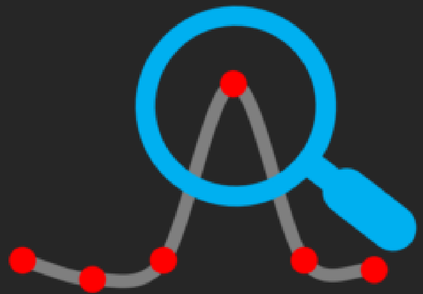
Trimming outliers using trees:
Winning solution of the Large-scale Energy Anomaly Detection (LEAD) competition



Chun Fu (PhD candidate in NUS),
Pandarasamy Arjunan, and
Clayton Miller

NUS
National University of Singapore

Large-scale Energy Anomaly Detection (LEAD) Competition

Figure 2: The user interface of our web-based anomaly annotation tool for energy time series.

- Based on the energy data set used in the ASHRAE - Great Energy Predictor III (2000+ power meters)

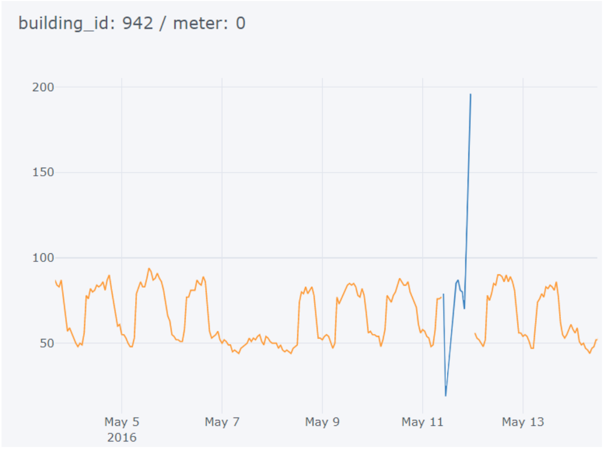
→Annotated with two types of anomalies:

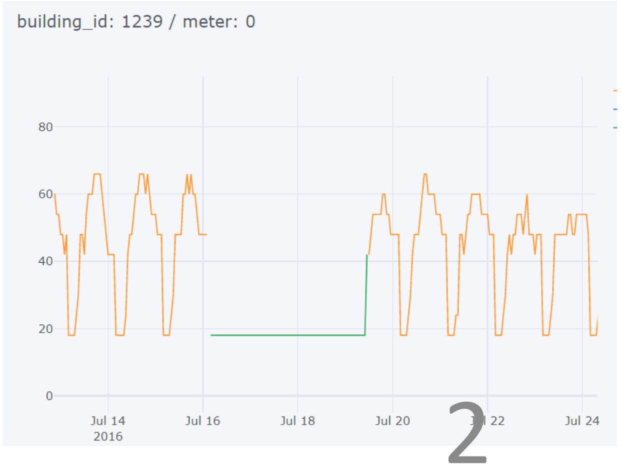
(1) Point anomalies:

(2) Sequential or collective anomalies

- Train dataset: 200 buildings throughout the entire year, with labels of either abnormal (1) or normal (0) usage
- Test dataset: 206 buildings without labels

