Application of Artificial Intelligence Techniques Enabling Coal Fired Power Plants the Ability to Achieve Higher Efficiency, Improved Availability, and Increased Reliability of Their Operation

Acknowledgment: "This material is based upon work supported by the Department of Energy Award Number DE-FE0031563."

Project Review

DE-FE0031563

Bill Spiesman, SparkCognition

13 May 2021

Project Milestones

	Milestone	Completion Date
M1	Project Kick-off Meeting	11/15/2018
M2	Historical Data Transfer from Plant Historian	6/26/2019
M3	Initial Model Development	9/30/2019
M4	Model Refinement based on SME Input	4/5/2021
M5	Annual Briefing	4/9/2019
M6	Completion of User Interface customization	4/30/2021
M7	Integration of User Interface and Completed Model into online system	4/30/2021
M8	Final Project briefing	9/30/2021

Project Overview

The primary objective of this engagement is to model and identify anomalous behavior in different modes of operations of coal-fired power plant assets.

The scope of effort involves leveraging existing sensor data feeds from the assets, and delivering a solution with the following objectives:

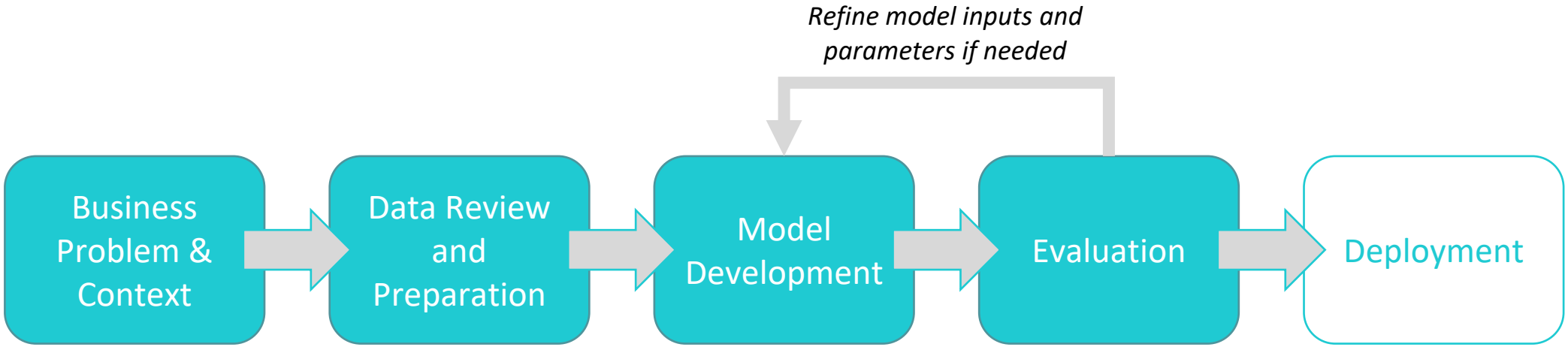
- Health assessment of scoped assets.
- Symptoms-based early warning of impending failures.
- Root-cause analysis providing a detailed view of anomalous behavior across features.

AI Strategy

- We find signatures of normal plant operational behavior using hundreds of sensor inputs.
- Using Machine Learning, we identify important data relationships that can flag critical operational changes.
- When unexpected changes occur, we generate an Alert, which includes key evidence in the form of sensor inputs that contributed to that Alert.

Data science process overview

SparkCognition employs a standard data science process that enables flexibility and quality in the model development and deployment phases.



Understand what the problem is, how it is currently handled and other relevant process details

Examine the data to build a baseline understanding in what the data represents, its quality, how it can be used. Scale and transform data as needed.

Train model(s) based on inputs identified in attempt to solve business problem

Evaluate model performance based on common statistical metrics given problem set

Prepare to operationalize the model by connecting to live data streams, deploying in software, integrating into workflow, etc.

Data Review

The following describes the data provided for this project.

The assets in-scope include:

- Steam Turbine (ST)
- Steam Turbine Generator (ST GEN)

Data received include:

- Nearly 3.5 years of historical sensor data for 704 tags across ST and ST GEN assets
- Historical log of maintenance events identified

Data Filtering: Model Definition

- Defining the asset high-level operating modes, each of which will be modeled.

Model	Definition	# of Tags
Fullspeed	• 3-TURB-SPEED-RPM > 3590	267
Lowspeed	• 3-TURB-SPEED-RPM <= 3590	267

Model	Definition	# of Tags
ON	• 3_GEN_NET_PWR_MW > 5	87
OFF	• 3_GEN_NET_PWR_MW <= 5	87

Data Filtering: Train/Test Split

Additionally, the data for each of the models must be split into mutually exclusive data sets for training and model evaluation.

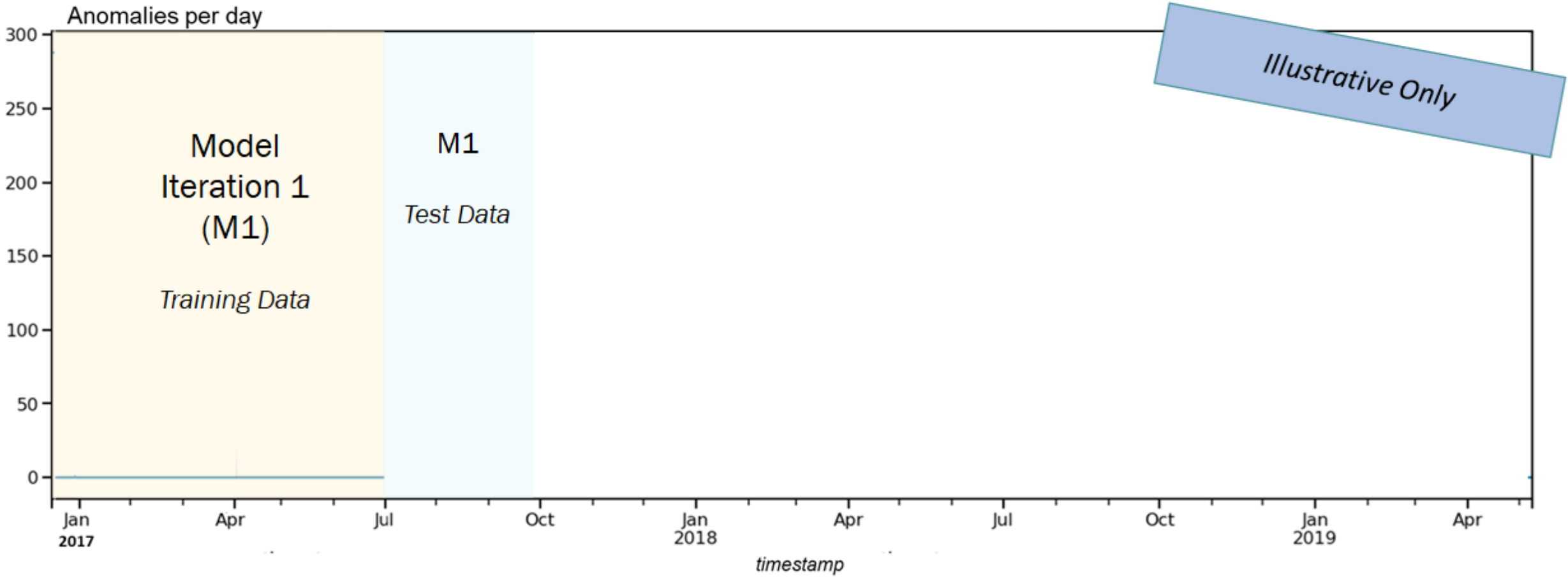
Training and testing periods are defined using an approach called **Sliding Window Back-testing**, which involves:

- Simulating live conditions by training multiple models on different windows of historical data and testing those models on "new" historical data.
- Sliding the window of training (model hyper parameter) with each stride having fixed amount of trained data. We use this if our focus is on recent past and not on distant past.

Sliding Window Back-testing is a common method used to define training and testing data sets in a way that mimics live deployment conditions.

