Growing Team Synergy for Analyzing IoT Streams for Risk Assessment

Nalini Ravishanker, University of Connecticut

Leading Women in Business and Industrial Statistics Webinar Series, March 23, 2021.

TEAM

Patrick Toman[†], Ahmed Soliman[†],
Nalini Ravishanker[†], Sanguthevar Rajasekaran[†]
Nathan Lally[*], Hunter D'Addeo[*]

[†]University of Connecticut, Storrs, CT, USA
[*]The Hartford Steam Boiler, Hartford, CT, USA


Our Team

Patrick Toman†  Nalini Ravishanker† 

Ahmed Soliman†  Sanguthevar Rajasekaran† 

Nathan Lally*  Hunter D'Addeo* 

†University of Connecticut, Storrs, CT, USA
*The Hartford Steam Boiler, Hartford, CT, USA

Outline

- Internet of Things (IoT) at HSB - Problem and Data
- Activity 1: A Deep Dive into Data Quality.
- Activity 2: Clustering and Classifying Aberrant Time Streams
- Activity 3: Anomaly Detection for Alerting Clients

Internet of Things (IoT) at HSB - Understanding the Problem and Data

What is IoT?

Networks of physical objects (buildings, vehicles, appliances) embedded with sensors and software.

Purpose: Connect with other devices and networks via the internet.

IoT and Insurance

There is growing interest among insurers to utilize IoT.

- Use vehicle sensors to classify low risk versus high risk drivers
- Use customer data to develop differential insurance pricing models
- Use real time monitoring/alert systems to reduce costly adverse events

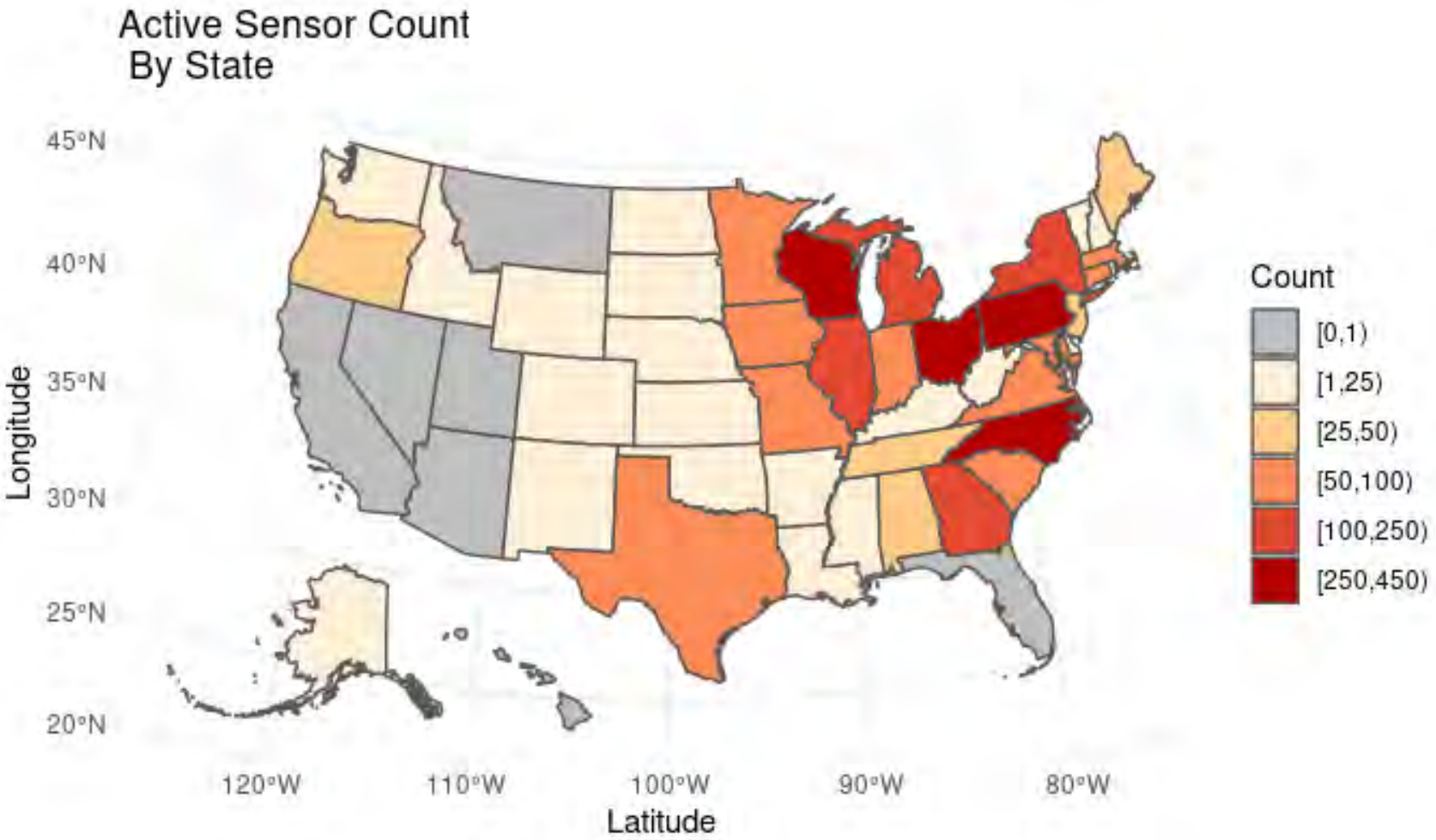
HSB IoT Freeze Loss Program

HSB leverages IoT technology to install temperature sensors in rooms with a high risk of water pipe burst (freeze loss).

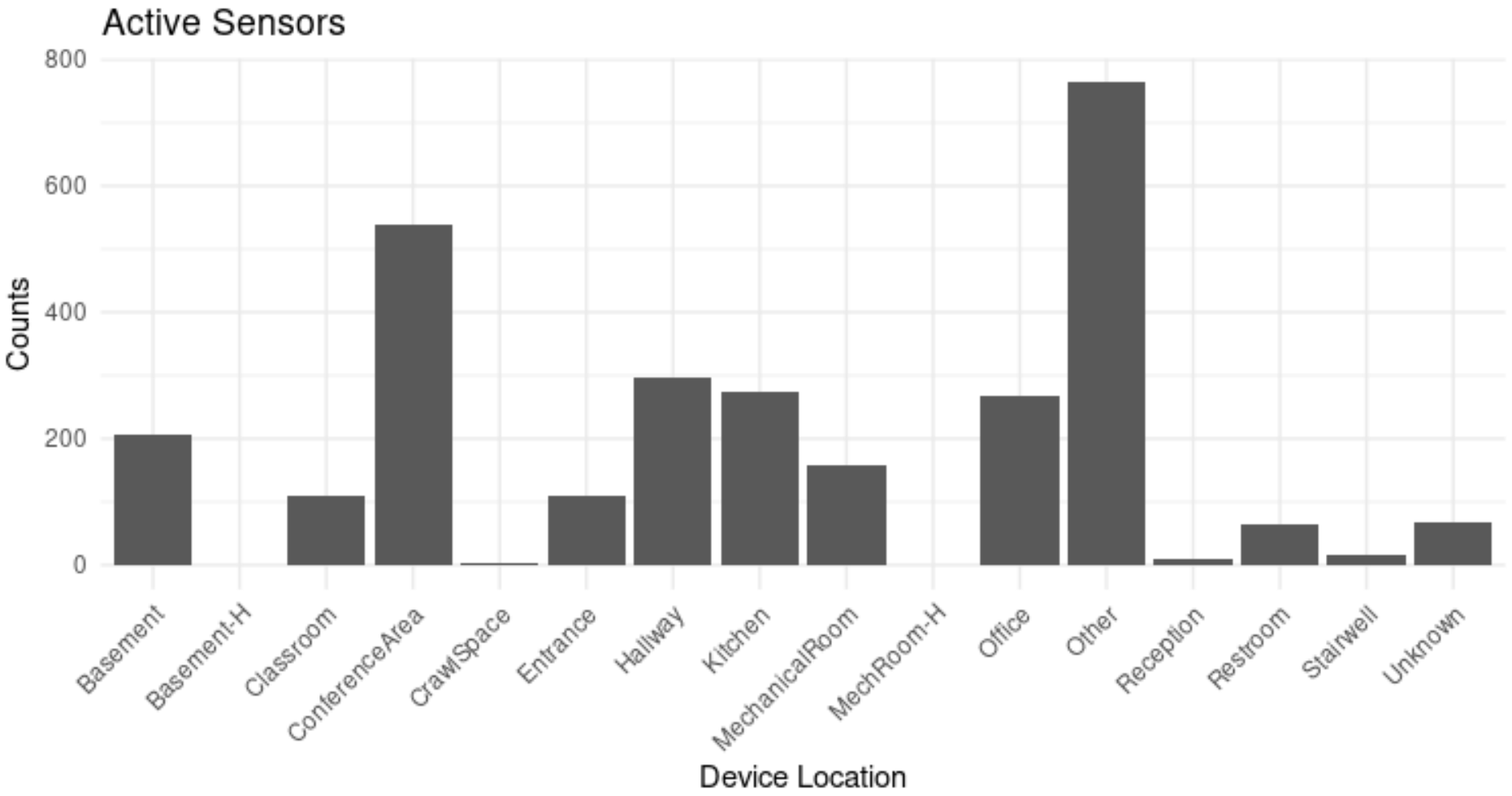
- Sensors are installed in buildings at risk of water pipe burst (freeze loss) across the United States.
- Buildings vary in terms of geography, building construction, heating system type, etc.
- Sensors are installed at varying locations inside the building, but typically within heated spaces.
- HSB employs a deterministic algorithm (RBA) to monitor for imminent freeze loss events.
- If there is an imminent threat of freeze loss, an alert is sent to customers.
- Currently, end users are mostly disengaged from the program.

