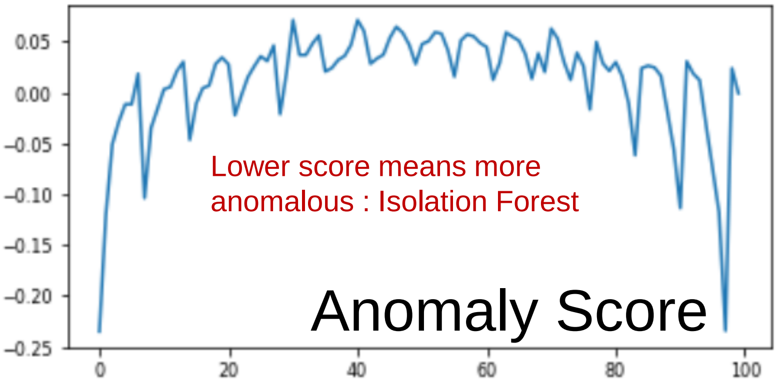
Topic IV:
API for Data Scientist

(Open Source) Anomaly Operators in Action

Algo 1. Let us code it using an open-source tool

```
import pandas as pd
seasonal_trend = pd.read_csv('seasonal+trend.csv')
from sklearn.ensemble import IsolationForest
isolation_operator = IsolationForest()
X = seasonal_trend[seasonal_trend.columns[-1]].values.reshape(-1,1)
isolation_operator.fit(X)
anomaly_X = isolation_operator.decision_function(X)
plt.plot(X)
plt.show()
plt.plot(anomaly_X)
```





Results are not as I was expecting …

Let us see what Expert has to say …

IID vs Time series

- Time series has structure
  - Stationarity (e.g., markov, exchangeability)
  - Typical stochastic process assumptions (e.g., independent increment as in Poisson process)
  - Mixtures of above

Don’t worry if you don’t know all these terminologies!

- Typical statistics involved
  - Transition probabilities
  - Event counts
  - Mean, variance, spectral density,…
  - Generally, likelihood ratio of some kind
  - Auto-correlation
- Try to exploit some of these structures in anomaly detection tasks to get better results
- Let us start with *time series windowing operator (“Flatten”)*

WindowAD : Anomaly Pipeline in Action

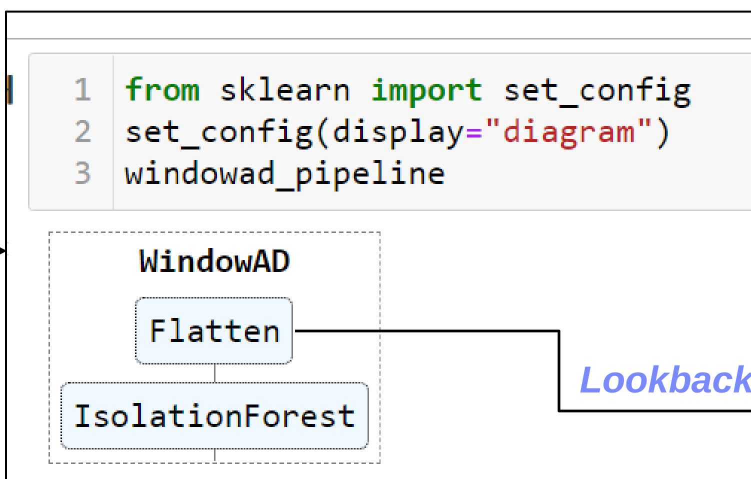
IBM

Algo 2. Let us code it using "***WindowAD***" pipeline

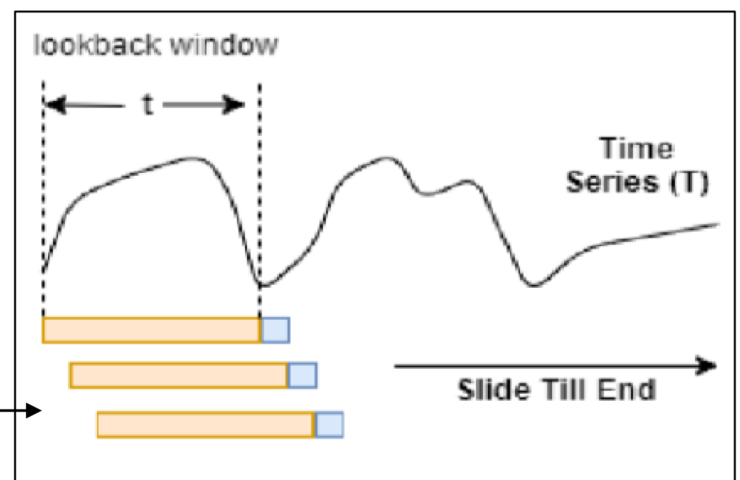
```
seasonal_trend = pd.read_csv('seasonal+trend.csv')
seasonal_trend = seasonal_trend.values
windowad_pipeline = WindowAD(steps=
                             [('flatten',Flatten()),
                              ('isolation',IsolationForest())],
                            lookback_win=5,
                            target_columns=[1],
                            feature_columns=[1],
                            scoring_method='iid')
windowad_pipeline.fit(X=seasonal_trend)

anomaly_label = windowad_pipeline.predict(X=seasonal_trend, prediction_type='training')
plt.plot(anomaly_label)
plt.show()

anomaly_score = windowad_pipeline.anomaly_score(X=seasonal_trend,
prediction_type='training')
plt.plot(anomaly_score)
```