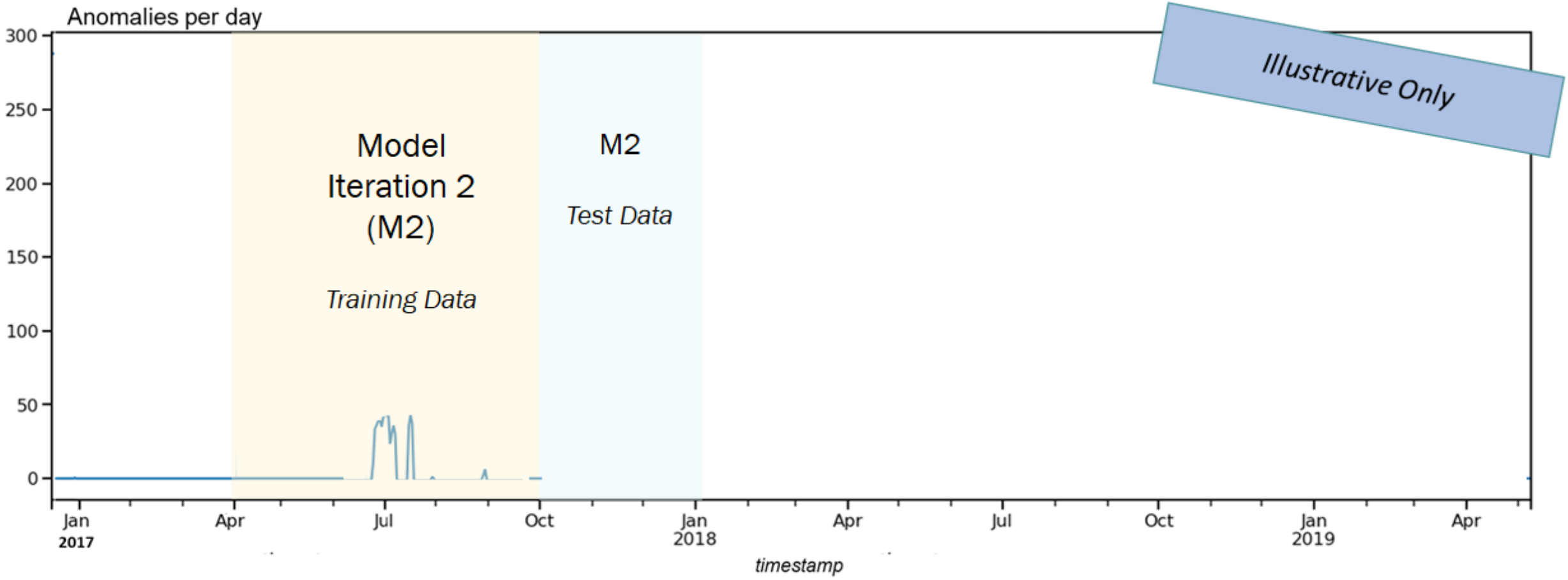
Sliding Window Back-Testing

The first training set is a defined window of 6 months from January through June 2017, followed by 3 months of testing data from July through September 2017.



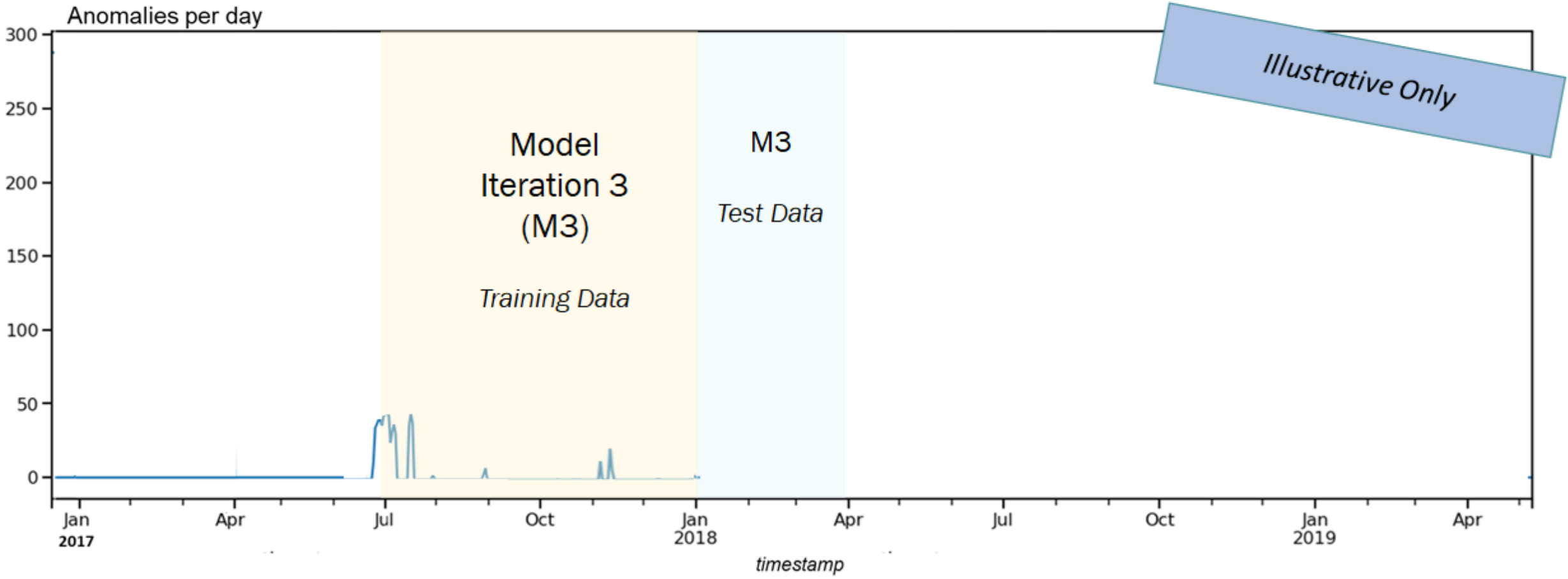
Sliding Window Back-Testing

Another model iteration is built by **sliding** both the training and testing windows by 1 month.
Training is now from Feb through July 2017, while testing is from Aug to Oct 2017.



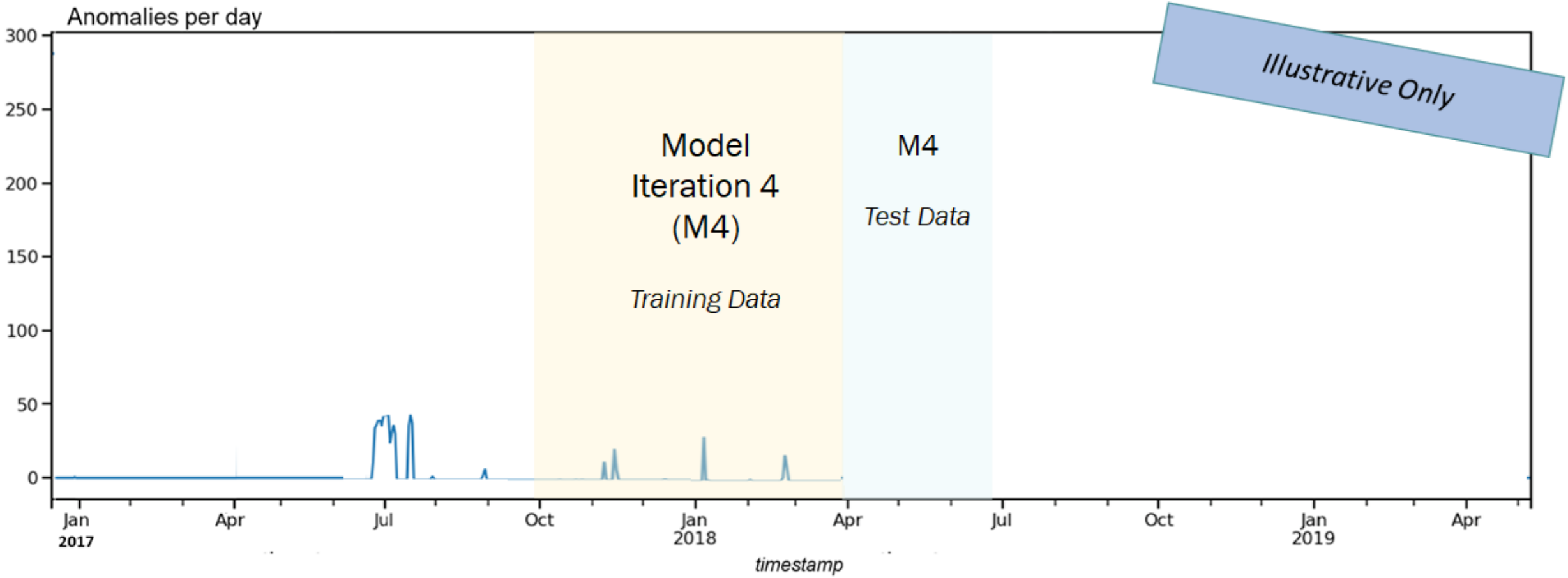
Sliding Window Back-Testing

A third iteration is built by sliding training and testing windows by another month.