Sensor Data Characteristics

- Dates range from 2018-10-01 until 2020-10-01
- Number of Installation Sites Analyzed: over 2842
- Number of States: 46
- Device Locations: 15
- Total Number of sensors: 2888


Active Sensor Count
By State
45°N
40°N
35°N
30°N
25°N
20°N
Longitude
120°W
110°W
100°W
90°W
80°W
Latitude
Count
[0,1)
[1,25)
[25,50)
[50,100)
[100,250)
[250,450)

Sensor Counts by Device Location



Sensor Temperatures Data Characteristics

- The sensors use a cellular network to relay temperature readings back in real time to HSB.
- Readings are measured and transmitted every 15 minutes.
- Temperature sensor accuracy published to be approximately $\pm 1.5^{\circ}$ F.
- Overall missingness of temperature readings across all sensors is about 11%.
- Nearly every sensor has some missingness.

Activity 1: A Deep Dive into Data Quality

Useful Reading

R. H. Shumway and D. S. Stoffer. Time Series Analysis and its Applications-with R Examples (2017).

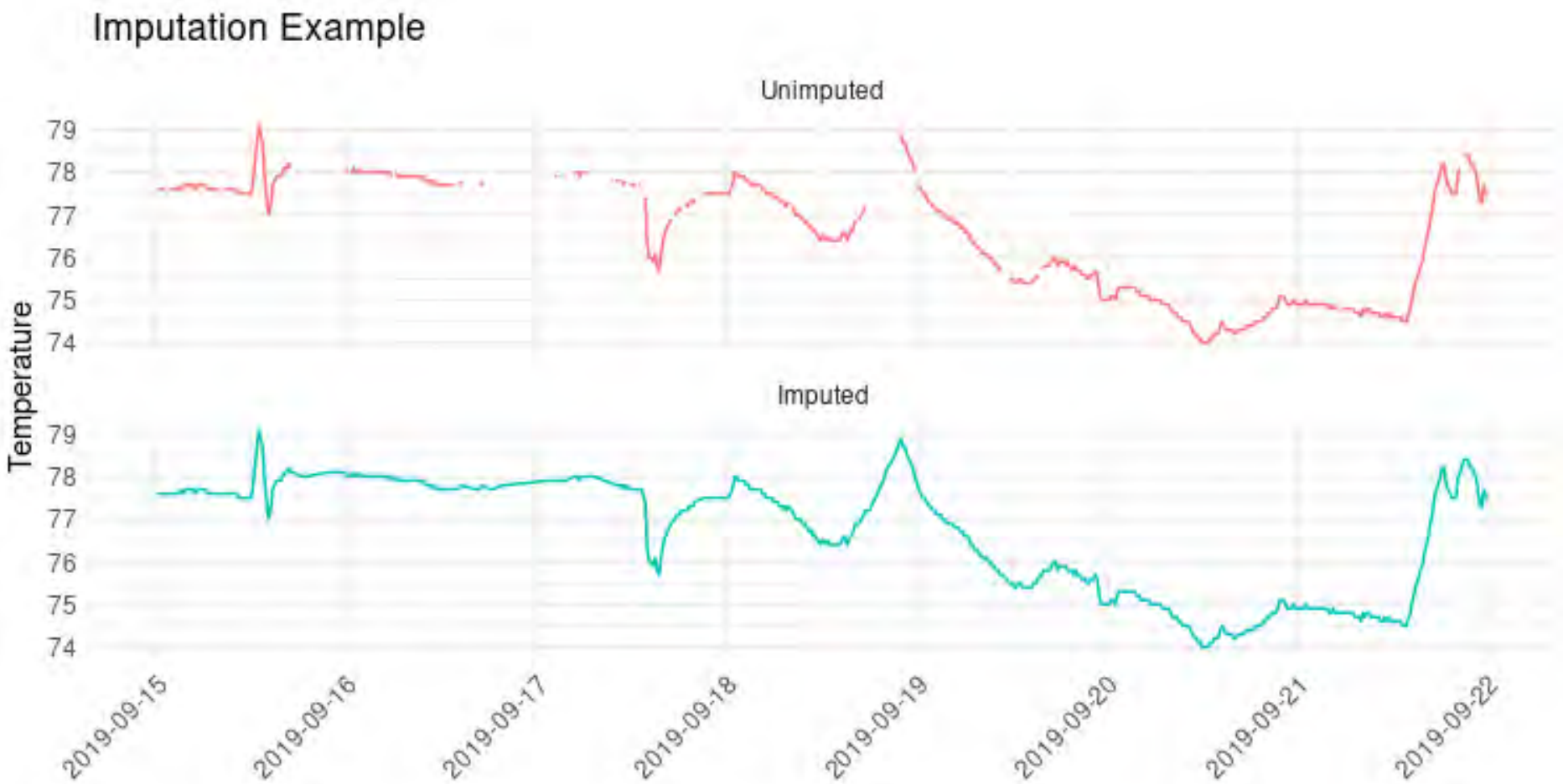
R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R", Journal of Statistical Software, 26(3), 2008.

S. Moritz and T. Bartz-Beielstein. "imputeTS: Time Series Missing Value Imputation in R." The R Journal (2017) 9:1, 207-218.

Missingness in IoT Data

For imputing missing data, we employ *locally estimated scatterplot smoothing* (LOESS), using the *forecast* library in R.

Imputation of Missing Data

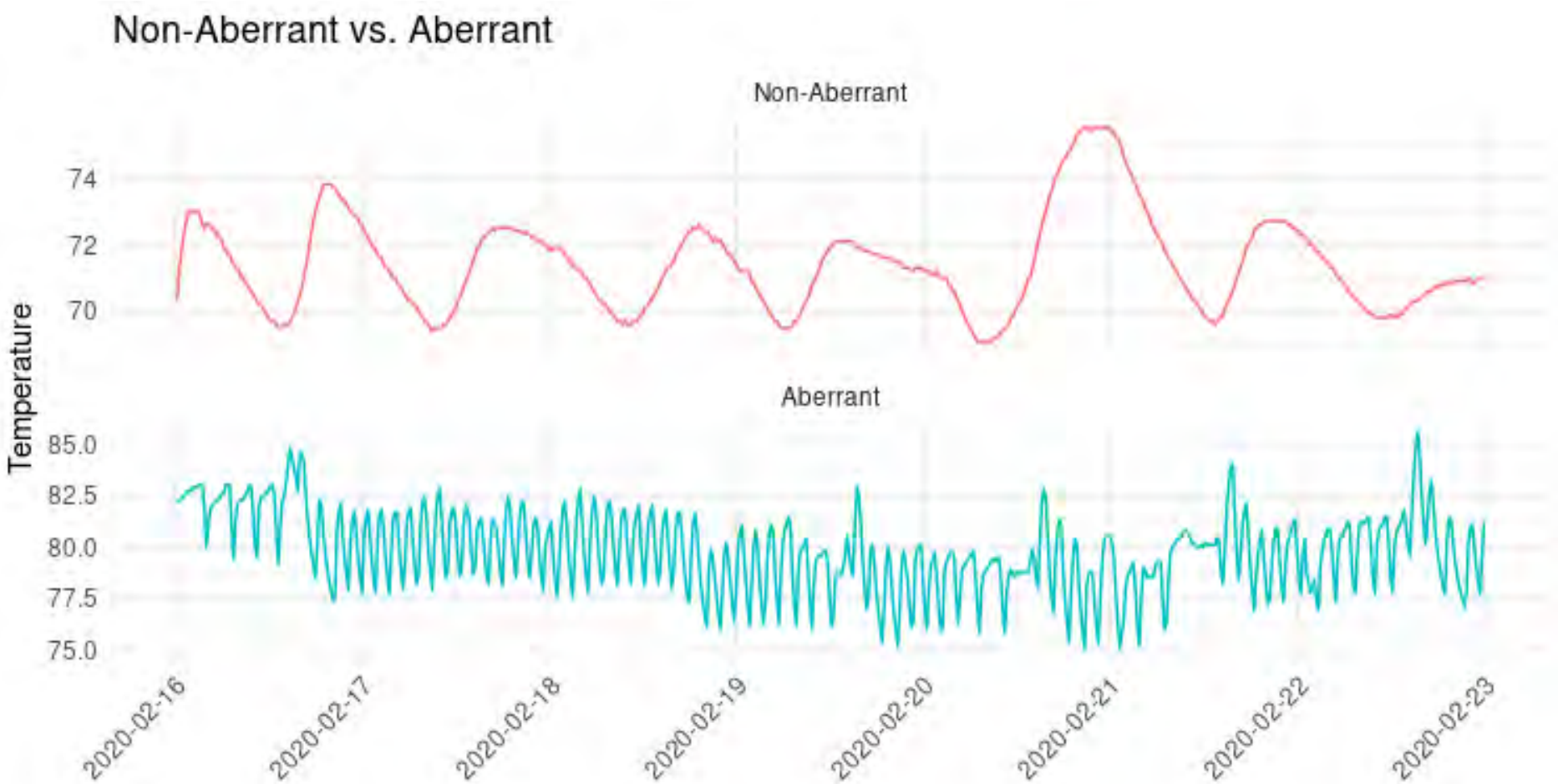


Quality of Sensor Temperature Streams

- For risk mitigation to be effective, the data streams must be of high quality.
- Does incoming data indicate it is from a properly installed and functioning sensor?
- Need to develop a way to detect data from faulty sensor streams, using patterns in historical time series data.