Lookback_win



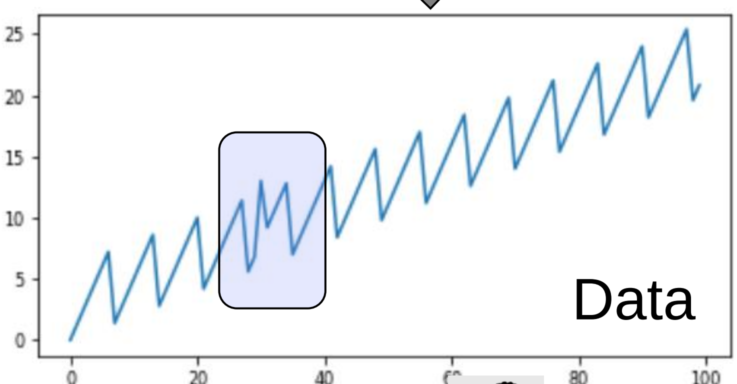
WindowAD : Anomaly Pipeline in Action

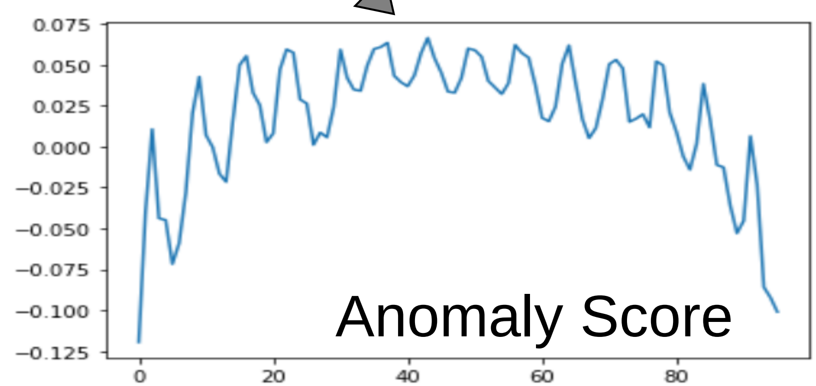
Algo 2. Let us code it using "***WindowAD***" pipeline

```
seasonal_trend = pd.read_csv('seasonal+trend.csv')
seasonal_trend = seasonal_trend.values
windowad_pipeline = WindowAD(steps=
                             [('flatten',Flatten()),
                              ('isolation',IsolationForest())],
                            lookback_win=5,
                            target_columns=[1],
                            feature_columns=[1],
                            scoring_method='iid')
windowad_pipeline.fit(X=seasonal_trend)

anomaly_label = windowad_pipeline.predict(X=seasonal_trend, prediction_type='training')
plt.plot(anomaly_label)
plt.show()

anomaly_score = windowad_pipeline.anomaly_score(X=seasonal_trend,
prediction_type='training')
plt.plot(anomaly_score)
```





Oops… Still not there yet…

WindowAD : Anomaly Pipeline in Action

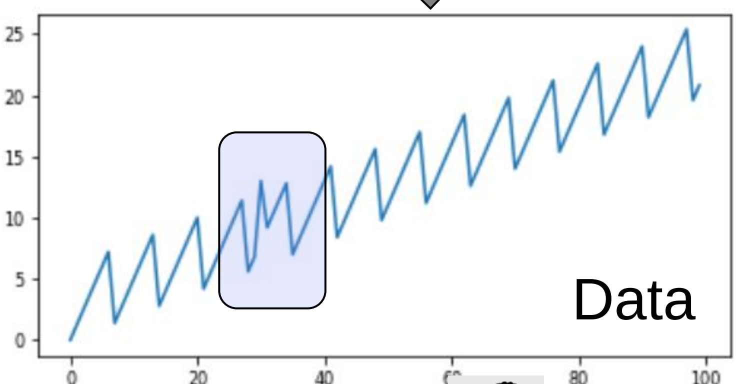
Algo 2. Let us code it using "***WindowAD***" pipeline

