TRANSIENT STATE DETECTION IN MACHINE DATA

JUNE 9, 2018

THE TEAM

Solar® Turbines
A Caterpillar Company

UC San Diego
JACOBS SCHOOL OF ENGINEERING

MEMBERS

Garrett Cheung
Data Engineer

Michael Galarnyk
Bookkeeper

Jared Goldsmith
Record Keeper

Jillian Jarrett
External Team Coordinator

Orysya Stus
Internal Team Coordinator

ADVISORS

Dr. Yoav Freund
University of California, San Diego

Dr. Chad Holcomb
Solar Turbines

SOLAR TURBINES

INSIGHT PLATFORM

THE CHALLENGE

To identify and distinguish transient states in Solar Turbines' packages.

THE MACHINERY

- ‘Packages’ are units comprised of compressors, combustors, turbines and application specific components
- Extremely complex machines with hundreds of moving parts
- Generally used for power generation and compression applications (pipelines)
- Single package has 200-600 features, time series data, events data, alerts data, etc

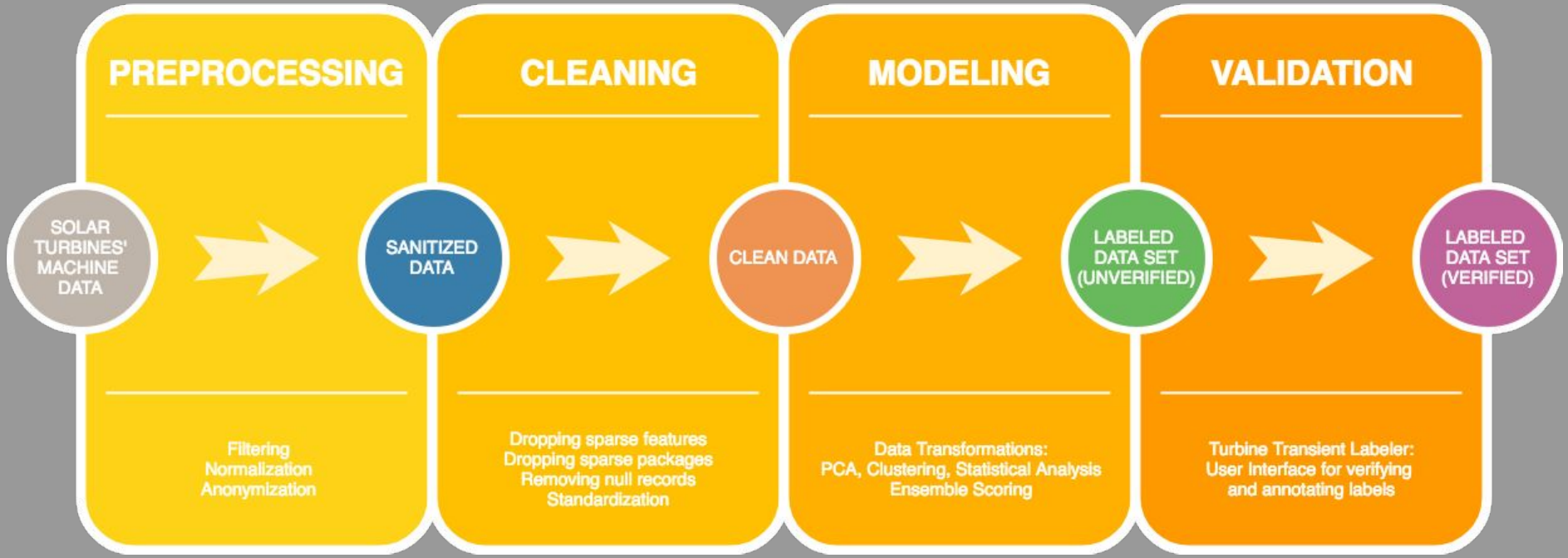
TRANSIENTS DEFINED

- Transients occur when the package is undergoing some sort of transition state
  - Speeding up or slowing down
  - Recalibrating to maintain a steady state
- In the industry, no universal definition exists
- Defined by our domain expert as “any instance where load (power output) changes by more than 25% in a 10 minute timeframe”

BUSINESS VALUE

- Improves Solar Turbines’ Alerting capability. Not all transition states are noteworthy, and proper identification can reduce the number of non-value added alerts reported to customers
- Failure analysis. It is believed that some transients are related to machine failures. To even begin testing this hypothesis, transients must fiTO Arst be identified

PROCESS OVERVIEW



THE DATA

RAW DATA

Data Specifications			
Number of Packages	**Timeframe**	**Data Resolution**	**Features**
72 (total) **Model1: 33** **Model2: 39**	**2 years** Start: 12/5/2015 End: 12/5/2017	**1 hr** and **10 minute** for both models	**Model1: 146 features** **Model2: 77 features** **Sensor readings:** pressures, temperatures, speed, displacement **Package basics:** timestamps, engine hours, package serial numbers, etc **Programmable logic controller (PLC) readings:** signals sent to the machine through the control system

DATA ACQUISITION

Data Preprocessing (Required to move off the Solar network)		
Filtering/Data Consistency	**Normalization**	**Anonymization**
On load data: machine operating close to max output **Gas data:** for dual fuel engines, data points where machine running on liquid fuel were removed **Consistent Column Names:** Only columns which were common across all packages for each model.	Alerting features were normalized by their global limits Other features were normalized by values suggested by domain experts Two features were not normalized: timestamps, and engine starts	**Package serial numbers aliased** **Column names aliased:** measurement types and descriptions are detailed in a data dictionary

DATA CONSIDERATIONS

- Time series analysis requires consistent and continuous data
- Given our time range and data resolution, what's the percentage of data we actually have?

