Shaping the Modern Era: A Deep Dive into AI and ML

DMLF Årsmøde 2023

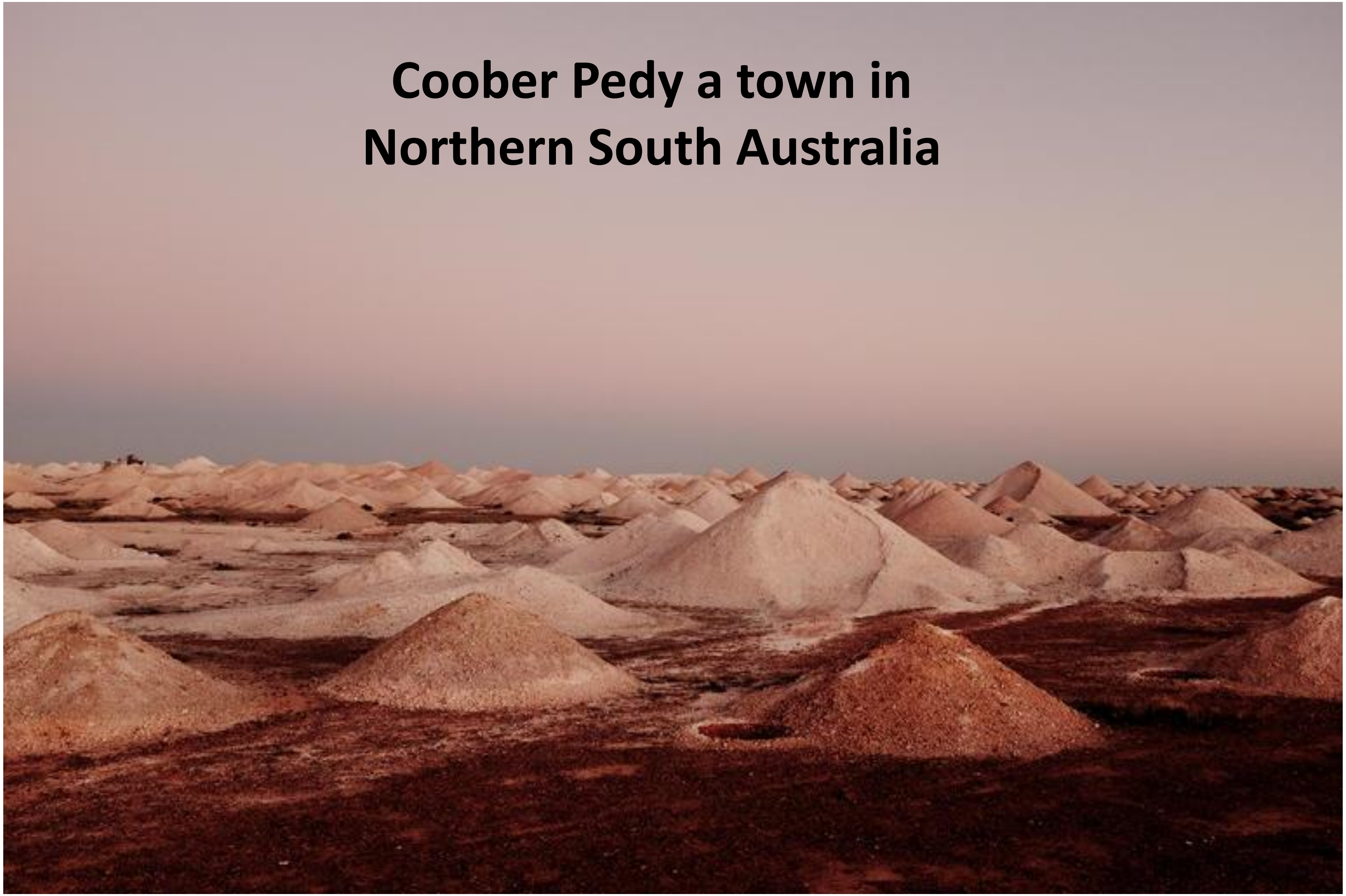
By Nadeem Iftikhar, PhD
University College of Northern Denmark (UCN)
naif@ucn.dk
28th of September 2023

Outline

- Inspiration
- Data
- Introduction to Artificial Intelligence (AI)
- Introduction to Machine Learning (ML)/Deep Learning (DL)
  - Supervised
  - Unsupervised
- Time Series
- Use Cases from SMVs
- ML Application Workflow
- Why Majority of AI/ML based Student Projects at Professional Academies Fail?
- Questions

Inspiration

Coober Pedy a town in Northern South Australia



into your DATA

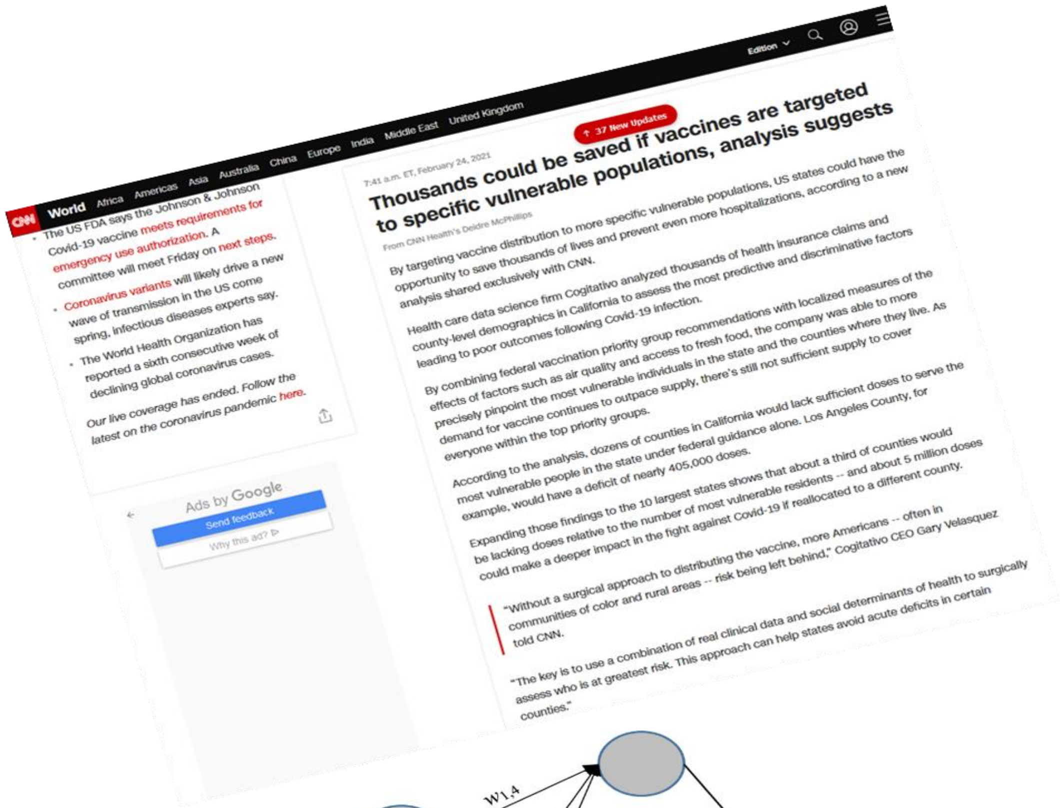
Dig deep enough and you will find Opals

Data

Data

- **Qualitative data**
  - Qualitative data is non-statistical and is typically unstructured or semi-structured
  - This data isn't necessarily measured using hard numbers used to develop graphs and charts
  - Qualitative data can be used to ask the question "why." It is investigative and is often open-ended until further research is conducted
- **Quantitative data**
  - Contrary to qualitative data, quantitative data is statistical and is typically structured in nature – meaning it is more rigid and defined
  - This data type is measured using numbers and values, making it a more suitable candidate for data analysis

Artificial Intelligence (AI)



CNN World Africa Americas Asia Australia China Europe India Middle East United Kingdom

Edition

- The US FDA says the Johnson & Johnson Covid-19 vaccine meets requirements for emergency use authorization. A committee will meet Friday on next steps.
- Coronavirus variants will likely drive a new wave of transmission in the US come spring, infectious diseases experts say.
- The World Health Organization has reported a sixth consecutive week of declining global coronavirus cases.

Our live coverage has ended. Follow the latest on the coronavirus pandemic here.

Ads by Google
Send feedback
Why this ad?

7:41 a.m. ET, February 24, 2021

37 New Updates

Thousands could be saved if vaccines are targeted to specific vulnerable populations, analysis suggests

From CNN Health's Deidre McPhillips

By targeting vaccine distribution to more specific vulnerable populations, US states could have the opportunity to save thousands of lives and prevent even more hospitalizations, according to a new analysis shared exclusively with CNN.

Health care data science firm Cogitativo analyzed thousands of health insurance claims and county-level demographics in California to assess the most predictive and discriminative factors leading to poor outcomes following Covid-19 infection.

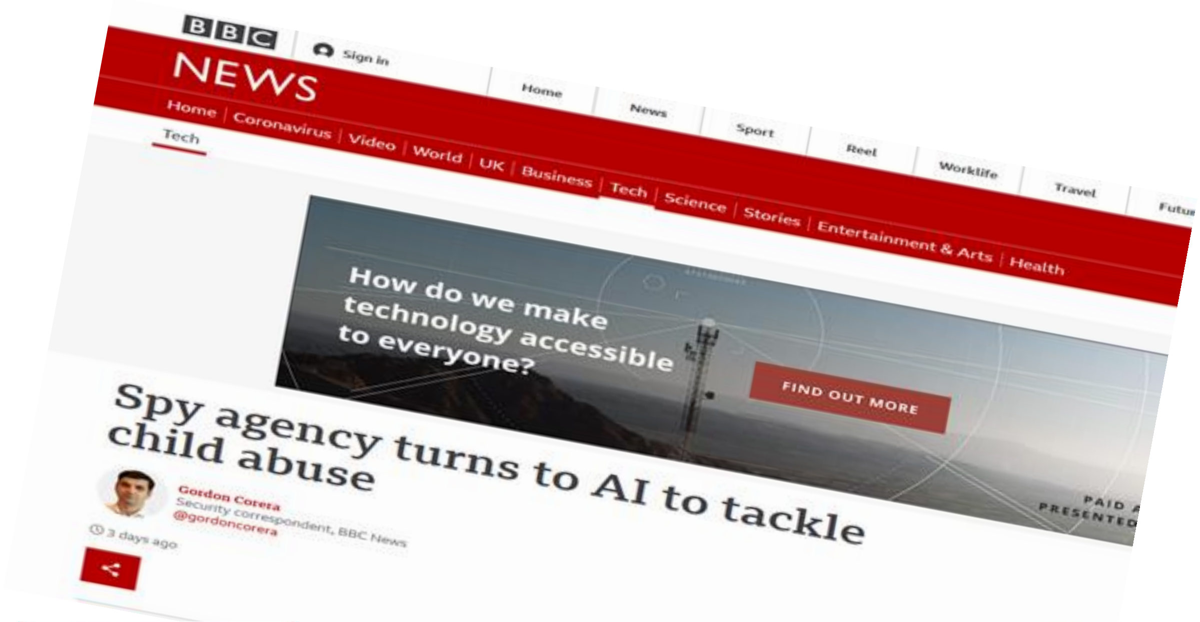
By combining federal vaccination priority group recommendations with localized measures of the effects of factors such as air quality and access to fresh food, the company was able to more precisely pinpoint the most vulnerable individuals in the state and the counties where they live. As demand for vaccine continues to outpace supply, there's still not sufficient supply to cover everyone within the top priority groups.

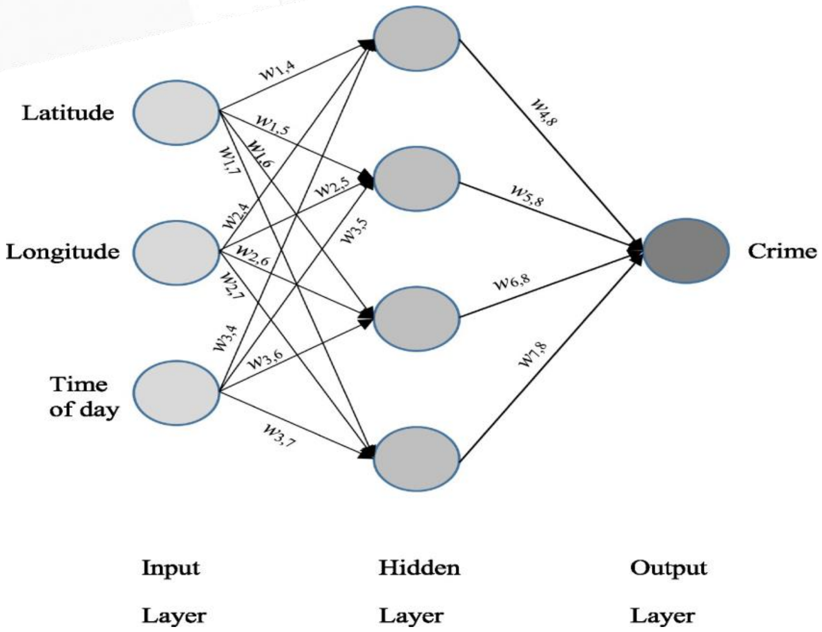
According to the analysis, dozens of counties in California would lack sufficient doses to serve the most vulnerable people in the state under federal guidance alone. Los Angeles County, for example, would have a deficit of nearly 405,000 doses.

Expanding those findings to the 10 largest states shows that about a third of counties would be lacking doses relative to the number of most vulnerable residents -- and about 5 million doses could make a deeper impact in the fight against Covid-19 if reallocated to a different county.

"Without a surgical approach to distributing the vaccine, more Americans -- often in communities of color and rural areas -- risk being left behind," Cogitativo CEO Gary Velasquez told CNN.

"The key is to use a combination of real clinical data and social determinants of health to surgically assess who is at greatest risk. This approach can help states avoid acute deficits in certain counties."





https://www.frontiersin.org/files/Articles/587943/fpsyg-12-587943-HTML/image_m/fpsyg-12-587943-g001.jpg

Artificial Intelligence (AI)-1

- Autonomy
  - The ability to perform tasks in complex environments without constant guidance by a user

- Adaptivity
  - The ability to improve performance by learning from experience

Artificial Intelligence (AI)-2

- Levels of AI
  - **Artificial Narrow Intelligence***
    - Handles only one task at a time
  - Artificial General Intelligence
  - Artificial Super Intelligence

*now, but we don’t know about future

Some Examples of AI

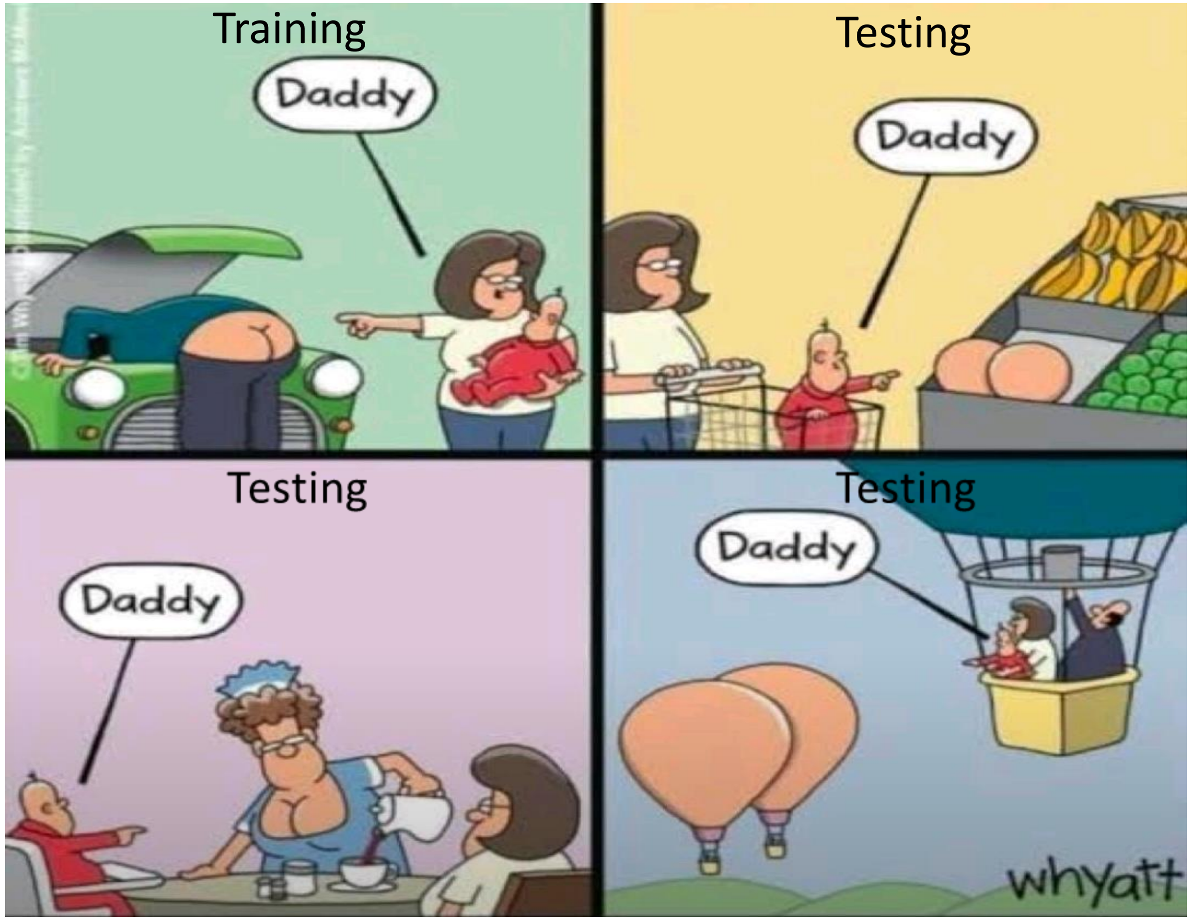
- ChatGPT
- Microsoft AI
- Self-driving cars
  - Find route
  - Computer vision
  - Decision making under uncertainty
- Content recommendation
  - Personalized information (Facebook, Twitter ..)
  - Recommendations (Spotify, Netflix …)

Types of AI

- **Artifical Intelligence (AI)**
  - AI is a popular field of computer science that concerns with building ``intelligent'' smart machines capable of performing intelligent tasks.
- **Machine Learning (ML)**
  - ML is a type of AI that enables machines to learn from data and deliver predictive models.
  - The ML is not dependent on any explicit programming but the data fed into it.
  - Based on the data you feed into ML algorithms and the training given to it, an output is delivered.
    - A predictive algorithm will create a predictive model.
- **Deep Learning (DL)**
  - DL is a subfield of ML that is concered with algorithms inspired by the human brain's structure and functions known as artifical neural networks.
    - A computer model can be taught using DL to run classification actions using pictures, texts, sounds etc. as input.