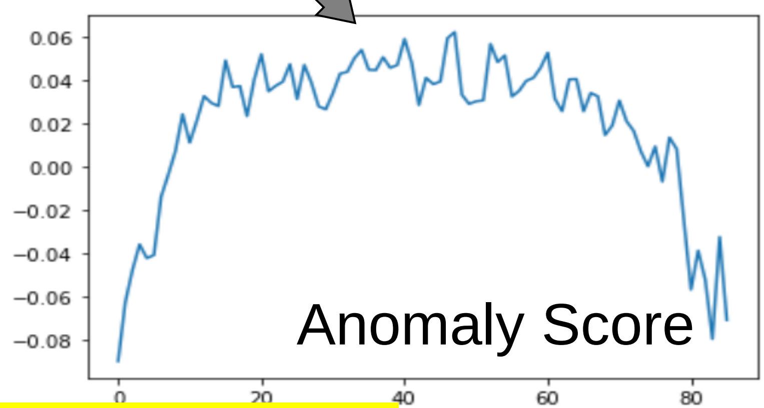
```
seasonal_trend = pd.read_csv('seasonal+trend.csv')
seasonal_trend = seasonal_trend.values
windowad_pipeline = WindowAD(steps=
                             [('flatten',Flatten()),
                              ('isolation',IsolationForest())],
                            lookback_win=15,
                            target_columns=[1],
                            feature_columns=[1],
                            scoring_method='iid')
windowad_pipeline.fit(X=seasonal_trend)

anomaly_label = windowad_pipeline.predict(X=seasonal_trend, prediction_type='training')
plt.plot(anomaly_label)
plt.show()

anomaly_score = windowad_pipeline.anomaly_score(X=seasonal_trend,
prediction_type='training')
plt.plot(anomaly_score)
```

Let me change Lookback





Oops… Still not there yet…

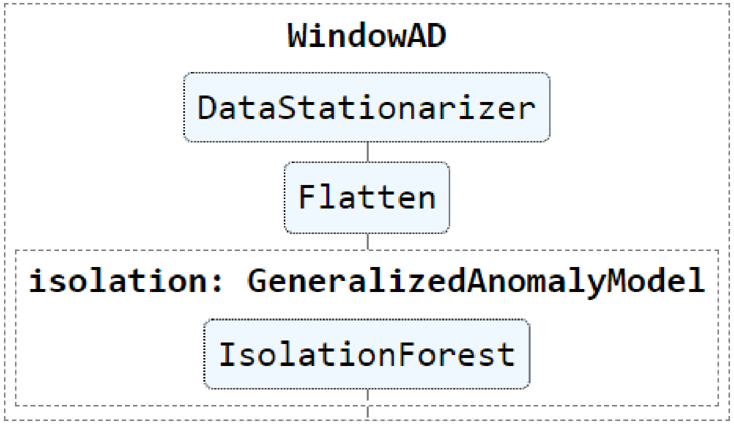
Steps taken so far to build anomaly model

Effort till so far….

- ***Algo 1.*** Tried IID version based on Isolation Forest
- ***Algo 2.*** Used Time Series Windowing based Approach with Isolation Forest
- ***Algo 2++***. Varied Lookback, but still no luck

- In Addition,
  – Why anomaly score is not high for the data points that are anomalous?
  – There is a disconnection across different anomaly algorithm

- Let us start with an extended version of ***WindowAD*** pipeline as follow:
  1. Use time series data standardization
  2. *time series windowing operator ("Flatten")*
  3. Then finally try to use an outlier detection algorithm