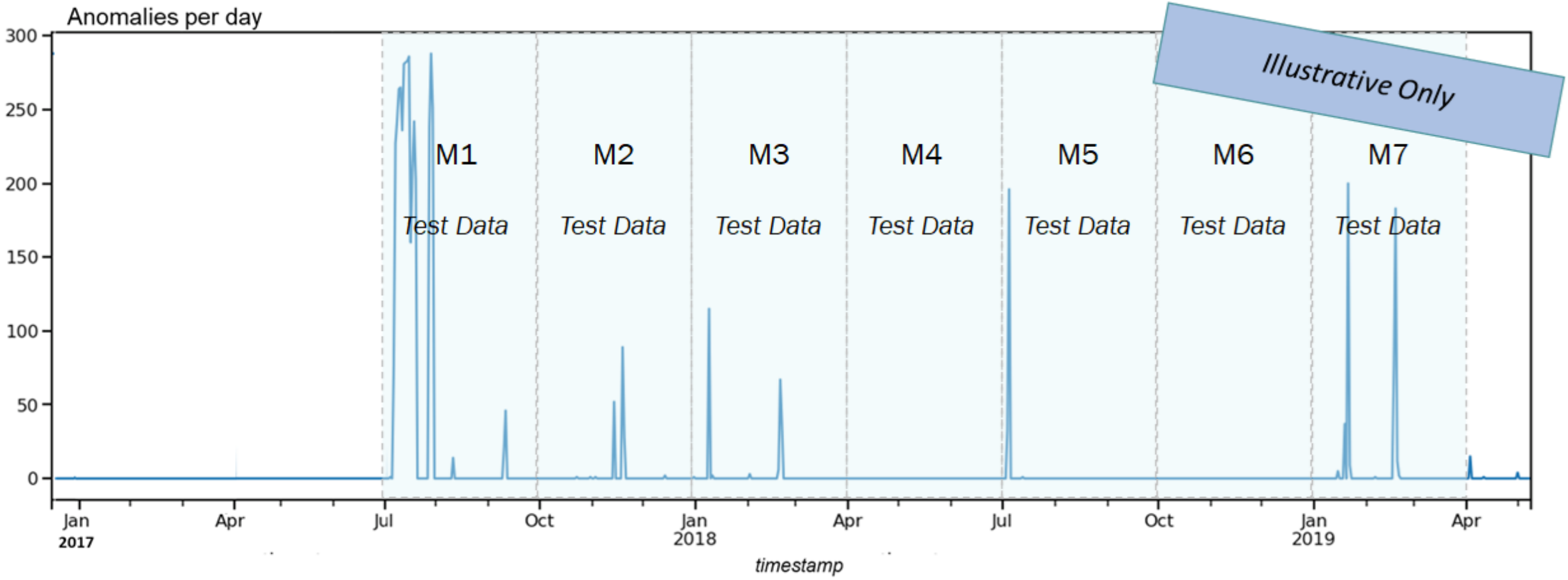
Sliding Window Back-Testing

And another.



Sliding Window Back-Testing

Until the end of the history and then all testing periods are stitched together for validation.



Modeling Training

The models developed involve several different machine learning (ML) approaches. Primarily, an **unsupervised** ML approach is used given the complexity of the assets and the limited availability of similar failures in the historical data.

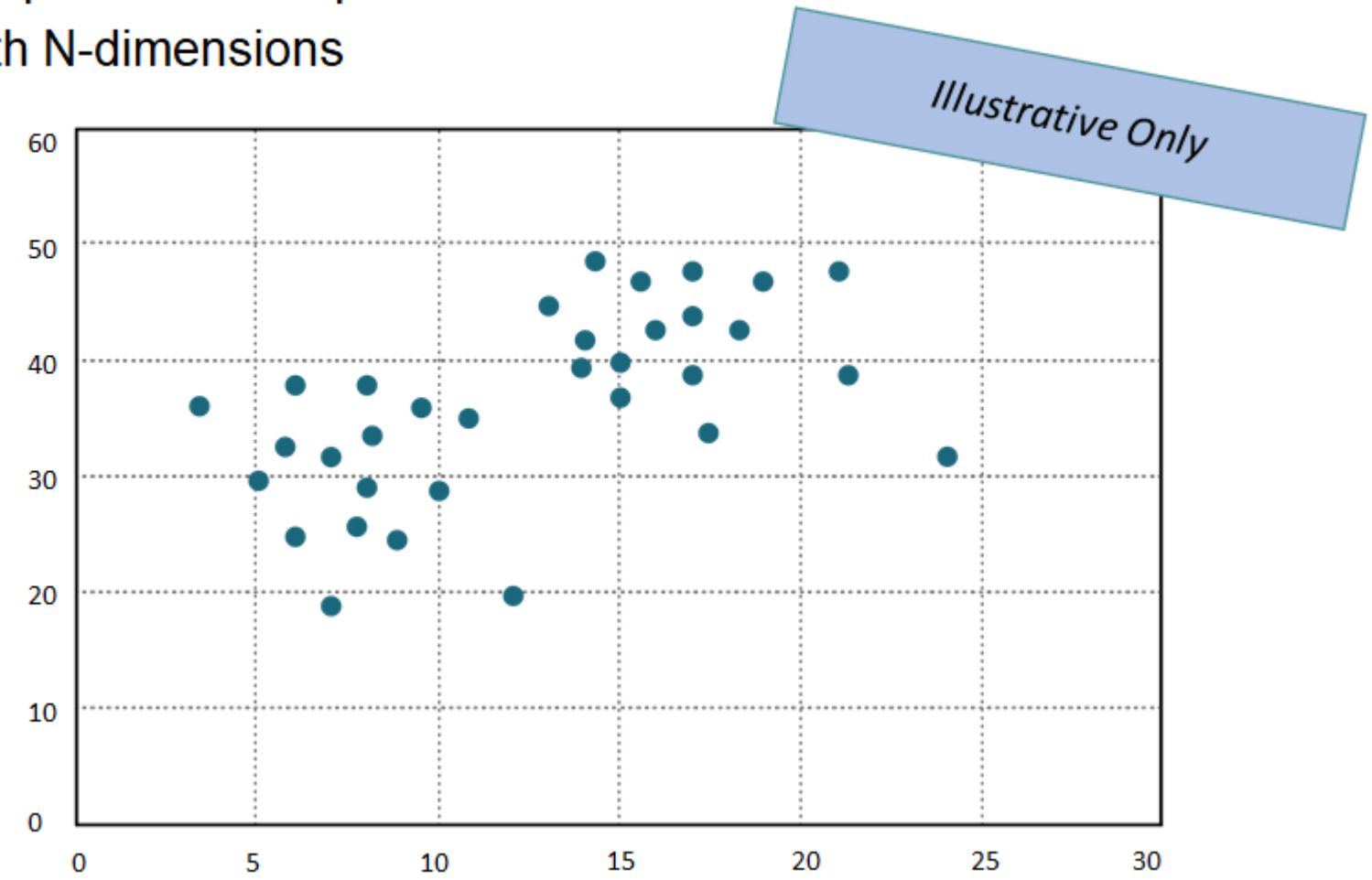
An ensemble of data science approaches are used to develop the final models, but the key components include:

- Clustering and anomaly detection
- Normal Behavior anomaly detection
- Alerting mechanism

Model Training: Clustering

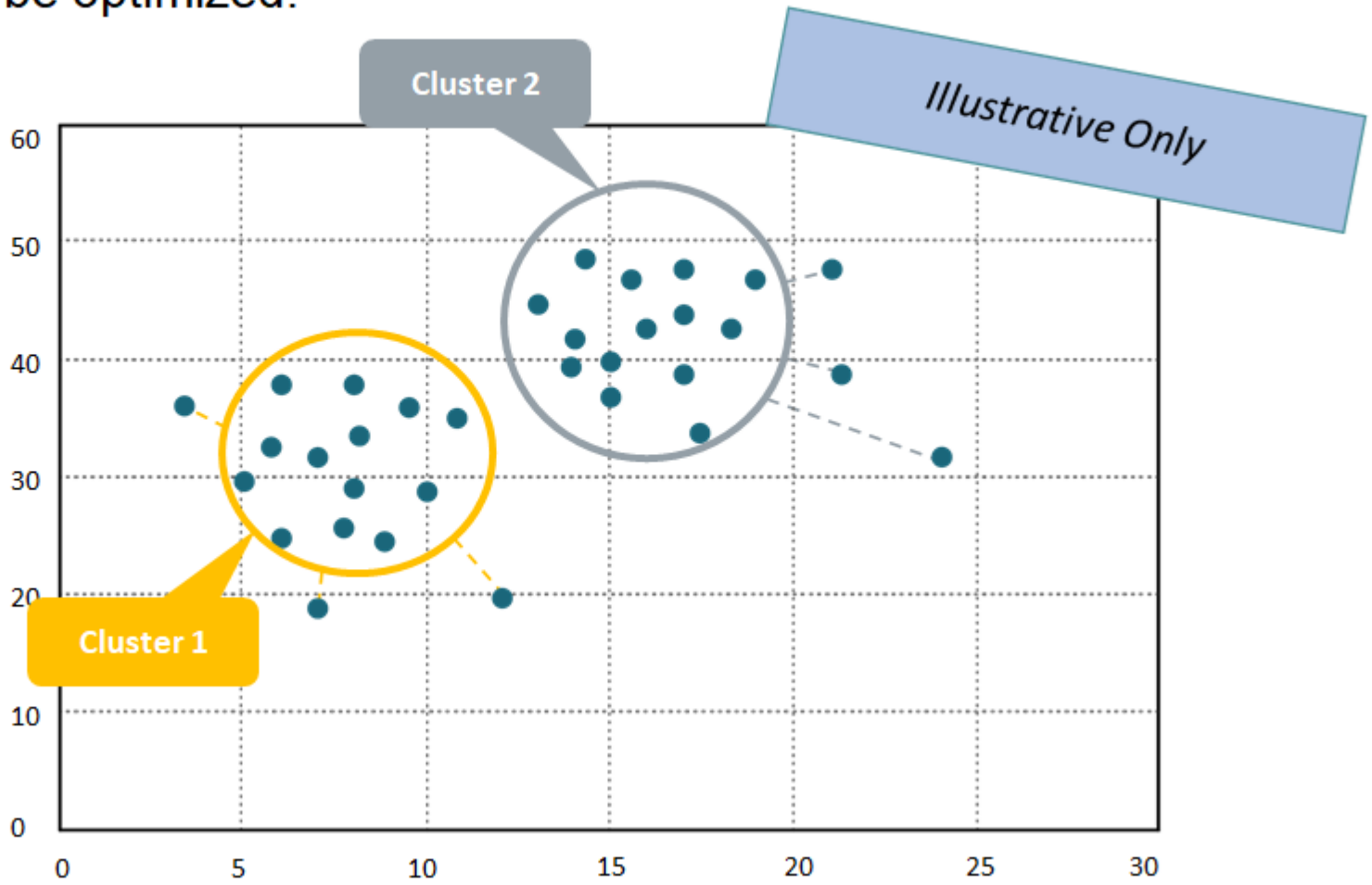
Clustering is an unsupervised approach used to detect groupings of data. This is the underlying ML approach used in this SparkPredict implementation.

Start with a dataset with N-dimensions



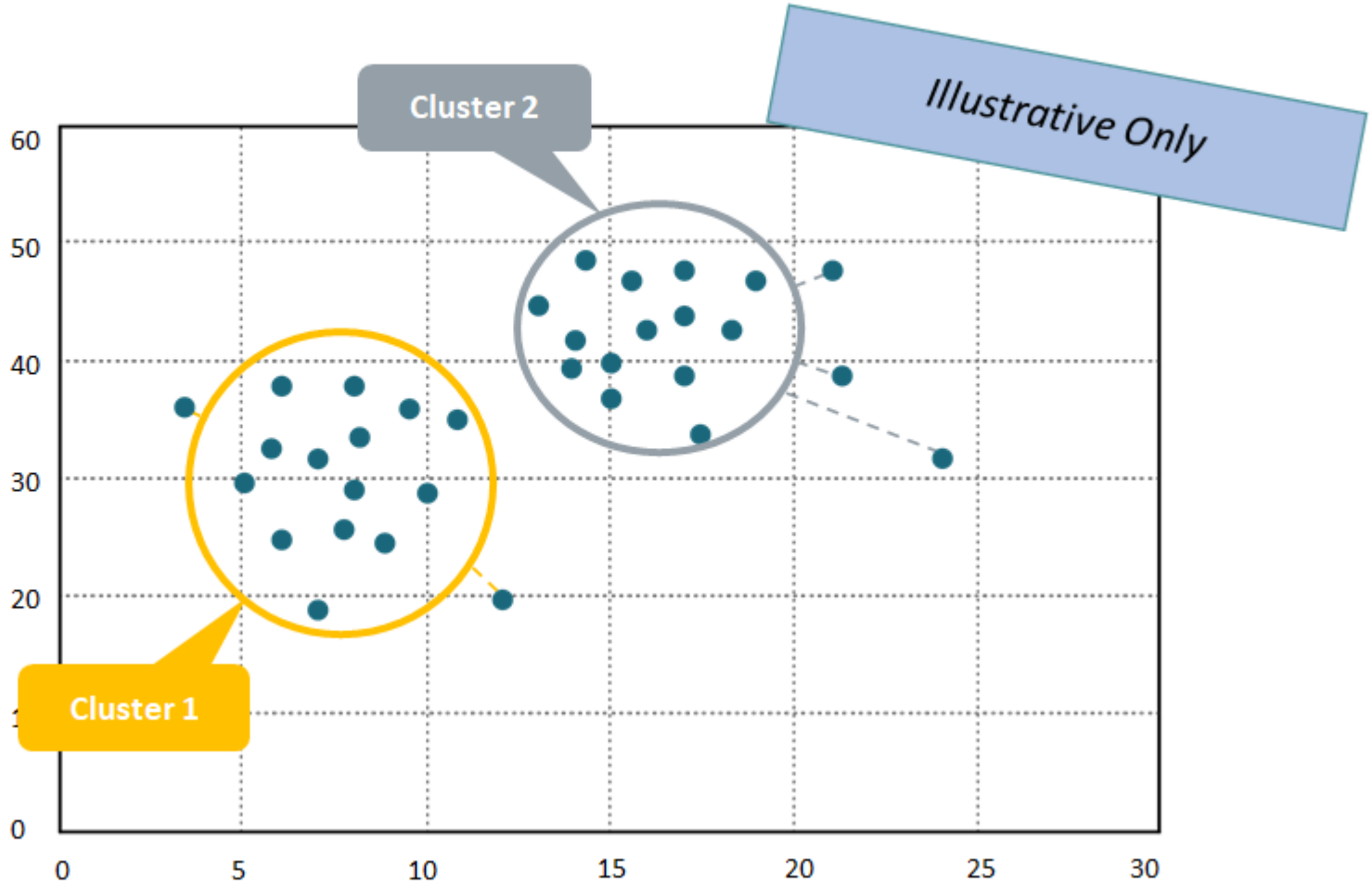
Model Training: Clustering

The computer takes a first attempt at classifying which groups of data go together. This is not a perfect fit and needs to be optimized.