Data Completeness(calculated per psn)	Model1	Model2
90th percentile	89.3%	94.9%
Median	48.6%	82.3%
25th percentile	24.8%	69.2%

- Availability is similar for packages on the high end but Model 2 data has many more higher availability datasets

UNDERSTANDING OUR DATASET

- Working without labels or domain expertise required extensive exploratory data analysis
- Early exploration revealed points in time where many of our features would spike or dip simultaneously
  - These were later verified to be ‘transient’ states
- Used unsupervised learning techniques to find meaning in our dataset, specifically with the goal of identifying transients

THE PIPELINE

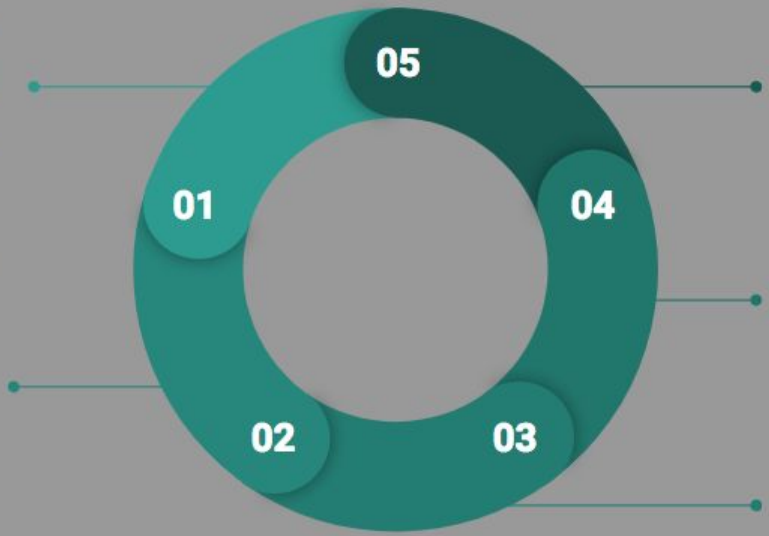
OUR PRODUCT

Raw Data Query

Querying our relational database for a subset of data to be processed in the pipeline.

Data Transformations

Data preprocessing (standardization, cleaning, etc) and transformation (PCA, clustering, statistical methods)



Visual Validation

Javascript dashboard allows domain experts to visualize and validate our model predictions. Dashboard will compare our model's predictions to known labels for a baseline accuracy score.

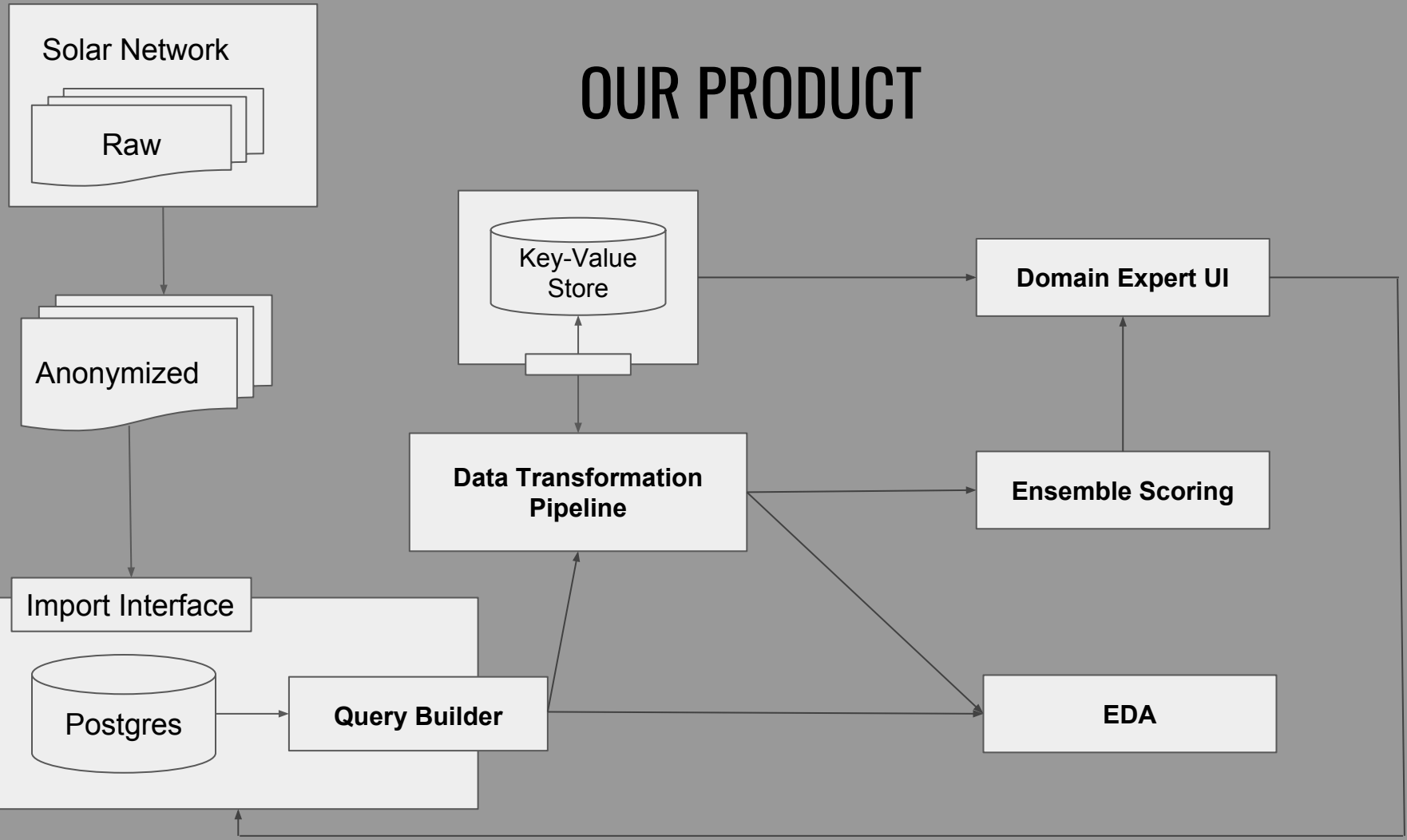
Transient Ensemble Scores

Normalize, combine scores, rank or classify data points are either transient or normal.