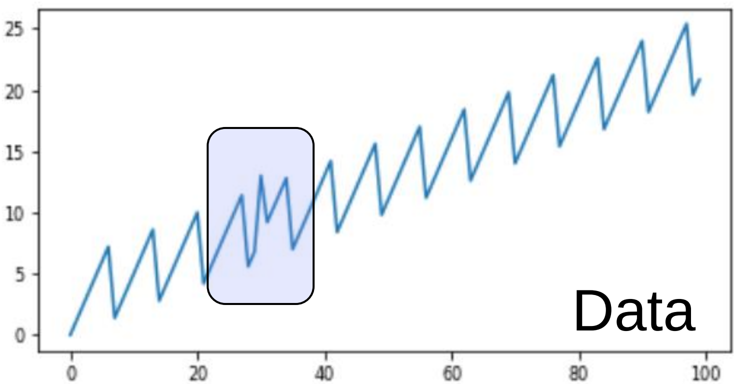
WindowAD : Anomaly Pipeline in Action

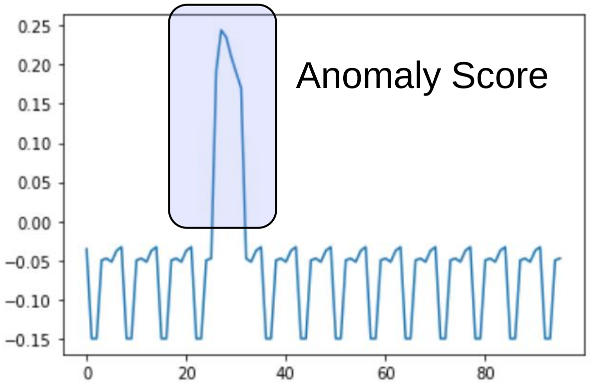
Algo 3. Let us code it using "***WindowAD***" pipeline

```
gam = GeneralizedAnomalyModel(base_learner=IsolationForest(),
                              predict_function='decision_function',
                              score_sign=-1)
seasonal_trend = pd.read_csv('seasonal+trend.csv')
seasonal_trend = seasonal_trend.values
windowad_pipeline = WindowAD(steps=
                             [("DataStat", DataStationarizer()),
                              ('flatten',Flatten()),
                              ('isolation',gam)],
                            lookback_win=5,
                            target_columns=[1],
                            feature_columns=[1],
                            scoring_method='iid')
windowad_pipeline.fit(X=seasonal_trend)

anomaly_score = windowad_pipeline.anomaly_score(X=seasonal_trend, prediction_type='training')
plt.plot(anomaly_score)
```

Great… But
Change lookback





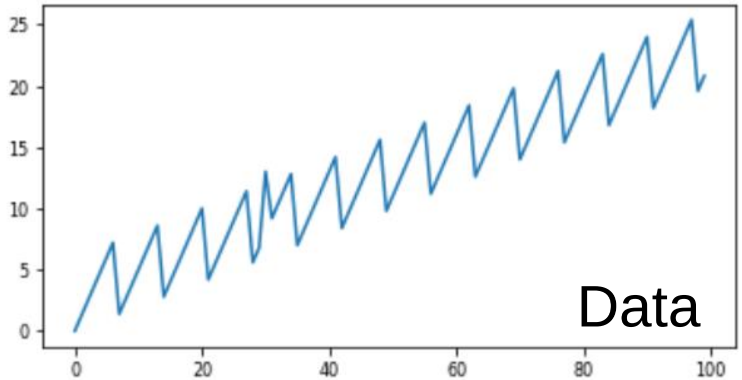
WindowAD : Anomaly Pipeline in Action

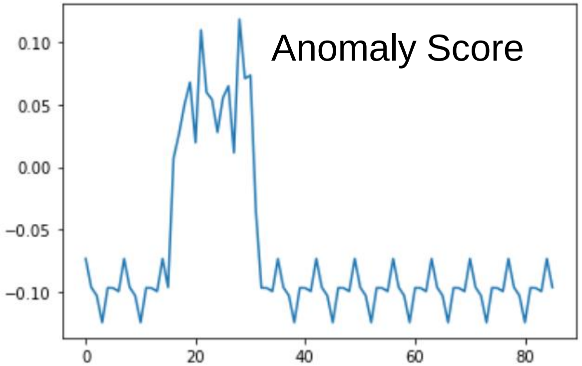
Algo 3. Let us code it using "***WindowAD***" pipeline