→Participants were required to predict labels in energy time series

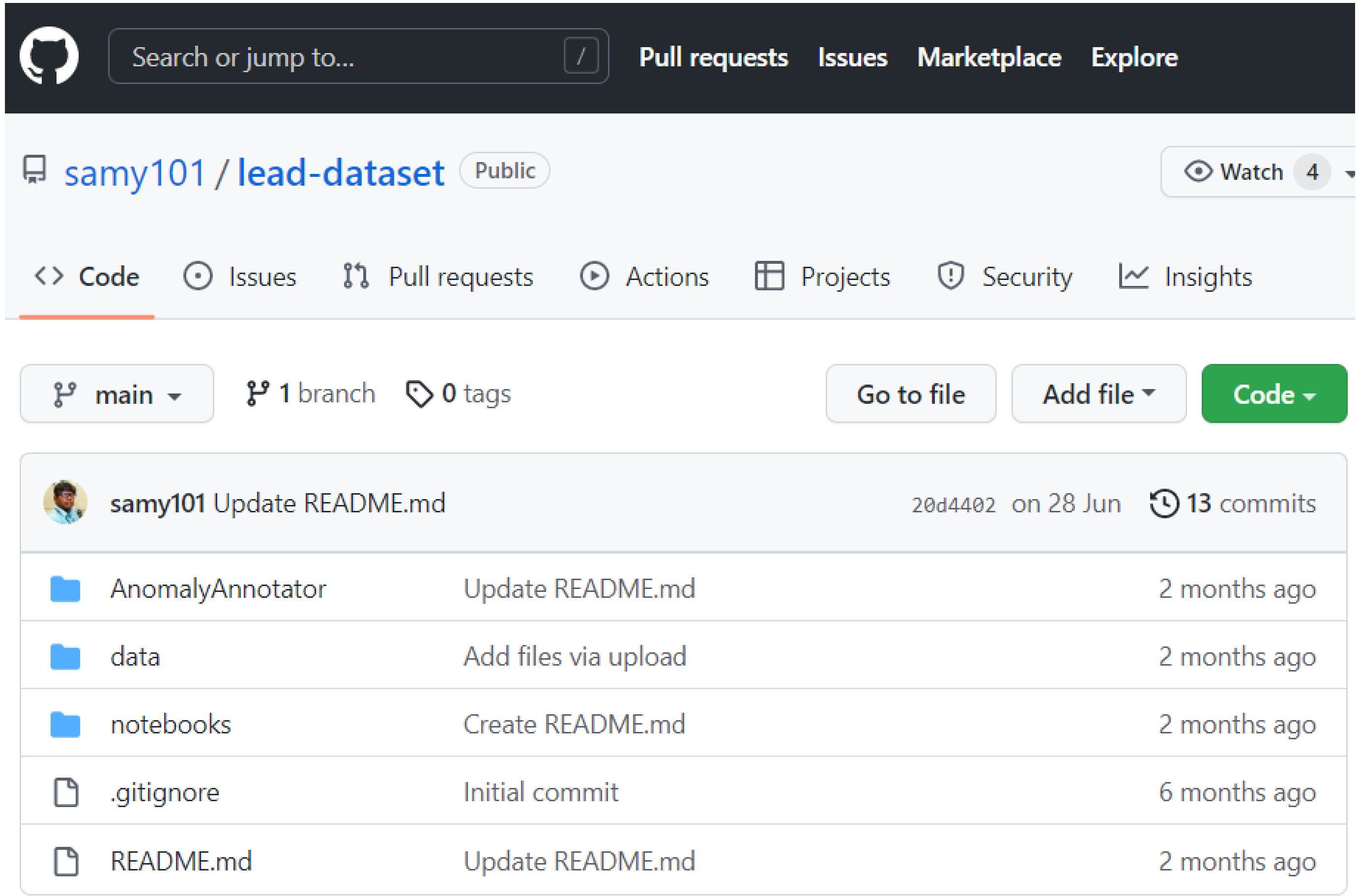




LEAD dataset

Github:

https://github.com/samy101/lead-dataset

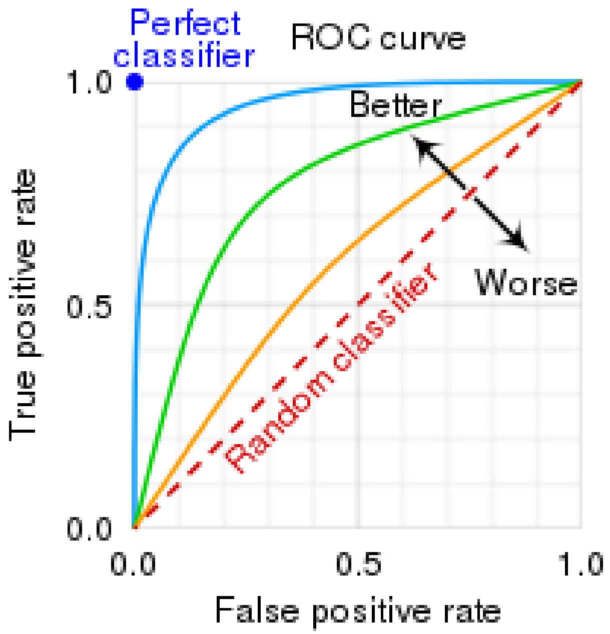


Paper of the dataset:

https://arxiv.org/abs/2203.17256

Evaluation metric used in the competition

AUC-ROC score
= The area under ROC Curve

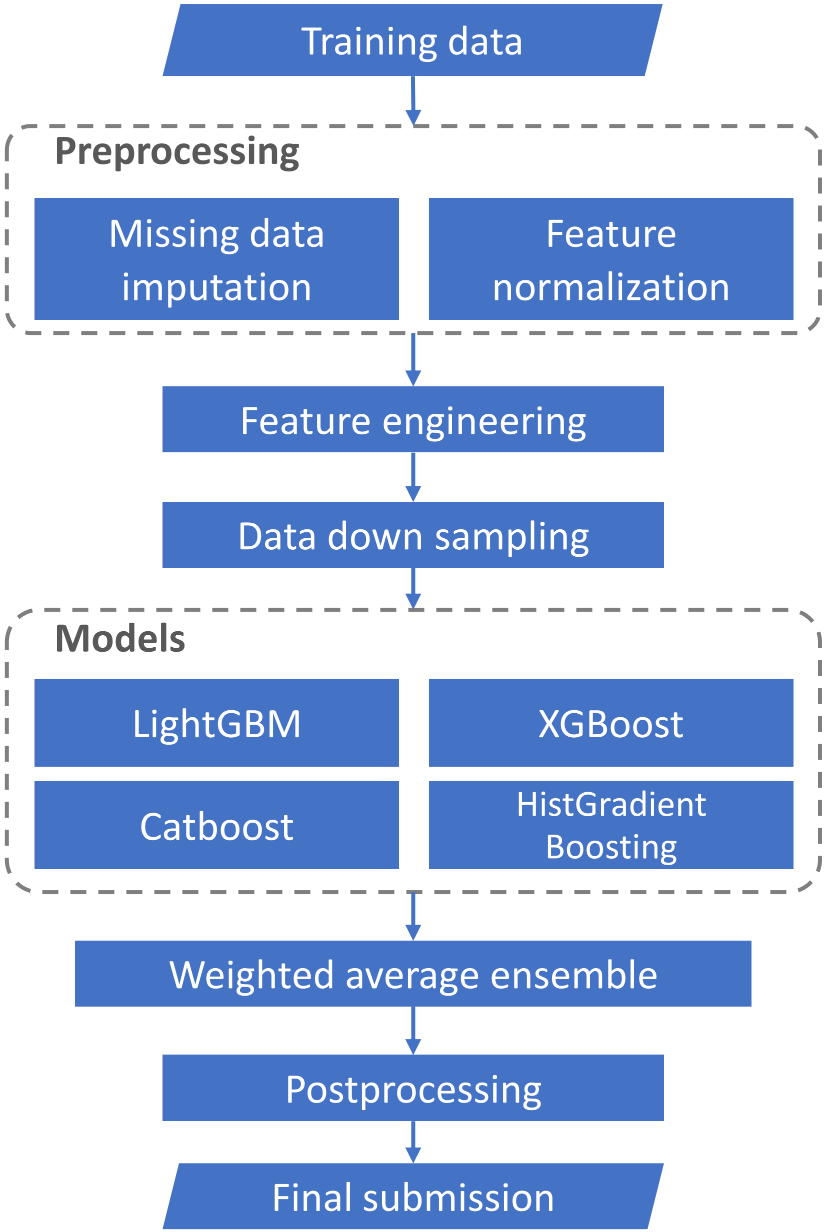


(Ref: https://en.wikipedia.org/wiki/Receiver_operating_characteristic)

AUC values	Test quality
0.9–1.0	Excellent
0.8–0.9	Very good
0.7–0.8	Good
0.6–0.7	Satisfactory
0.5–0.6	Unsatisfactory

Overview of the winning solution

- **Data preprocessing**
  - Missing values (NaN) were replaced with the median value of each time series
- **Feature engineering**
  - Building meta data and weather data
  - Temporal features (e.g., hour, weekday, and day of year)
  - Target encoding features
  - Value-change features
- **Modeling**
  - Train/valid split by *building_id* to ensure the valid data were unseen during training
  - Downsampling training dataset to solve data imbalance (~5% of anomalies)
  - Model ensembling via simple averaging: XGBosst, LightGBM, CatBoost, and HistGradientBoosting (weight of 0.25 for each)
- **Postprocessing**
  - Set zeros to rows with 1.0 of meter_reading
  - Set zeros to start and end points of time series