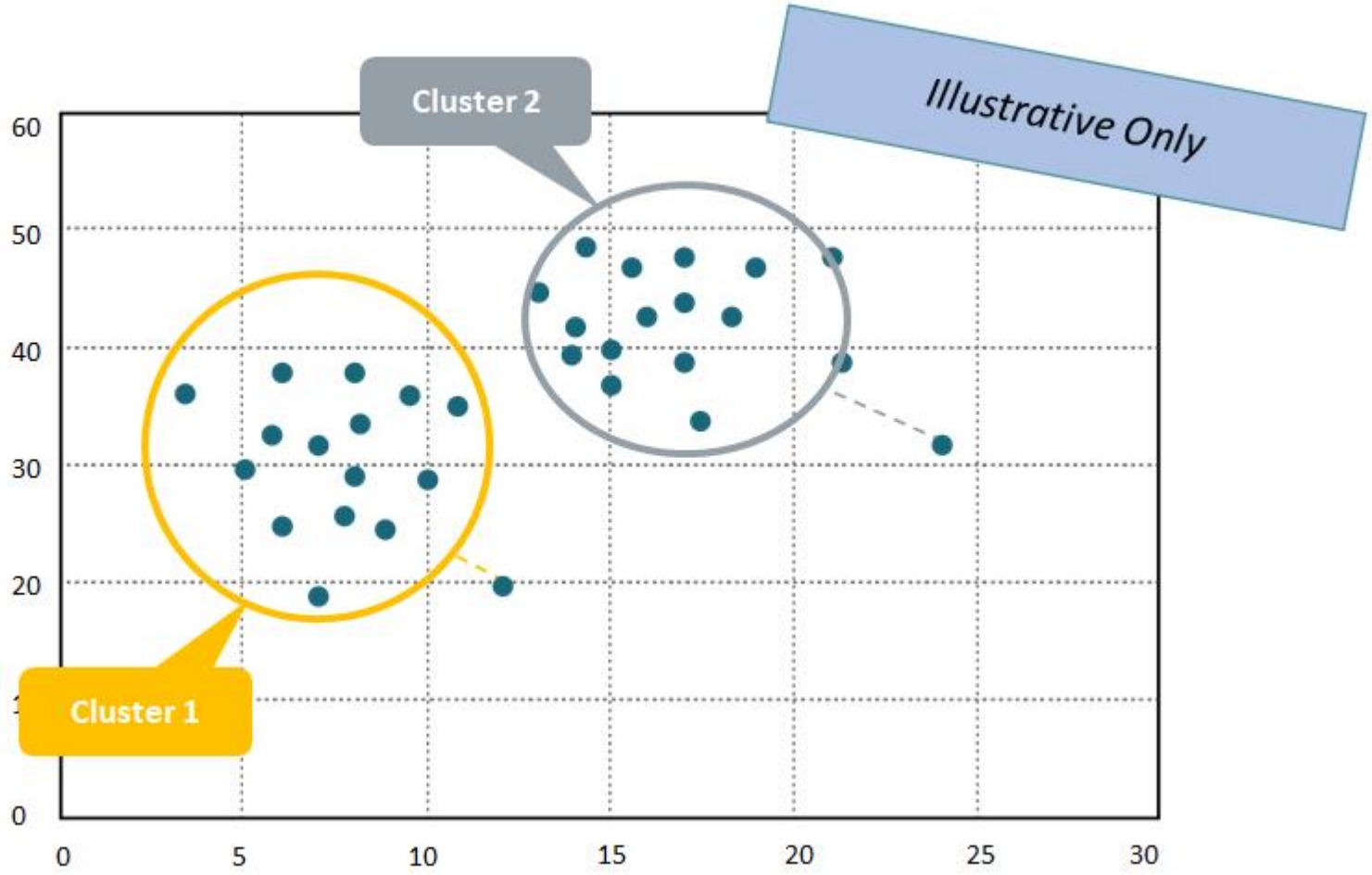
Model Training: Clustering

It then iterates to try and optimize how data is grouped.



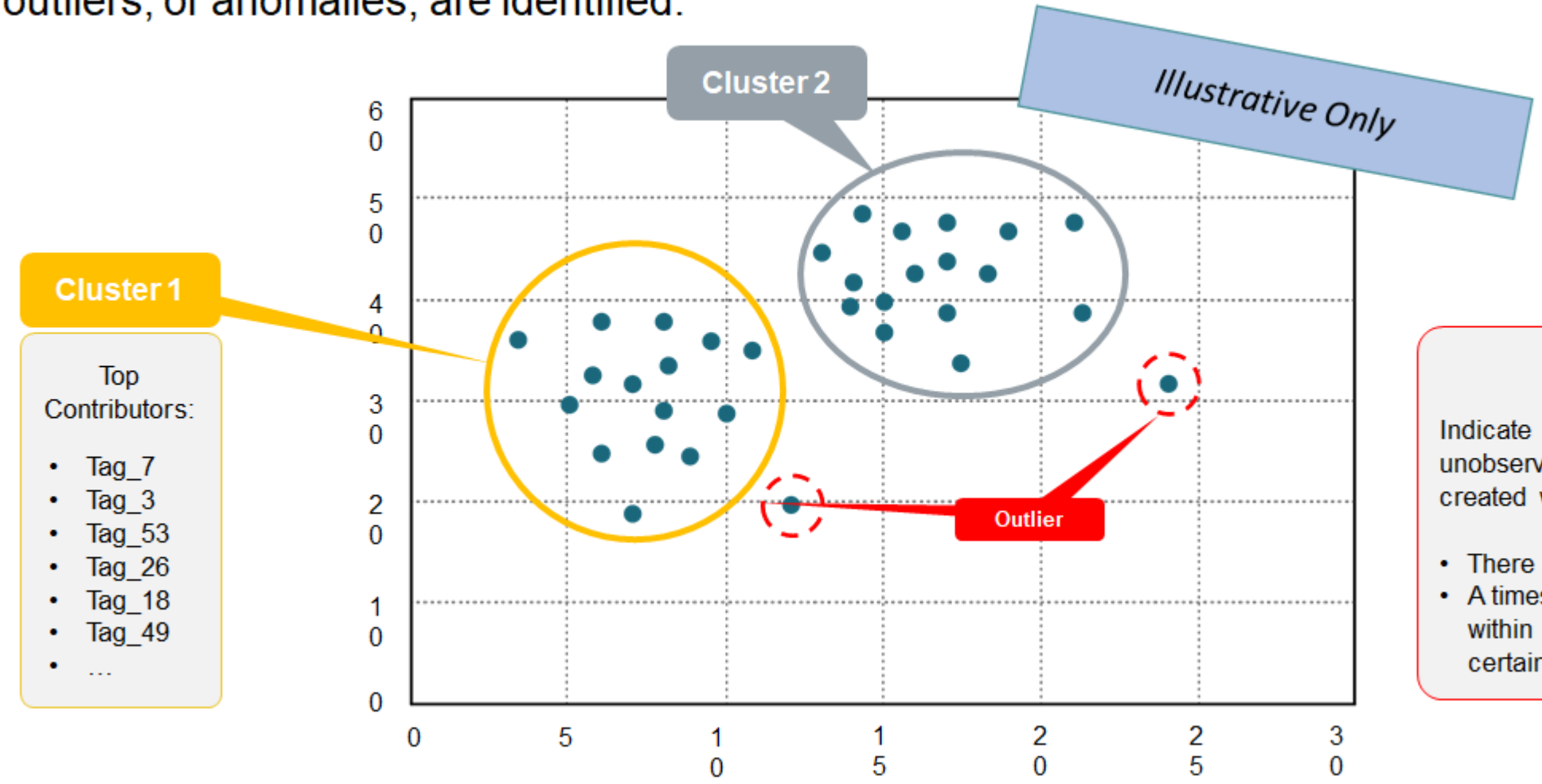
Model Training: Clustering

And continues to do so….