Data Quality: 2020/02/16 - 2020/02/23

- Top Plot: Smooth, almost seasonal patterns
- Bottom Plot: Temperature shifts of $\pm 5^{\circ}$ F occurring frequently throughout the day.



Goal: Detect faulty data streams

Aberrant sensor streams can be due to

- improper sensor installation (leading theory)
- sensor failure
- possible tampering with the sensor (i.e., moving it from location to location)

By being able to identify and label these series we gain the ability to

- identify whether certain installation locations coincide with greater frequency of aberrant behavior
- develop an online alert system to inform customers of faulty sensors/improper installation
- improve unsupervised anomaly detection methods by removing these observations from the historical data.

We develop a method to detect faulty data streams based on **time series clustering and classification** methods.

Activity 2: Clustering and Classifying Aberrant Streams

Useful Reading

S. Aghabozorgi, A. L. Shirkhorshidi, and T. Y. Wah. "Time-series clustering–a decade review." In: Information Systems 53, 16–38, (2015).

J. A. Hartigan and M. A. Wong, M. A. "Algorithm AS 136: A k-Means Clustering Algorithm." Journal of the Royal Statistical Society, Series C. 28 (1): 100–108 (1979).

A. D. Choukaria and P. Nagabhushan. "Adaptive dissimilarity index for measuring time series proximity". In:Advances in Data Analysis and Classification 1 (Feb. 2007), pp. 5–21.doi:10.1007/s11634-006-0004-6.

J. Lines, S. Taylor, and A. Bagnall. "HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles for Time Series Classification". In: 2016 IEEE 16th International Conference on Data Mining (ICDM) 2016, pp. 1041–1046.

A few items to recall ...

A **supervised machine learning (ML) algorithm** relies on labeled input data to learn a function that produces an appropriate output when given new unlabeled data. Used for classification or regression problems.

An **unsupervised ML algorithm** uses input data without any labels to learn a function that allows us to make predictions given some new unlabeled data. Used in clustering.

K-means is an unsupervised algorithm, used in clustering.

K-**Nearest Neighbor (KNN) algorithm** is a supervised algorithm, used in classification.

Different **distance metrics** may be used, based on context.

K-means: a review

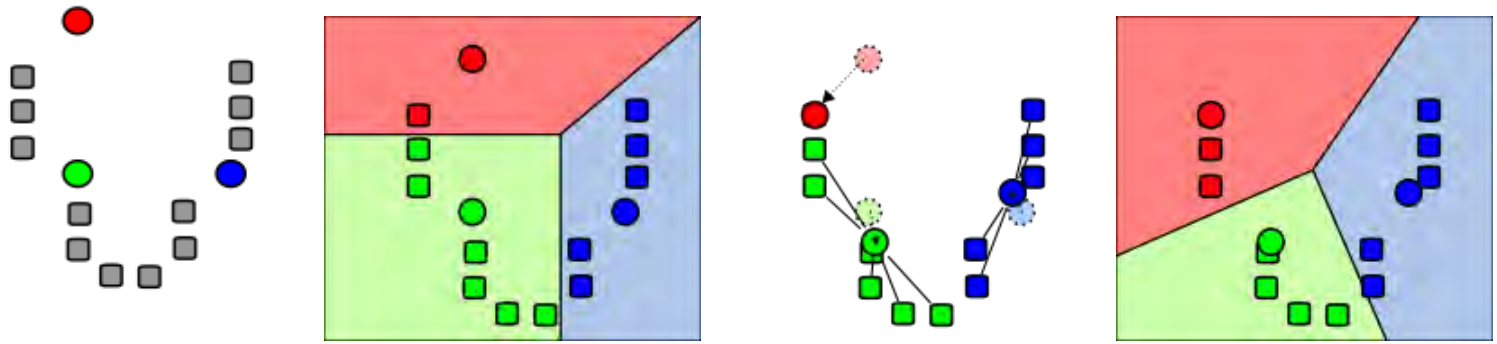


Figure: Source: https://en.wikipedia.org/wiki/K-means-clustering

1. k initial means (in this case k=3) are randomly generated within the data domain (shown in color).
2. k clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.
3. The centroid of each of the k clusters becomes the new mean.
4. Steps 2 and 3 are repeated until convergence has been reached

KNN Classification: a review

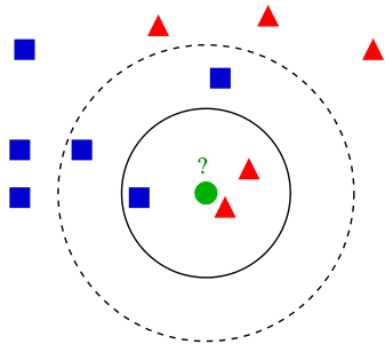


Figure: Source: https://en.wikipedia.org/wiki/K-nearest-neighbors-algorithm

Test sample (green dot) should be classified either to blue squares or to red triangles. If k = 3 (solid line circle), it is assigned to red triangles because there are 2 triangles and only 1 square inside the inner circle. If k = 5 (dashed line circle), it is assigned to the blue squares (3 squares vs. 2 triangles inside the outer circle).

Clustering Sensor Temperature Subsequences

- **Goal:** Cluster sensor temperature streams into groups with similar behavior, including the 'aberrant' group, based on the observed behavior (patterns) in standardized time series data.
- Series that are fall into the 'aberrant cluster' will be appended to an 'aberrant' library we will discuss later.
- We must define an appropriate dissimilarity metric and choose a clustering algorithm.
- We use a dissimilarity measure called *DCORT* (Chouakria and Nagabhushan, 2007) which incorporates dynamic time warping to accommodate misalignment.

Dynamic Time Warping: a review

- Often, two time series may exhibit very similar patterns over time, but may be 'phase shifted' from one another.
- In such cases, the Euclidean distance (which can only map the t^{th} point from one series to the t^{th} point of the other series) will not be adequate.
- The 'dynamic time warping distance' overcomes this misalignment by finding the optimal alignment or 'warp' path between the two series (Senin, 2008 for a review).

DTW in Freeze Loss Context - An Example

- Consider two work places that turn on the heat as the workday starts, but one starts at 8 AM and the other at 11 AM.
- We expect the temperature readings to have the same pattern, i.e., a temperature increase, but the peaks will be at different times (i.e., phase shifted).