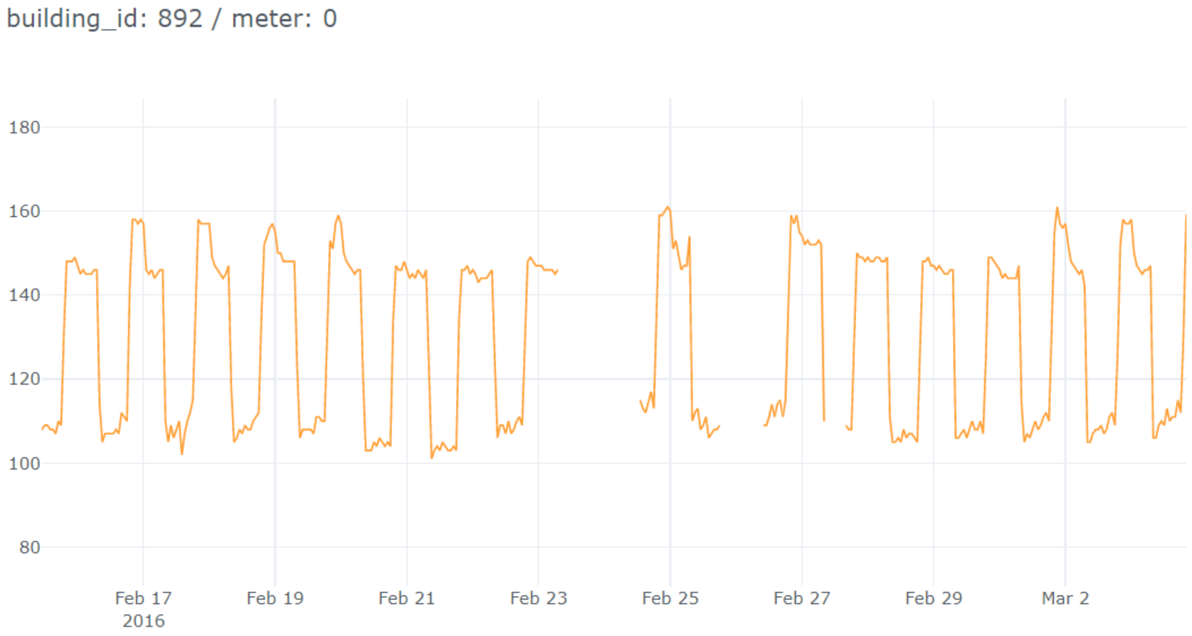


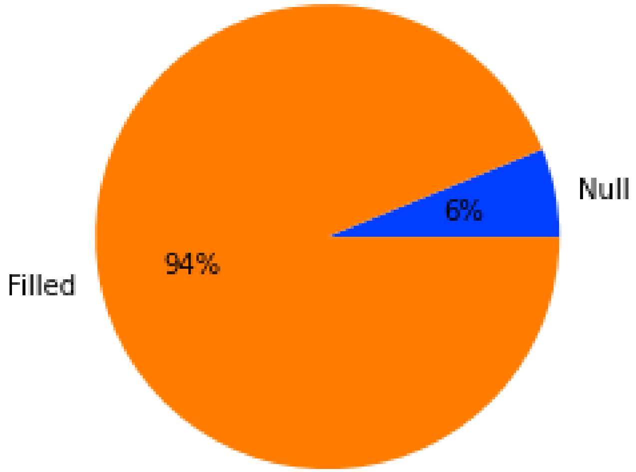
Data preprocessing

- About 6% of the values are null value in *meter_reading* column

→Missing values (NaN) were replaced with the median value of each time series

- No anomalies were removed because the goal of this competition is anomaly detection





Feature engineering

- Original features of provided dataset: 57 features