Machine Learning (ML)

Machine Learning (ML)



Ref: Machine Learning Community: A LinkedIn group

ML

- Machine learning
  - Systems that improve their performance in each task with more and more experience or data

https://www.youtube.com/watch?v=QFyM3w95fXI&t=16s

AI vs ML

AI

- **Fixed outcomes:** When there is a small or fixed number of outcomes.
- **Risk of error:** The penalty of error is too high to risk false positives and therefore only rules—which will be 100 percent accurate—should be implemented.
- **Not planning for ML:** If those maintaining the system don't have machine learning knowledge and the business does not have plans to source for it moving forward.

ML

- **Simple rules don't apply:** When there is no easily definable way to solve a task using simple rules
- **Speed of change:** When situations, scenarios, and data are changing faster than the ability to continually write new rules.
- **Natural language processing:** Tasks that call for an understanding of language, or natural language processing. Since there are an infinite number of ways to say something.

Types of Machine Learning (ML)

- Supervised Learning
- Unsupervised Learning

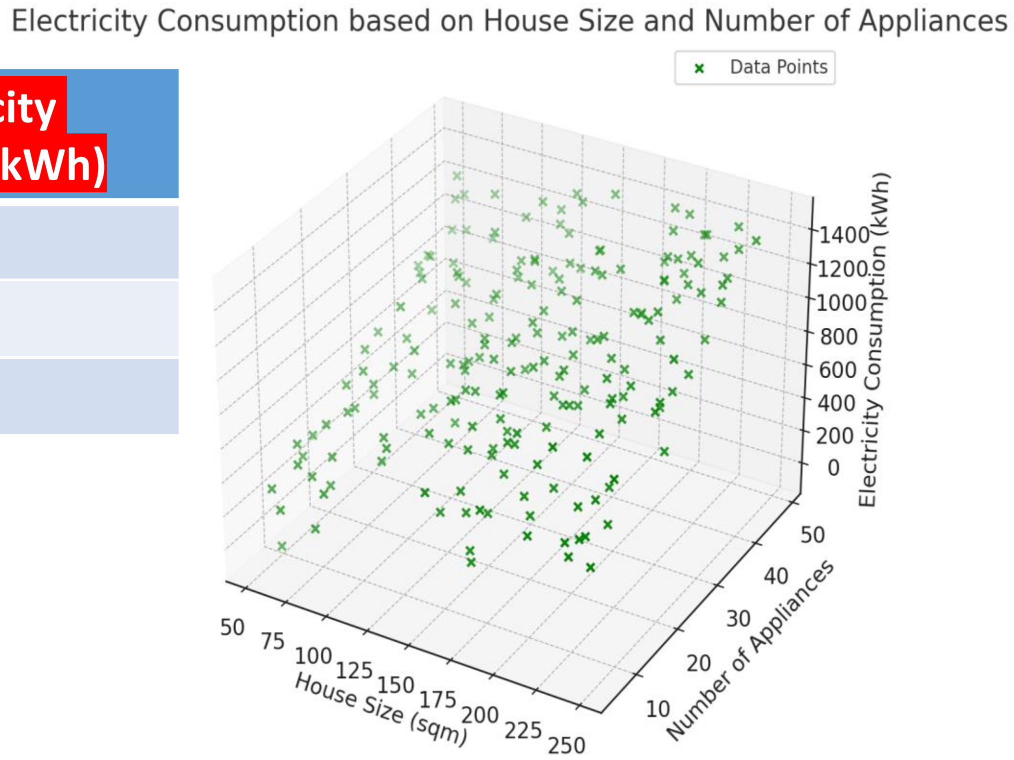
Machine Learning

- Supervised Learning
  - If we have a set of **labeled data,** we take this data and train the supervised machine learning model. Once the model is trained, we predict the results from the sample or the data in which the results are unknown.
    - For example, an algorithm would be trained with pictures of dogs and other things, all labeled by humans, and the machine would learn ways to identify pictures of dogs on its own.
- Unsupervised Learning
  - If we provided with **unlabeled data**, then we could apply unsupervised learning to predict pattern in that data.
    - A program looks for patterns in *unlabeled data*. Unsupervised machine learning can find patterns or trends that people aren't explicitly looking for. For example, an unsupervised machine learning program could look through online sales data and identify different types of clients making purchases.

Supervised Learning Example (Regression)

- Predict the annual electricity consumption of households in Denmark based on the size of the house (in square meters) and the number of electronic appliances in the house.

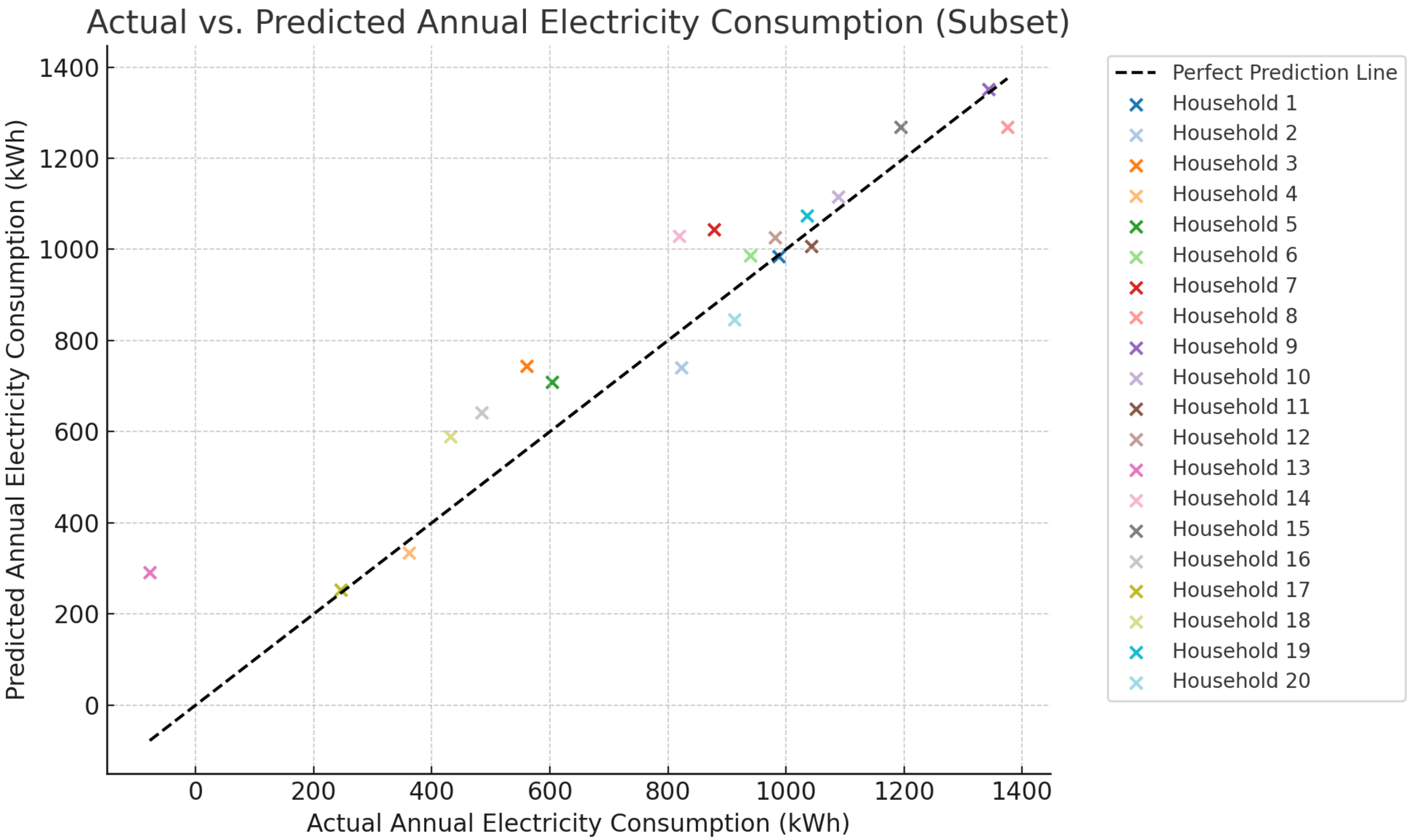
House Size (sqm)	Number of Appliances	Annual Electricity Consumption (kWh)
208.9	38	1259.7
150.5	37	1054.4
68.4	12	545.0



Multiple Linear Regression:

Electricity Consumption Model (kWh)=

2.04×House Size (sqm)+18.93×Number

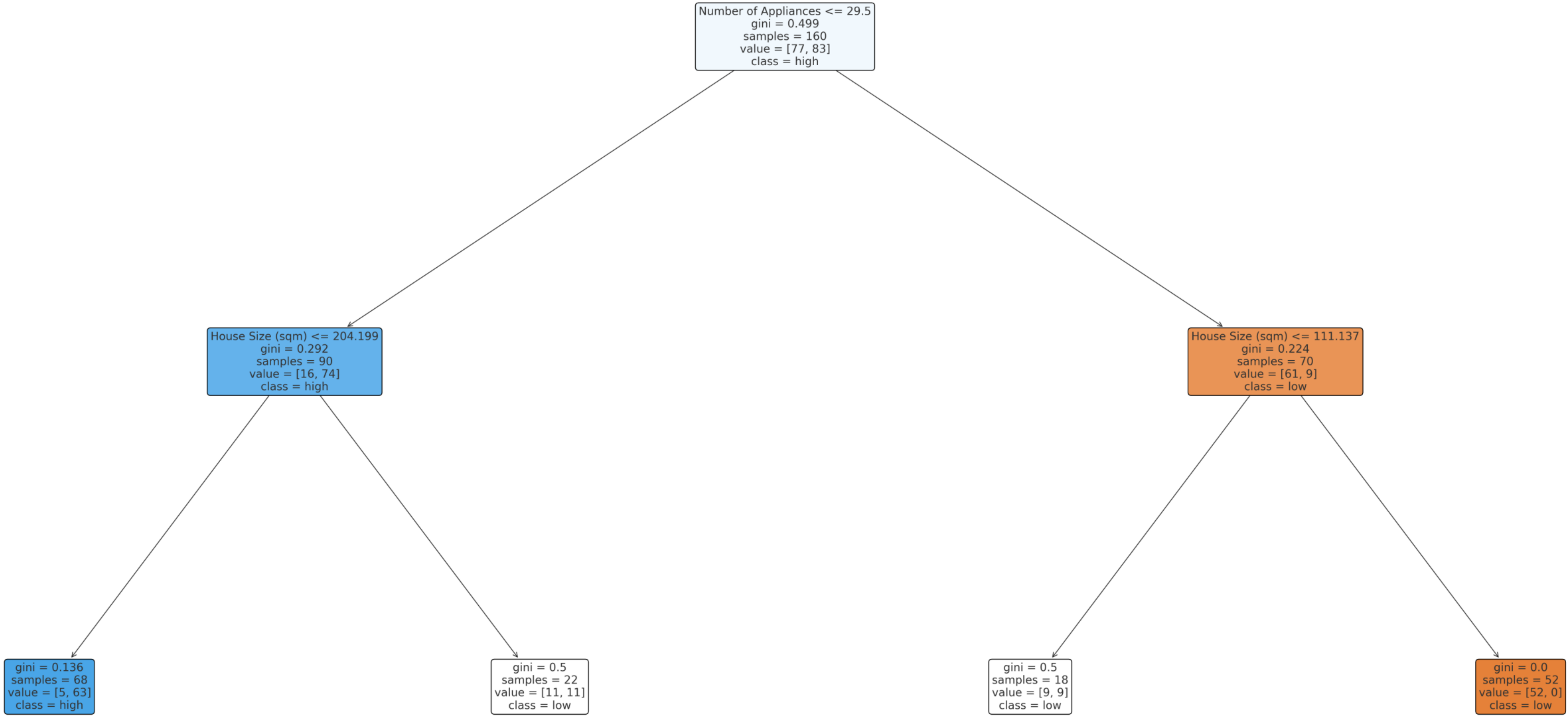

Actual vs. Predicted Annual Electricity Consumption (Subset)
Predicted Annual Electricity Consumption (kWh)
Actual Annual Electricity Consumption (kWh)
0
200
400
600
800
1000
1200
1400
Perfect Prediction Line
Household 1
Household 2
Household 3
Household 4
Household 5
Household 6
Household 7
Household 8
Household 9
Household 10
Household 11
Household 12
Household 13
Household 14
Household 15
Household 16
Household 17
Household 18
Household 19
Household 20

Supervised Learning Example (Classification)

- Predict whether a household has "high" or "low" electricity consumption based on the size of the house (in square meters) and the number of electronic appliances in the house.

House Size (sqm)	Number of Appliances	Annual Electricity Consumption (kWh)	Consumption Label
208.9	38	1259.7	high
150.2	37	1054.4	high
68.4	12	545.0	low

Decision Tree



Unsupervised Learning Example (Clustering)

- The task is to discover the structure of the data (by grouping / *clustering* similar items).

https://foolproofliving.com/spring-mix-salad/

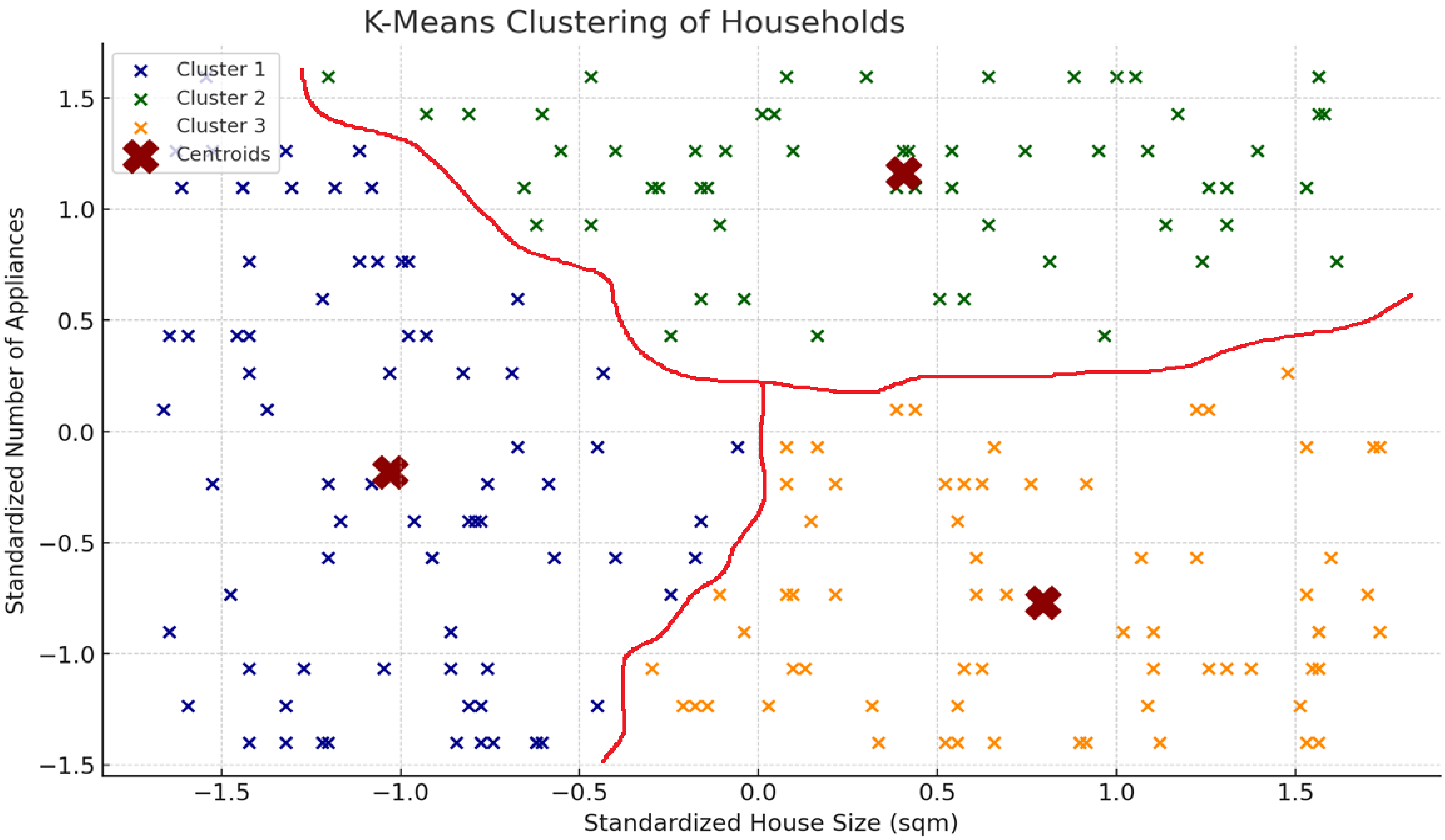
→

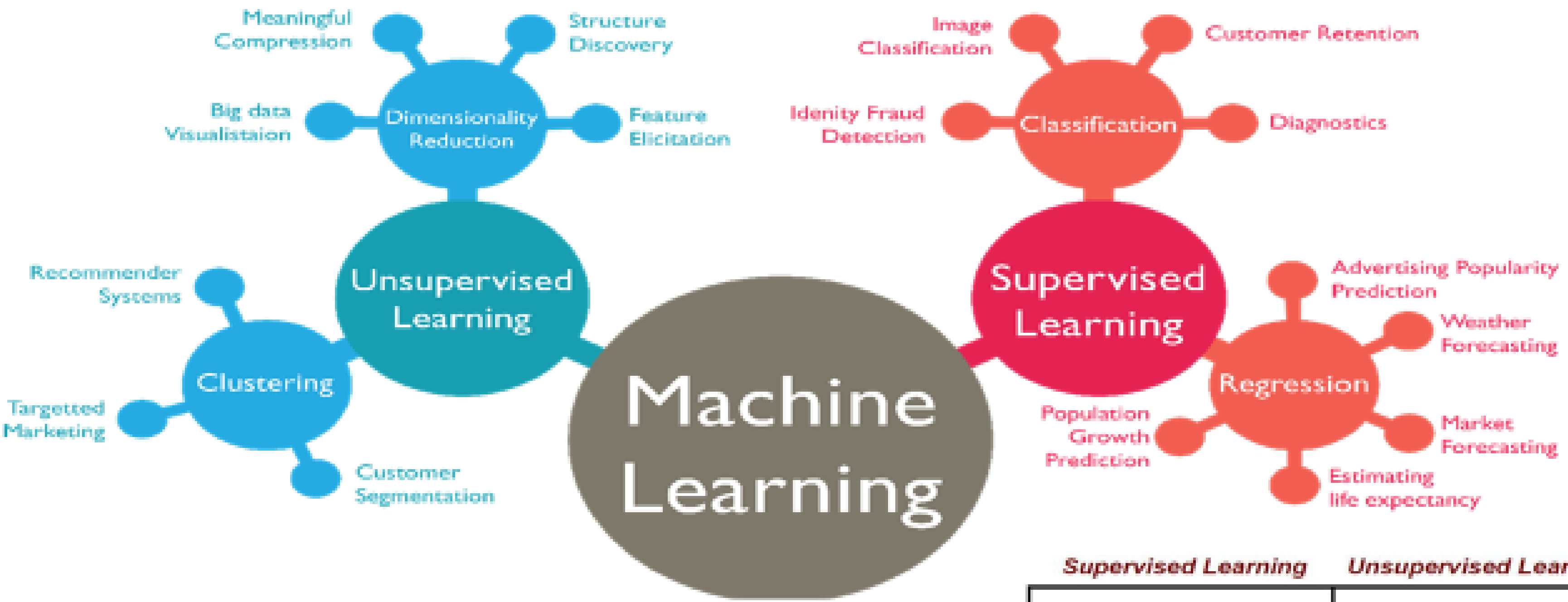
https://analyticsindiamag.com/clustering-techniques-every-data-science-beginner-should-swear-by/

Unsupervised Learning Example (Clustering)-2

- let's group households based on their attributes: house size and number of appliances. The idea would be to determine if there are distinct clusters (or groups) of households that have similar characteristics.