Phase Shifted Series

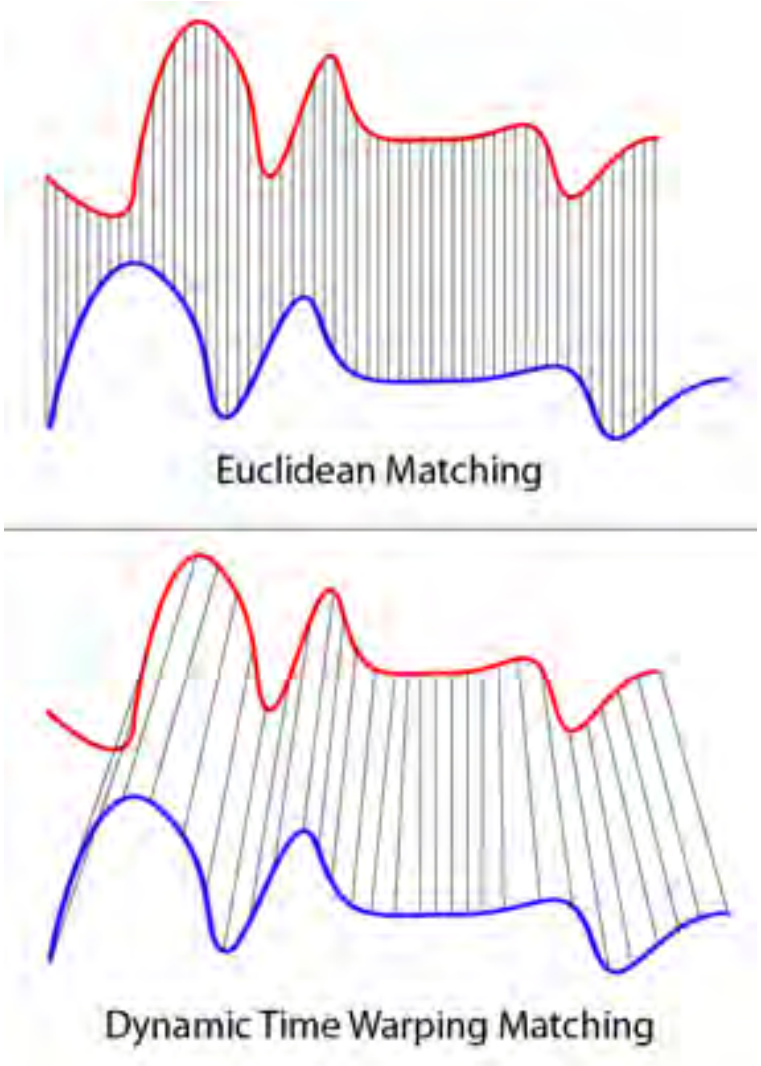


Figure: Source: https://commons.wikimedia.org

DCORT dissimilarity measure

$$d_{CORT}(X_T, Y_T) = \phi_k[CORT(X_T, Y_T)] \cdot \delta_{DTW}(X_T, Y_T)$$

- $CORT(X_T, Y_T) = \frac{\sum_{i=1}^{t-1}(x_{t+1}-x_t)(y_{t+1}-y_t)}{\sqrt{\sum_{t=1}^{t-1}(x_{t+1}-x_t)^2}\sqrt{\sum_{t=1}^{t-1}(y_{t+1}-y_t)^2}}$
- $\phi(.) = \frac{2}{1+e^{kx}}$
- $\delta_{DTW} = \min_{r \in M}\left(\sum_{t=1}^{M}|x_{a_t} - y_{b_t}|\right)$ - Dynamic Time Warping Distance.
- $k \geq 0$ - an adaptive tuning function that controls the proportion of the similarity determined by $\delta_{DTW}(.)$ and $CORT(.)$.

Application to IoT Sensor Temperatures Data

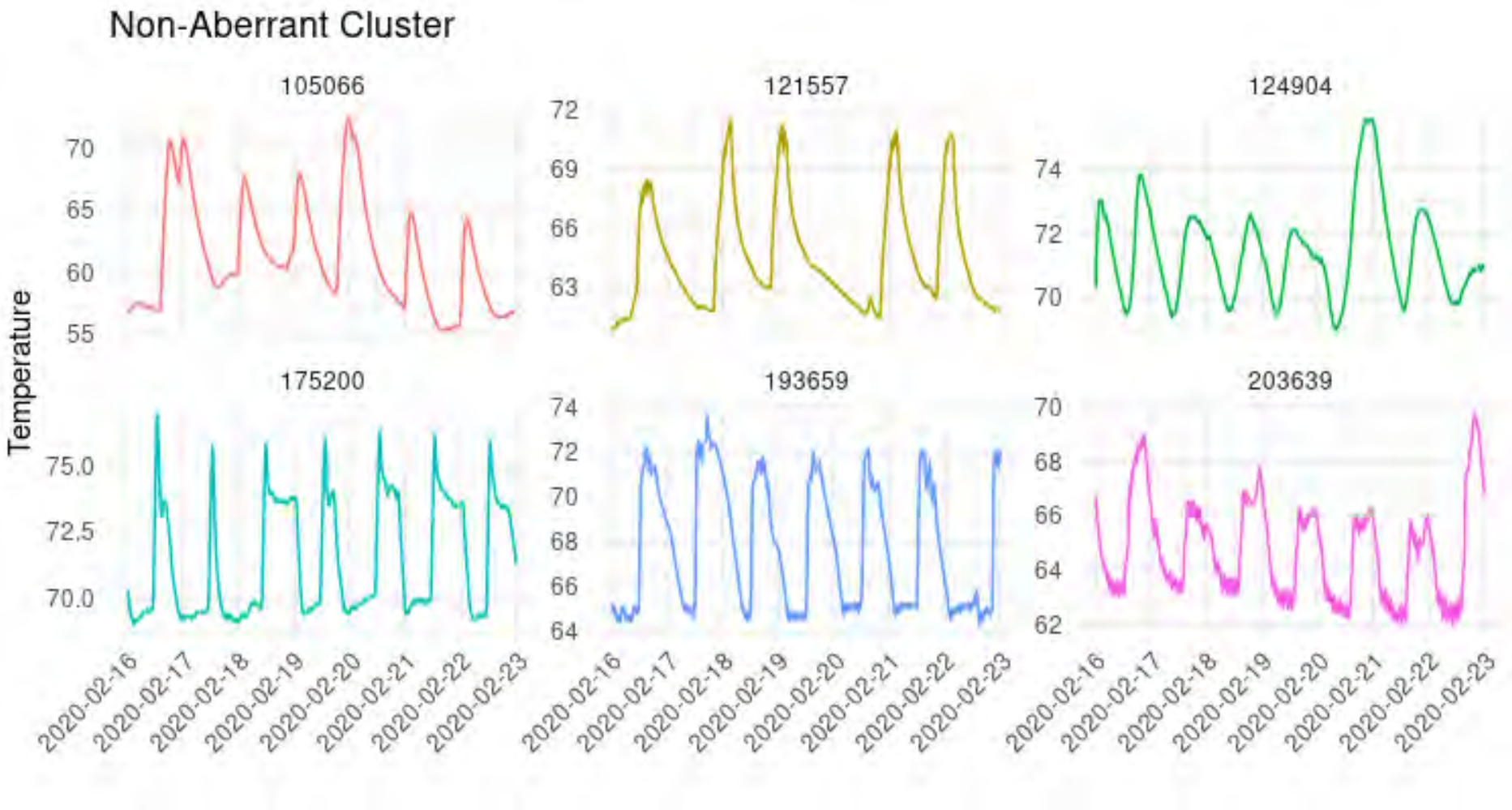
- We apply the clustering algorithm to a training set of 50% of the active sensors on a weekly basis - a week runs from Sunday to Sunday.
- Weekly segmentation is sensible because we expect to see recurring patterns of behavior on a weekly basis due to user demographics.

Example: Week 72 Clustering Results

- For illustration, we will focus on the clustering results for sensors during week 72 (2020/02/16 - 2020/23/02)
- Elbow plots, the gap statistic, and the silhouette index indicate that k = 8 clusters is reasonable for this week.
- Visualization of average cluster profiles and randomly selected series in each cluster help confirm the clusters are homogeneous within their groups.
- We find that 57 series fall into the aberrant class for this particular week.

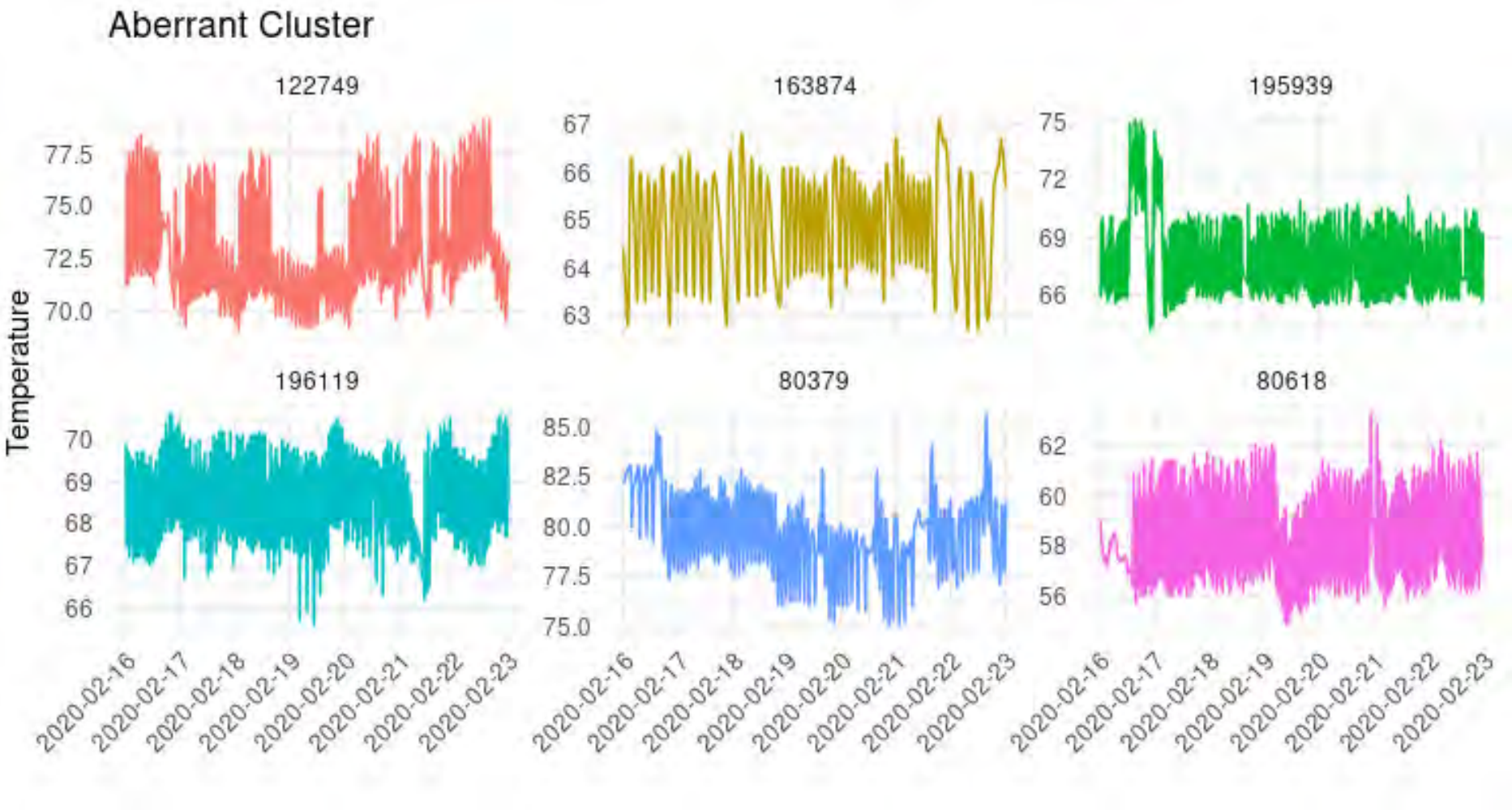
Cluster 4 - A 'Non-Aberrant' Cluster

- 6 Randomly selected sensors.