  - Building meta data and weather data
  - Temporal features (e.g., hour, weekday, and day of year)
  - Target encoding features (created by winning team in GEPIII)
- These features are created for building energy prediction task

→Might not be suitable to the task of anomaly detection

- To strengthen the identification of anomaly detection…

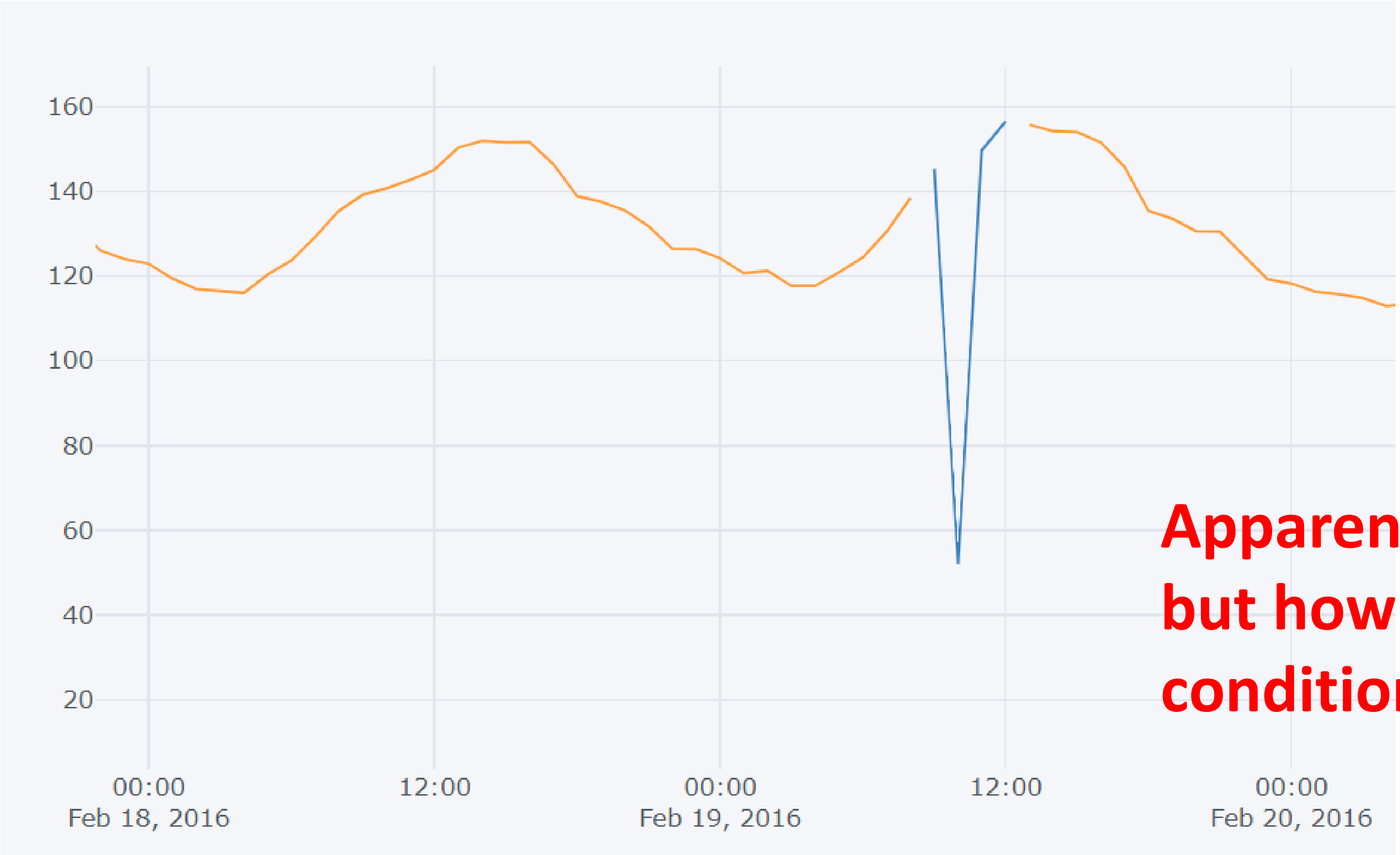
→ We need new features!