House Size (sqm)	Number of Appliances	Annual Electricity Consumption (kWh)
208.9	38	1259.7
150.5	37	1054.4
68.4	12	545.0


K-Means Clustering of Households
Cluster 1
Cluster 2
Cluster 3
Centroids
Standardized Number of Appliances
Standardized House Size (sqm)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5



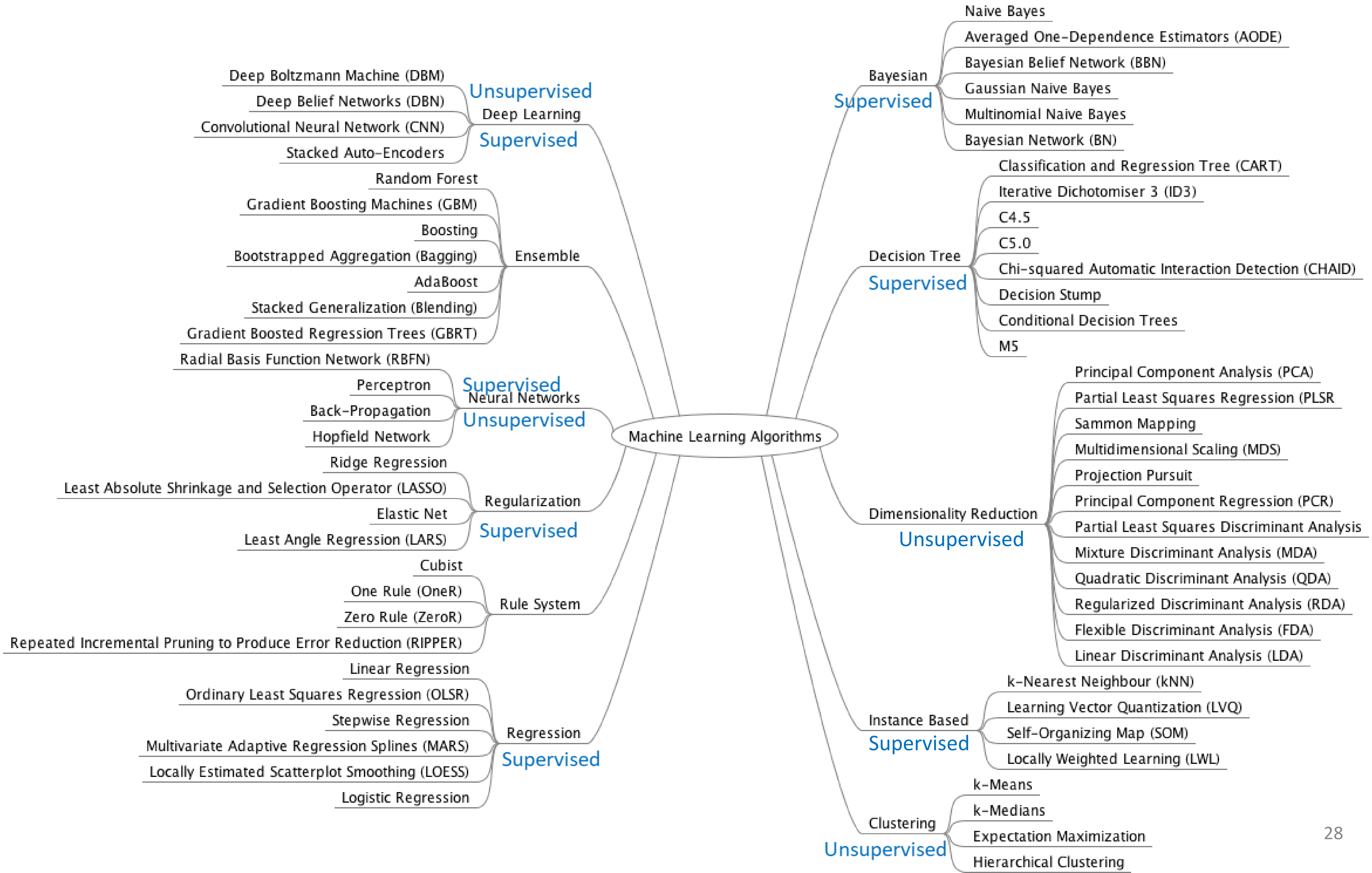
	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Supervised and Unsupervised learning

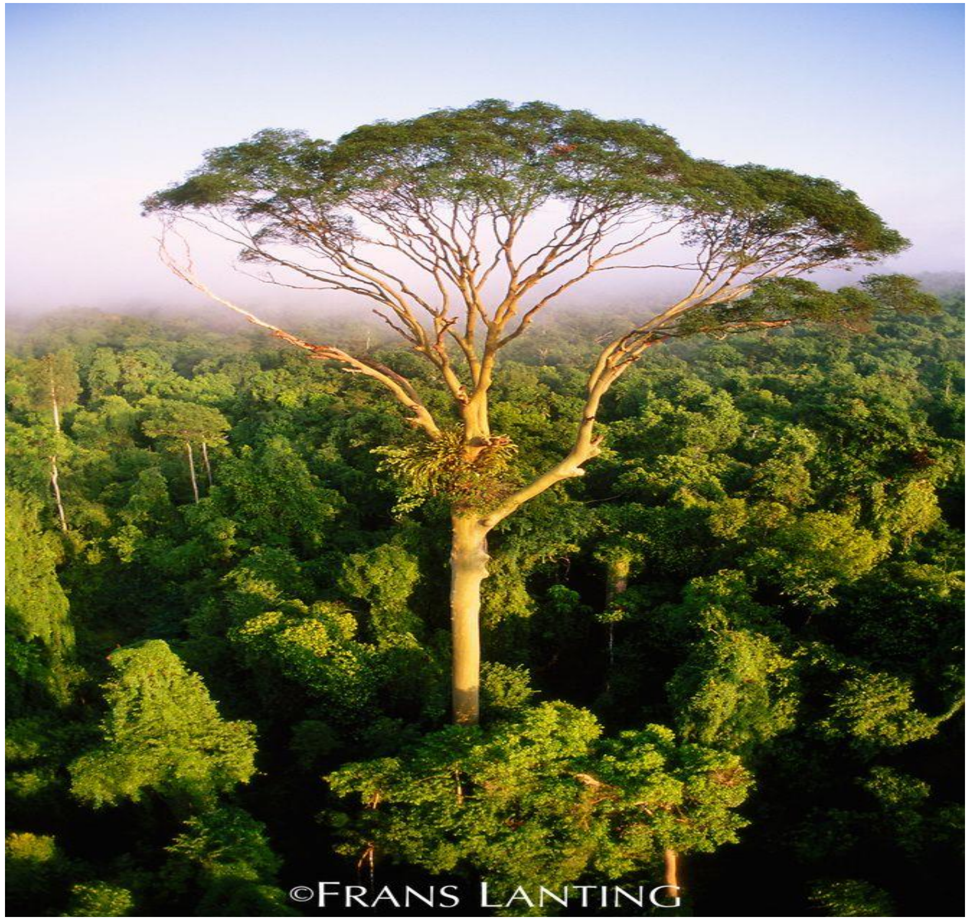
So, the biggest difference between supervised and unsupervised learning is that supervised learning deals with labeled data while unsupervised learning deals with unlabeled data. In supervised learning, we have machine learning algorithms for classification and regression.In unsupervised learning, we have methods such as clustering.

https://medium.com/@keerthanahosalli/machine-learning-1-2-e5cb5d08f7ae

29

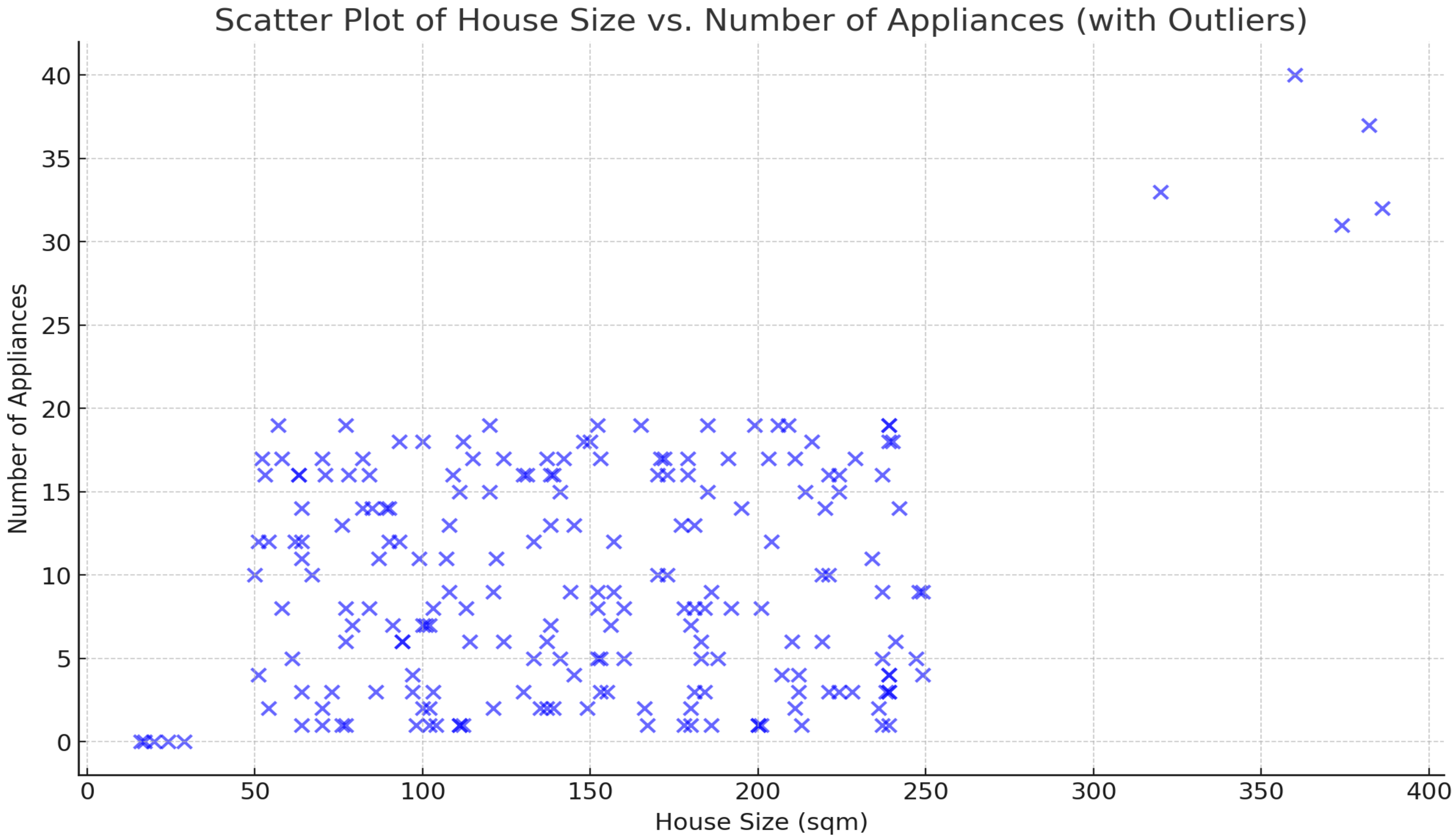

Machine Learning Algorithms
Deep Learning
Unsupervised
Supervised
Deep Boltzmann Machine (DBM)
Deep Belief Networks (DBN)
Convolutional Neural Network (CNN)
Stacked Auto-Encoders
Ensemble
Random Forest
Gradient Boosting Machines (GBM)
Boosting
Bootstrapped Aggregation (Bagging)
AdaBoost
Stacked Generalization (Blending)
Gradient Boosted Regression Trees (GBRT)
Neural Networks
Supervised
Unsupervised
Radial Basis Function Network (RBFN)
Perceptron
Back-Propagation
Hopfield Network
Regularization
Supervised
Ridge Regression
Least Absolute Shrinkage and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Rule System
Cubist
One Rule (OneR)
Zero Rule (ZeroR)
Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
Regression
Supervised
Linear Regression
Ordinary Least Squares Regression (OLSR)
Stepwise Regression
Multivariate Adaptive Regression Splines (MARS)
Locally Estimated Scatterplot Smoothing (LOESS)
Logistic Regression
Bayesian
Supervised
Naive Bayes
Averaged One-Dependence Estimators (AODE)
Bayesian Belief Network (BBN)
Gaussian Naive Bayes
Multinomial Naive Bayes
Bayesian Network (BN)
Decision Tree
Supervised
Classification and Regression Tree (CART)
Iterative Dichotomiser 3 (ID3)
C4.5
C5.0
Chi-squared Automatic Interaction Detection (CHAID)
Decision Stump
Conditional Decision Trees
M5
Dimensionality Reduction
Unsupervised
Principal Component Analysis (PCA)
Partial Least Squares Regression (PLSR
Sammon Mapping
Multidimensional Scaling (MDS)
Projection Pursuit
Principal Component Regression (PCR)
Partial Least Squares Discriminant Analysis
Mixture Discriminant Analysis (MDA)
Quadratic Discriminant Analysis (QDA)
Regularized Discriminant Analysis (RDA)
Flexible Discriminant Analysis (FDA)
Linear Discriminant Analysis (LDA)
Instance Based
Supervised
k-Nearest Neighbour (kNN)
Learning Vector Quantization (LVQ)
Self-Organizing Map (SOM)
Locally Weighted Learning (LWL)
Clustering
Unsupervised
k-Means
k-Medians
Expectation Maximization
Hierarchical Clustering

Outlier



https://www.pinterest.dk/pin/frans-lanting-sabah-borneo-tall-tree-rising-above-rainforest-canopy--121949102397261067/

*Outliers are normally removed during data processing


Scatter Plot of House Size vs. Number of Appliances (with Outliers)
Number of Appliances
House Size (sqm)
0
5
10
15
20
25
30
35
40
0
50
100
150
200
250
300
350
400

Anomaly

Abnormal Pattern

VS

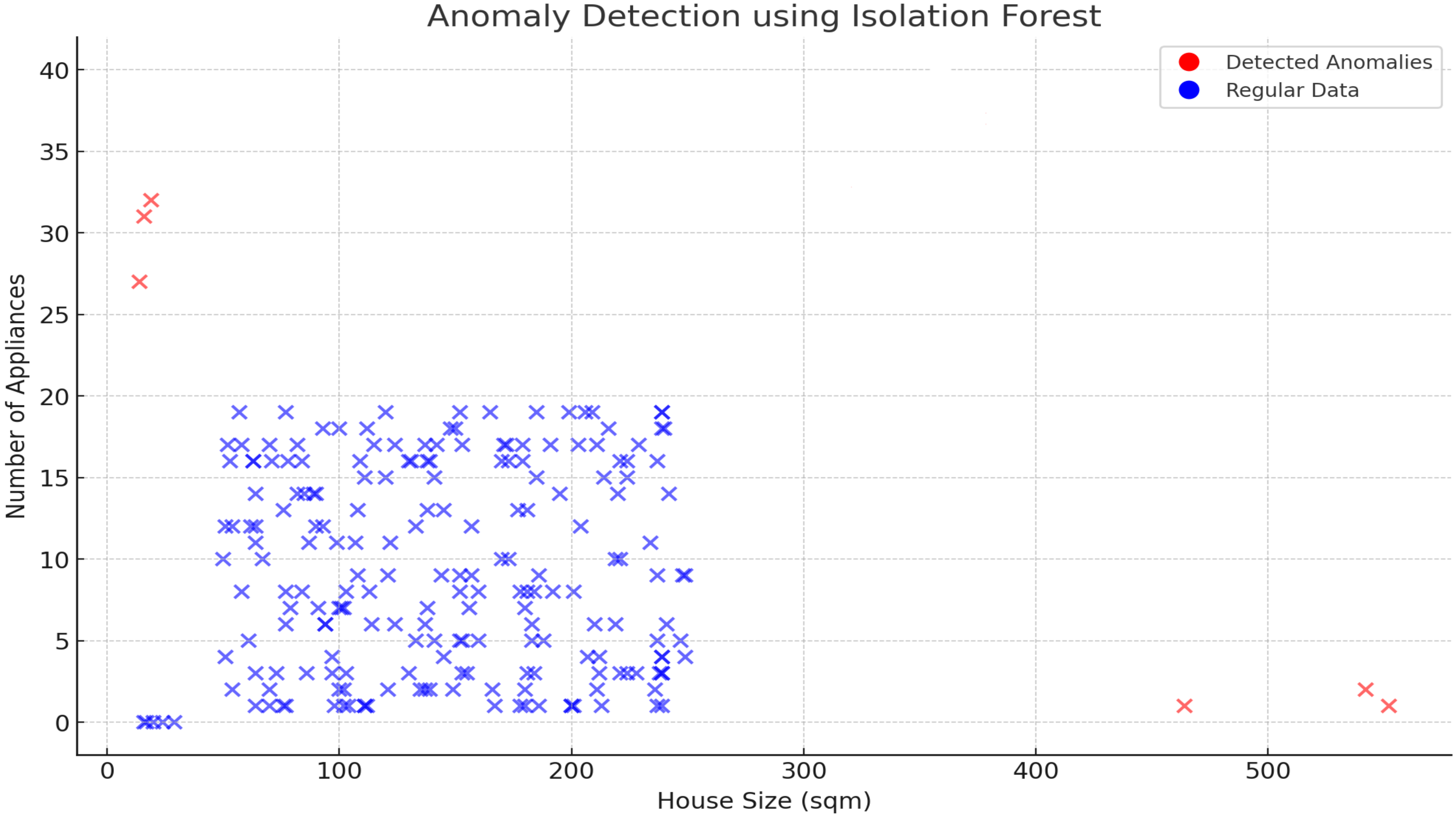
Normal Pattern

Rebild Bakker

Anomaly Detection

- Anomaly detection is a technique used to identify unusual patterns that do not conform to expected behavior. In the context of our dataset, we want to detect households that have unusual electricity consumption patterns based on their house size and the number of appliances.