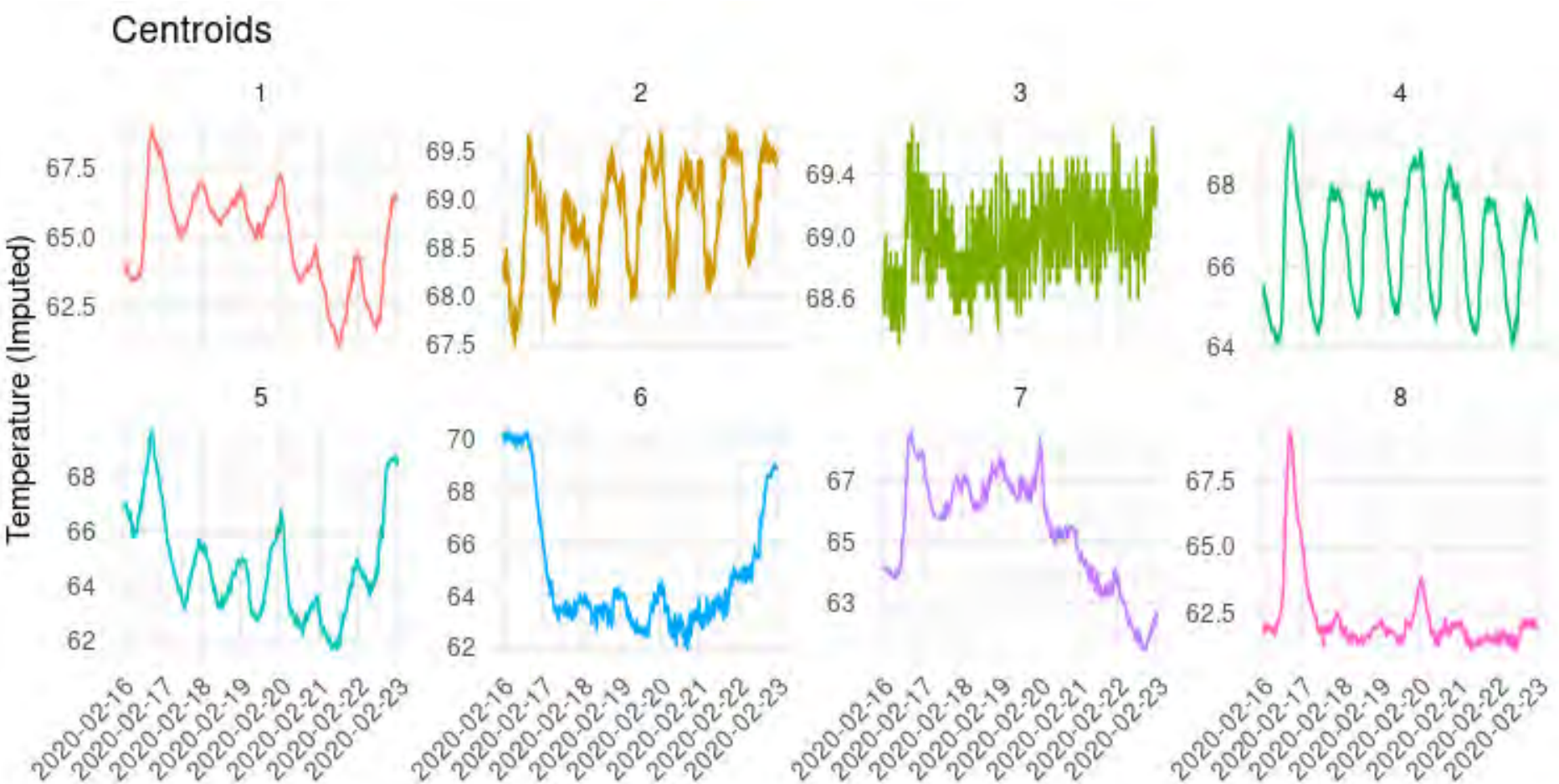
Cluster 3 - The 'Aberrant' Cluster

- 6 Randomly selected sensors.



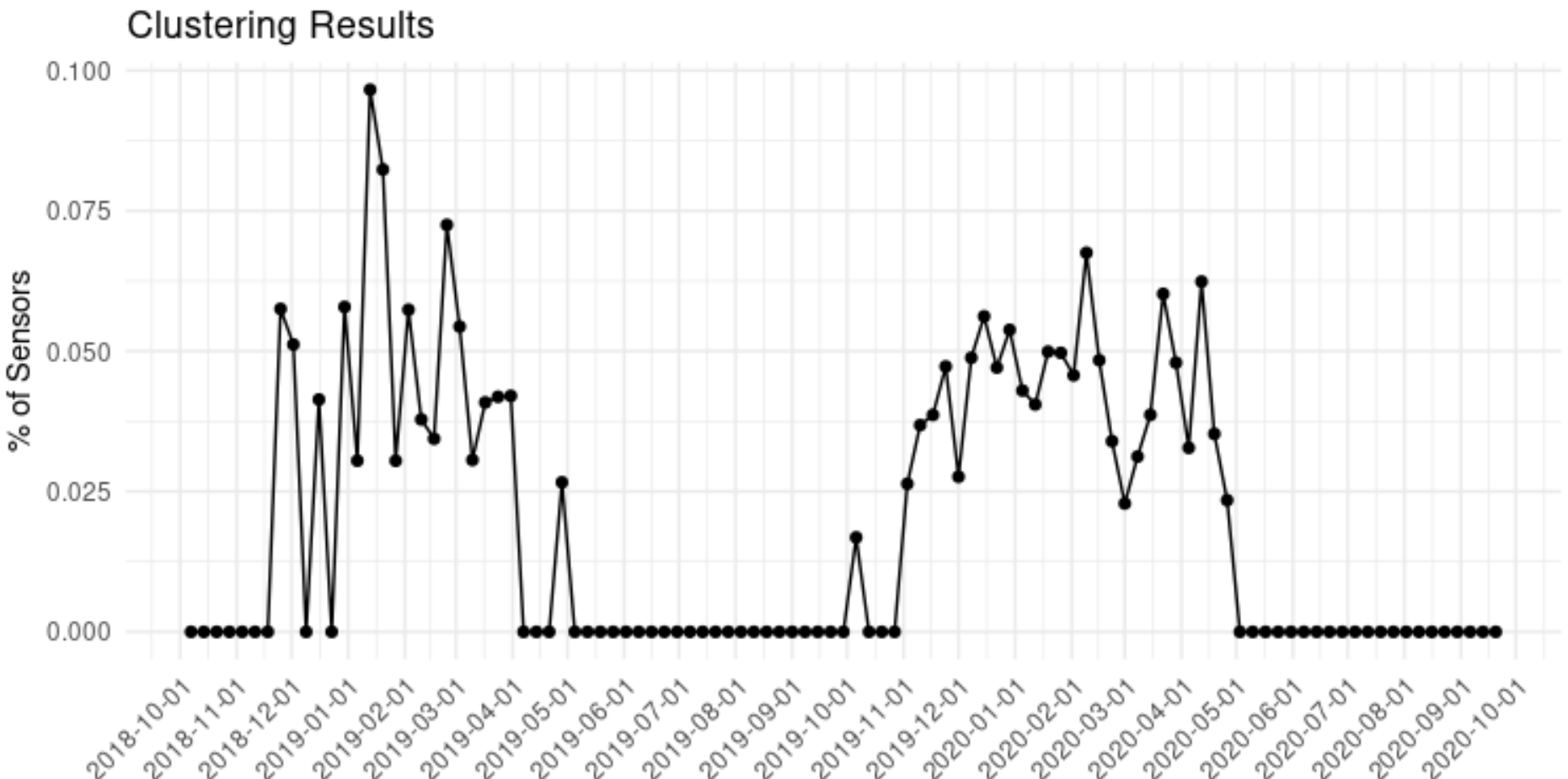
Average Profile

- The average profile or 'centroid' for each cluster is found by taking the average of all time series at each time point within a given clusters.
- Cluster 3 (our 'aberrant' class) has very noisy profile that is not exhibited in the other clusters.



Overall Clustering Results

- Overall, $\approx 0 - 10\%$ of all segments are grouped into a cluster of aberrant series in any given week.
- In addition, a clear seasonal pattern emerges, where the percentage of sensors behaving erratically increases during colder months



Next Goal: Classification and Aberrant Time Series Library

- Leverage information from the clustering results to develop a classification system that will perform the following:
  1. Construct/dynamically update an aberrant series library containing weekly time series segments flagged as aberrant by classification or clustering algorithm, flagging novel behavioral patterns.
  2. Assign labels to all other non-aberrant series based on their similarity to existing clusters.