Original features

Category	Descriptions of features
Energy use	Meter readings from power meters.
Building meta	Basic information of buildings. (e.g., site_id, building_id, primary_use, square_feet, year_built, and floor_count)
Weather data	Onsite measurements of weather conditions. (e.g., air_temperature, cloud_coverage, dew_temperature, precip_depth_1_hr, sea_level_pressure, wind_direction, and wind_speed)
Temporal feature	Derived features from timestamps. (e.g., hour, weekday, and day of year)
Target encoding feature	Average values of the target variable aggregated by category (e.g., average values grouped by building_id)
Value-change feature	Changes of time-series values in the form of difference or ratio (e.g., the increase or decrease of value compared to previous hour)

Table 1: Features for developing anomaly classification model

Feature engineering



Apparently, there is an anomaly, but how to quantify the condition and create a feature?

Feature engineering

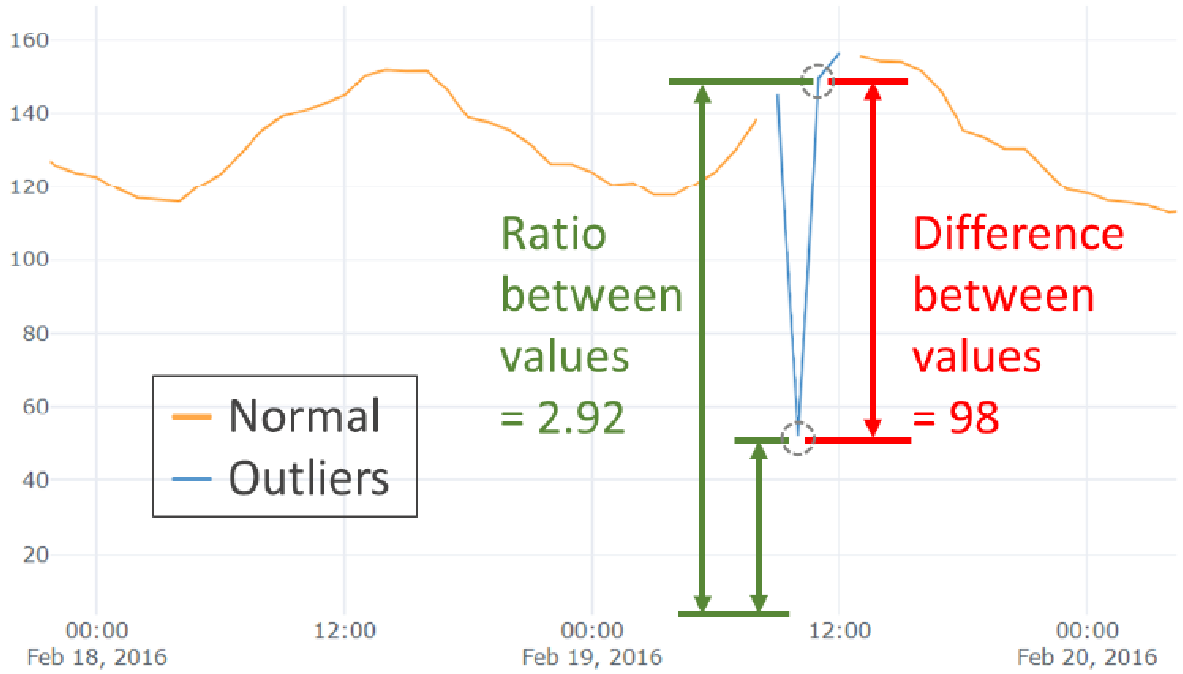


Figure 2: Illustration of calculating value-change features: (1) Value change in difference (red) and (2) Value change in ratio (green)

$$\boxed{\text{Value change in difference} = X(t) - X(t-s)}$$

$$\boxed{\text{Value change in ratio} = \frac{X(t)+1}{X(t-s)+1}}$$

To avoid zeros in denominator

- t = timestamp
- s = shift of timesteps

How feature engineering affects classification result

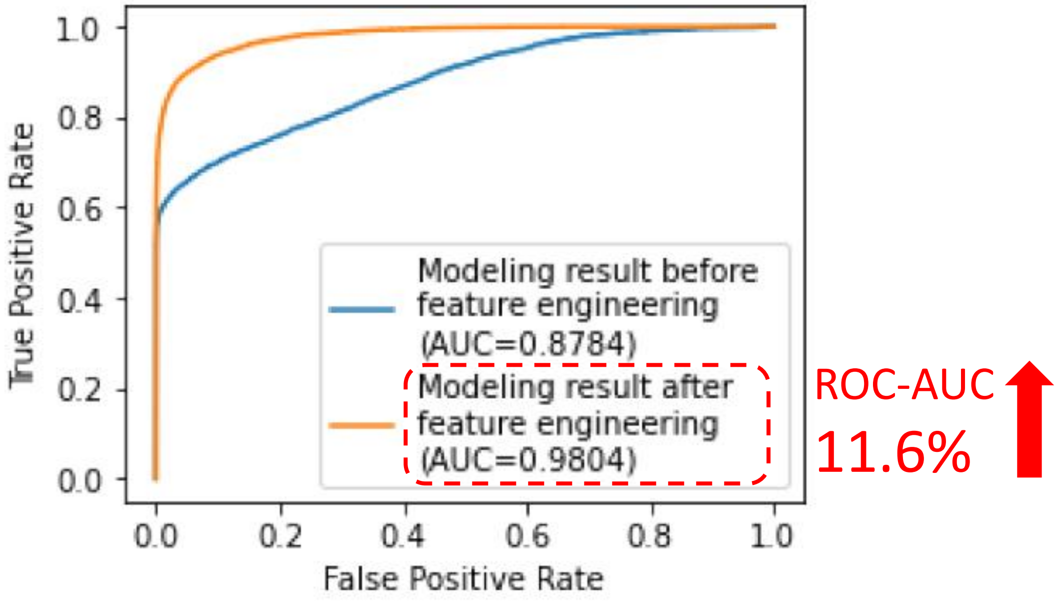


Figure 4: ROC curve and AUC-score before and after feature engineering

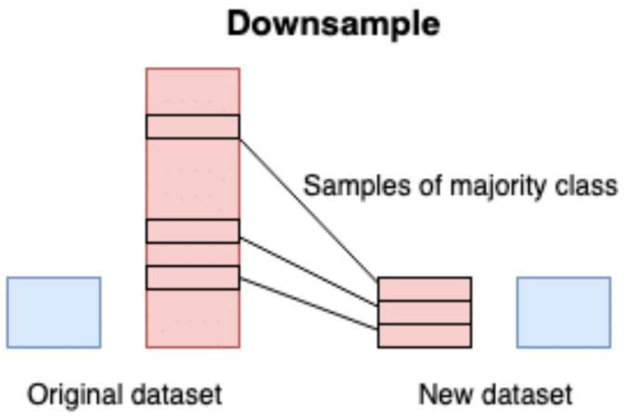
	feature	category	importance
0	building_id	Building meta	219
1	value_chg_ratio_1	Value-change feat.	144
2	value_chg_ratio_-1	Value-change feat.	127
3	meter_reading	Energy use	98
4	dayofyear	Temporal feat.	95
5	square_feet	Building meta	85
6	gte_building_id	Target encoding	63
7	value_chg_ratio_-168	Value-change feat.	61
8	value_chg_ratio_2	Value-change feat.	54
9	gte_meter_primary_use	Target encoding	54