House Size (sqm)	Number of Appliances	Annual Electricity Consumption (kWh)
208.9	38	1259.7
150.5	37	1054.4
68.4	12	545.0

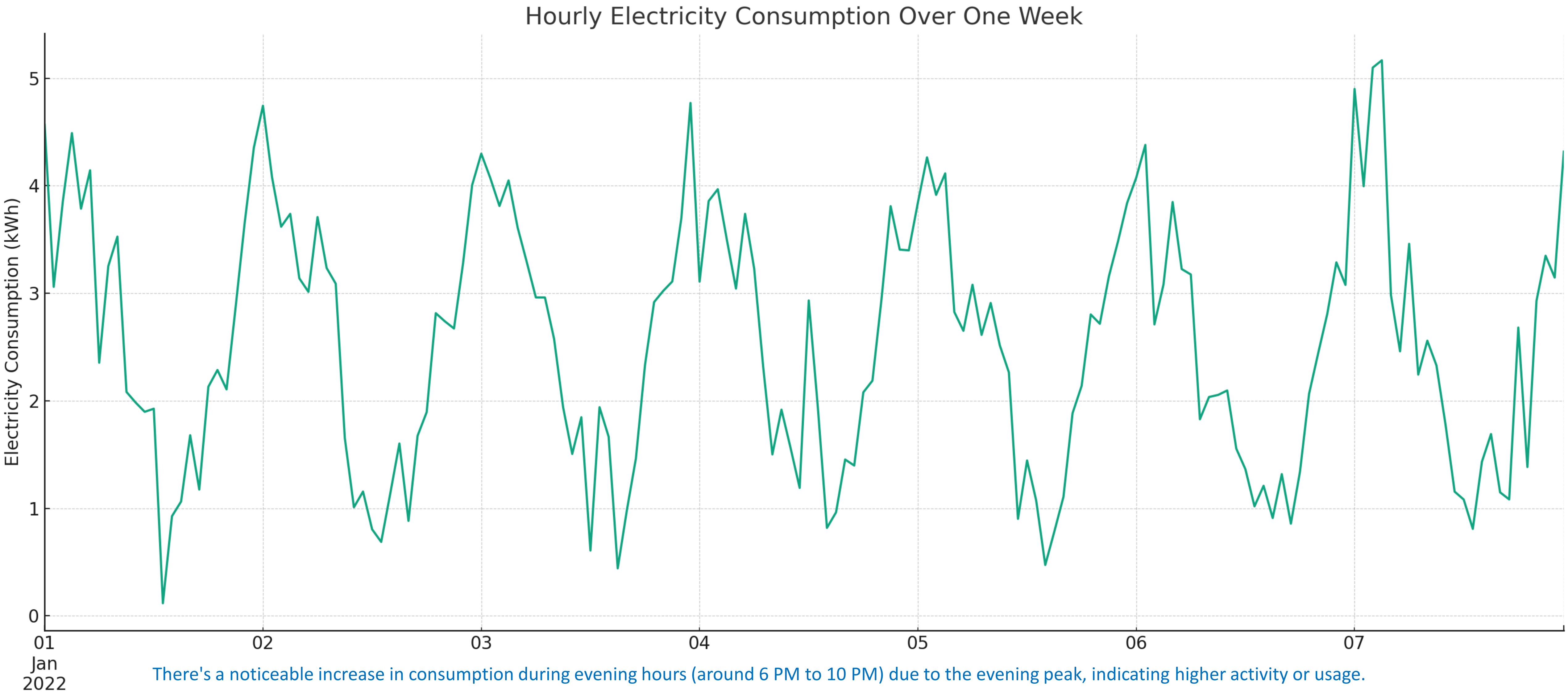

Anomaly Detection using Isolation Forest
Detected Anomalies
Regular Data
Number of Appliances
House Size (sqm)
0
5
10
15
20
25
30
35
40
0
100
200
300
400
500

Time Series

Time Series

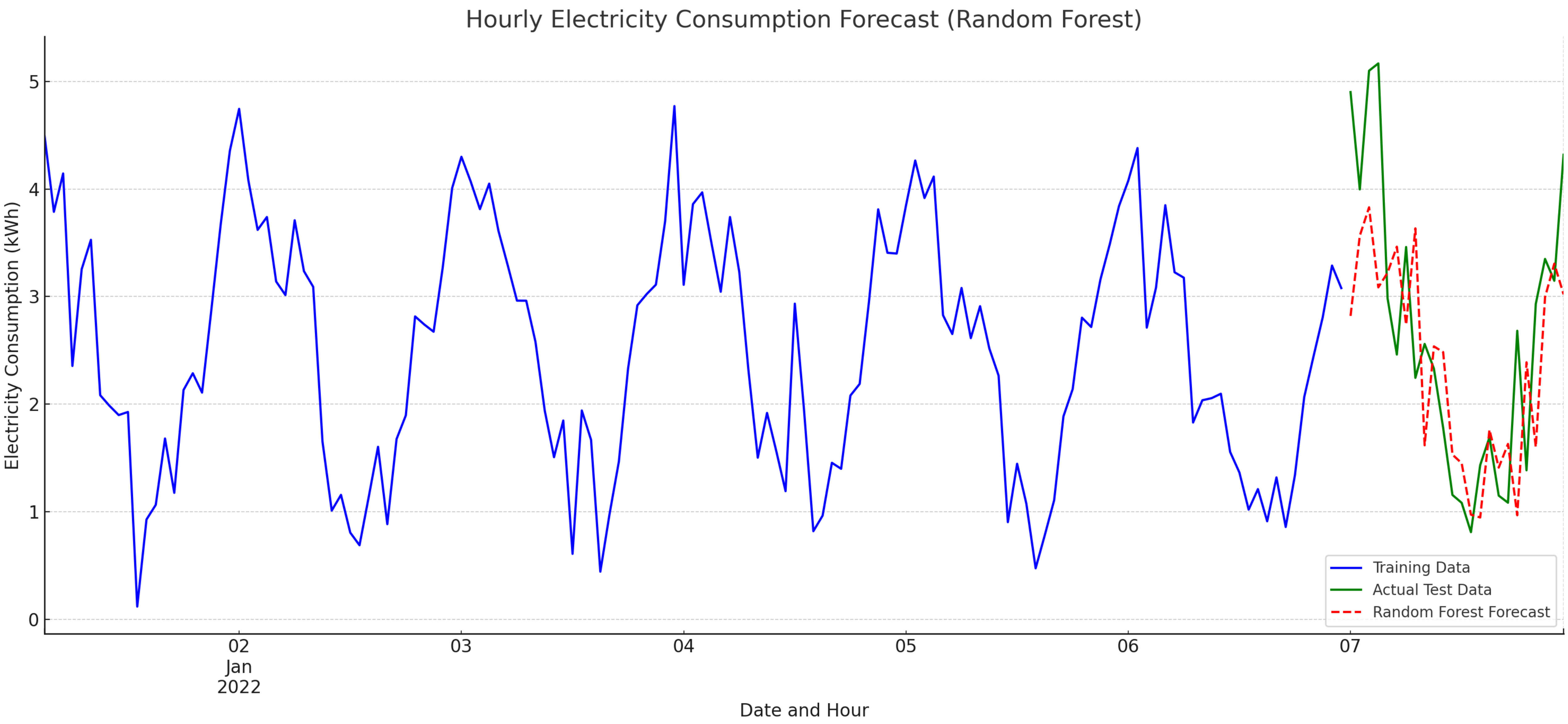
- An hourly time series dataset for a single household's electricity consumption over a period of one week (168 hours). This will give us a more granular view of consumption patterns.

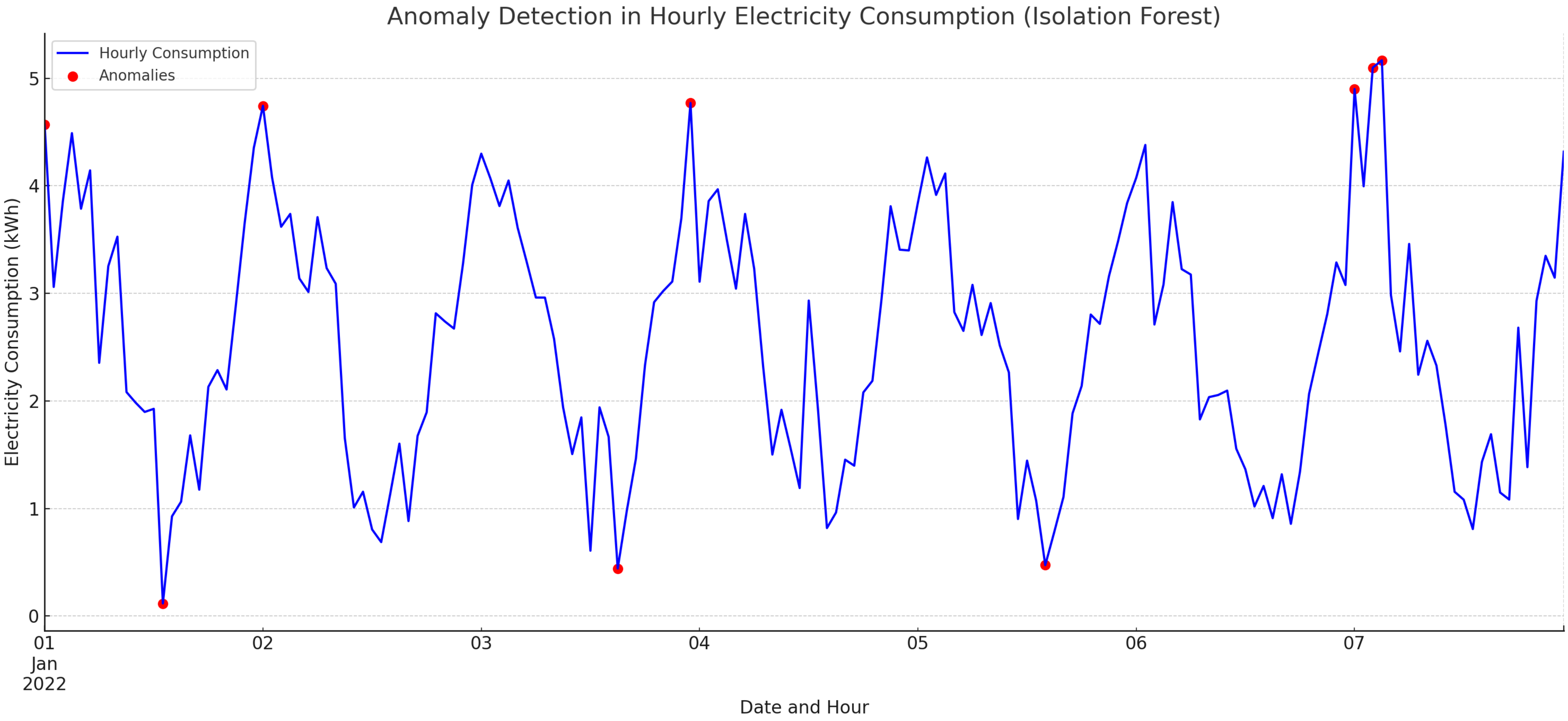
Date and Time	Hourly Consumption (kWh)
2022-01-01 00:00:00	4.567 kWh
2022-01-01 01:00:00	3.060 kWh
2022-01-01 02:00:00	3.854 kWh
2022-01-01 03:00:00	4.490 kWh
2022-01-01 04:00:00	3.787 kWh



There's a noticeable increase in consumption during evening hours (around 6 PM to 10 PM) due to the evening peak, indicating higher activity or usage.

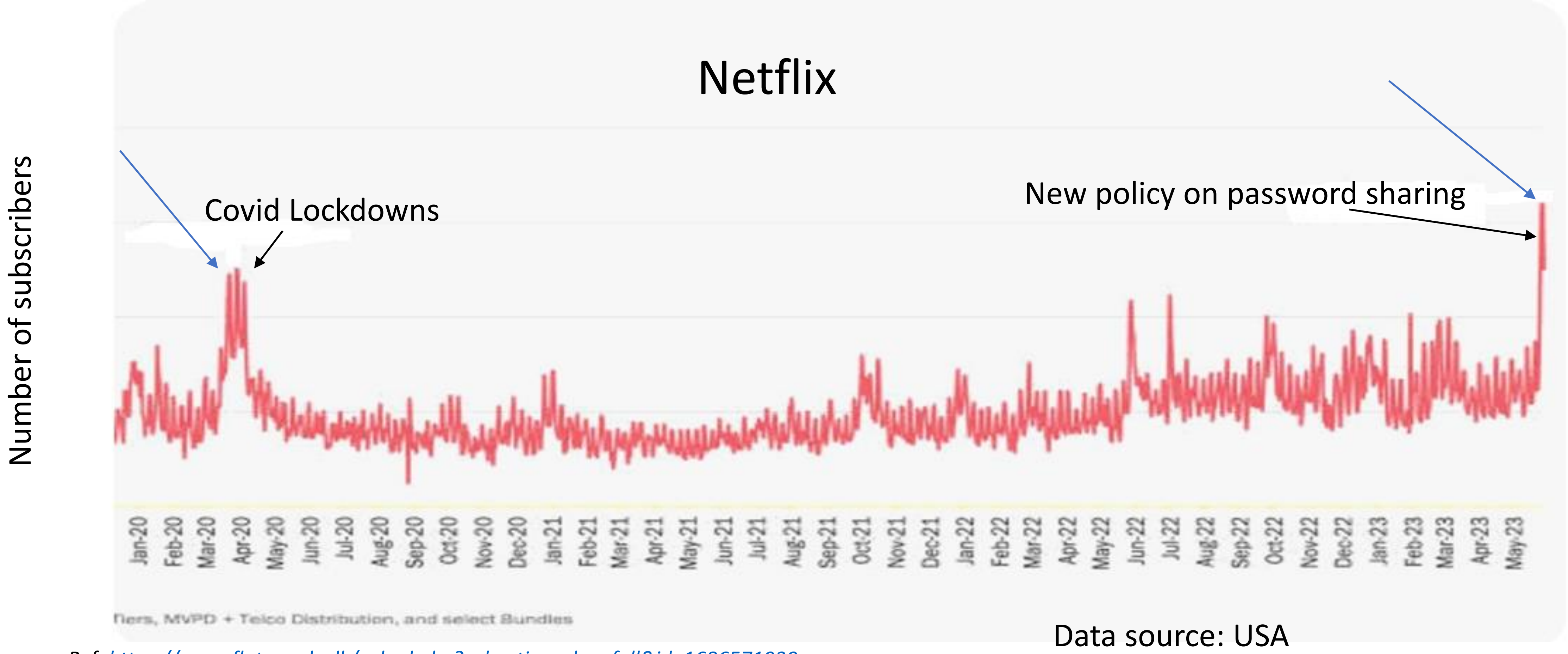
There's a dip in consumption during the early morning hours (around 1 AM to 5 AM) when most people are likely asleep, resulting in lower electricity usage.