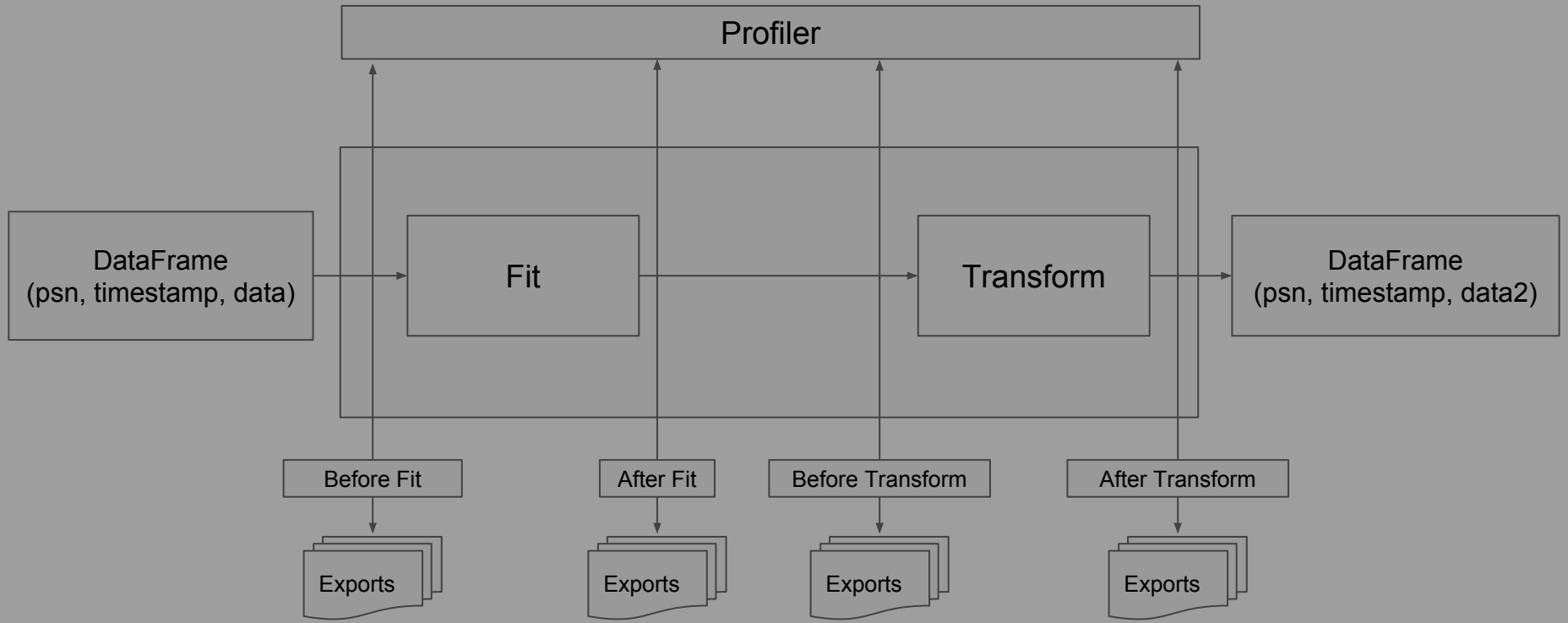
Derived Feature Cache

Allows for scalability and reproducibility of our data pipeline.