Figure 5: Feature importance of the 10 most influential features exported by LightGBM

Modeling

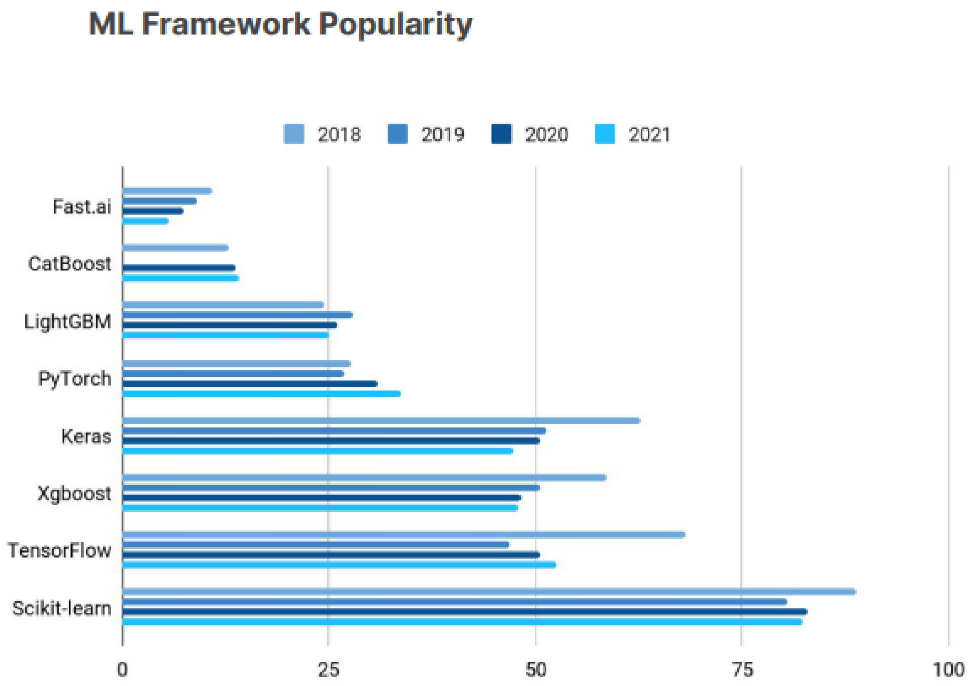
- Data splitting method
  - Train/validation was split by *building_id* to ensure the valid data were unseen during training
  - Use validation dataset to evaluate modeling strategies
- Data downsampling
  - Data imbalance:
    ~5% of abnormal data
  - Random sampling of normal data to make proportions of two labels equal

	Number of power meters
Train	160
Validation	40
Test	206



Modeling

- Tree-based classification models
  - The most popular and powerful tool for classification problems with tabular data
  - Among many tree-based models, few popular ones were chosen: LightGBM, XGBoost, Catboost and HistGradientBoosting
- Model ensembling



(Kaggle's State of Machine Learning and Data Science 2021)

	AUC-ROC score	
	Train	Test
LightGBM	0.9981	0.9804
XGBoost	1.0000	0.9809
Catboost	0.9999	0.9798
Hist Gradient Boosting	0.9975	0.9804
Weighted average ensemble	**0.9998**	**0.9828**