Classification Problem

- Training (labeled) Data: $\mathcal{D} = \{(\mathbf{X}_1, y_1), \ldots, (\mathbf{X}_n, y_n)\}$.
  $\mathbf{X}_i \in \mathcal{R}^p$. Sensor temperatures for the i^{th} sensor.
  $y_i \in \{1, ..., J\}$. Classify label given to a particular time series segment by the DCORT clustering algorithm.
- Holdout (unlabeled) Data: $\mathbf{X}_0 \in \mathcal{R}^p$. A new vector of time series data that is unlabelled
- Goal: Given $\mathbf{X_0}$, predict y_0: assign a label.

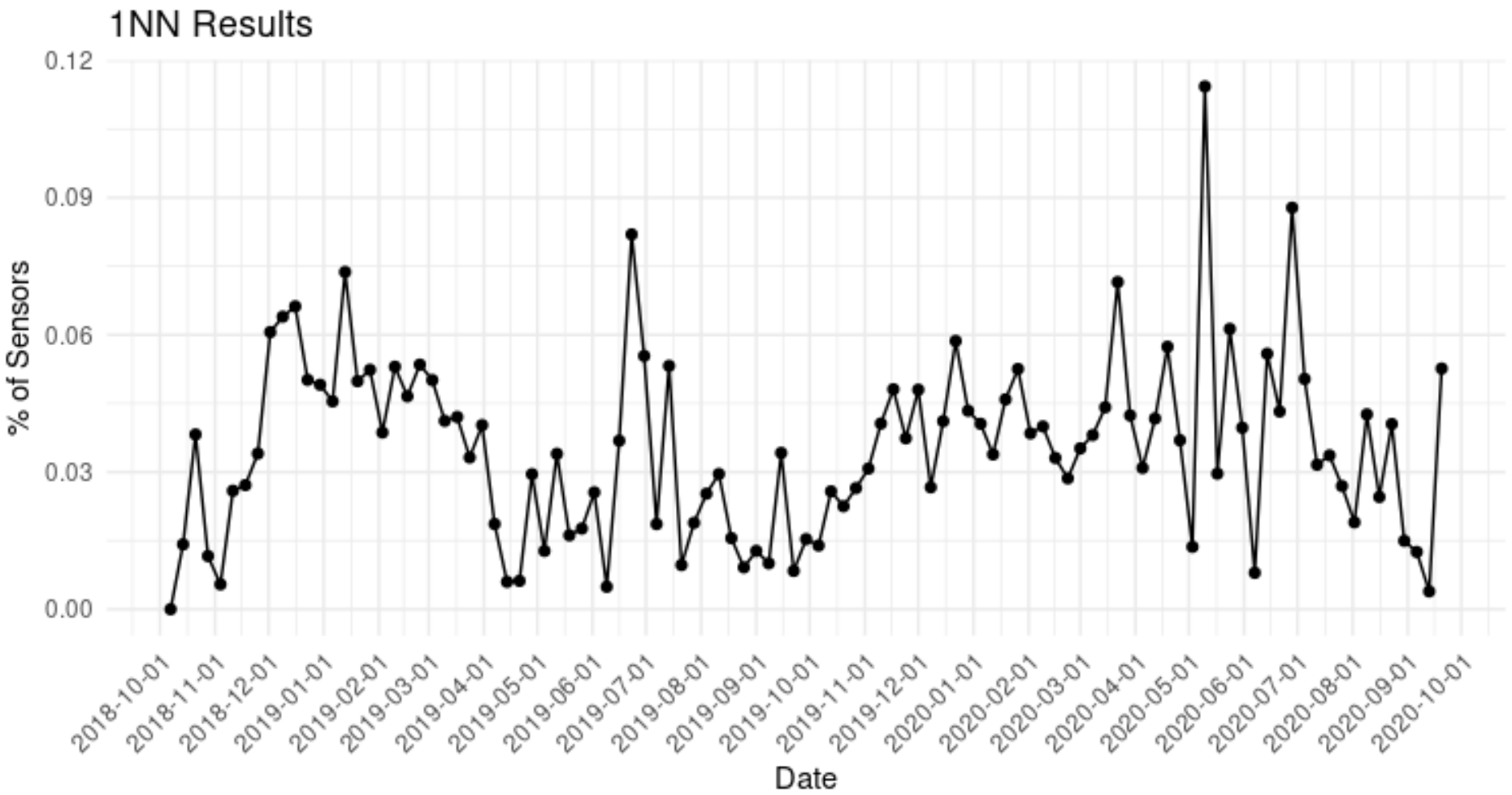
1NN Classification for HSB Data

- We apply 1NN-DCORT-DTW on a weekly basis from 2018-10-01 to 2020-10-01 to all sensors that have not already been clustered
  - Training Set: Clustered data from a user selected week (week 72 in this case) plus an initial aberrant time series library consisting of 261 weekly segments most representative of the aberrant class
- For weeks 1:103 (excluding training set from week 72)
  1. Calculate DCORT distance between each holdout series and entire training set.
  2. Find minimum 1NN-DCORT distance
  3. Label holdout series with label of 1NN
     - ★ If: 1NN is from aberrant class, append segment to aberrant library.
     - ★ Else: Label as non-aberrant
  4. Check whether any series have 1NN distance $> \theta$ where θ is a user specified threshold. ($\theta = 600$ in our case)

Results for HSB Data

- On any given week, $\approx 0 - 11.4\%$ of sensors in the holdout set are labeled as aberrant according to the 1NN classifier
- Cross-validation shows that the method 1NN-DCORT is extremely accurate in terms of classifying the aberrant series with an average prediction accuracy of 94%
- Similar to the clustering results, the 1NN classifier finds that the frequency of aberrant behavior decreases in the winter months

1NN Results


1NN Results
0.12
0.09
0.06
0.03
0.00
% of Sensors
2018-10-01
2018-11-01
2018-12-01
2019-01-01
2019-02-01
2019-03-01
2019-04-01
2019-05-01
2019-06-01
2019-07-01
2019-08-01
2019-09-01
2019-10-01
2019-11-01
2019-12-01
2020-01-01
2020-02-01
2020-03-01
2020-04-01
2020-05-01
2020-06-01
2020-07-01
2020-08-01
2020-09-01
2020-10-01
Date

Activity 3: Anomaly Detection for Alerting Clients

Useful Reading

R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," arxiv, vol. abs/1901.03407, 2019.

A. Cook, G. Misirli, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," IEEE Internet of Things Journal, vol. 7, pp. 6481–6494,2020.

F. Giannoni, M. Mancini, and F. Marinelli, "Anomaly detection models for IoT time series data," arxiv, vol. abs/1812.00890, 2018.

F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in 2008 Eighth IEEE International Conference on Data Mining. IEEE, 2008, pp. 413–422.

S.Hariri, M.CarrascoKind,andR.J.Brunner, "Extended Isolation Forest," IEEE Transactions on Knowledge and Data Engineering, 2019.

Isolation Forest Algorithm (IFA): a review

What is IFA and how does it work?

- Isolation Forest* Algorithm (IFA) is an effective and efficient anomaly detection algorithm.
- How it works:
  - Assume anomalous data points are few and have attribute values that are very different from those of normal points.
  - Randomly select a feature.
  - Randomly select a split value between the minimum and maximum values of the selected feature.
  - Recursive partitioning can be represented by a tree structure. The number of splittings required to isolate a point is equivalent to the path length from the root node to the terminating node.
  - IF builds an ensemble of "Isolation Trees" (iTrees) for the data set; anomalies are points that have shorter average path lengths on the iTrees.
  - The average path length to isolate an observation using these trees is used as an estimate for the **anomaly score**.