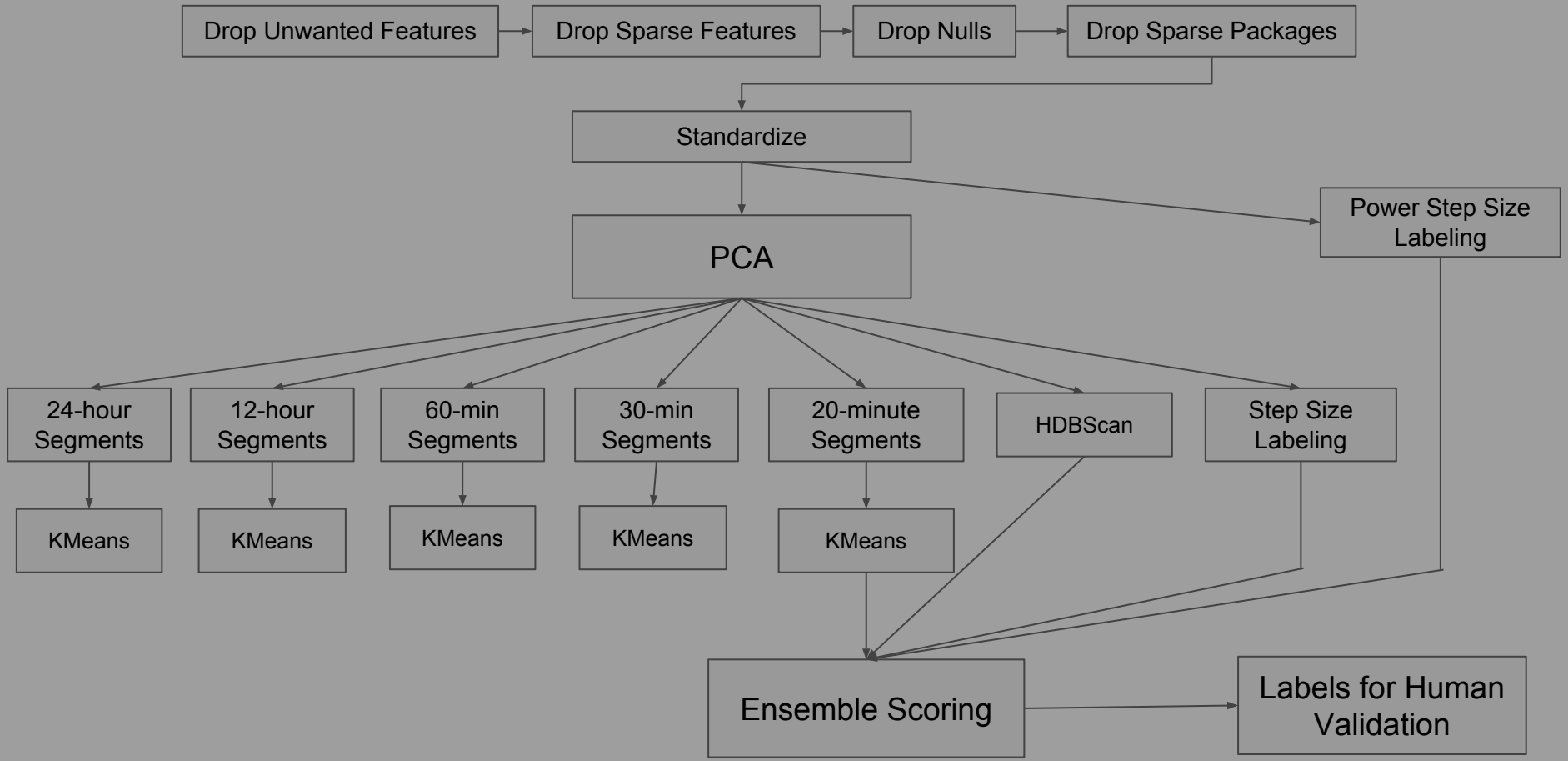
OUR PRODUCT


Solar Network
Raw
Anonymized
Key-Value Store
Domain Expert UI
Data Transformation Pipeline
Ensemble Scoring
Import Interface
Postgres
Query Builder
EDA

TRANSFORMATION



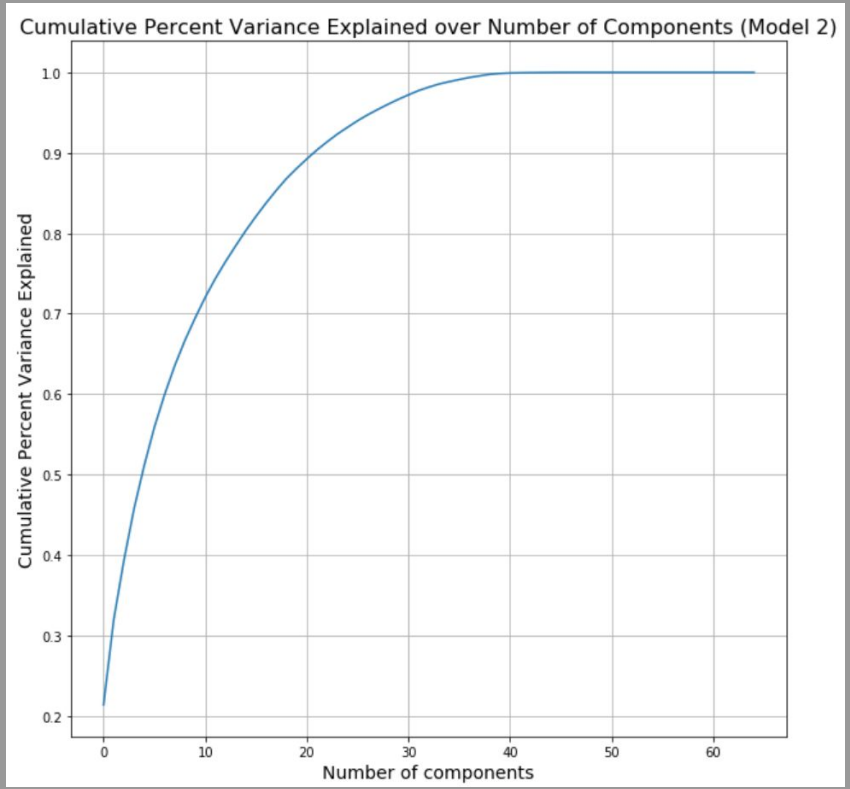
DATA TRANSFORMATION PIPELINE



DATA TRANSFORMATION

PRINCIPAL COMPONENTS ANALYSIS

- PCA by fleet (all Model2 packages)
- 20 principal components retain 88.02% of the dataset’s variance
- Helped us identify important features in our dataset



KEY FINDINGS THROUGH PCA (MODEL 2)

Principal Component	Primary Contributing Features	Percentage of Variance Explained	Cumulative Percentage of Variance Explained
1	**Load (Power output): temperatures, power**	**21.35%**	**21.35%**
2	**Controller signals: command, position**	**10.59%**	**31.94%**
3	**Pressure**	**7.36%**	**39.30%**
4	**Unclear**	**6.45%**	**45.75%**
5	**Temperatures**	**5.37%**	**51.12%**