Model Training: Clustering

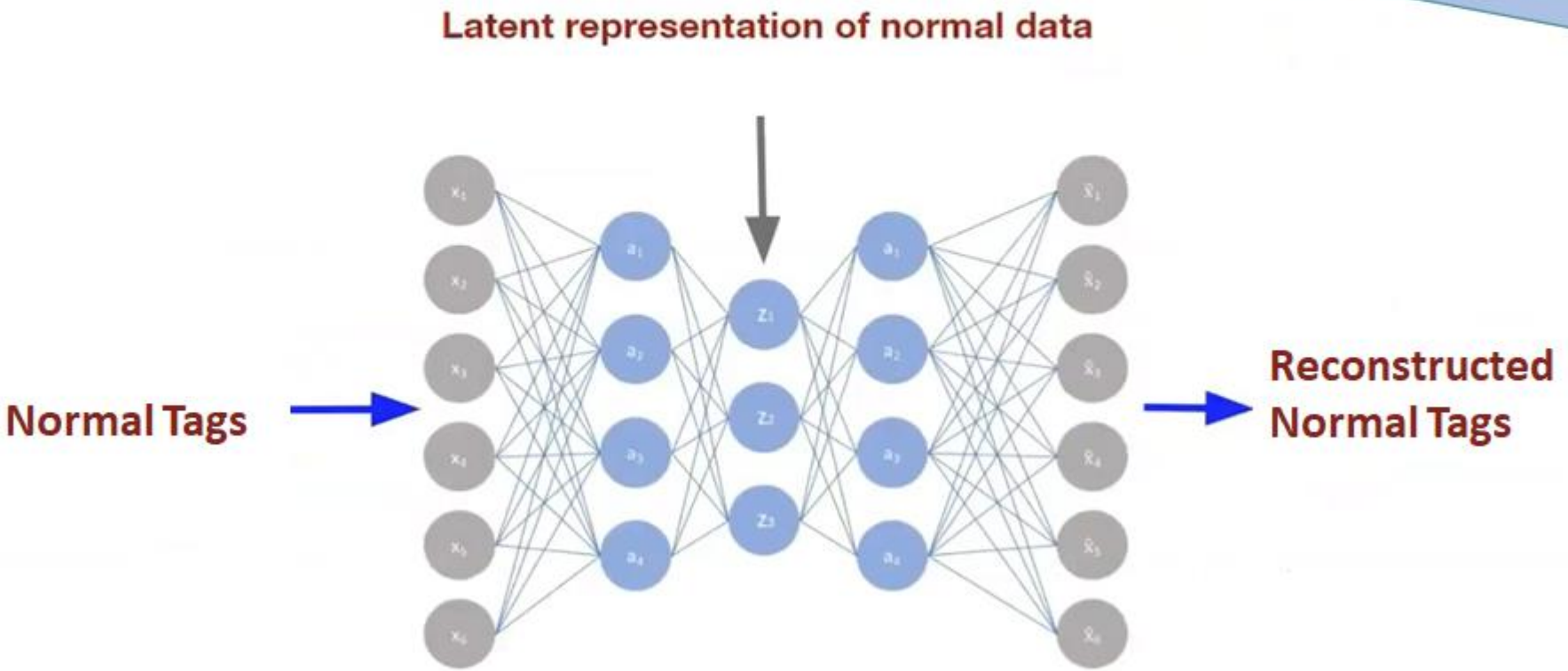
Until it has optimally clustered the data. Notice that it does not always include all the data and that outliers, or anomalies, are identified.



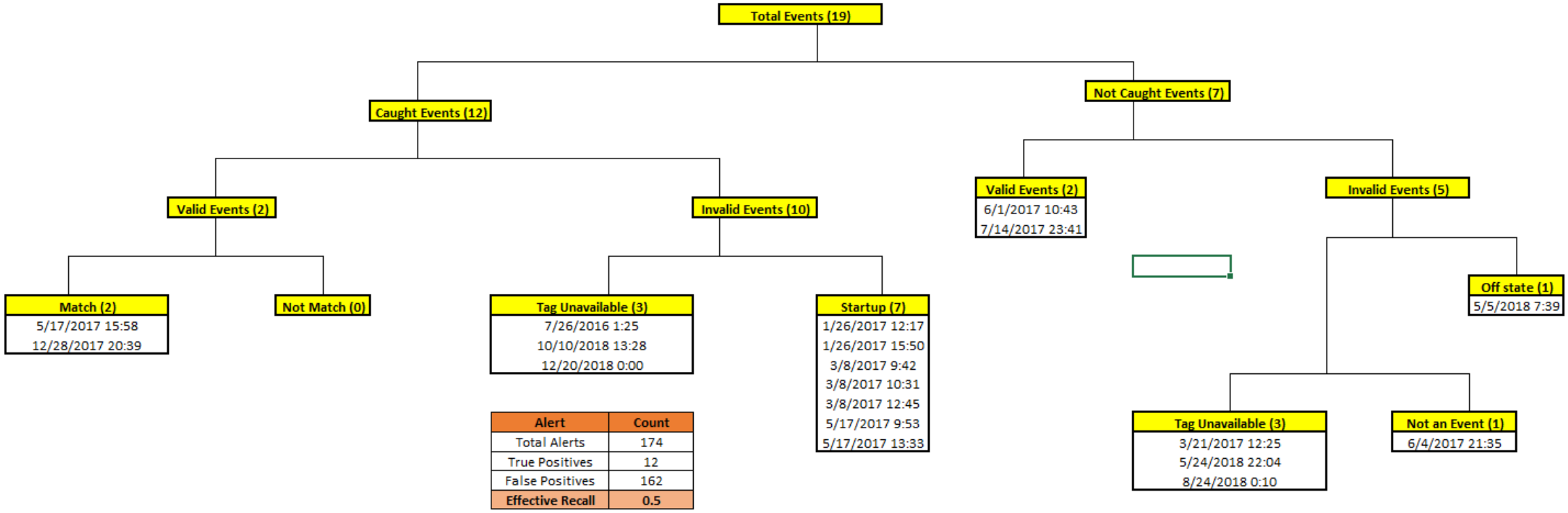
Model Training: Normal Behavior Modeling

An Auto-Encoder is used to reduce the number of nodes needed to describe normal data behavior, but it will be very sensitive to subsequent non-normal changes.

Illustrative Only



Steam Turbine Validation Results: Clustering



Alert	Count
Total Alerts	174
True Positives	12
False Positives	162
Effective Recall	0.5

```
**********************************************************************************************
Hyper - Model Evaluation for Model_feat_selct
**********************************************************************************************
Configuration     Model_feat_selct_thresholding_estimator__alpha...
True Positive                                                    12
False Negative                                                    7
False Positive                                                  162
Total_alerts                                                    174
Name: 13, dtype: object
```

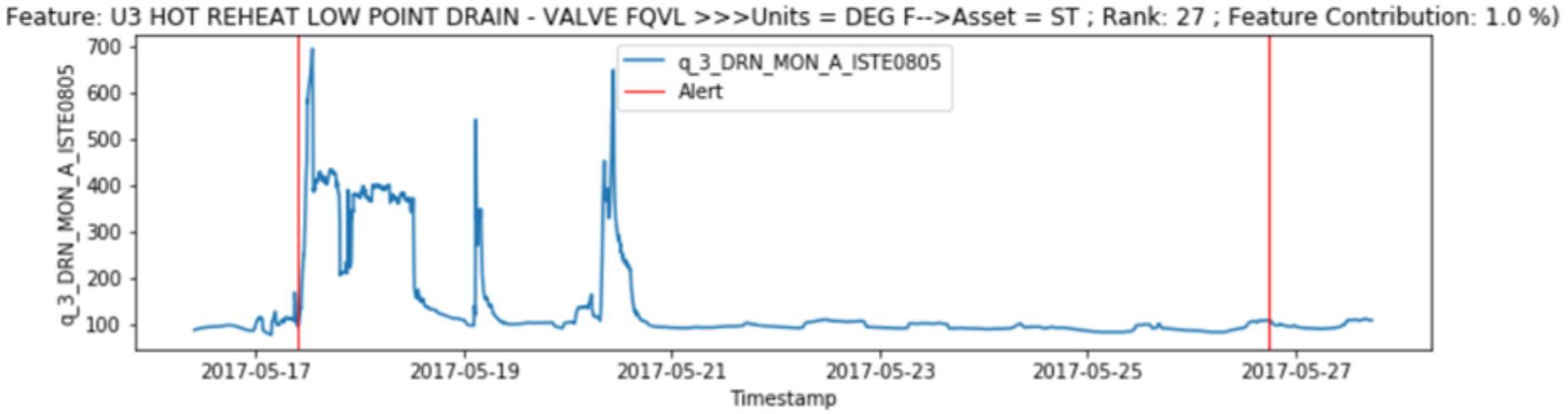
Steam Turbine Event Example

Event Log Entry:

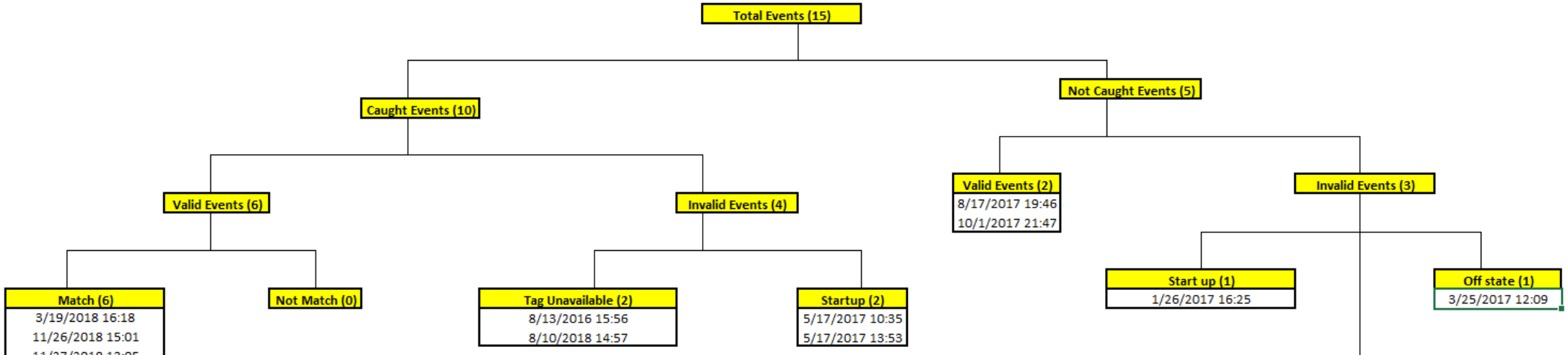
Plant Schererâ€™s model SCH3_UNIT-DRAIN_OPER indicates the following OIS tags for drains that are either increasing in temperature or are at least 200 deg above expected: {3-DRN-MON-A-ISTE0803}, {3-DRN-MON-A-ISTE0804}, {3-DRN-MON-A-ISTE0805}, {3-HOT-REH-WYE-DRN-T}, and {3-HOT-REH-RH-DRN-T}. I sent an email to the plant distribution list. [JET]

Alert Lead Time:

$6^{h}12^{m}$



Generator Validation Results: NBM



Alert	Count
Total Alerts	30
True Positives	10
False Positives	20
Effective Recall	0.75

Best Model

```
print("*****************************************************************************************")
print("Hyper - Model Evaluation for Model_feat_selct")
print("*****************************************************************************************")
df = pd.read_csv('/home/jvaghela/Projects/DoE/Models/STGEN/NBM/Model_feat_selct/STGEN_all_cases.csv')
df[(df['True Positive'] == 10) & (df['Total_alerts'] == df[df['True Positive'] == 10]['Total_alerts'].min())].iloc[0,:]
```

```
*****************************************************************************************
Hyper - Model Evaluation for Model_feat_selct
*****************************************************************************************
Configuration     Model_feat_selct_thresholding_estimator__alpha...
True Positive                                                    10
False Negative                                                    5
False Positive                                                   20
Total_alerts                                                     30
Name: 8, dtype: object
```

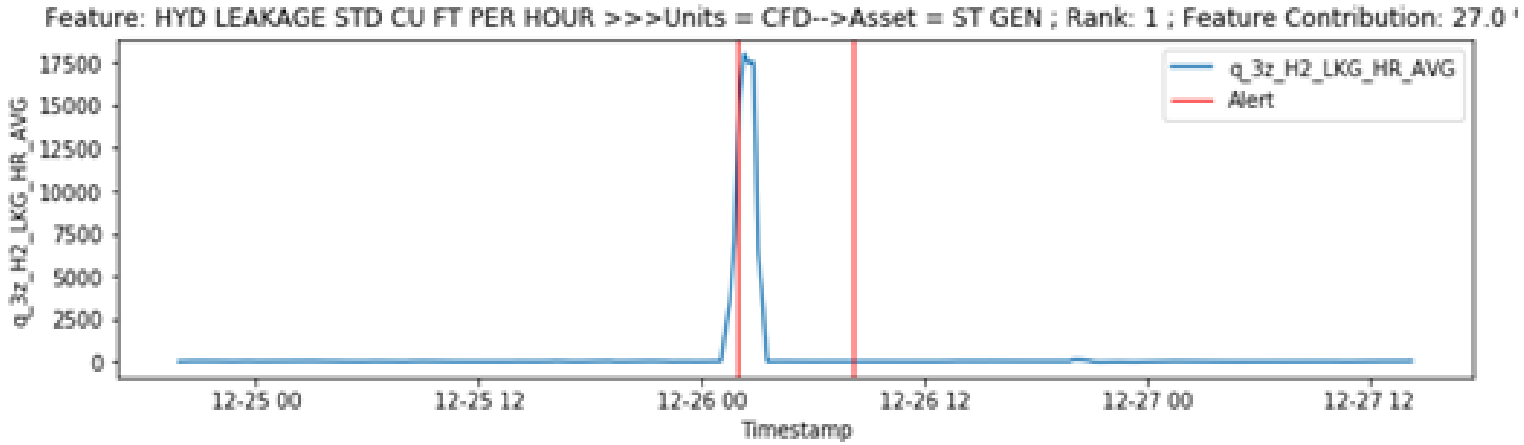
Generator Event Example

Event Log Entry:

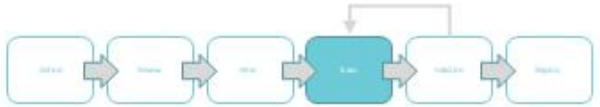
0158 Called Scherer 3 Operator regarding the Generator Hydrogen Hour Leakage (3z-H2-LKG-HR-AVG) increasing to ~ 5000 CFD. Operator informed scavenging Hydrogen to increase the purity. (BWG)

Alert Lead Time:

$2^h 42^m$



Model Output: Alerting



Below shows an example of how the model outputs appear in the latest version of SparkPredict.