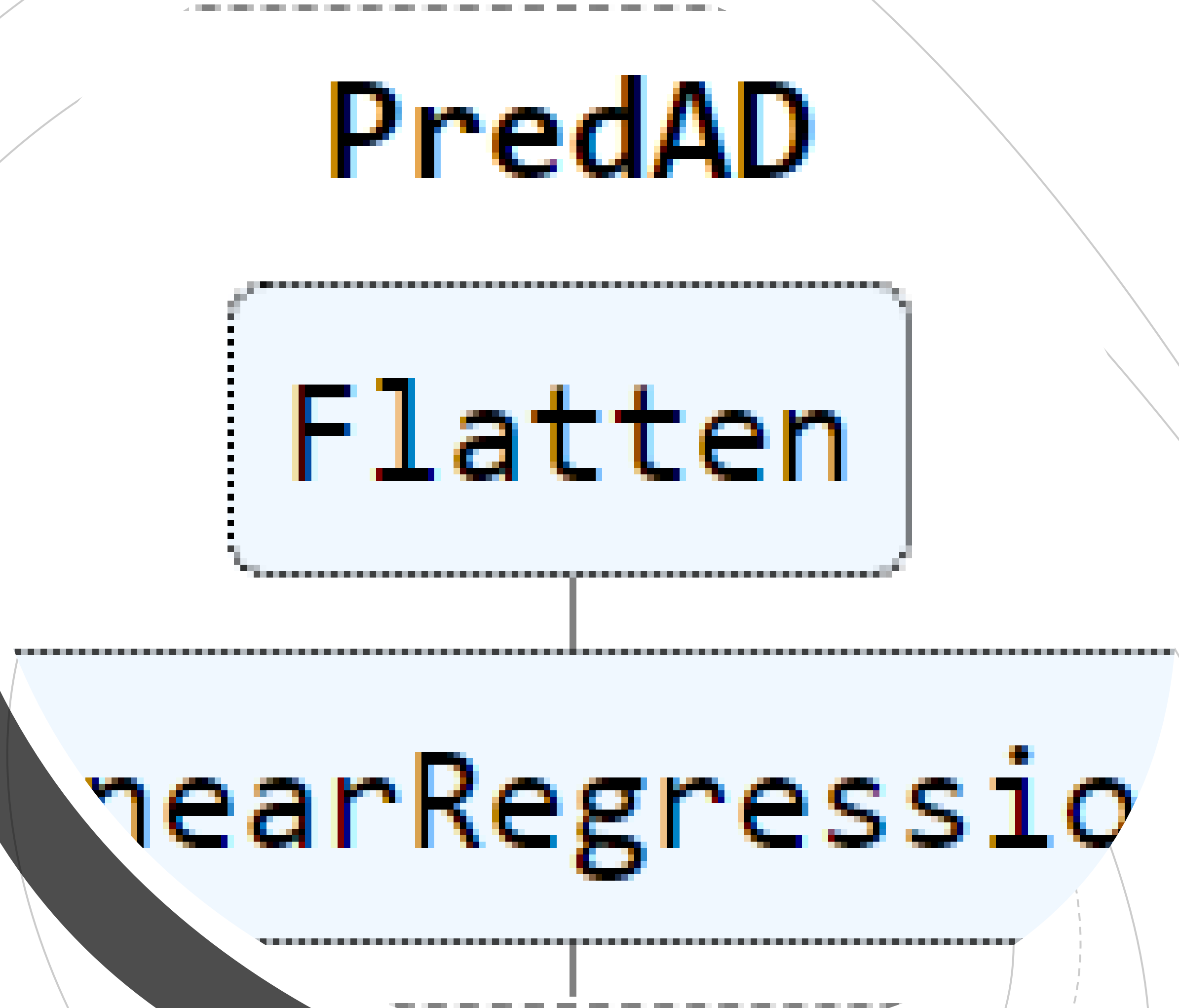
```
gam = GeneralizedAnomalyModel(base_learner=IsolationForest(),
                              predict_function='decision_function',
                              score_sign=-1)
seasonal_trend = pd.read_csv('seasonal+trend.csv')
seasonal_trend = seasonal_trend.values
windowad_pipeline = WindowAD(steps=
                             [("DataStat", DataStationarizer()),
                              ('flatten',Flatten()),
                              ('isolation',gam)],
                            lookback_win=15,
                            target_columns=[1],
                            feature_columns=[1],
                            scoring_method='iid')
windowad_pipeline.fit(X=seasonal_trend)

anomaly_score = windowad_pipeline.anomaly_score(X=seasonal_trend, prediction_type='training')
plt.plot(anomaly_score)
```

We are still good



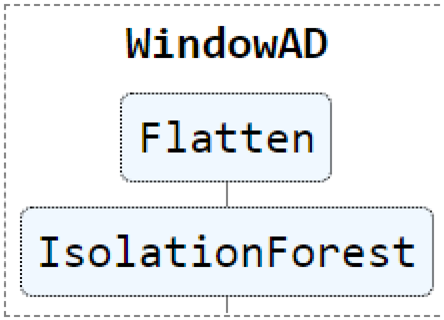




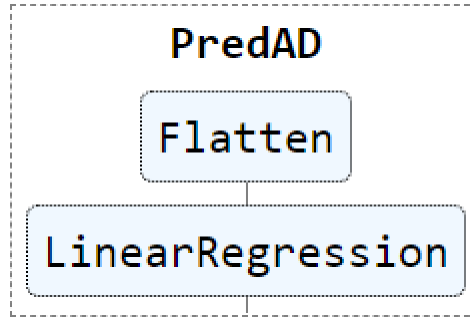
AnomalyKits Design Consideration

AnomalyKits Design Consideration

- Switching between different anomaly pipelines is made easy




```
WindowAD(steps=
         [('flatten',Flatten()),
          ('isolation',IsolationForest())],
        lookback_win=15,
        target_columns=[1],
        feature_columns=[1],
        scoring_method='iid')
```




```
PredAD(steps=[("flatten", Flatten()),
              ("linearregression", LinearRegression())
             ],
       lookback_win=5,
       feature_columns=[1],
       target_columns=[1],
       pred_win=1)
```