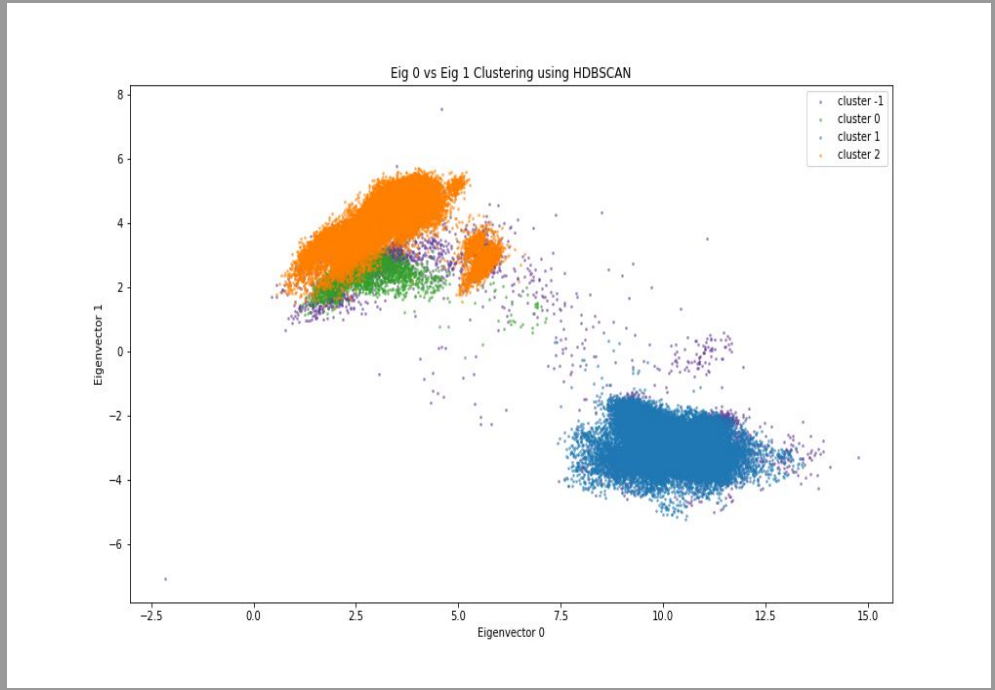
STATISTICAL METHODOLOGIES

Function Name	Inputs	Outputs	Description/Application
StepSize	pd.DataFrame, rolling window length, threshold (z-score)	Boolean pd.DataFrame	Calculates a rolling mean and rolling standard deviation for each column, and returns true for any data points outside n standard deviations (threshold). Applied to transformed features.
PowerJump	pd.DataFrame, column of interest, threshold (%)	Boolean pd.DataFrame	Returns true for all points where column spikes or dips more than the defined threshold. Applied to raw features.
KinkFinder	pd.DataFrame, Threshold (%)	Boolean pd.DataFrame	Returns true for any clusters with (max - min)/ max > threshold. Applied to time split clusters.

CLUSTERING 10 MINUTE DATA - HDBSCAN

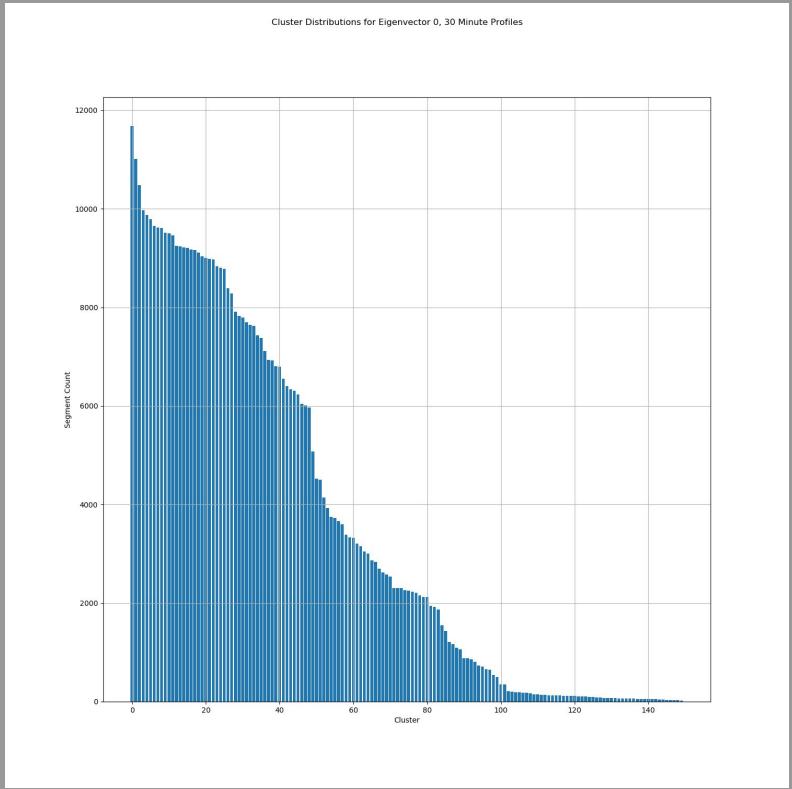
Hierarchical Density-Based Spatial Clustering of Applications with Noise

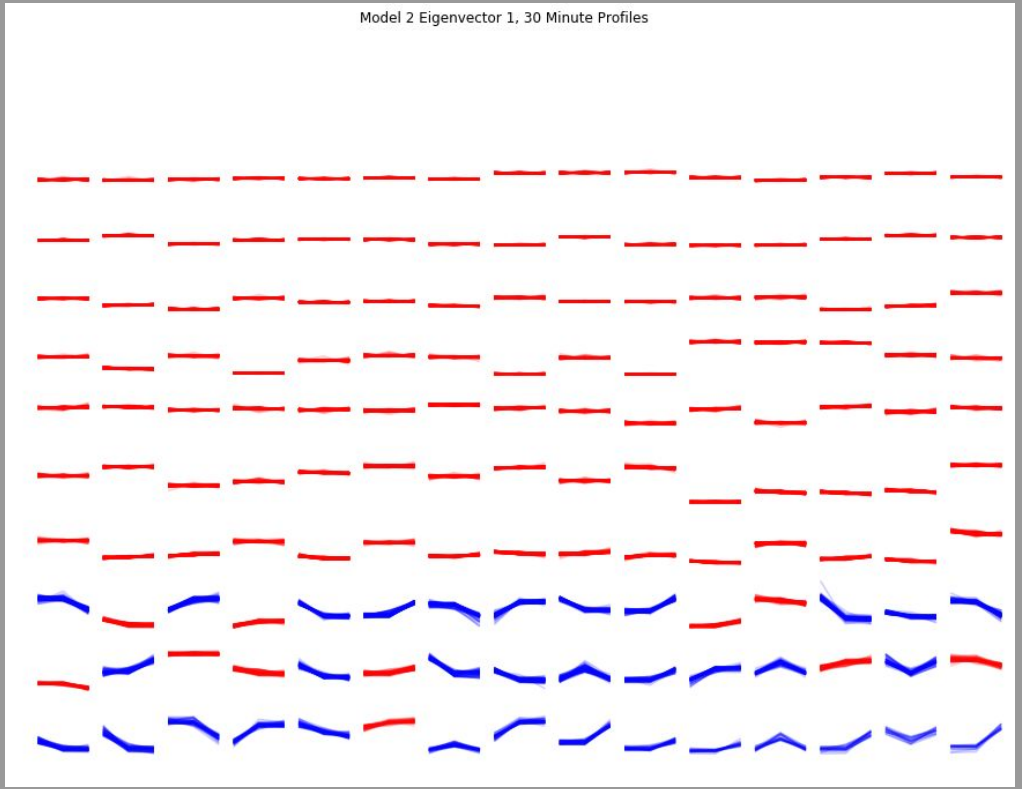
- Similar to DBSCAN but allows for clusters of varying density
- Unsupervised: does not require number of clusters to be specified
- Does not label all points (noise)
- Retains high performance speed on larger datasets