Table 2: AUC-ROC scores of tree-based models and ensemble model

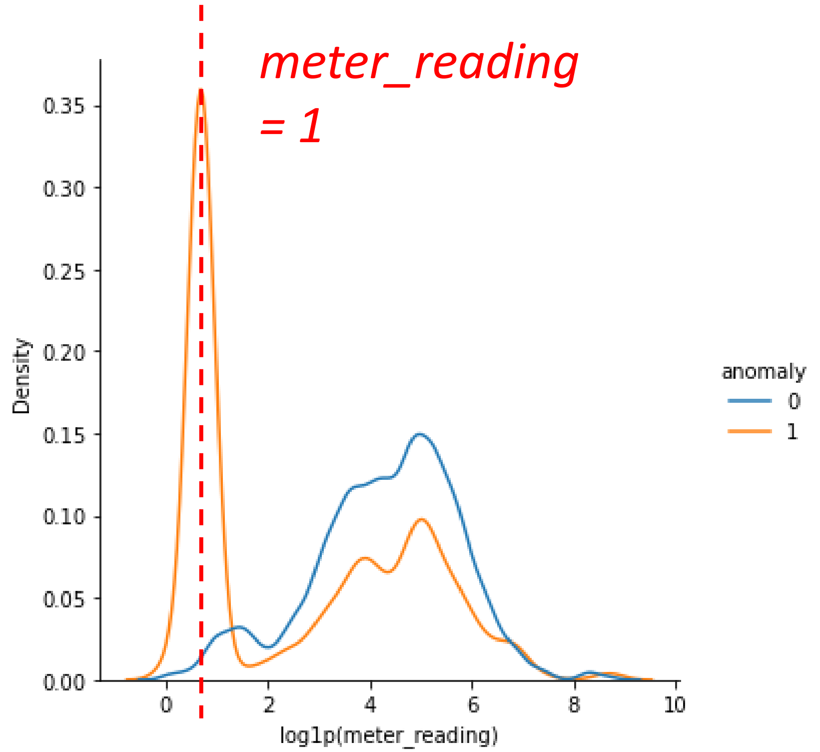
Post-processing

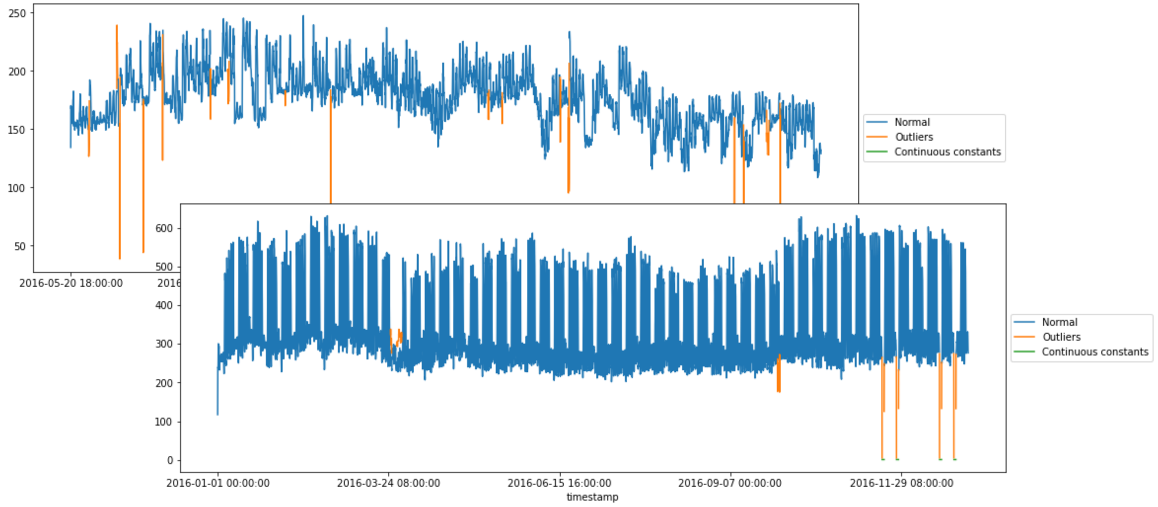
- Nearly 100% of the points with *meter_reading* equal to one are anomalies

→Set prediction to 1 (abnormal) for rows with *meter_reading* value of 1

- Also, most energy time series start and end without anomalies

→Set prediction to 0 (normal) for start and end points of time series





Overview of public solutions in competition

Table 3: List of publicly available shared solutions and their modeling strategies

Team / Author	Public score	Private score	Preprocessing techninques	Features (count)	Modeling strategies
Proposed	0.9734	0.9866	Normalization, imputation, and downsampling	Raw, V-C (169)	Ensemble: LightGBM, XGBoost, CatBoost, Hist Gradient Boosting
Abhishek Maurya	0.8794	0.9237	Normalization, imputation, and downsampling	Raw (31)	XGBoost
Abdallah El-Sawy	0.7633	0.8189	Imputation	Raw (10)	Ensemble: KNN, DT, ET
FabioDalForno	0.7275	0.7566	Normalization, imputation	Raw, V-C (6)	Random Forest
Yoda	0.7105	0.7433	-	Raw (33)	XGBoost
shafiullah	0.6022	0.6242	Imputation	Raw (19)	XGBoost

Raw = Features from raw dataset; V-C = Value-change features

Conclusion

- Value-change features are effective in capturing context in time series!
  - Beneficial for the task of detecting anomalies
  - Especially for tree-based models, which are unable to extract features
- Benchmark of supervised learning in anomaly detection of energy data
  - As the first anomaly detection competition for a large number of power meters

  →Benchmark for future research in anomaly detection of energy data!
  - The AUC-ROC score of 0.9866 in anomaly detection has established a fairly high classification performance benchmark in field of building energy, especially it's trained on only 200 power meters

Future work

- Labeling rate v.s. classification performance:
  - How many labeled data are required for training a good-performance anomaly classification model (e.g., 0.95 of AUC-ROC score)?
  - If the number of power meters used to train the model changes, at what point does the model's performance plateau?
- Generalizability across sites/countries:
  - Could classification model trained on labeled energy data from one site well predict anomalies at another unseen site?

Thanks for your attention.

- Email: patrickfu0302@gmail.com