20th Mediterranean Communication and Computer Networking Conference (MedComNet 2022)
1-3 June 2022, Paphos, Cyprus

Cleaning Matters! Preprocessing-enhanced Anomaly Detection and Classification in Mobile Networks

*Juan Marcos Ramírez, †Pablo Rojo, ‡Fernando Díez, ***Vincenzo Mancuso,** *Antonio Fernández Anta

***IMDEA Networks Institute,** † Nokia CNS, Spain, ‡Universidad Politécnica de Madrid

Developing the Science of Networks

Outline

- Background and motivation
- The KLNX methodology
- Experiments
  - The Nokia dataset
  - The synthetic dataset
- Conclusions

Background and motivation

- The growth of cellular technologies (5G and the emergence of 6G) has relied on complex communication infrastructures that collect information from several heterogeneous system components.
- This scenario requires the development of automatic and intelligent tools to detect performance anomalies in mobile networks.
- It is also necessary to identify components that could induce network performance anomalies for taking the corresponding corrective actions.
- Monitor and control tools should resort to explainable machine learning models enabling the identification of relevant parameters that describe both the network's normal behavior and performance loss scenarios.

Performance anomalies in mobile networks

- **Performance anomalies:** Unexpected operation scenarios significantly deviating from the performance for which the network was designed.

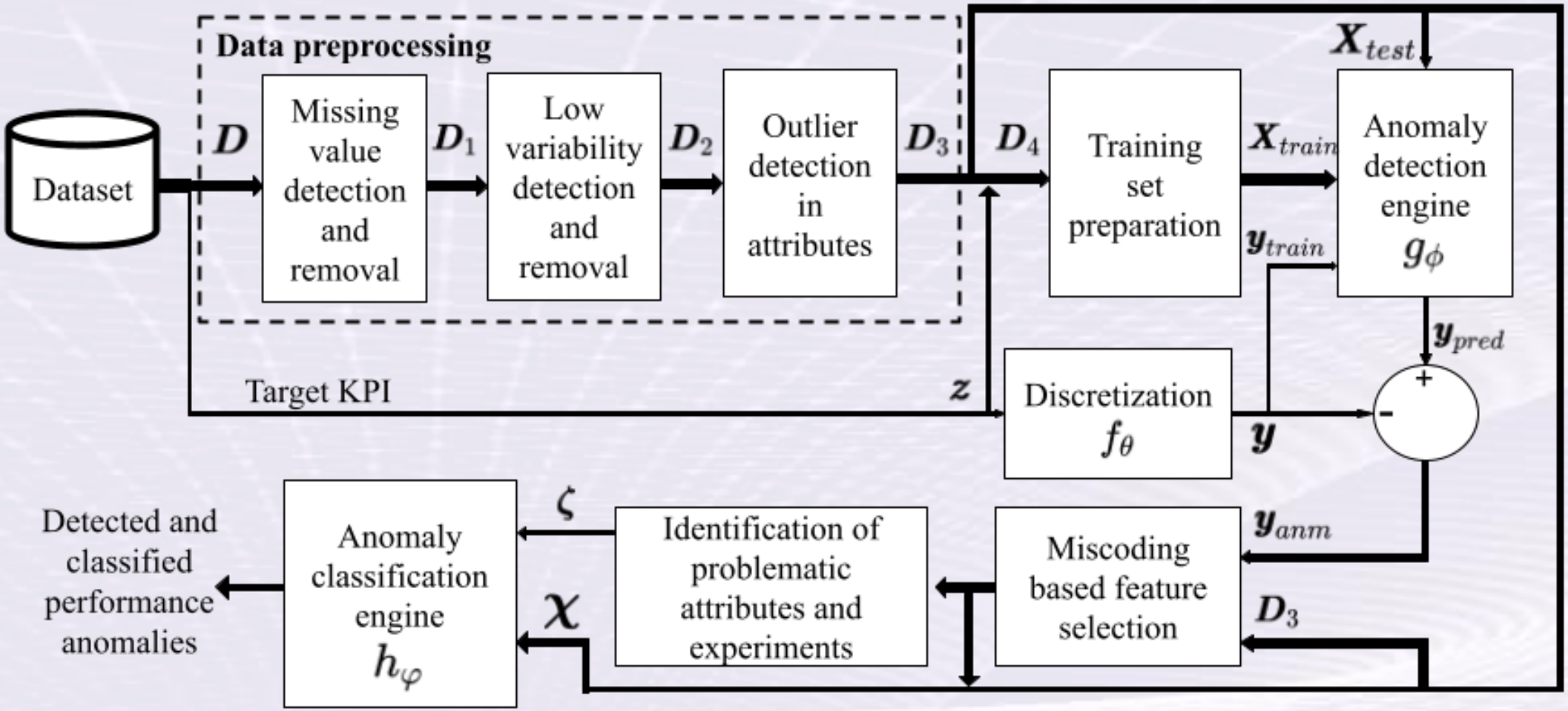
- Previous anomaly detection methods via pattern recognition models [1,2]:

  – They are designed to solve specific problems (detect security anomalies, detect traffic time series anomalies, etc.).

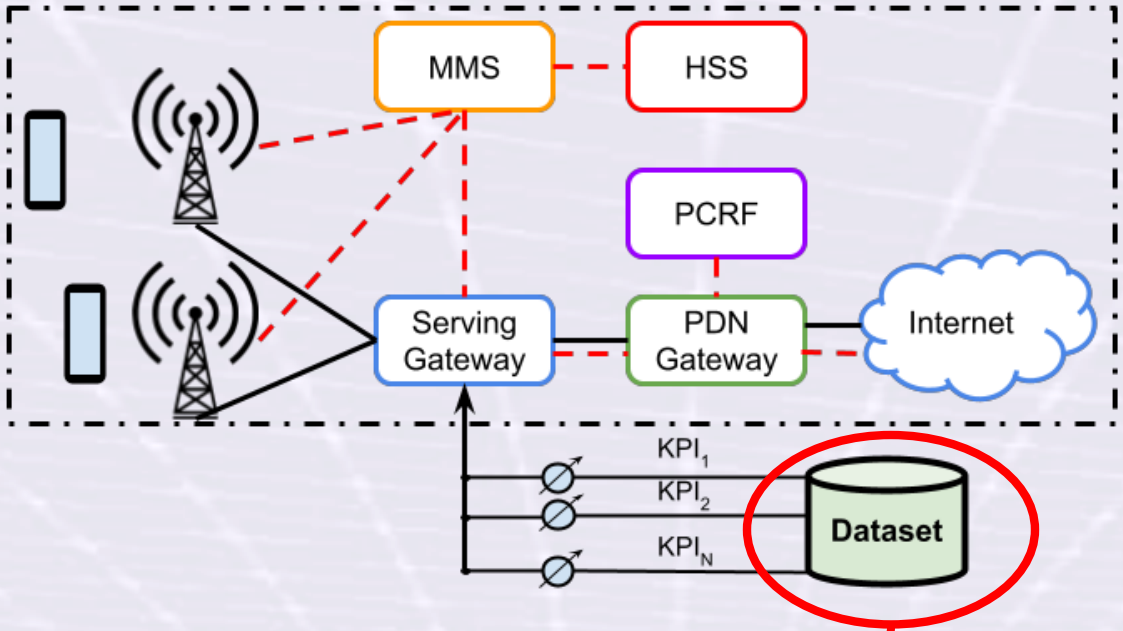
  – They require fine hyperparameter tuning.

  – They do not identify the causes of network anomalies.

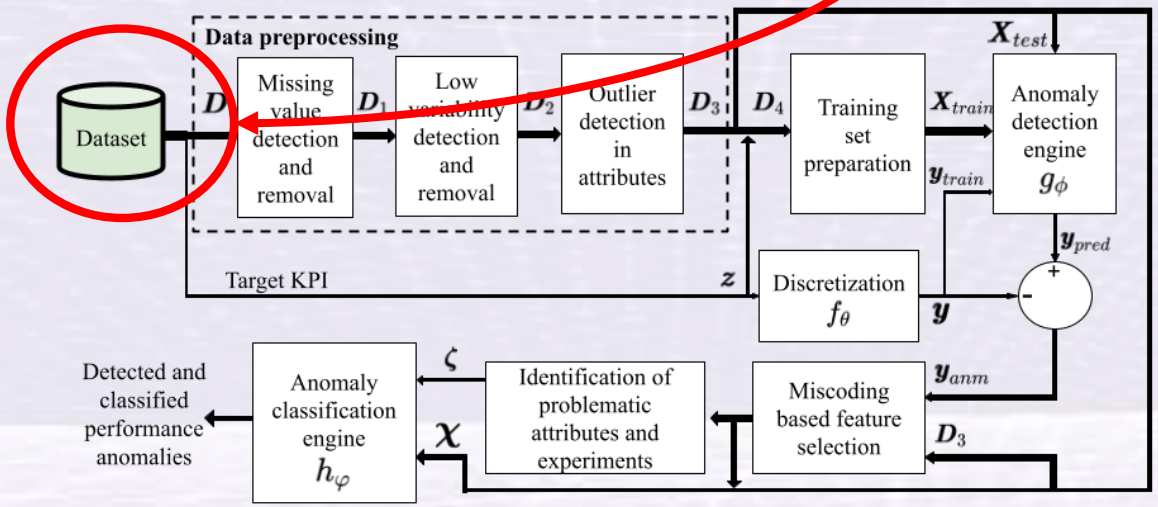
The clean and explainable (KLNX) methodology