Hourly Electricity Consumption Forecast (Random Forest)
Electricity Consumption (kWh)
0
1
2
3
4
5
02
Jan
2022
03
04
05
06
07
Date and Hour
Training Data
Actual Test Data
Random Forest Forecast



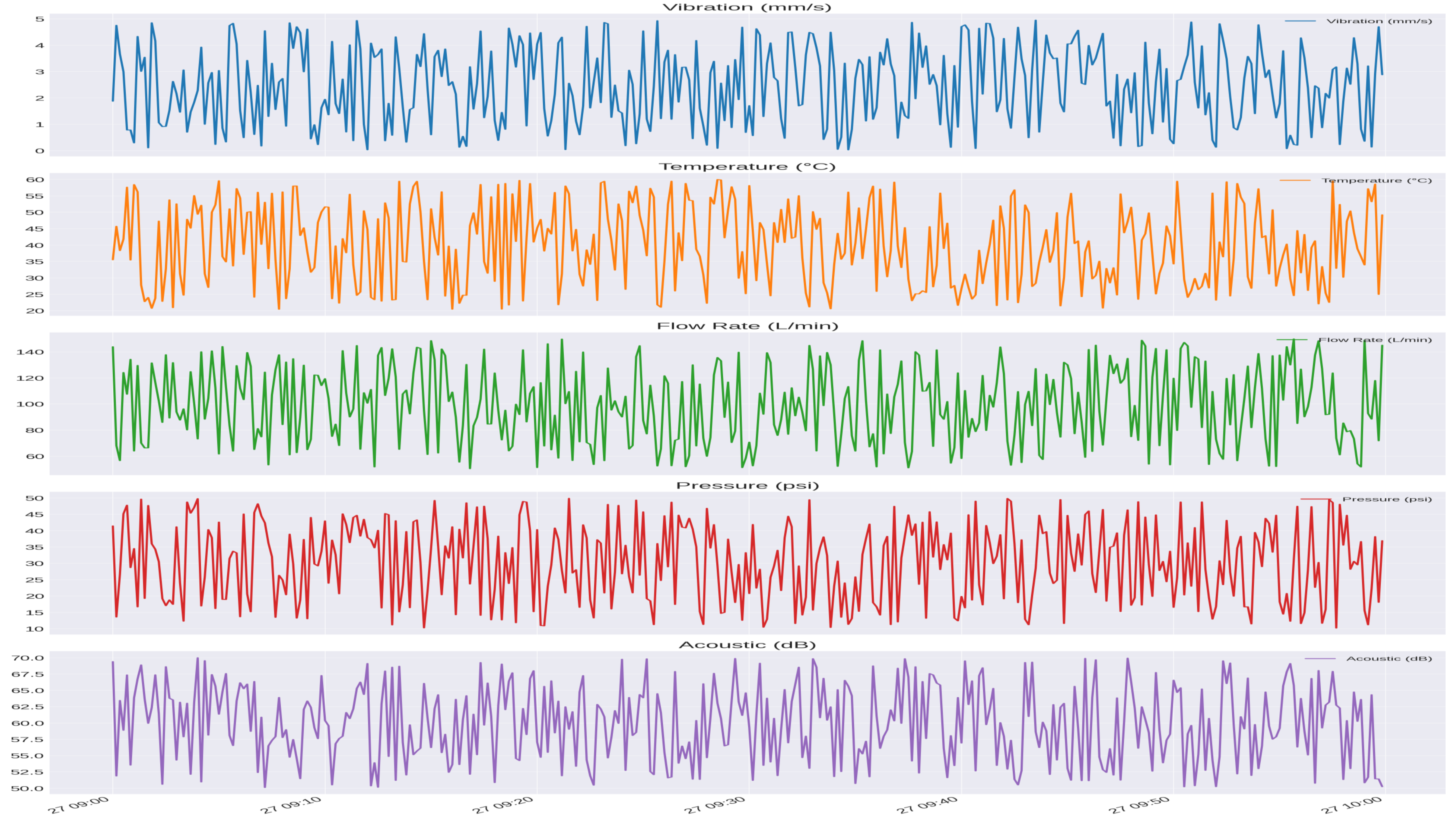
Anomaly Detection in Hourly Electricity Consumption (Isolation Forest)
Hourly Consumption
Anomalies
Electricity Consumption (kWh)
0
1
2
3
4
5
01
Jan
2022
02
03
04
05
06
07
Date and Hour

Time-series Analysis



Ref: https://www.flatpanels.dk/nyhed.php?subaction=showfull&id=1686571939

Example



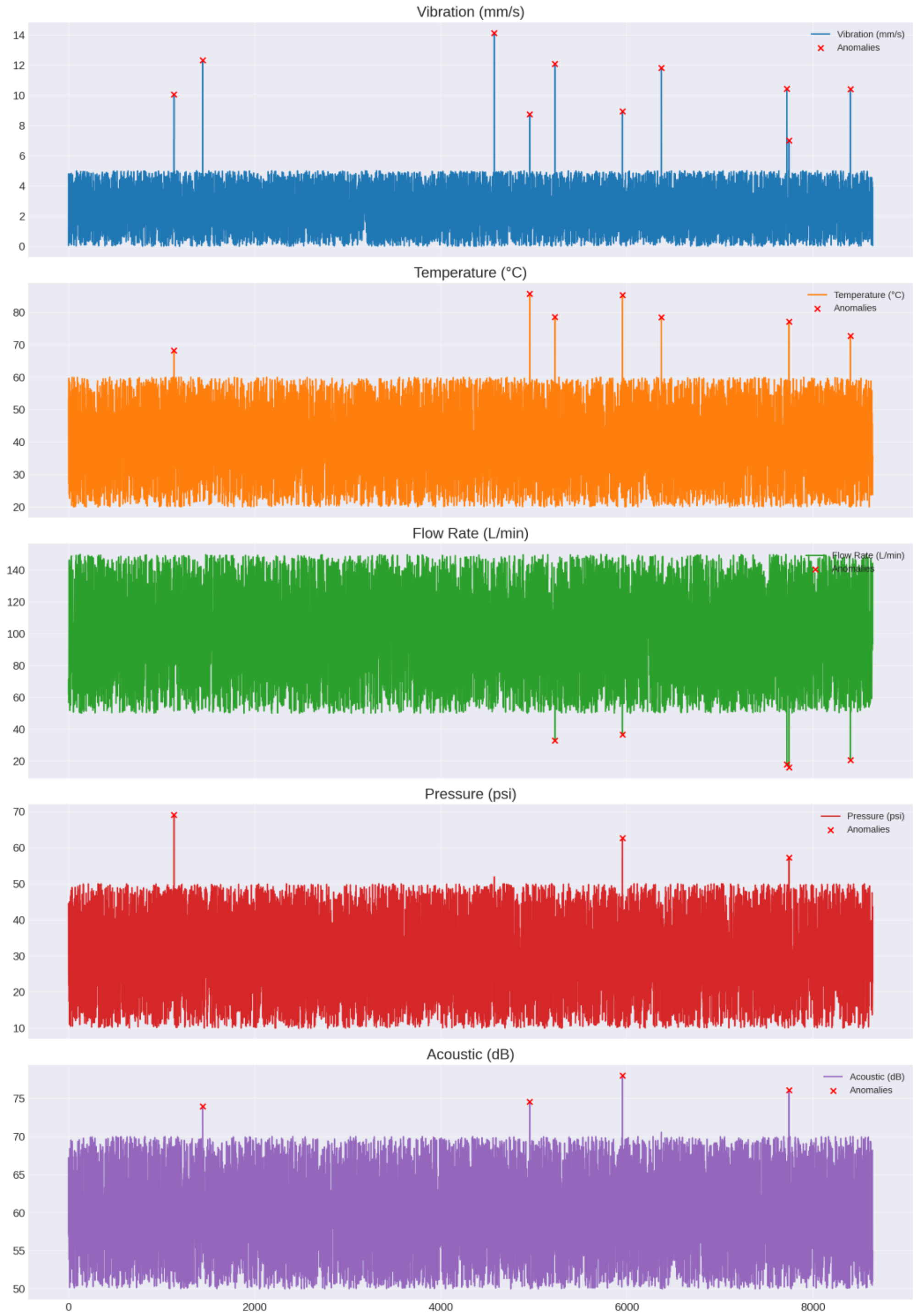
Vibration (mm/s)
Temperature (°C)
Flow Rate (L/min)
Pressure (psi)
Acoustic (dB)
27 09:00
27 09:10
27 09:20
27 09:30
27 09:40
27 09:50
27 10:00

Traditional (Non-Machine Learning) Approaches:

- **Threshold-Based Monitoring**: We could define static thresholds based on domain knowledge or by analyzing historical data. For instance, if the vibration of a machine should never exceed a certain level under normal conditions, we can set an upper limit and alert whenever this threshold is crossed.

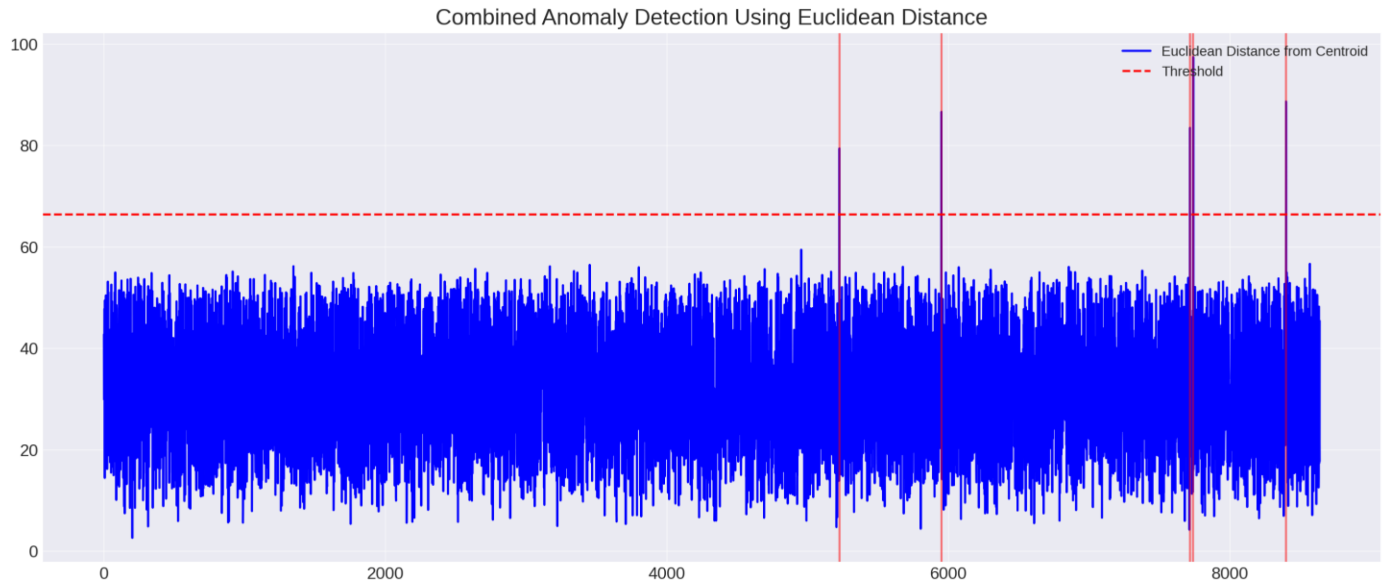
- **Statistical Methods:** Techniques like the Z-score or the IQR (Interquartile Range) can be used to detect outliers based on statistical properties of the data.

- **Rolling Statistics:** We can compute statistics (e.g., mean, standard deviation) on a rolling window and flag data points that deviate significantly from these metrics.



Threshold-Based Anomaly Detection Algorithm:

1. **Compute Mean and Standard Deviation**:
   - For each sensor, calculate the mean and standard deviation based on the "normal" data.
2. **Define Thresholds**:
   - Upper threshold: $\text{mean} + k \times \text{standard deviation}$
   - Lower threshold: $\text{mean} - k \times \text{standard deviation}$
   - Where k is a constant. We'll use $k = 2$ for our example (i.e., values beyond two standard deviations from the mean).
3. **Detect Anomalies**:
   - For each data point in the test data, check if it lies outside the defined thresholds. If it does, mark it as an anomaly.

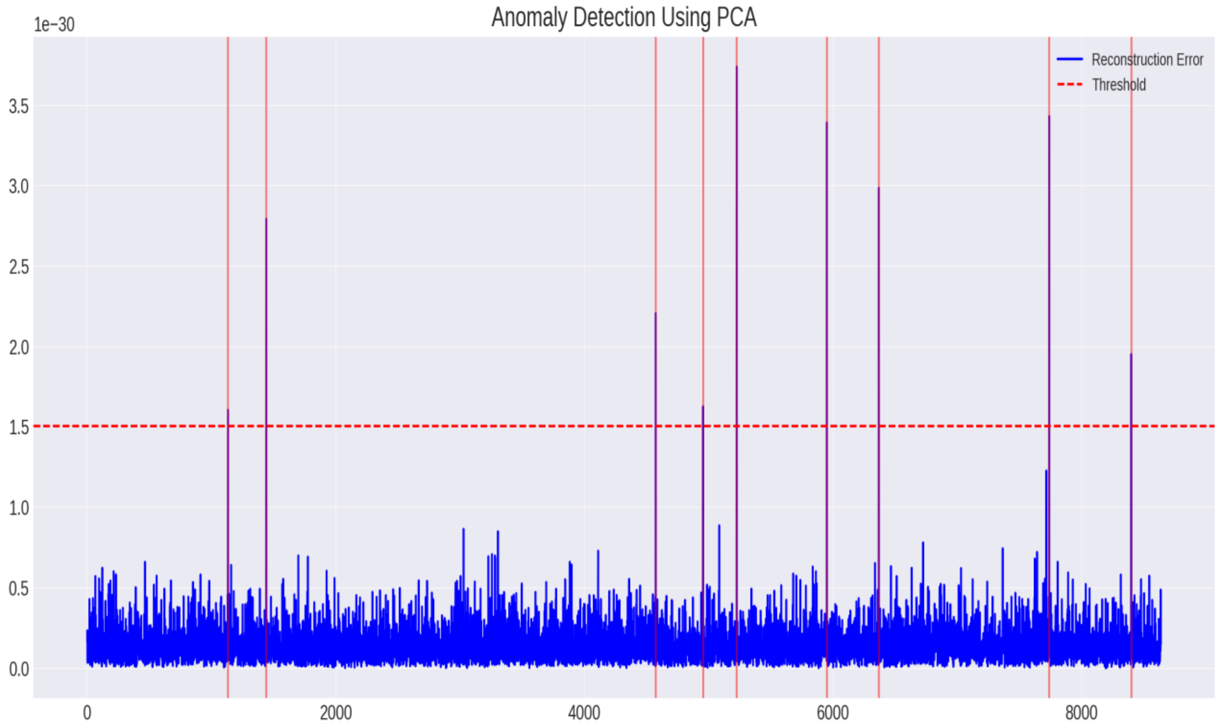


Euclidean Distance: Calculate the Euclidean distance of a test data point (a vector of sensor readings) from the centroid (mean vector) of the normal data. If this distance exceeds a predefined threshold, flag the point as an anomaly.

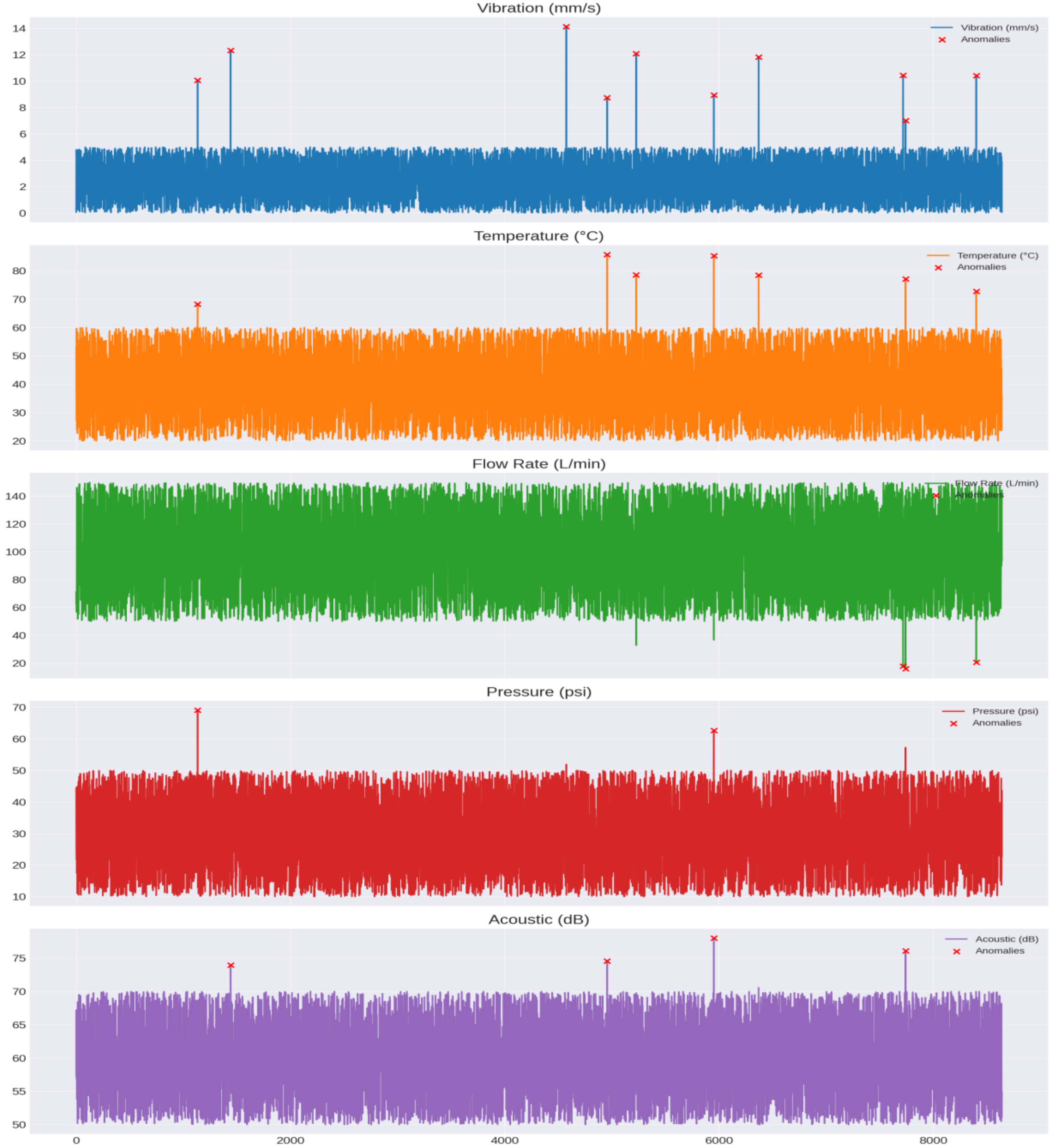
Unsupervised (Machine Learning based) Approaches:

- **Clustering**: Algorithms like K-means can be used to group data points. If a new data point doesn't belong closely to any cluster, it could be an anomaly.

- **Dimensionality Reduction**: Techniques like PCA (Principal Component Analysis) can be used to reduce the dimensionality of the dataset. Reconstruction errors can be used to detect anomalies.