An alert is triggered when the risk scores persistently breach the threshold and alerting logic

Illustrative Only

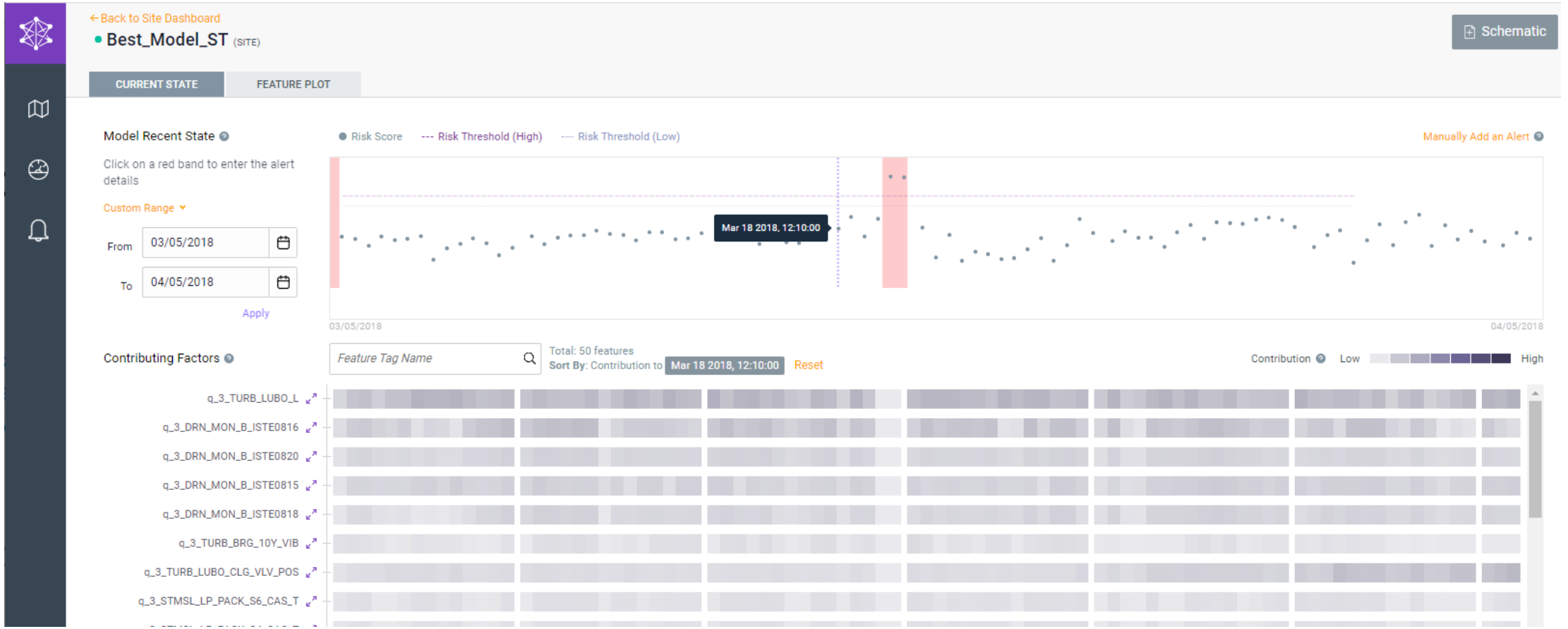
*A threshold is defined and tuned based on distribution of **training** risk scores.*

The model also generates an ordered list of contributina factors

SparkPredict® by SparkCognition

User Interface

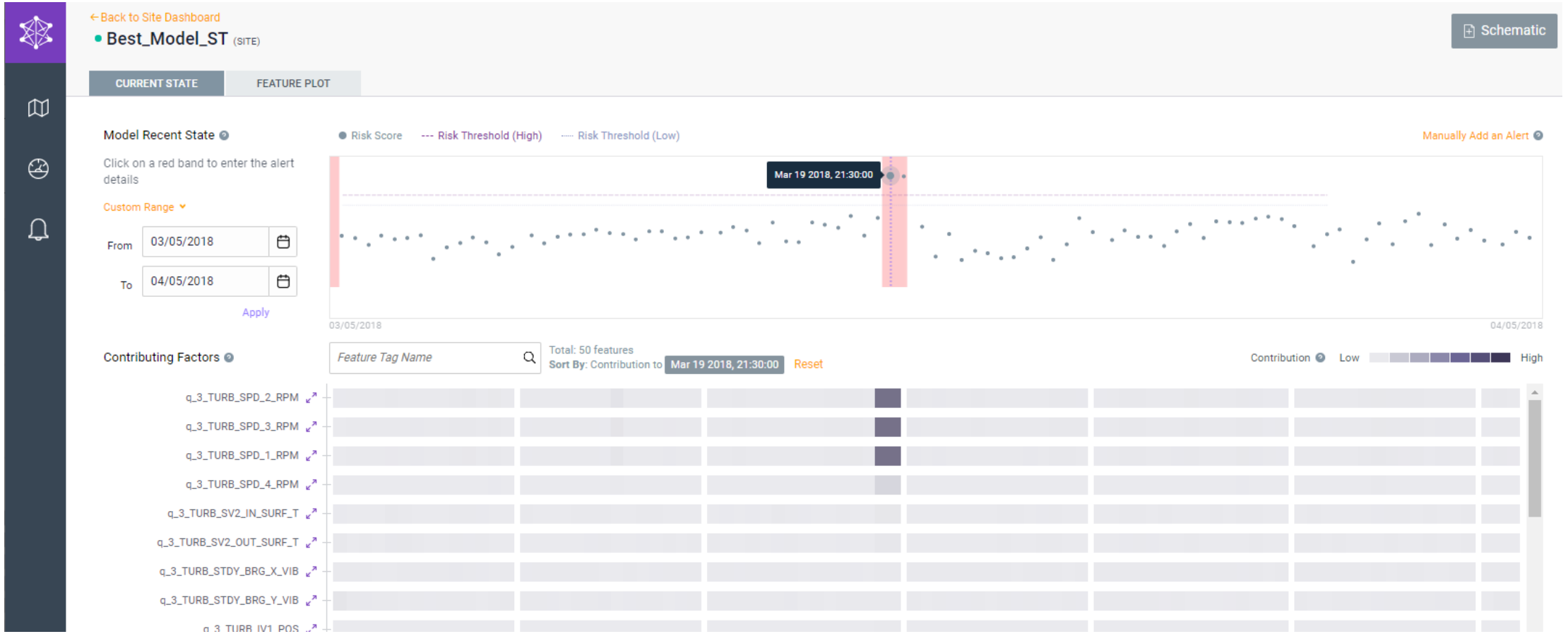
Spectrum plot of contributors to Risk Level ***before*** event



SparkPredict® by SparkCognition

User Interface

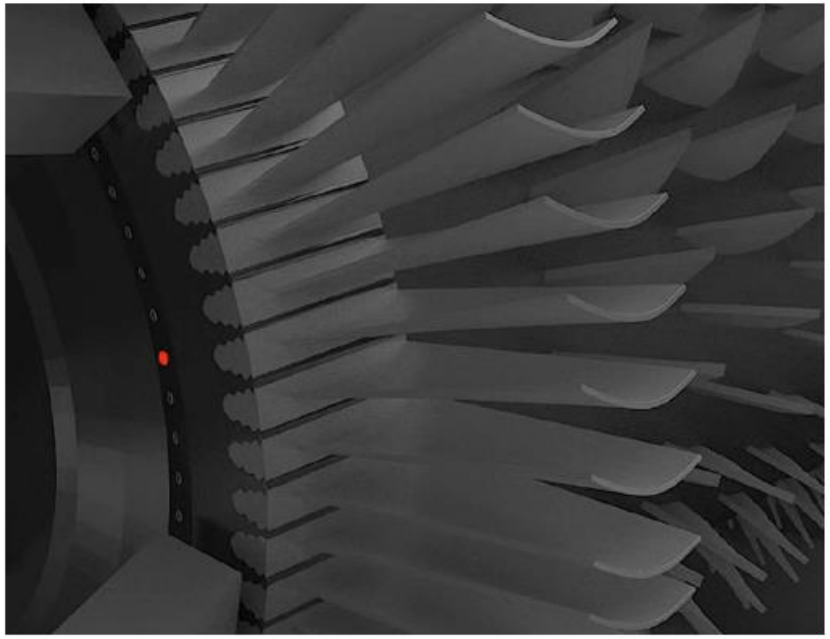
Spectrum plot of contributors to Risk Level ***during*** event



Identified never-before-seen issue

Background

- Unsupervised techniques to understand operational conditions and flag anomalies
- Asset is a GE Steam Turbine put into operation 1/1/1997
- Unit Power: 800 MWatt
- Detected anomaly with one month lead time
- Failure was a manufacturing defect unlikely to occur again
- Correctly pointed to problem area of turbine
- Immanent failure missed by conventional monitoring



Results

With the SparkPredict® product, the utility:

- Safeguarded a turbine investment worth USD $2B
- Predicted a unique failure one month in advance
- Averted roughly $500K in repairs

Questions?