- Point 1. unifying framework for anomaly detection methods
  - 100+ different anomaly pipelines can be constructed using 4 exemplar anomaly pipelines

- Point 2: framework for reducing the anomaly detection delay time
  - Scoring methodology for post analysis of anomaly score

Steps taken so far to build a good anomaly model

- AnomalyKits bring many such "*cool*", "*technical*" and key "*differentiator*" capabilities
  - Formalization and Implementation of sklearn based exemplars anomaly pipelines to cover wide range of anomaly detection approaches
    - PredAD,
    - DeepAD,
    - WindowAD,
    - RelationshipAD,
    - ReconstructAD, …
  - Support for Dynamic and Static anomaly thresholding techniques to generate anomaly labels and alert
    - Point Anomaly : Q-Score, Chi-Square Test, Sliding-Window Threshold, Adaptive Sliding-Window Threshold, …
    - Contextual Anomaly
  - Anomaly DAG constructs to conduct multiple pipeline evaluations for discovering the best options for a given dataset and Parameter tuning
    - Unsupervised Model Ranking for IID data : EM Score, AL Score, MV Score
  - Support for Data-Driven Intelligent Lookback Generation Capability
    - AIC Score based, BIC score based, T-Statistic based, Model-CV based, cross-validation based, …

Topic IV:
Web based Anomaly Detection System

API Hub:

Anomaly Detection Service

A Web based Anomaly Detection Service

Key differentiators

- Support Univariate and Multi-Variate time series
- Support Batch and Train-Test Anomaly Scoring
- Capable to support upto 100 active users
- Away from Installation and infrastructure free
- Use from anywhere

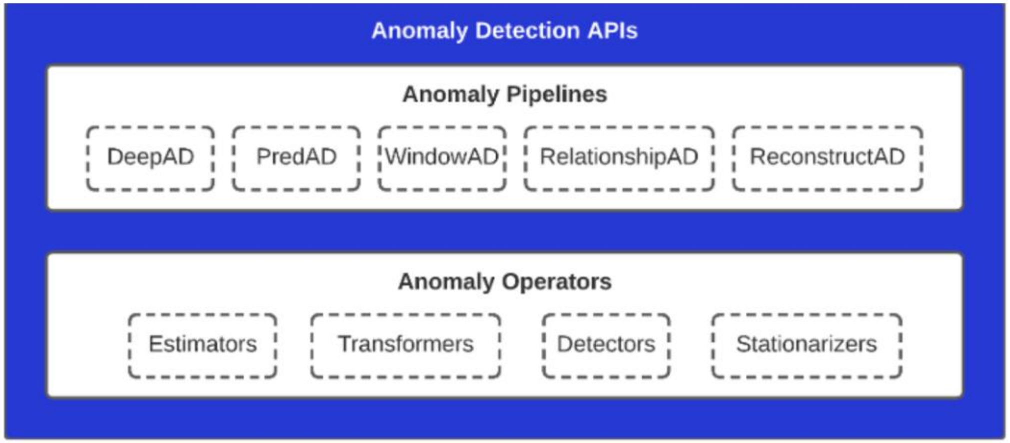


Figure 5: Anomaly pipeline stack

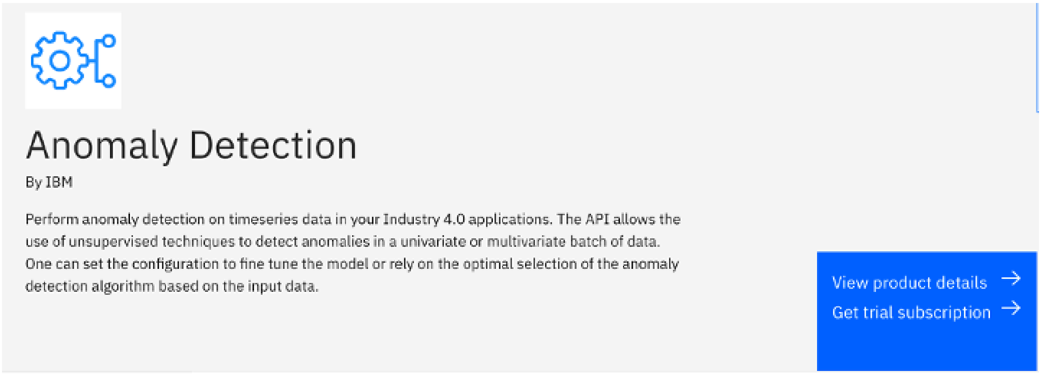


Figure 6: Anomaly Detection API Hub

API Hub:

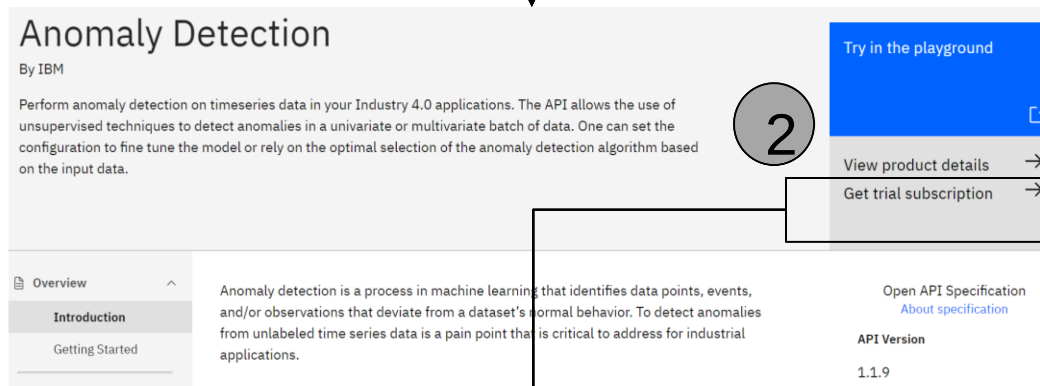
Anomaly Detection Service

A Web based Anomaly Detection Service

Key differentiators

- Support Univariate and Multi-Variate time series
- Support Batch and Train-Test Anomaly Scoring
- Capable to support upto 100 active users
- Away from Installation and infrastructure free
- Use from anywhere

1 https://developer.ibm.com/apis/catalog/ai4industry--anomaly-detection-product/Introduction



2

https://developer.ibm.com/profile/myapis/507688319

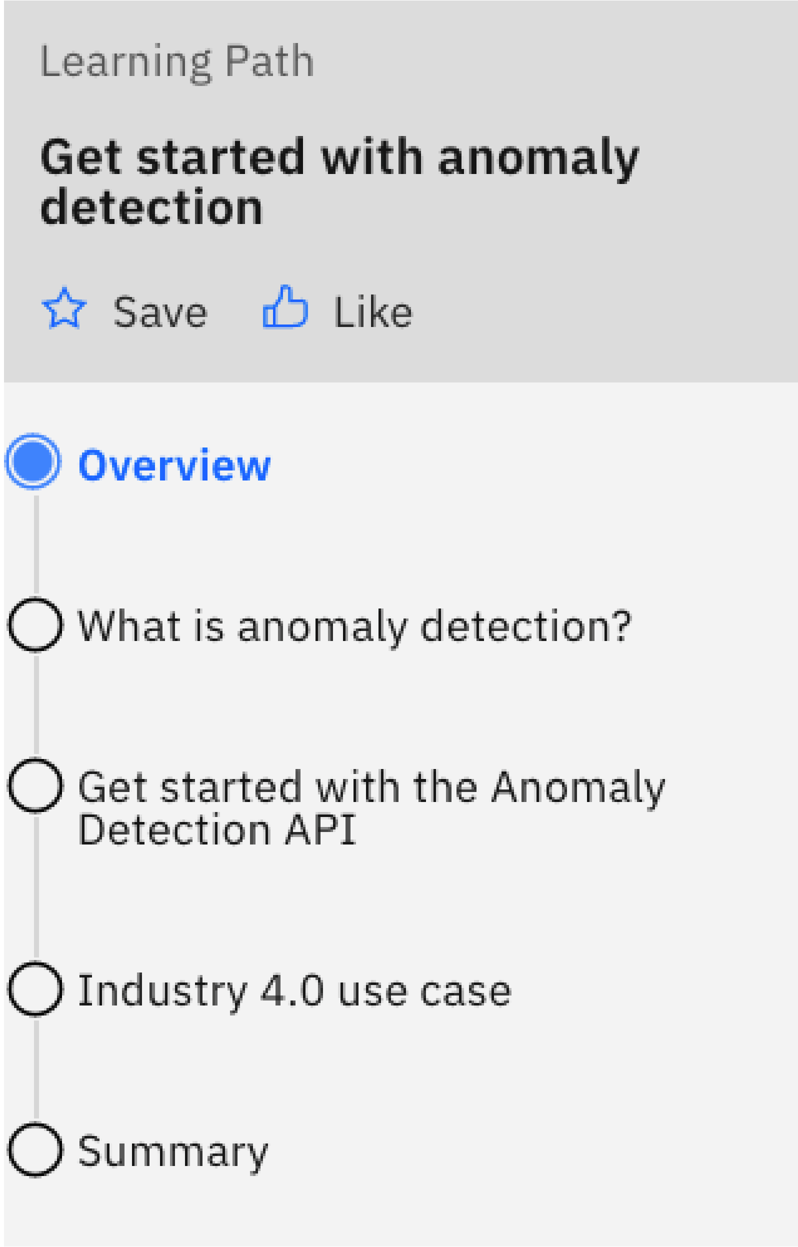
3

Anomaly Detection Learning Path

- https://developer.ibm.com/learningpaths/get-started-anomaly-detection-api/

- Introduction to anomaly detection
- Examples of data and types of anomalies
- Getting started with anomaly detection on the IBM API Developer Hub
- Industry 4.0 use case