DBSCAN (Density-Based Spatial Clustering of Applications with Noise)



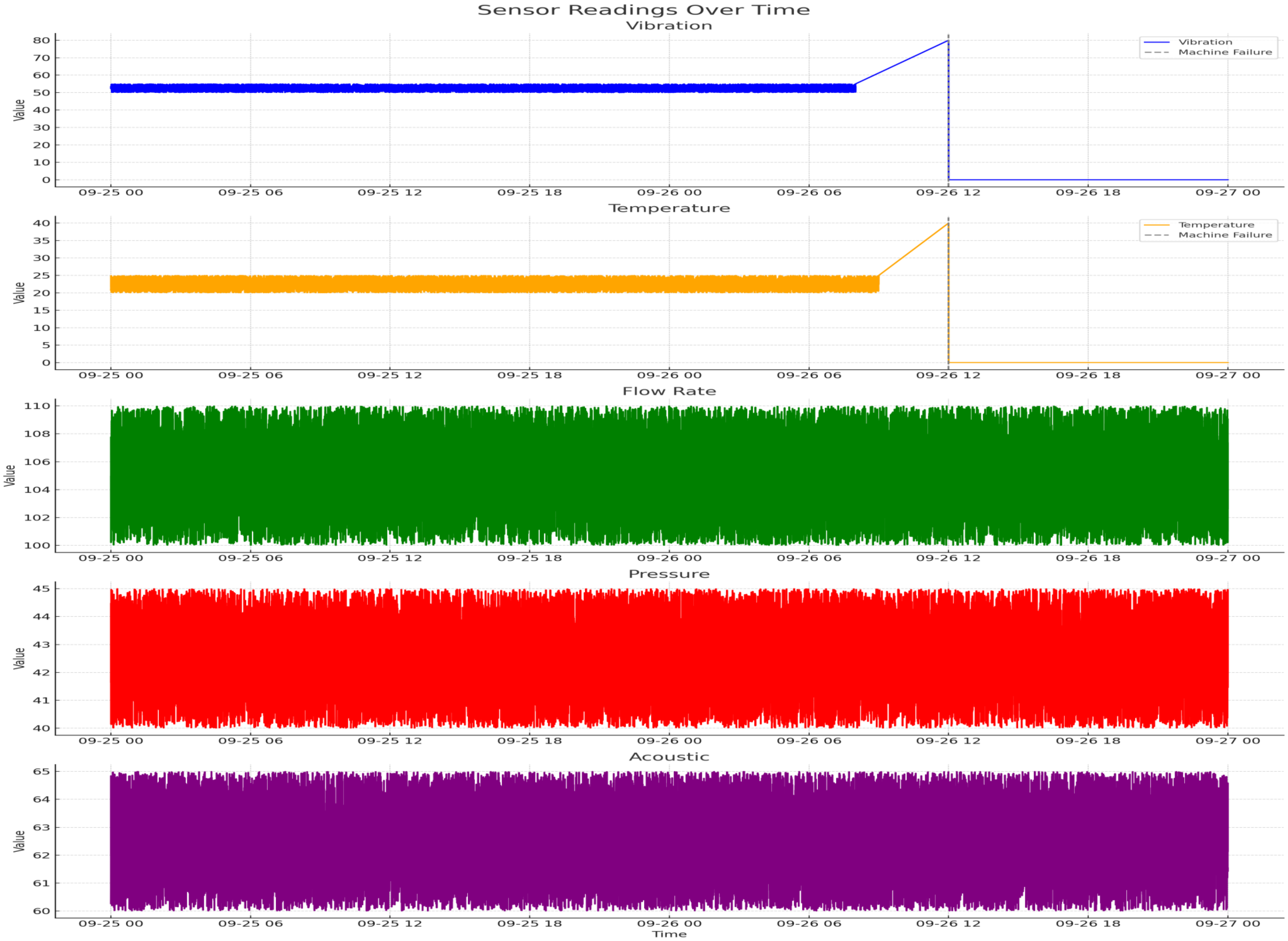
Time Series Analysis

- **Descriptive Analysis**
  - Compute mean, median, standard deviation etc. and/or identify patterns, trends and seasonality
- **Feature Engineering**
  - Rolling mean, roalling standard deviation for each feature over a window of previous observations
- **Anomaly Detection**
  - Detect unusual patterns that do not conform to expected behavior. This is essential for early fault detection in machines or systems
- **Forecasting**
  - Predict future values of the parameters based on historical data
- **Causality Analysis**
  - Understand if changes in one parameter cause changes in another (e.g., does increased pressure lead to increased temperature?)
- **Frequency Analysis**
  - Decomcope the time series into its frequency components to understand periodic behaviors better
- **Condition Monitoring & Predictive Maintenance**
  - Based on the time series data, predict when the machine might fail or require maintenance
- **Optimization**
  - Find optimal operating conditions for the machine by analyzing the relationships between the parameters
- **Simulation & What-If Analysis**
  - Simulate different scenarios to see the potential impact on the parameters (e.g., what if the pressure increases by 10%?)

Predictive Maintenance (PM)

- Is all about predicting when equipment will fail so that maintenance can be performed just in time to avoid unplanned downtime.
- The goal is to prevent unplanned reactive maintenance without incurring costs associated with doing too much preventive maintenance.

Goal: predicts if a machine is about to fail in the near future (e.g., in the next *20 minutes*)


Sensor Readings Over Time
Vibration
Vibration
Machine Failure
Temperature
Temperature
Machine Failure
Flow Rate
Pressure
Acoustic
Value
Time
09-25 00
09-25 06
09-25 12
09-25 18
09-26 00
09-26 06
09-26 12
09-26 18
09-27 00

Predictive Maintenance Algorithm:

1. Data Collection:

Let D be our dataset where each observation d_i is a tuple (t, v, T, a, p, f_r):

$$D = \{d_1, d_2, ..., d_n\}$$

Where:

- t is the timestamp.
- v is the vibration reading.
- T is the temperature reading.
- a is the acoustic reading.
- p is the pressure reading.
- f_r is the flow rate reading.

2. Labeling:

For each observation d_i at time t, assign a label y_i based on whether a machine failure occurs within a certain window W following that observation:

$$y_i = \begin{cases} 1 & \text{if machine fails within window } W \text{ after } t \\ 0 & \text{otherwise} \end{cases}$$

3. Feature Engineering:

For each observation d_i at time t, compute rolling mean and standard deviation for all the sensor readings over different windows w. For instance, for the vibration reading:

$$\text{RollingMean}(v, w) = \frac{1}{w} \sum_{j=i-w+1}^{i} v_j$$

$$\text{RollingStd}(v, w) = \sqrt{\frac{1}{w} \sum_{j=i-w+1}^{i} (v_j - \text{RollingMean}(v, w))^2}$$

Similar calculations will be made for T, a, p, and f_r.

4. Model Training:

Train a model M using features from the feature engineering step and labels y:

$$M : f(D) \rightarrow y$$

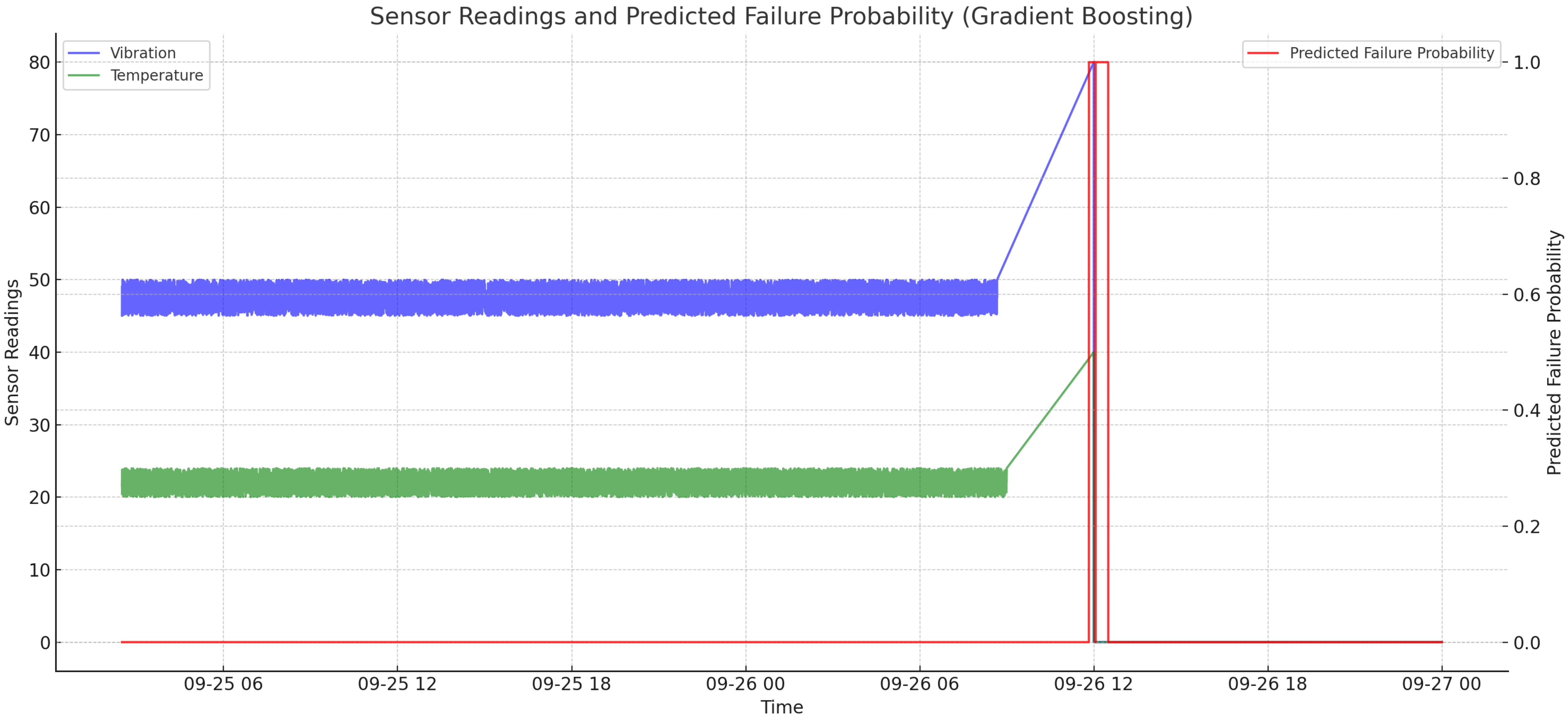
Where $f(D)$ represents the feature set derived from D.

5. Prediction:

For a new observation d without a label, the predicted failure probability P is given by:

$$P(d) = M(f(d))$$

If $P(d) > \theta$, where θ is a threshold, raise an alert indicating imminent failure.


Sensor Readings and Predicted Failure Probability (Gradient Boosting)
Vibration
Temperature
Predicted Failure Probability
Sensor Readings
Predicted Failure Probability
Time
0
10
20
30
40
50
60
70
80
0.0
0.2
0.4
0.6
0.8
1.0
09-25 06
09-25 12
09-25 18
09-26 00
09-26 06
09-26 12
09-26 18
09-27 00

Composite Health Index

- The Composite Health Index (CHI) is a synthesized metric derived from multiple sensor readings or diagnostic indicators to represent the overall health or condition of a system, machine, or equipment. The primary purpose of CHI is to simplify the multidimensional health information of a system into a single, easily interpretable metric

Composite Health Index Calculation:

The CHI at time t, denoted by $\mathrm{CHI}(t)$, is a combination of the standardized readings from all sensors. A simple approach, which we've used, is to take the arithmetic mean:

$$\mathrm{CHI}(t) = \frac{1}{m} \sum_{i=1}^{m} s_i'(t)$$

Here, m is the total number of sensors.

This CHI represents a single metric derived from all sensors that indicates the health status of the machine at time t.

Case Studies with SMVs

Case 1: DOLLE Loftladder

Issue:

Based on historical consequences: Decision of when and how to prevent break down

Stigemaskine 1 (Phoenix radioline)

Maskine: 01

Input word 1, Binær: 0000000000100101

Indgang 0101:	Maskine er startet
Indgang 0102:	Taktføler, ved indløb (B95.0)
Indgang 0103:	Taktføler ved udløb (B157.6)
Indgang 0104:	Skæv vange ved bore- eller møtrikstation
Indgang 0105:	Fejl på en af skruemaskinerne
Indgang 0106:	Alarm på stigemaskine

Date	Time	Indgang 0101	Indgang 0102	Indgang 0103	Indgang 0104	Indgang 0105	Indgang 0106
31-01-2018	15:37:56	0	0	0	0	0	0
31-01-2018	15:42:26	1	0	0	0	0	0
31-01-2018	15:43:08	1	1	0	0	0	0
31-01-2018	15:43:09	1	0	0	0	0	0
31-01-2018	15:43:15	1	1	0	0	0	0
31-01-2018	15:43:17	1	0	0	0	0	0
31-01-2018	15:43:23	1	1	0	0	0	0
31-01-2018	15:43:25	1	0	0	0	0	0
31-01-2018	15:43:31	1	1	0	0	0	0
31-01-2018	15:43:33	1	0	0	0	0	0
31-01-2018	15:43:38	1	1	0	0	0	0
31-01-2018	15:43:39	1	0	0	0	0	0
31-01-2018	15:43:55	1	1	0	0	0	0
31-01-2018	15:44:03	1	0	0	0	0	0
31-01-2018	15:44:06	1	1	0	0	0	0
31-01-2018	15:44:11	1	0	0	0	0	0
31-01-2018	15:44:27	0	0	0	0	0	0
31-01-2018	15:45:07	1	0	0	0	0	0
31-01-2018	15:47:52	0	0	0	0	0	0
31-01-2018	15:49:34	1	0	0	0	0	0
31-01-2018	15:49:41	1	1	0	0	0	0
31-01-2018	15:49:52	1	0	0	0	0	0
31-01-2018	15:49:55	1	1	0	0	0	0
31-01-2018	15:49:59	1	0	0	0	0	0
31-01-2018	15:50:02	1	1	0	0	0	0
31-01-2018	15:50:04	1	0	0	0	0	0
31-01-2018	15:50:09	1	1	0	0	0	0
31-01-2018	15:50:11	1	0	0	0	0	0
31-01-2018	15:50:17	1	1	0	0	0	0
31-01-2018	15:50:19	1	0	0	0	0	0
31-01-2018	15:52:38	1	0	1	0	0	0
31-01-2018	15:52:42	1	0	0	0	0	0
31-01-2018	15:53:03	1	0	1	0	0	0
31-01-2018	15:53:09	1	0	0	0	0	0
31-01-2018	15:53:32	1	0	1	0	0	0
31-01-2018	15:53:36	1	0	0	0	0	0
31-01-2018	15:53:57	1	0	1	0	0	0
31-01-2018	15:54:03	1	0	0	0	0	0

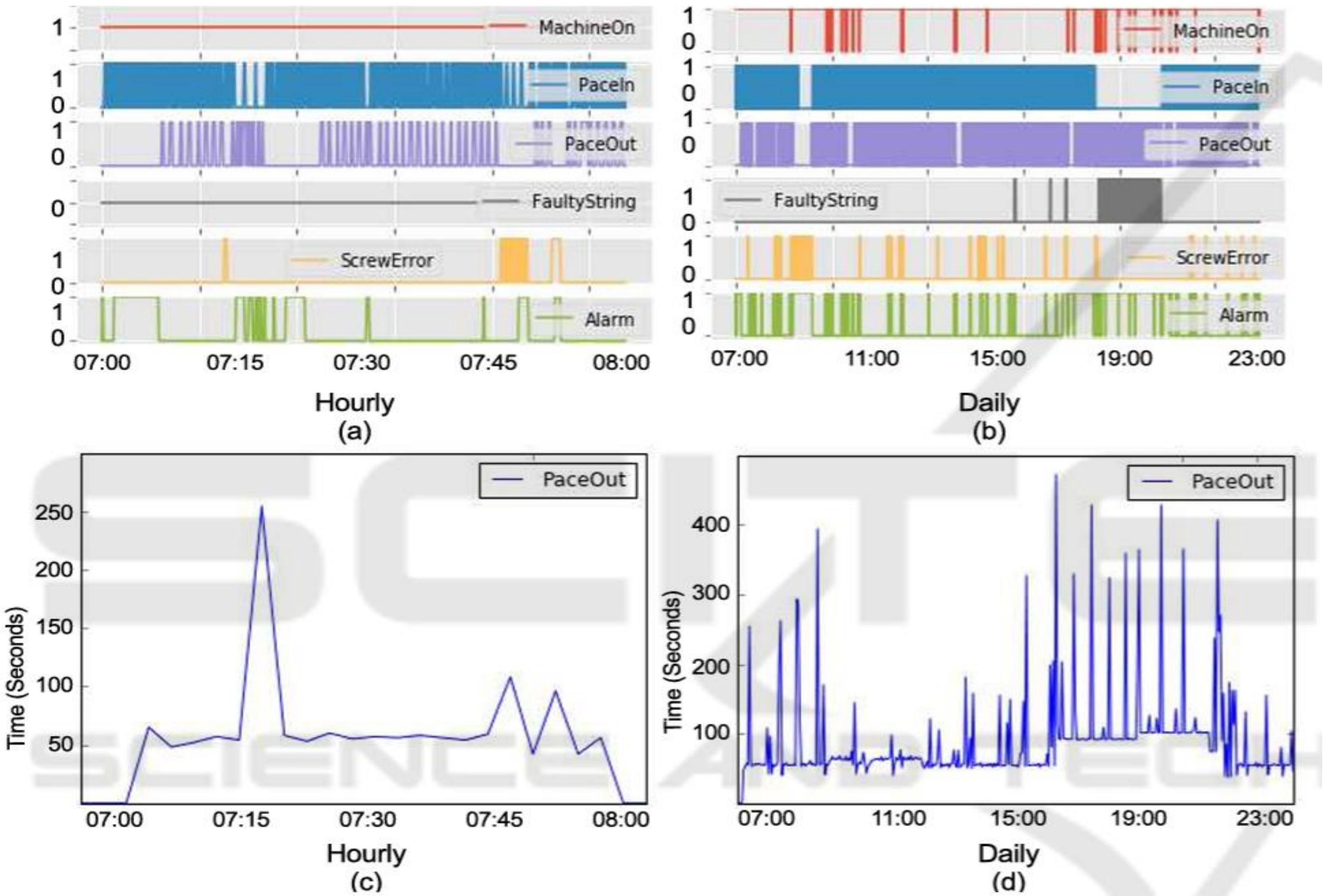


Figure 3: Sensor and alarm data overview.

Iftikhar, N.; Baattrup-Andersen, T.; Nordbjerg, F.; Bobolea, E. and Radu, P. (2019). **Data Analytics for Smart Manufacturing: A Case Study**. In *Proceedings of the 8th International Conference on Data Science, Technology and Applications - DATA*; ISBN 978-989-758-377-3; ISSN 2184-285X, SciTePress, pages 392-399. DOI: 10.5220/0008116203920399

Input:

- **Dataset D**: A set of ordered tuples $(t_1, m_1, p_{in1}, p_{out1}, f_1, s_1, a_1, h_1), (t_2, m_2, p_{in2}, p_{out2}, f_2, s_2, a_2, h_2), \ldots$ where:
  - t represents the timestamp.
  - m represents the 'MachineON/OFF' attribute.
  - p_{in} represents the 'PaceIn' attribute.
  - p_{out} represents the 'PaceOut' attribute.
  - f represents the 'FaultyString' attribute.
  - s represents the 'ScrewMachineError' attribute.
  - a represents the 'Alarm' attribute.
  - h represents the 'HourOfDay' attribute.
- **Window Size w**: An integer representing the size of the rolling window.
- **Prediction Lead Time k**: An integer representing the desired lead time for predictions.

Output:

- A predictive model M trained to forecast machine stops k minutes in advance using the rolling window features.