*Not to be confused with Random Forest

IFA: a review

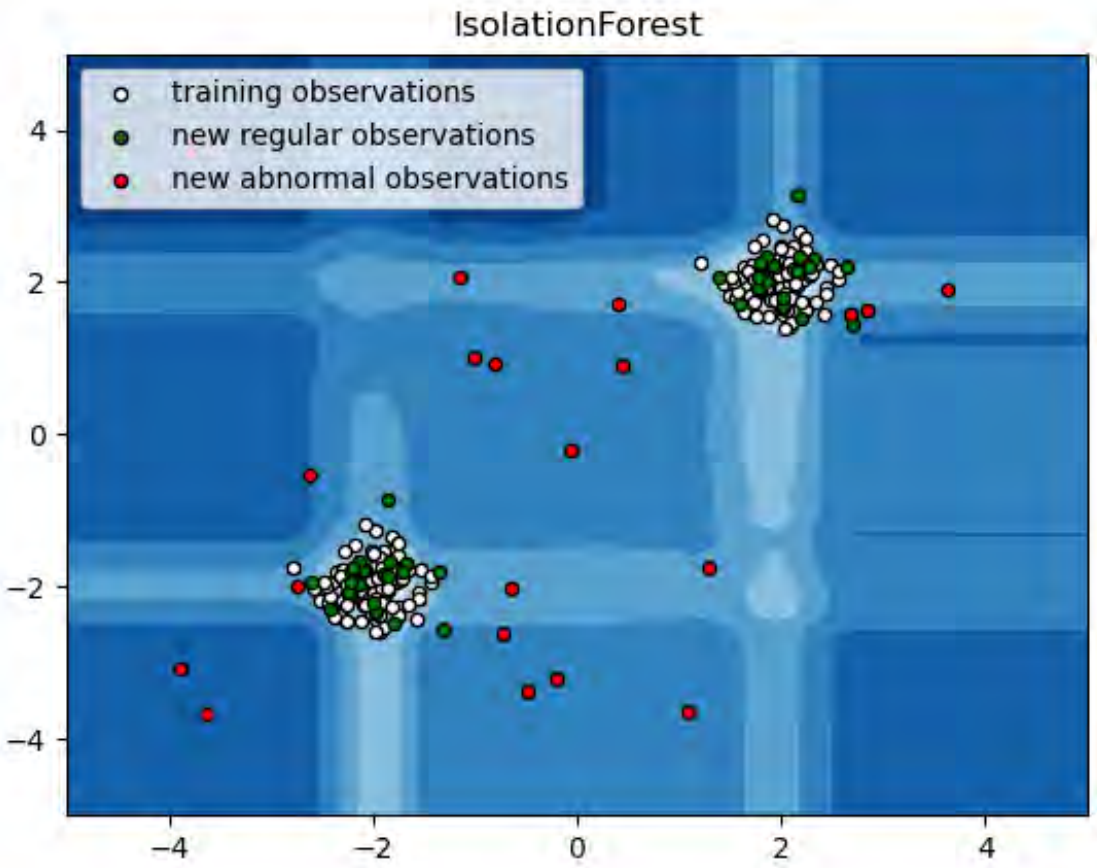


Figure: Source: https://scikit-learn.org/stable/auto-examples/

Applying IFA to HSB Data

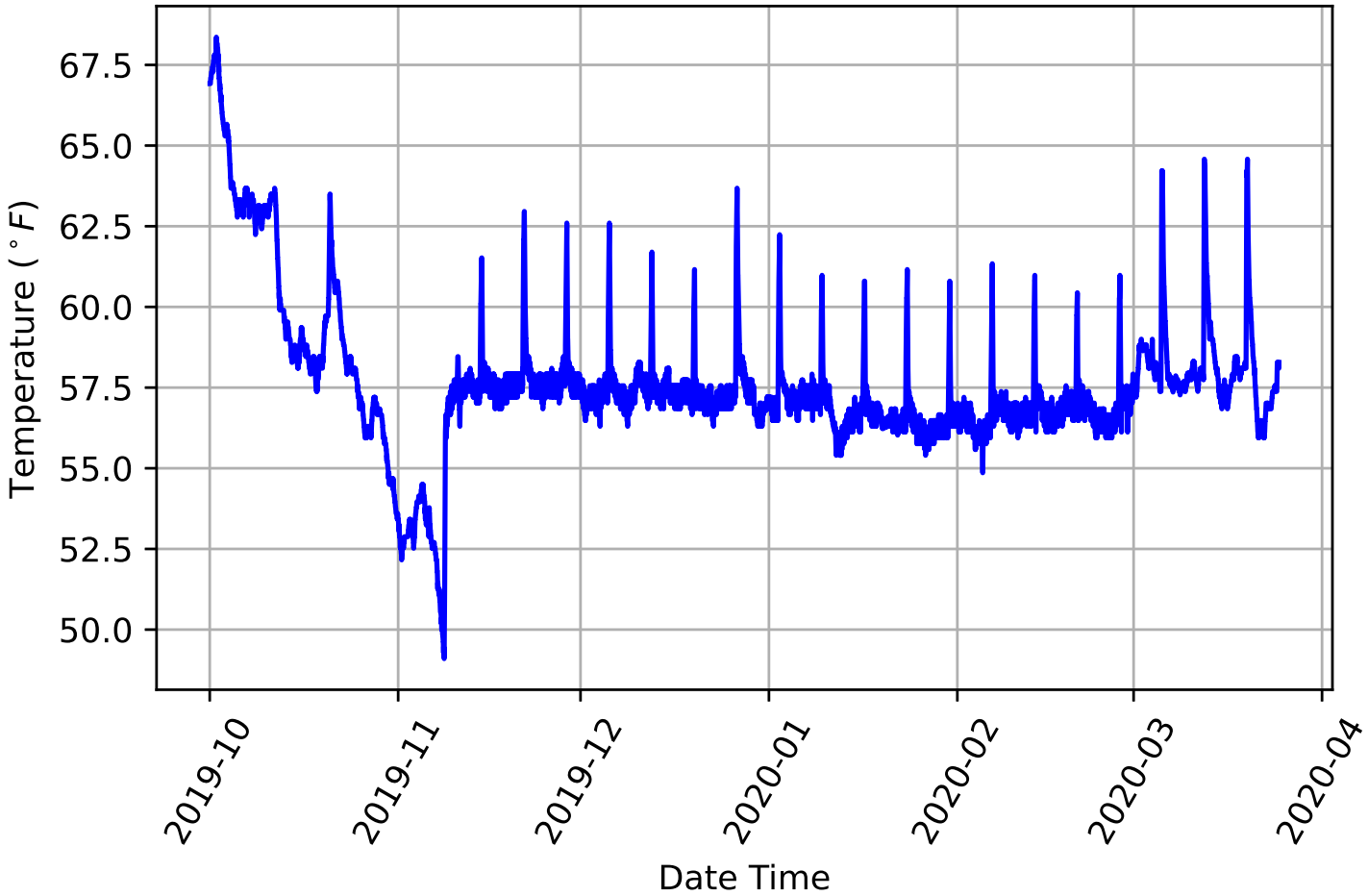


Figure: Temperatures for a random sensor node.

Applying IFA to HSB Data

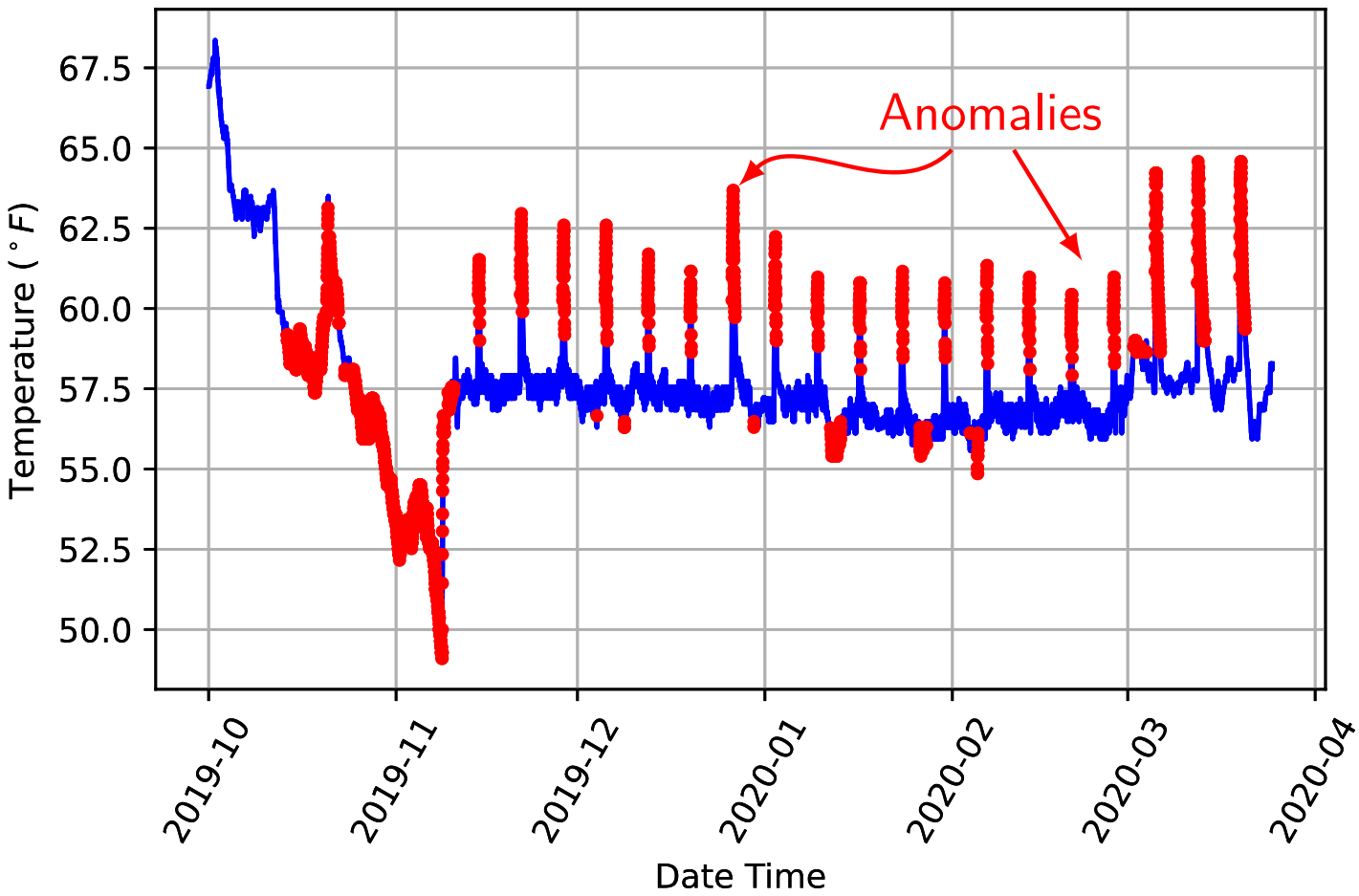


Figure: A sample result of applying the original IFA on a node where Number of anomalies $= 3,310/16,885$ (19.60%)

Drawbacks of IFA for HSB data analysis

Applying IFA to HSB data has led to unsatisfactory results:

- Too many anomalies flagged (very high False Positive Rate)
- Most anomalies are irrelevant to pipe-freeze protection

This motivated us to customize the IF algorithm ...