McInnes L, Healy J. *Accelerated Hierarchical Density Based Clustering* In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, pp 33-42. 2017

CLUSTERING N-10 MINUTE SEGMENTS - K-MEANS

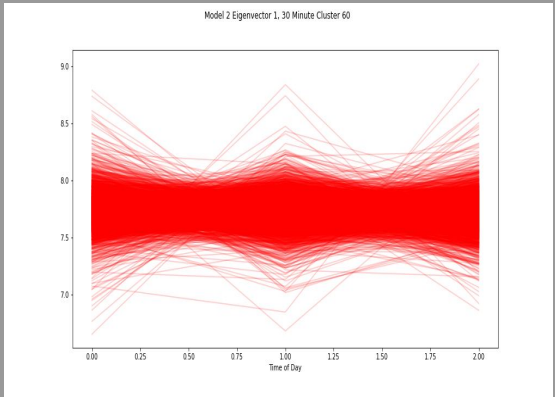


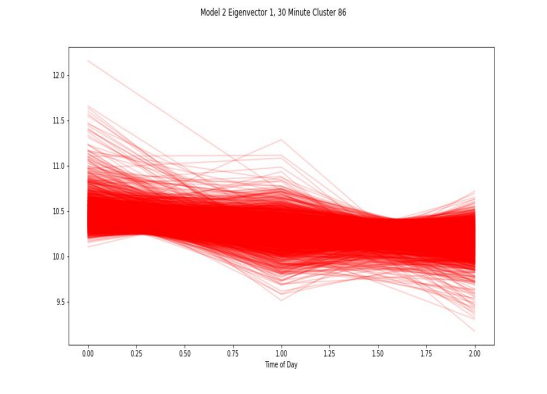


CLUSTER ANALYSIS

- Determine which clusters (30 min segments) and which packages are most similar to one another.
- Cluster maps using euclidean distance as the distance metric was used.
- Repeating analysis for eigenvectors 1-4.

CLUSTER ANALYSIS - RESULTS

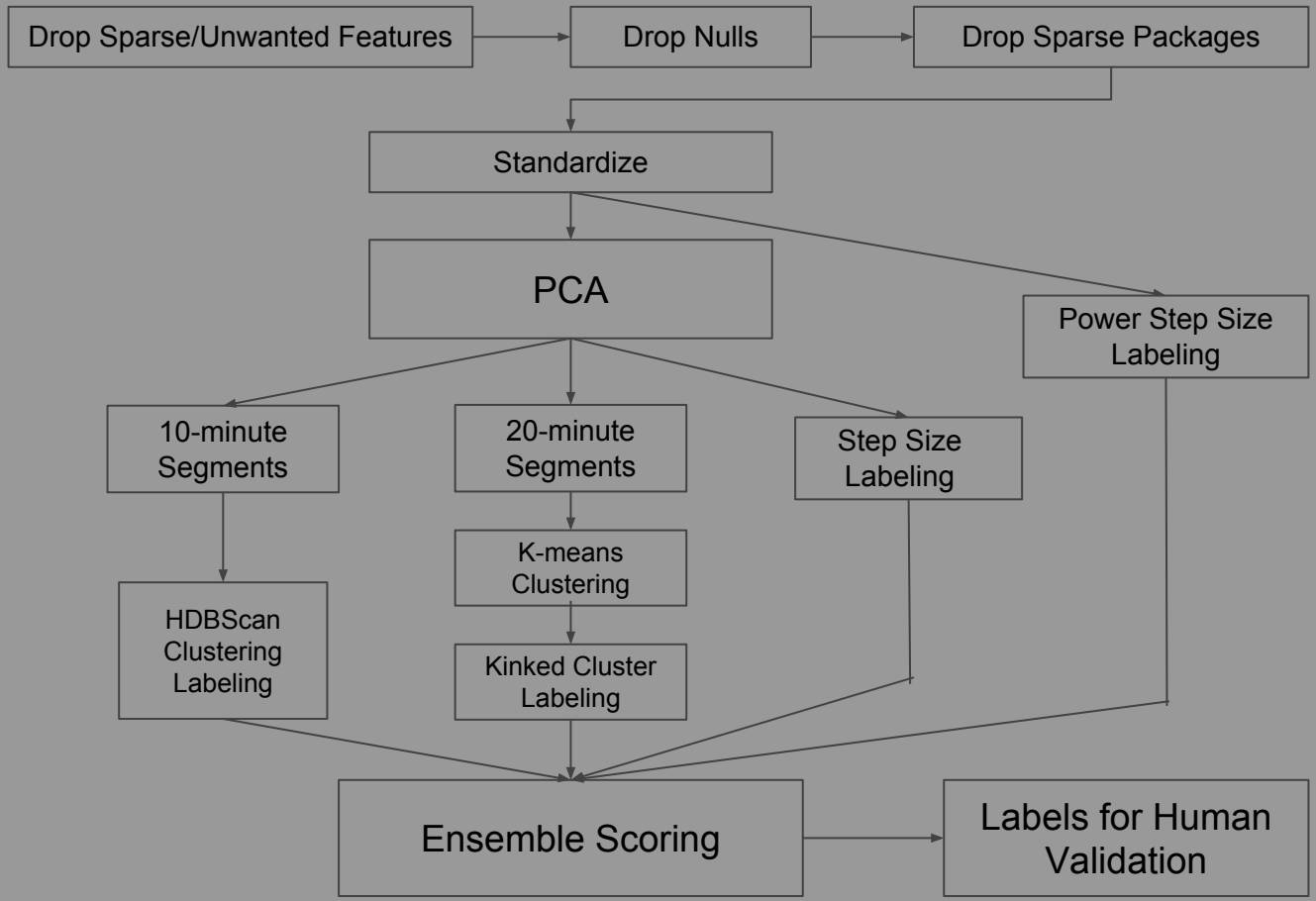




- Clusters close to another another based on euclidean distance: 60, 86
  - The shape of the data is similar.
- Packages close in euclidean distance: 63, 50, 51
  - Might suggest that at the same site, operated by the same customers for the same application.