Method:

1. **Ensure Chronological Ordering**:
   - Arrange the dataset D in increasing order based on the timestamp t.
2. **Compute Rolling Window Features**:
   - For each attribute x in the dataset D and for each timestamp t_i:
     - **Rolling Mean**: $\mu_{x,t_i} = \frac{1}{w} \sum_{j=0}^{w-1} x_{t_{i-j}}$
     - **Rolling Standard Deviation**: $\sigma_{x,t_i} = \sqrt{\frac{1}{w} \sum_{j=0}^{w-1} (x_{t_{i-j}} - \mu_{x,t_i})^2}$
     - For binary attributes, **Rolling Sum**: $\Sigma_{x,t_i} = \sum_{j=0}^{w-1} x_{t_{i-j}}$
3. **Prepare the Target Variable**:
   - Define the future machine status as: $m'_{t_i} = m_{t_{i+k}}$.
4. **Data Partitioning**:
   - Divide D into two subsets: Training dataset D_{train} (comprising approximately 80% of the tuples) and Test dataset D_{test} (comprising the remaining 20%).
5. **Model Selection, Training, and Evaluation**:
   - Choose an appropriate machine learning model M.
   - Train M using the rolling window features from D_{train} to predict m'.
   - Evaluate M on D_{test} to measure its predictive accuracy.

	00:00	00:01	00:02	00:03	00:04	00:05	00:06	00:07	00:08	00:09	00:10	00:11	00:12	00:13	00:14	00:15	00:16	00:17	00:18	00:19	00:20	00:21	00:22	00:23	00:24	00:25	00:26	00:27	00:28	00:29	00:30	00:31	00:32	00:33	00:34	00:35
1	0	1	0	1	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0	0	1	0	0	1	1	1	0	0	0	1	1	0	0
2	0	1	0	1	1	1	0	0	0	1	1	0	1	0	0	0	1	0	0	1	1	0	1	1	1	1	1	1	0	0	1	0	1	1	0	0
3	0	1	0	1	1	1	0	0	1	0	0	1	1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	1	1	0	1	1	1	0
4	1	1	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	1	1	0	1	1	0	0	0	1	1	0	1	0	1	1	0
5	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	1

	00:36	00:37	00:38	00:39	00:40	00:41	00:42	00:43	00:44	00:45	00:46	00:47	00:48	00:49	00:50	00:51	00:52	00:53	00:54	00:55	00:56	00:57	00:58	00:59	01:00	01:01	01:02	01:03	01:04	01:05	01:06	01:07	01:08	01:09	01:10	01:11
1	1	1	1	1	0	0	1	0	1	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	1	0	1	1	0	0	1	1	1	1
2	1	0	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	1	0	1	1	1	1
3	1	1	1	0	1	1	1	0	1	1	1	1	0	0	1	1	0	1	0	0	1	1	1	1	0	1	0	0	1	1	0	1	0	0	0	1
4	0	0	0	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	1	1
5	1	0	0	1	0	1	0	0	1	1	0	0	1	1	1	1	0	1	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	0	1	1

	00:00	00:01	00:02	00:03	00:04	00:05	00:06	00:07	00:08	00:09	00:10	00:11	00:12	00:13	00:14	00:15	00:16	00:17	00:18	00:19	00:20	00:21	00:22	00:23	00:24	00:25	00:26	00:27	00:28	00:29	00:30	00:31	00:32	00:33	00:34	00:35
1	0	1	0	1	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0	0	1	0	0	1	1	1	0	0	0	1	1	0	0
2	0	1	0	1	1	1	0	0	0	1	1	0	1	0	0	0	1	0	0	1	1	0	1	1	1	1	1	1	0	0	1	0	1	1	0	0
3	0	1	0	1	1	1	0	0	1	0	0	1	1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	1	1	0	1	1	1	0
4	1	1	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	1	1	0	1	1	0	0	0	1	1	0	1	0	1	1	0
5	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	1

	00:36	00:37	00:38	00:39	00:40	00:41	00:42	00:43	00:44	00:45	00:46	00:47	00:48	00:49	00:50	00:51	00:52	00:53	00:54	00:55	00:56	00:57	00:58	00:59	01:00	01:01	01:02	01:03	01:04	01:05	01:06	01:07	01:08	01:09	01:10	01:11
1	1	1	1	1	0	0	1	0	1	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	1	0	1	1	0	0	1	1	1	1
2	1	0	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	1	0	1	1	1	1
3	1	1	1	0	1	1	1	0	1	1	1	1	0	0	1	1	0	1	0	0	1	1	1	1	0	1	0	0	1	1	0	1	0	0	0	1
4	0	0	0	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	1	1
5	1	0	0	1	0	1	0	0	1	1	0	0	1	1	1	1	0	1	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	0	1	1

	00:00	00:01	00:02	00:03	00:04	00:05	00:06	00:07	00:08	00:09	00:10	00:11	00:12	00:13	00:14	00:15	00:16	00:17	00:18	00:19	00:20	00:21	00:22	00:23	00:24	00:25	00:26	00:27	00:28	00:29	00:30	00:31	00:32	00:33	00:34	00:35
1	0	1	0	1	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0	0	1	0	0	1	1	1	0	0	0	1	1	0	0
2	0	1	0	1	1	1	0	0	0	1	1	0	1	0	0	0	1	0	0	1	1	0	1	1	1	1	1	1	0	0	1	0	1	1	0	0
3	0	1	0	1	1	1	0	0	1	0	0	1	1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	1	1	0	1	1	1	0
4	1	1	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	1	1	0	1	1	0	0	0	1	1	0	1	0	1	1	0
5	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	1

	00:36	00:37	00:38	00:39	00:40	00:41	00:42	00:43	00:44	00:45	00:46	00:47	00:48	00:49	00:50	00:51	00:52	00:53	00:54	00:55	00:56	00:57	00:58	00:59	01:00	01:01	01:02	01:03	01:04	01:05	01:06	01:07	01:08	01:09	01:10	01:11
1	1	1	1	1	0	0	1	0	1	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	1	0	1	1	0	0	1	1	1	1
2	1	0	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	1	0	1	1	1	1
3	1	1	1	0	1	1	1	0	1	1	1	1	0	0	1	1	0	1	0	0	1	1	1	1	0	1	0	0	1	1	0	1	0	0	0	1
4	0	0	0	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0	1	1
5	1	0	0	1	0	1	0	0	1	1	0	0	1	1	1	1	0	1	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	0	1	1

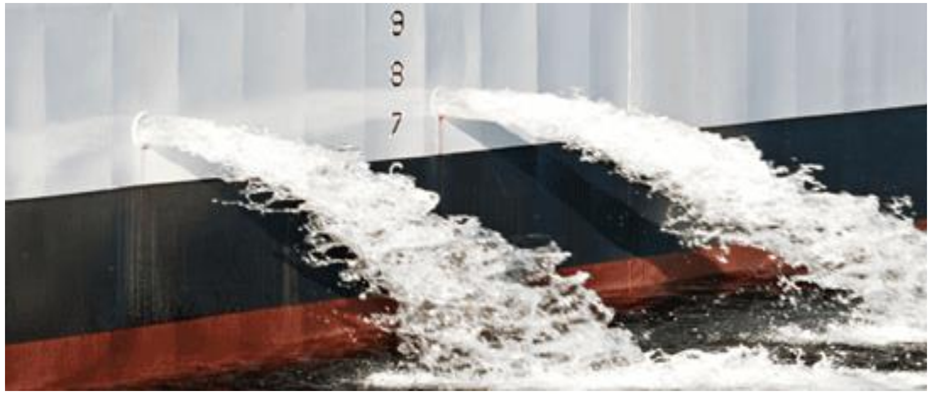
Iftikhar, N., Nordbjerg, F.E., Baattrup-Andersen, T. and Jeppesen, K., 2020. **Industry 4.0: sensor data analysis using machine learning**. In Data Management Technologies and Applications: 8th International Conference, DATA 2019, Prague, Czech Republic, July 26–28, 2019, Revised Selected Papers 8 (pp. 37-58). Springer International Publishing.

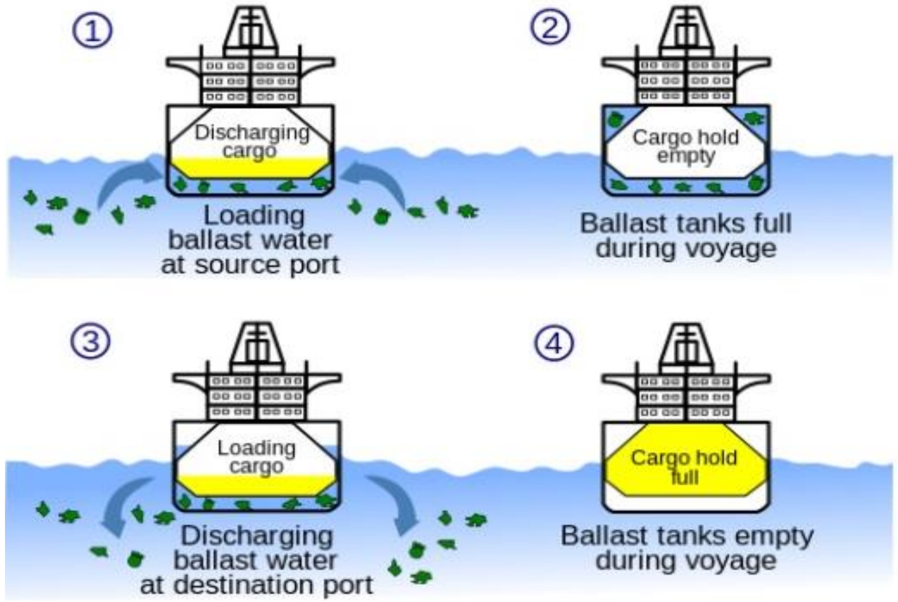
Timestamp	Engine_Temperature (°C)	Oil_Pressure (psi)	Fuel_Level (%)	Battery_Voltage (V)	Car_Status	**Shifted_Car_Status (Target)**
2023-01-01 08:00	90.5	40	30	5.5	1	**0** # Predicted breakdown 60 minutes in advance
2023-01-01 08:10	91.0	41	29	4.6	1	**0**
2023-01-01 08:20	92.0	42	28	3.7	1	**0**
2023-01-01 08:30	93.5	43	27	2.8	1	**0**
2023-01-01 08:40	95.0	44	26	1.9	1	**0**
2023-01-01 08:50	96.0	45	25	0.0	**0**	**0** # Car breaks down
2023-01-01 11:00	60.0	30	24	50.0	1	1 # Car restarts
2023-01-01 11:10	61.0	30	23	49.0	1	1
2023-01-01 11:20	62.0	31	22	48.1	1	1
2023-01-01 11:30	63.0	32	21	47.4	1	1
2023-01-01 11:40	64.0	32	20	46.5	1	1

Case 2: DESMI Ocean Guard A/S

- *ballast water treatment systems*

(Specializes in the development, manufacturing, sale and service of IMO and USCG certified Ballast Water Management & Treatment Systems)





- While ballast water is essential for safe and efficient modern shipping operations, it may pose serious ecological, economic and health problems due to the multitude of marine species carried in ships' ballast water. These include bacteria, microbes, small invertebrates, eggs, cysts and larvae of various species.
- **Issue: How it is made sure that water that is discharged into the sea is clean enough based on IMO & UGCS standards.**

Proposed Solution:
We proposed a solution based on *online machine learning* to make sure that the water that is released into the sea has fulfilled the International Maritime Organization (IMO) & United State Geological Survey (USCG) standards (i.e. the filtration plant is delivering optimal performance).

Case 2 Continuted...

Online Machine Learning for Adaptive Ballast Water Management

Nadeem Iftikhar
University College of Northern Denmark
Aalborg, Denmark
naif@ucn.dk

Yi-Chen Lin
University College of Northern Denmark
Aalborg, Denmark
yichenlintaiwan@gmail.com

Xiufeng Liu
Technical University of Denmark
Kgs. Lyngby, Denmark
xiuli@dtu.dk

Finn Ebertsen Nordbjerg
University College of Northern Denmark
Aalborg, Denmark
fen@ucn.dk

Algorithm 1: Online Machine Learning with Model Switching

Input: A stream of sensor data $X = \{x_1, x_2, ..., x_n\}$ from ships and ports, a training strategy S
Output: A stream of predictions $\hat{Y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_n\}$ and their confidence intervals $\hat{C} = \{\hat{c}_1, \hat{c}_2, ..., \hat{c}_n\}$

```
Initialize a set of candidate machine learning models M with
  random parameters θ;
Initialize a null best model m*;
Initialize a null training trigger T;
for i = 1 to n do
    Receive a new input x_i;
    Predict the output ŷ_i = m*(x_i; θ) and its confidence
      interval ĉ_i using the best model;
    Output ŷ_i and ĉ_i;
    Update the training trigger T based on the training
      strategy S;
    if T is activated then
        Train or update the models M using the available
          data (X, Y);
        Update the parameters θ;
        Evaluate the models M using different metrics;
        Select the best model m* from M based on the
          metrics and the suitability for the ship-port pair;
    end
end
```

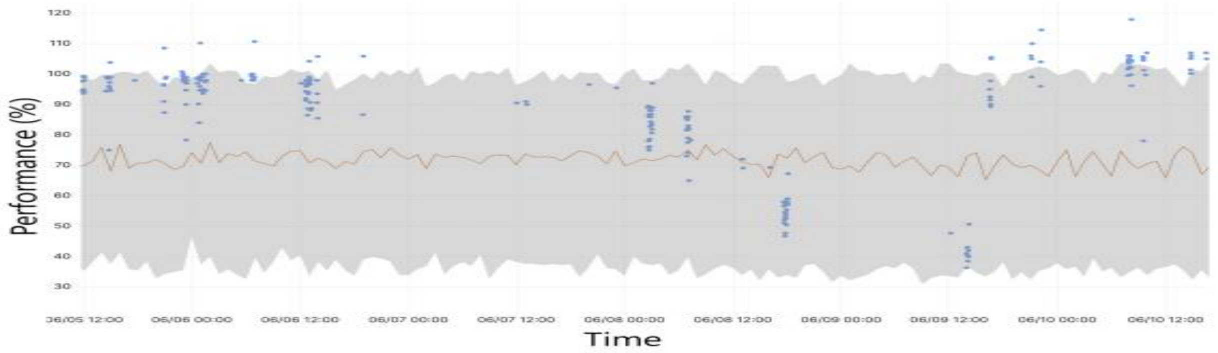


Figure 7: One of the ship's live and forecast data visualized with time-based line charts

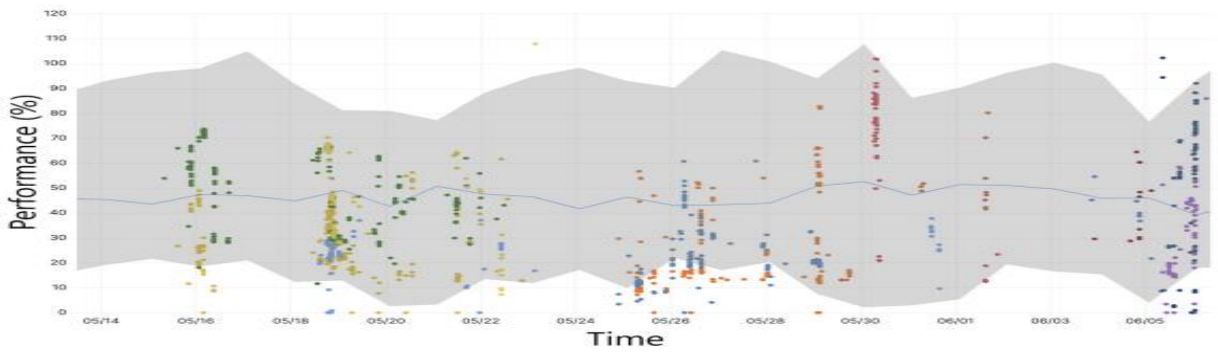


Figure 8: One of the port's live and forecast data visualized with time-based line charts

ML Workflow

Machine Learning Application Workflow

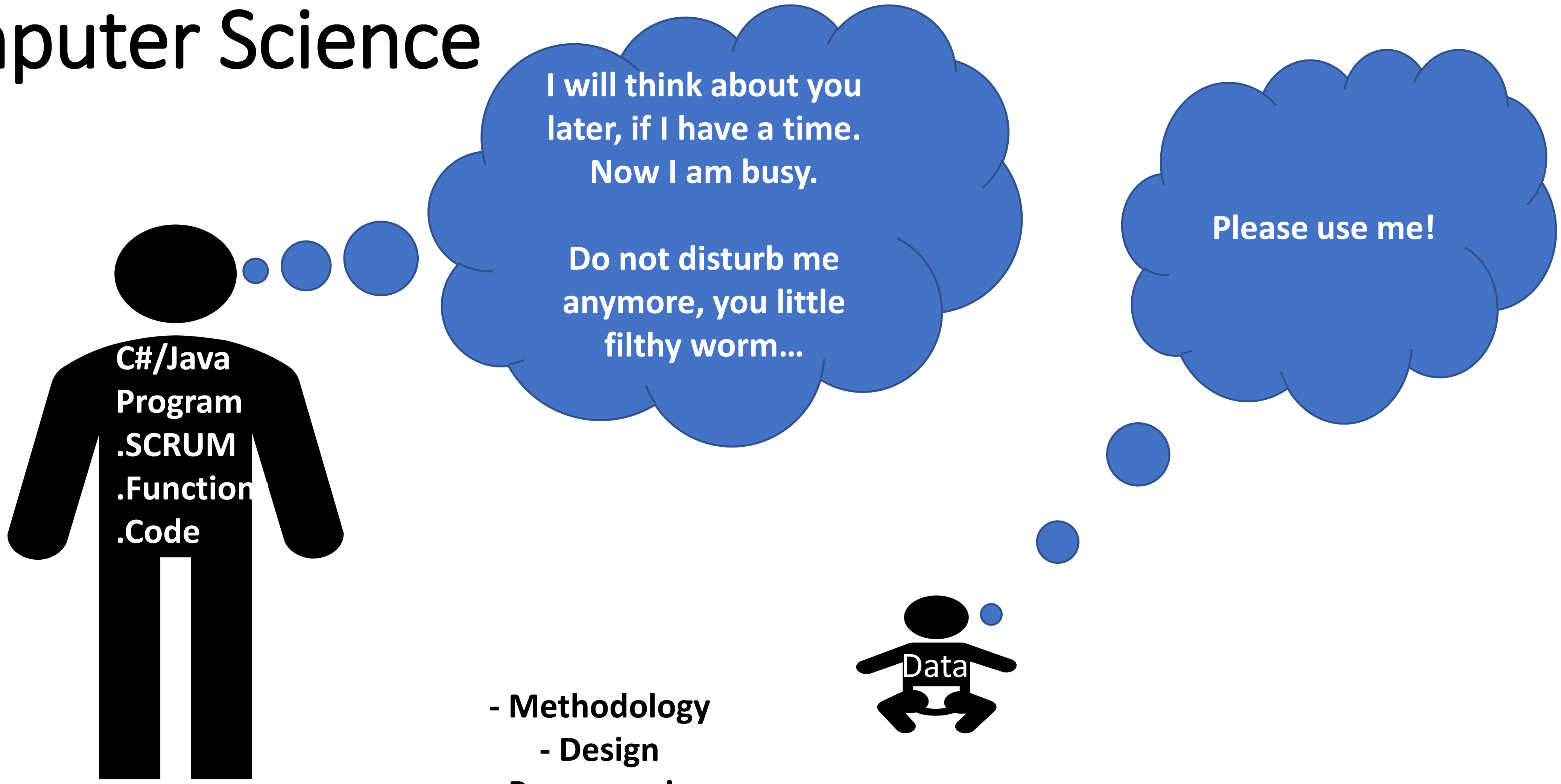
- Data and problem definition: The first step is to ask interesting questions. What is the problem you are trying solve? Why is it important? Which format of result answers your question? Is this a simple yes/no answer? Do you need to pick one of the available questions?
- Data collection: Once you have a problem to tackle, you will need the data. Ask yourself what kind of data will help you answer the question. Can you get the data from the available sources? Will you have to combine multiple sources? Do you have to generate the data? Are there any sampling biases? How much data will be required?
- Data preprocessing: The first data preprocessing task is data cleaning. For example, filling missing values, smoothing noisy data, removing outliers, and resolving consistencies. This is usually followed by integration of multiple data sources and data transformation to a specific range (normalization), to value bins (discretized intervals), and to reduce the number of dimensions.