Tailoring a Custom IFA-based Solution for HSB

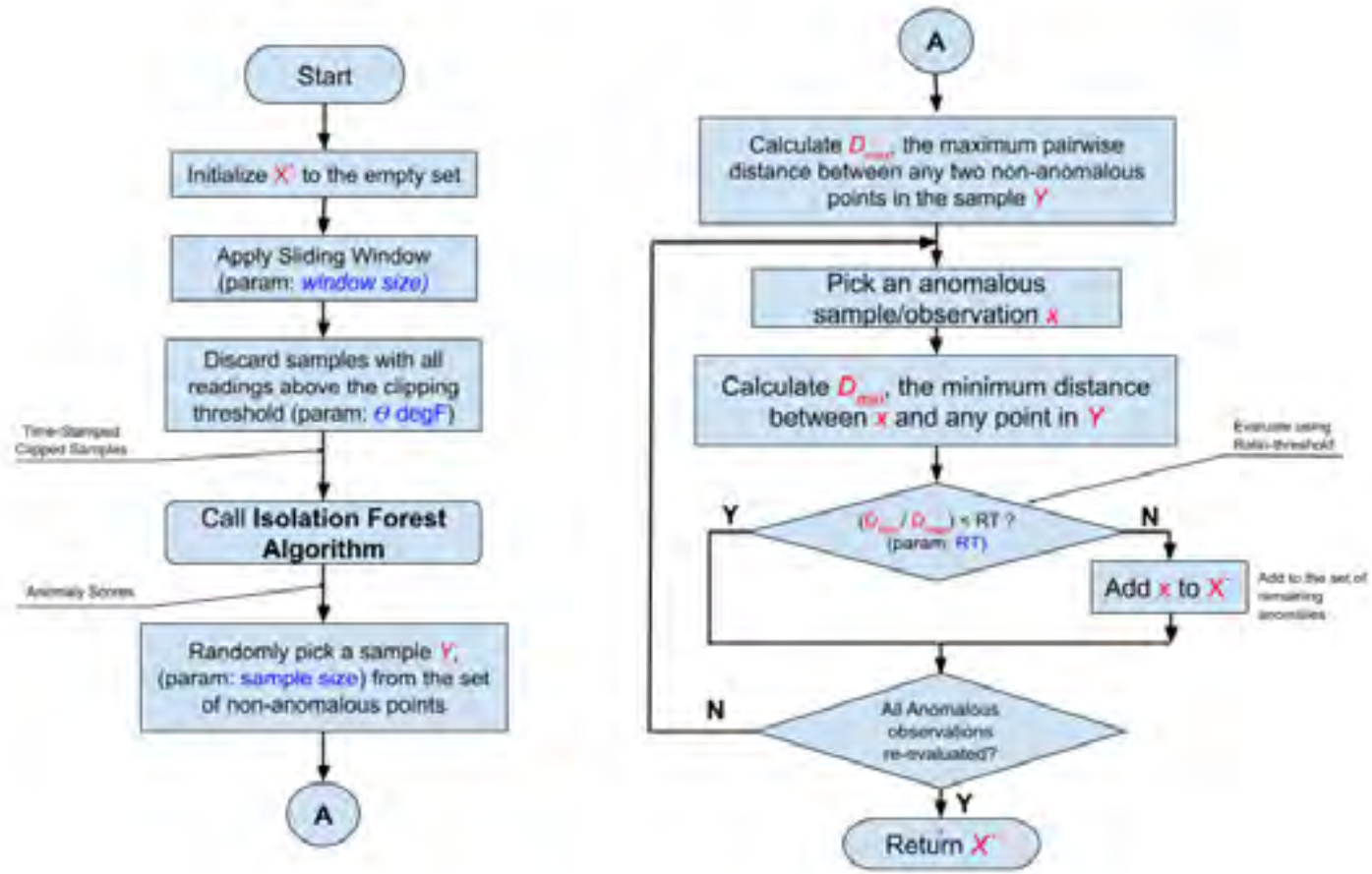


Figure: A flowchart for IFA+ algorithm (our proposed solution)

Note: For confidentiality reasons, some parameters/values have been disguised.

Leveraging outdoor/ambient temperatures

- **Goal:** Improve performance by leveraging the combined information extracted from indoor sensor data and outdoor ambient temperatures denoted by T_{amb}
- Towards that goal, three attempts/techniques have been implemented (summarized on the next slide)
- Only presenting in details the latest tweaks that produced the best results later on the upcoming slides

Attempts to leverage ambient temperature

Three algorithms to use ambient temperature in the anomaly detection process

- **IF+2:** Same window size for processing T_{int} and T_{amb}
  Features vector is a combination of all T_{amb} and T_{int} readings at the current window location

Attempts to leverage ambient temperature

Three algorithms to use ambient temperature in the anomaly detection process

- **IF+2:** Same window size for processing T_{int} and T_{amb}
  Features vector is a combination of all T_{amb} and T_{int} readings at the current window location
- **IF+3:** Again, same window size for processing T_{int} and T_{amb}
  Features vector is a combination of all T_{int} readings and the average value of the T_{amb} readings at the current window location

Attempts to leverage ambient temperature

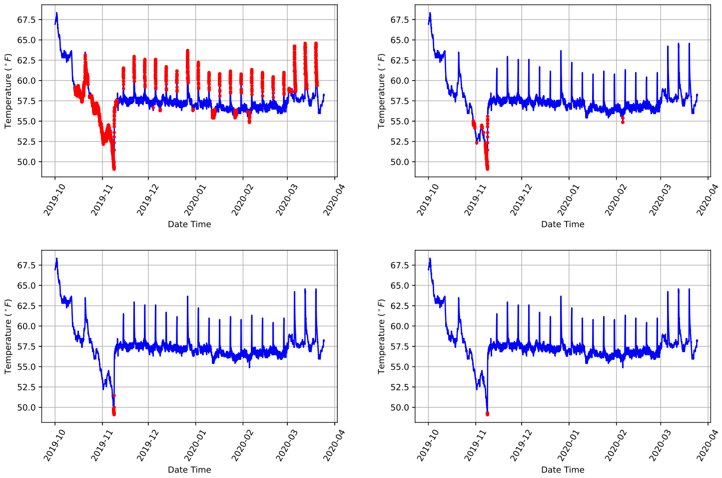
Three algorithms to use ambient temperature in the anomaly detection process

- **IF+2:** Same window size for processing T_{int} and T_{amb}
  Features vector is a combination of all T_{amb} and T_{int} readings at the current window location
- **IF+3:** Again, same window size for processing T_{int} and T_{amb}
  Features vector is a combination of all T_{int} readings and the average value of the T_{amb} readings at the current window location
- **IF+4:** T_{amb} is censored before applying the moving average sliding window
  Features vector is formed as in IF+3
  Filtering: Use of different weights for T_{int} and T_{amb}

After a few trials,... IFA+4

Improve performance by leveraging the combined information extracted from indoor sensor data and outdoor ambient temperatures, denoted by T_{amb}. **IFA+4** consists of the following three steps:

1. **Preprocessing**
   - Censoring of T_{amb} signal
   - Applying moving average window to censored signal
2. **Building the Isolation Forest model**
   - First 4 features $\leftarrow T_{int}$
   - Fifth feature $\leftarrow$ last value of the moving average window
3. **Postprocessing**
   - Use different weights for T_{int} and T_{amb} during distance-based filtering



67.5
65.0
62.5
60.0
57.5
55.0
52.5
50.0
Temperature (° F)
2019-10
2019-11
2019-12
2020-01
2020-02
2020-03
2020-04
Date Time

Results (IFA+4 vs. RBA)

Number of anomalies detected by each algorithm.

Description	Number of nodes	Percentage
Total	806	100.00%
Normal	772	95.78%
Flagged by HSB Alg.	33	4.09%
Flagged by IF+4	8	0.99%

Table: Number and percentages of anomalies

About the Team Synergy

- Synergy between Statistics and Computer Science at UConn and professionals from the HSB (whose parent company is the Munich Re Group).
- Weekly meetings both between groups and within groups using a virtual platform.
- PhD students get to learn how to take methods/code to production in a business environment.
- Faculty members learn to do a deep dive into each other's domain.
- Happily, this is an on-going venture, and we hope to explore more exciting intersections of Domain, Stat and CS -
  Suppose a client got an alert. Can we use the data only to infer whether or not an action was taken by the client?

Thank You!