THE MODEL

COMBINING AND UTILIZING OUR BEST RESULTS



ENSEMBLE SCORING

- Score each of our four detection methods' outputs to be between 0 and 1, where 0=normal and 1=transient
- Combine our scores to give each (psn, timestamp) a TRANSIENT_SCORE
- Defined a threshold, TRANSIENT THRESHOLD, that we use to delineate transients from normal in our ensemble model
  - TRANSIENT_THRESHOLD = 2
- If TRANSIENT SCORE > TRANSIENT THRESHOLD, our model labels that (psn, timestamp) to be a transient occurrence

MODEL FINE TUNING AND VALIDATION

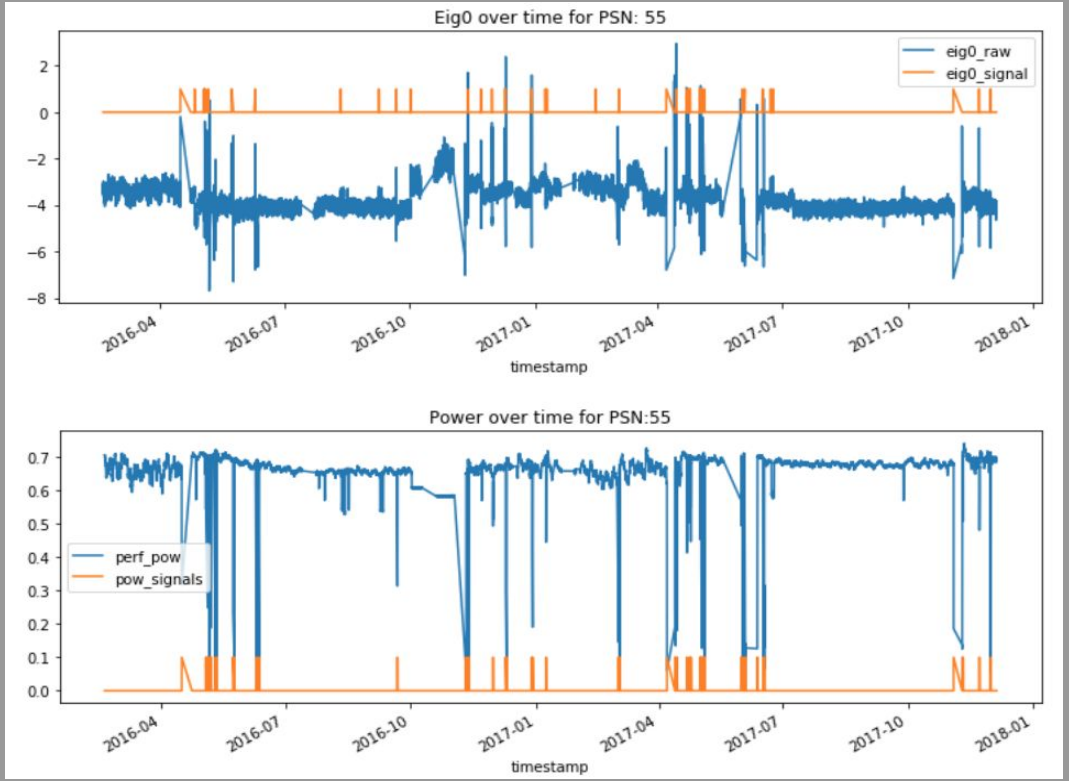
USING DELTA POWER TO IDENTIFY TRANSIENTS

- Power is the amount of energy transferred per unit time
- It is one of our raw features for Model2, and can serve as a baseline for comparison
- Labeled data points with a 25% change in power
- Hard to ascertain which model is “better”
  - Power: simple model, utilizes one feature from the raw data, so it is subject to sensor errors, etc
  - Eigenvectors: more complex, difficult to interpret. Comprised of multiple features and more likely to catch changes that a single feature might miss
- Caveat: Misses emissions mode related transients

MODEL FINE TUNING

TOP PLOT: Eig0 over time. Yellow spikes = transients as detected by our stepsize function (rolling window = 1 day, threshold=5

BOTTOM PLOT: Power over time. Yellow spikes =transients detected using powerjump function (threshold=0.25)



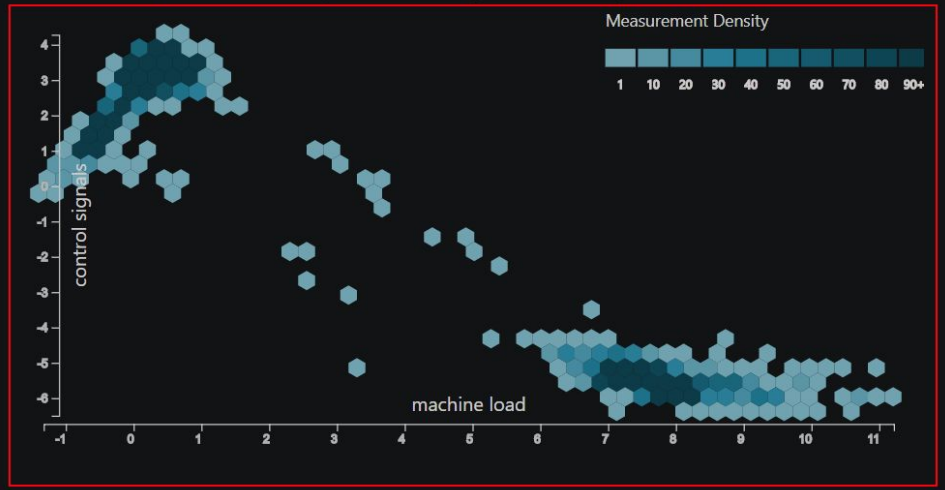
Turbine Transient Labeler

PSN 41 | Label Types ...