Machine Learning Application Work Flow

- Data analysis and modeling with unsupervised and supervised learning: Data analysis and modeling includes unsupervised and supervised machine learning, statistical inference, and prediction. A wide variety of machine learning algorithms are available, including k-nearest neighbors, naïve Bayes, decision trees, support vector machines, logistic regression, k-means, and so on. The choice of method to be deployed depends on the problem definition discussed in the first step and the type of collected data. The final product of this step is a model inferred from the data.
- Evaluation: The main issue models built with machine learning face is how well they model the underlying data—if a model is too specific, that is, it over fits to the data used for training, it is quite possible that it will not perform well on a new data. The model can be too generic, meaning that it under fits the training data. For example, when asked how the weather is in California, it always answers sunny, which is indeed correct most of the time. However, such a model is not really useful for making valid predictions.

Why Majority of AI/ML based Student Project Fail..

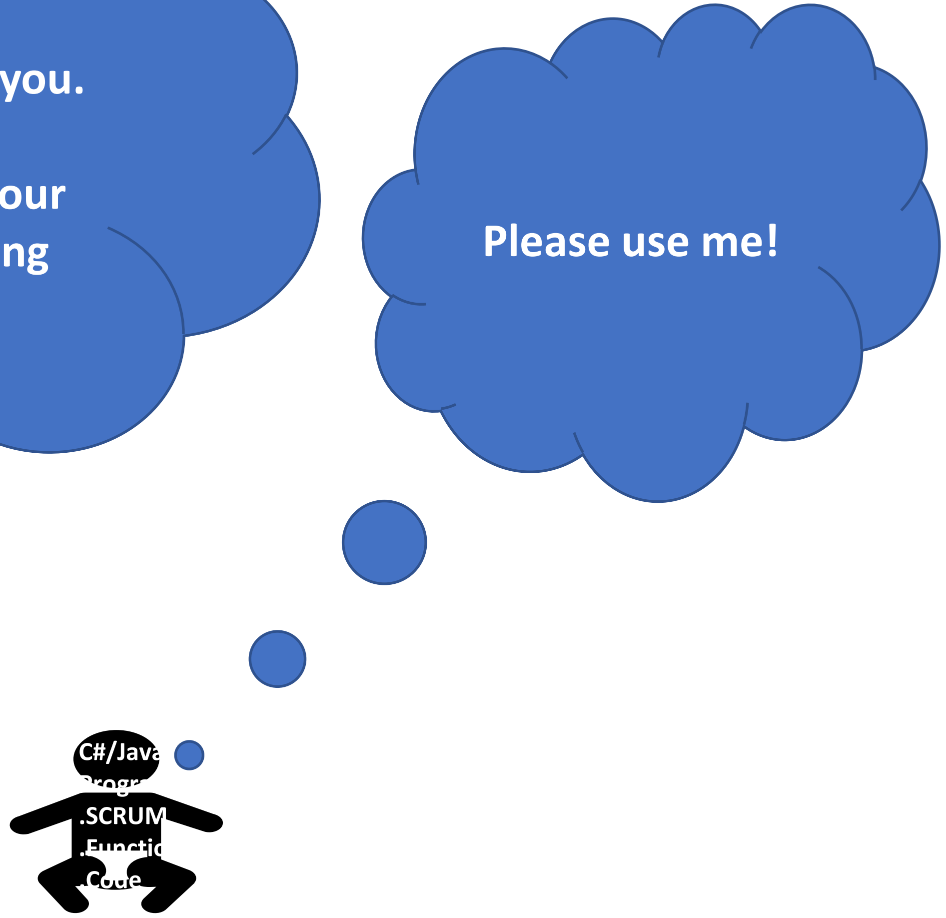
Computer Science



- Methodology
- Design
- Programming
- Databases
- Testing

Data Science

- Databases
- Statistics
- Machine Learning
- Programming
- Methodology/Design
- Testing



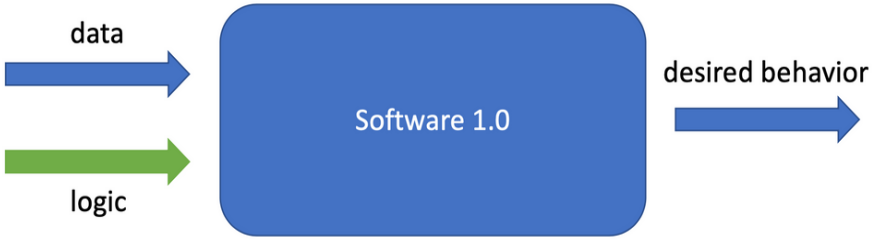
Be Ready for the Change - When we start working with AI/Machine learning then we must change the way we think, analyze, plan, solve and test

- **Start with data**
  - Preprocess and clean the data.
  - Select and construct appropriate features.
  - Select an appropriate model family.
  - Optimize model hyperparameters.
  - Postprocess machine learning models.
  - Critically analyze the results obtained.
- **Methodology**
  - Cross-industry standard process for data mining (CRISP-DM)
  - Agile CRISP-DM
  - Cognitive Project Management for AI (CPMAI)

https://www.forbes.com/sites/cognitiveworld/2020/01/19/why-agile-methodologies-miss-the-mark-for-ai--ml-projects/?sh=6898954f21ea

Be Ready for the Change - When we start working with AI/Machine learning then we must change the way we think, analyze, plan, solve and test Cont.

- **Testing**



Traditional Software Systems Testing
e.g., unit, regression, integration testing

Testing Machine Learning-based Systems (model tests)
e.g., invariance, directional expectation, minimum functional testing

https://www.jeremyjordan.me/testing-ml/

Be Ready for the Change - When we start working with AI/Machine learning then we must change the way we think, analyze, plan, solve and test Cont.

Problem:

Predict the electricity consumption of a household based on the number of daylight hours. The assumption here is that during shorter daylight hours (winter months), households use more electricity for lighting.

Traditional approach:

```
def test_predict_consumption_traditional():
    # Test for daylight hours less than 8
    assert predict_consumption_traditional(7) == 130  # 100 base + 30 for lighting
    assert predict_consumption_traditional(5) == 130  # 100 base + 30 for lighting

    # Test for daylight hours 8 or more
    assert predict_consumption_traditional(8) == 100  # just the base value
    assert predict_consumption_traditional(10) == 100  # just the base value
```

Be Ready for the Change - When we start working with AI/Machine learning then we must change the way we think, analyze, plan, solve and test Cont.

Linear Regression:

Linear regression aims to fit a line to data points such that the sum of the squared distances of the points from the line is minimized.

Given data, $y = mx + b$, where:

- y is the output (e.g., electricity consumption).
- x is the input feature (e.g., daylight hours).
- m is the slope of the line.
- b is the y-intercept.

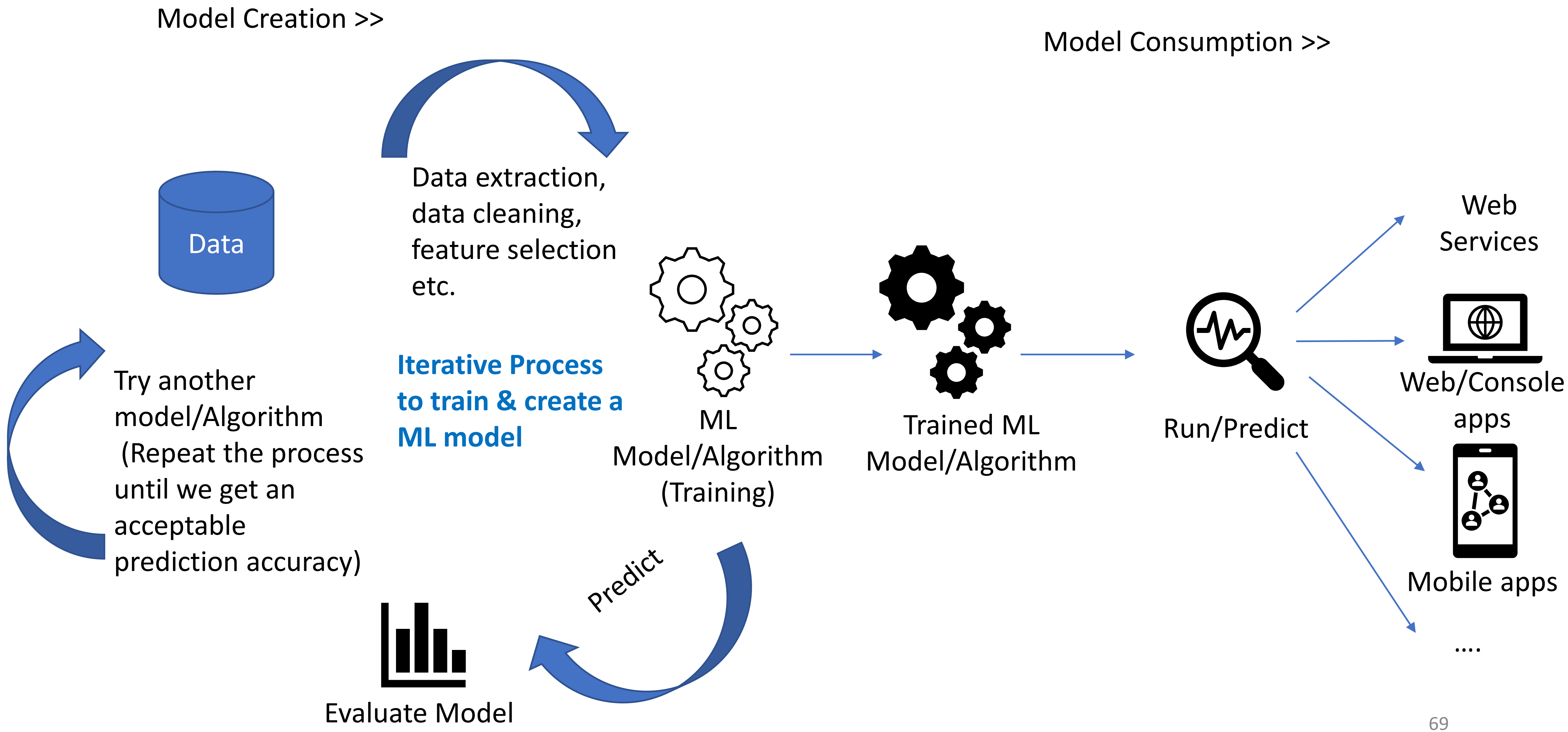
Here, m and b are the parameters that the model will "learn" or optimize during training.

Machine Learning Approach:

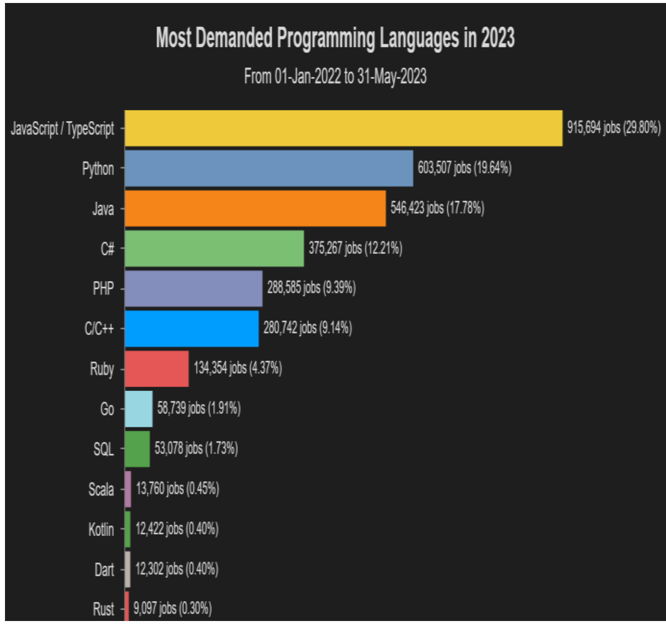
```
# Test data
test_daylight_hours = np.array([6.5, 8.5, 10.5]).reshape(-1, 1)
actual_consumptions = np.array([127.5, 107.5, 92.5])  # actual consumptions

def test_predict_consumption_ml():
    predictions = model.predict(test_daylight_hours)
    for i, prediction in enumerate(predictions):
        # Assert that the error is within an acceptable range, let's say 10 units for this example
        assert abs(prediction - actual_consumptions[i]) <= 10
```

ML lifecycle



Top Programming Languages for AI/ML



Python → C#

R Java/JavaScript Scala

References

- https://course.elementsofai.com

Adopting Artificial Intelligence in Danish SMEs: Barriers to Become a Data Driven Company, Its Solutions and Benefits

Nadeem Iftikhar[a] and Finn Ebertsen Nordbjerg
University College of Northern Denmark, Aalborg 9200, Denmark

Keywords: Industry 4.0, Artificial Intelligence, SMEs, Smart Manufacturing.

Abstract: Artificial intelligence allows small and medium-sized enterprises (SMEs) in the manufacturing sector to improve performance, reduce downtime and increase productivity. SMEs in Denmark are still struggling to implement artificial intelligence based strategies since they face a range of challenges, such as business applications, data availability, organizational culture towards the acceptance of new technologies, investment in new technologies, skills gap, development process and effective strategy. In the beginning, the paper describes the challenges faced by SMEs in adopting artificial intelligence. Then, the paper suggests solutions to overcome these challenges and discusses the importance of artificial intelligence as well as the opportunities it offers to SMEs.

1 INTRODUCTION

Industry 4.0, also called fourth industrial revolution, is the integration of IT and production systems. The IT systems compromise of artificial intelligence, industrial robots, sensors, alarms, IoT, cloud computing, image analysis, inventory management, data analysis and mood/behaviour analysis. The production systems include but not limited to enterprise resource planning (ERP), manufacturing execution system (MES), control and hardware. The integration of IT systems with production systems provides new insights from previously hidden information and allows for better decision making. Further, since 5G reduces response times from minutes to milliseconds, artificial intelligence (AI) has become the main driving force behind this industrial revolution in order to help enterprises to improve yield, quality, performance, efficiency and to decrease product waste, downtime, production and maintenance cost. AI is an broader term that covers a wide range of concepts and technologies, including machine learning (ML) and deep learning (DL). A survey of more than 500 Danish companies (Colotla et al., 2016) concluded that most of the companies are willing to change their business models to adopt AI in order to become more productive and to improve customer as well as product services. In spite of that most of the SMEs are left behind in this race of embracing AI due to numerous obstacles, just as, lack of expert knowledge, capabilities and funds. Aiming to be a front-runner in the use of AI, Denmark has formulated a *National Strategy for Artificial Intelligence*[1] to boost research and development in AI. The main aims of this strategy is to establish better collaboration between researchers and businesses, start new education programmes on AI and raise an investment pool of DKK 1.5 (€ 0.2) billion based on public-private partnership to help enterprises adopt AI (O'Dwyer, 2019).

To summarize, the main contributions of this paper are as follow:

- Discussing the main barriers for adopting AI.
- Suggesting the activities to be performed before adopting AI and investigating what needs to be done to prepare SMEs for AI adoption.
- Presenting the values of AI for SMEs.

The paper is structured as follows. Section 2 explains the motivation behind this paper. Section 3 introduces the research methodology. Section 4 presents the challenges for successfully implementing AI in SMEs. Section 5 suggests the solutions to deal with these challenges. Section 6 presents the benefits of AI for SMEs. Section 7 concludes the paper and points to the future directions.

[a] https://orcid.org/0000-0003-4872-8546

[1] https://en.digst.dk/policy-and-strategy/denmark-s-national-strategy-for-artificial-intelligence

Iftikhar, N. and Nordbjerg, F.
Adopting Artificial Intelligence in Danish SMEs: Barriers to Become a Data Driven Company, Its Solutions and Benefits.
DOI: 10.5220/0010691800003062
In Proceedings of the 2nd International Conference on Innovative Intelligent Industrial Production and Logistics (IN4PL 2021), pages 131-136
ISBN: 978-989-758-535-7

131

Proceedings of the Changeable, Agile, Reconfigurable and Virtual Production Conference and the World Mass Customization & Personalization Conference
World Mass Customization & Personalization Conference
↳ CARV 2021, MCPC 2021: **Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems** pp 448–456 | Cite as

Home › Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems › Conference paper

Implementing Machine Learning in Small and Medium-Sized Manufacturing Enterprises

Nadeem Iftikhar & Finn Ebertsen Nordbjerg

Conference paper | First Online: 01 November 2021

Part of the Lecture Notes in Mechanical Engineering book series (LNME)

Abstract

Large enterprises in the world are making substantial investments to adopt smart manufacturing technologies (Industry 4.0). Machine learning is one of the main driving forces behind this industrial revolution. On the other hand, small and medium-sized enterprises (SMEs) in the manufacturing sector are falling behind. Thus, for SMEs, there is an urgent need for adopting a machine learning based approach to gain competitive advantage. Machine learning helps SMEs to improve efficiency, catch manufacturing defects, predict machine failures, reduce unplanned downtime and increase productivity. This paper elaborates on machine learning project development life cycle for manufacturing SMEs. Furthermore, the paper presents a real-life case study of a medium-sized manufacturing company. Finally, the paper offers new insights and suggestions for other SMEs to adopt machine learning successfully.

Sections | References

Abstract
Notes

The End
Thank You

Open for collaborations
naif@ucn.dk

Questions?