- Skill level: Beginner to Intermediate
- Estimated time to complete: 1 hour



Code patterns

- IBM/anomaly-detection-code-pattern: Sample Jupyter Notebook for playing around with the Anomaly Detection service made available on API Hub (github.com)
- https://github.com/IBM/anomaly-detection-code-pattern
- Contains univariate and multivariate time series anomaly detection code patterns with the IBM Anomaly Detection API Service, with step by step instructions
- Python-based visualizations and analysis of the returned anomaly scores along with other results

Unsupervised Anomaly Detection in Multivariate Time Series Data

Many applications require being able to decide whether a new observation belongs to the same distribution as existing observations (it is an inlier), or should be considered as different (it is an outlier). Often, this ability is used to monitor the Assets.

The workflow of this notebook is as follows:

1. Provide Credential.
2. Load Dataset.
3. Compose Anomaly Service and Submit Job.
4. Monitor Job
5. Result Analysis

Credentials

This notebook requires two credentials. Please obtain your own credentials when customizing this notebook for your own work. Please visit **Anomaly Detection @ IBM** for trial subscription.

```
# Credentials required for running notebook

Client_ID = "replace-with-valid-client-ID"
Client_Secret = "replace-with-valid-client-Secret"
```

API Hub: Anomaly Detection Service Catalog

Anomaly Pipelines	Category	Supported Anomaly Estimators
WindowAD	Machine Learning	IsolationForest NearestNeighbor SyntheticRandomForestTrainer MinCovDet AnomalyEnsembler
PredAD/DeepAD	Mixed	-
ReconstructionAD	Deep Learning	DNN_AutoEncoder Seq2seq_AutoEncoder CNN_AutoEncoder DNN_VariationalAutoEncoder
RelationshipAD (Multi-Variate Only)	Machine Learning & Statistical	Covariance GMM_L0 GMM_L1 MachineTranslation

* Data Transformation related components are not included in the table

Key References

- D. Patel, G. Ganapavarapu, S. Jayaraman, S. Lin, A. Bhamidipaty, J. Kalagnanam. AnomalyKits: AnomalyKiTS: Anomaly Detection Toolkit for Time Series, AAAI-2022 Demo
- D Patel, S Shrivastava, W Gifford, S Siegel, J Kalagnanam, C Reddy: Smart-ML: A System for Machine Learning Model Exploration using Pipeline Graph, IEEE BigData 2020
- D Patel, SY Shah, N Zhou, S Shrivastava, A Iyengar, A Bhamidipaty, J Kalagnanam: FLOps: On Learning Important Time Series Features for Real-Valued Prediction, IEEE BigData 2020
- B Vinzamuri, E Khabiri, A Bhamidipaty, G Mckim, B Gandhi: An End-to-End Context Aware Anomaly Detection System. IEEE BigData 2020
- B. Vinzamuri, E. Khabiri, and A. Bhamidipaty: An Unsupervised Framework for Semantics Driven Causal Explanations for Anomalies, ISWC 2020
- TI Robert J. Baseman, Dzung T. Phan, Dhavalkumar C. Patel, Fateh A. Tipu: Applications of Gaussian Graphical Models for Process Control, Advanced Process Control Conference (APC), 2019
- T Idé, DT Phan, J Kalagnanam: Multi-task Multi-modal Models for Collective Anomaly Detection, ICDM 17
- DT Phan, T Idé: L0-Regularized Sparsity for Probabilistic Mixture Models, SIAM International Conference on Data Mining (SDM19), 2019
- DT Phan, LM Nguyen, NH Nguyen, JR Kalagnanam: Pruning Deep Neural Networks with L0-constrained Optimization, ICDM 2020
- DT Phan, M Menickelly: On the Solution of L0-Constrained Sparse Inverse Covariance Estimation Problems, INFORMS Journal on Computing, 2020
- DT Phan, T Idé, J Kalagnanam, M Menickelly, K Scheinberg: A Novel l0-constrained Gaussian Graphical Model for Anomaly Localization, ICDMW 2017