15484 data points across 114 days | 0 labeled transients | 0 data points selected | 0 data points verified

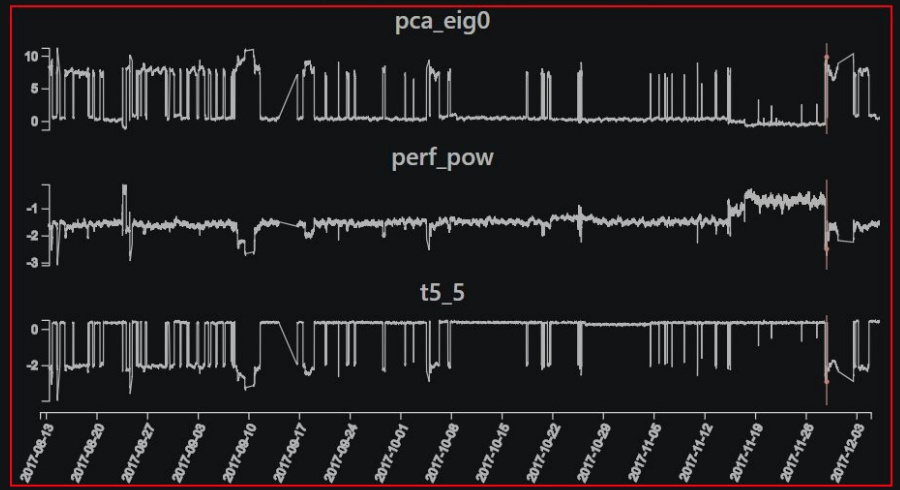
Reduced Space

X-Axis Feature | Y-Axis Feature



Time Series

Composite Features | Machine Tags



Operating Load Profiles

Similar packages: 66 34 64 67 65

44.01%	18.92%	11.41%	10.30%	7.04%	3.54%	2.09%	0.89%	0.46%	0.43%	0.41%	0.25%	0.12%	0.06%	0.05%	0.03%
136,180 minutes	58,540 minutes	35,320 minutes	31,860 minutes	21,780 minutes	10,960 minutes	6,460 minutes	2,740 minutes	1,420 minutes	1,340 minutes	1,280 minutes	780 minutes	380 minutes	200 minutes	140 minutes	80 minutes

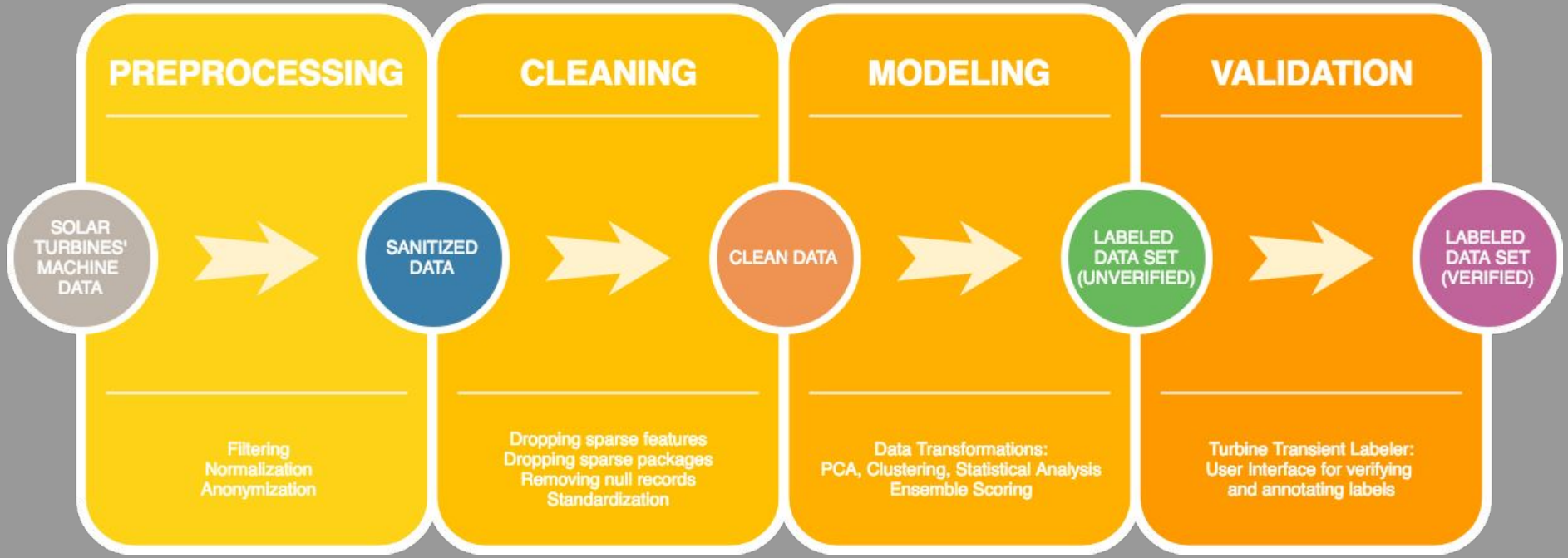
DEMO

FUTURE DIRECTIONS

LONGER TERM GOALS

- DASHBOARD GAMIFICATION
  - Incorporating elements to engage users and encourage their participation
  - High scorers, easter eggs, etc.
- SHUTDOWN ANALYSIS
  - Mapping our transients to unplanned shutdowns and see if there exists a correlation
- TRANSIENT PREDICTION AND SUBTYPING
  - Supervised learning techniques
- INTEGRATION AT SOLAR TURBINES

SUMMARY


PREPROCESSING
CLEANING
MODELING
VALIDATION
SOLAR TURBINES' MACHINE DATA
SANITIZED DATA
CLEAN DATA
LABELED DATA SET (UNVERIFIED)
LABELED DATA SET (VERIFIED)
Filtering
Normalization
Anonymization
Dropping sparse features
Dropping sparse packages
Removing null records
Standardization
Data Transformations:
PCA, Clustering, Statistical Analysis
Ensemble Scoring
Turbine Transient Labeler:
User Interface for verifying
and annotating labels

THANK YOU