Measurement campaigns



The KLNX methodology

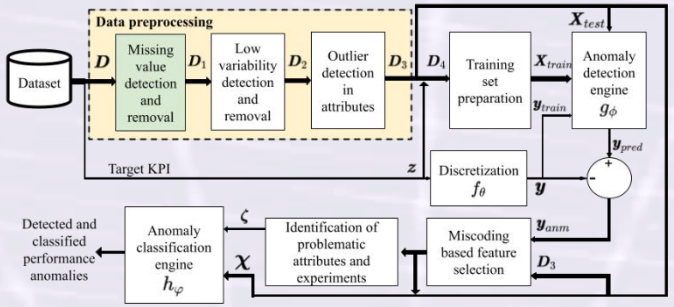


Datasets

- Operators and regulators often perform tests to evaluate the Quality of Service (QoS) provided by mobile networks.
- Tests collect key performance indicators (KPIs) of different network aspects.
  - TCP, radio performance, routing, etc.
- Test acquire tens hundreds of end-to-end of individual cases (experiments or samples) and hundreds of KPIs (attributes or features)
- **The KLNX method detects and classifies performance anomalies from KPI data.**

Data preprocessing

- **Missing value detection:** Identifies and removes attributes with many null cells.



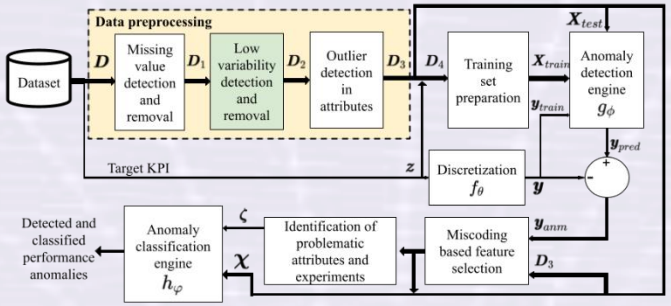
ack_b2a	End_SINR	Start_SINR	pure_ack	initial_cw	RTT
1	5.8	-10.1	23		100
1	3.7	-5.3	40		139
5	Nan	0.3	9		201
4	Nan	-7.9	75	5	283
7	Nan	-3.8	53	3	139
2	Nan	0.1	42	5	195
6	2.9	-11.3	13		321

Empty cells

Nan in numeric cells

Remove columns

- **Low variability detection:** Detects and discards attributes with zero standard deviation.

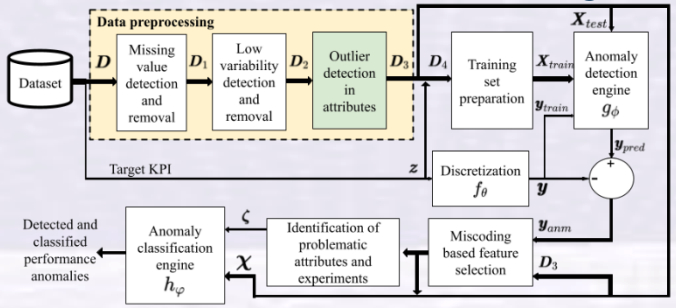


ack_b2a	End_SINR	Start_SINR	pure_ack	initial_cw	RTT
1	5.8	-10.1	23	2	100
1	3.7	-10.1	40	7	100
5	8.5	-10.1	9	9	100
4	5.8	-10.1	75	5	100
7	0.9	-10.1	53	3	100
2	1,6	-10.1	42	5	100
6	2.9	-10.1	13	5	100

Low variability columns

Remove columns

- **Outlier detection in attributes:** Recognizes and removes attributes with many outliers.



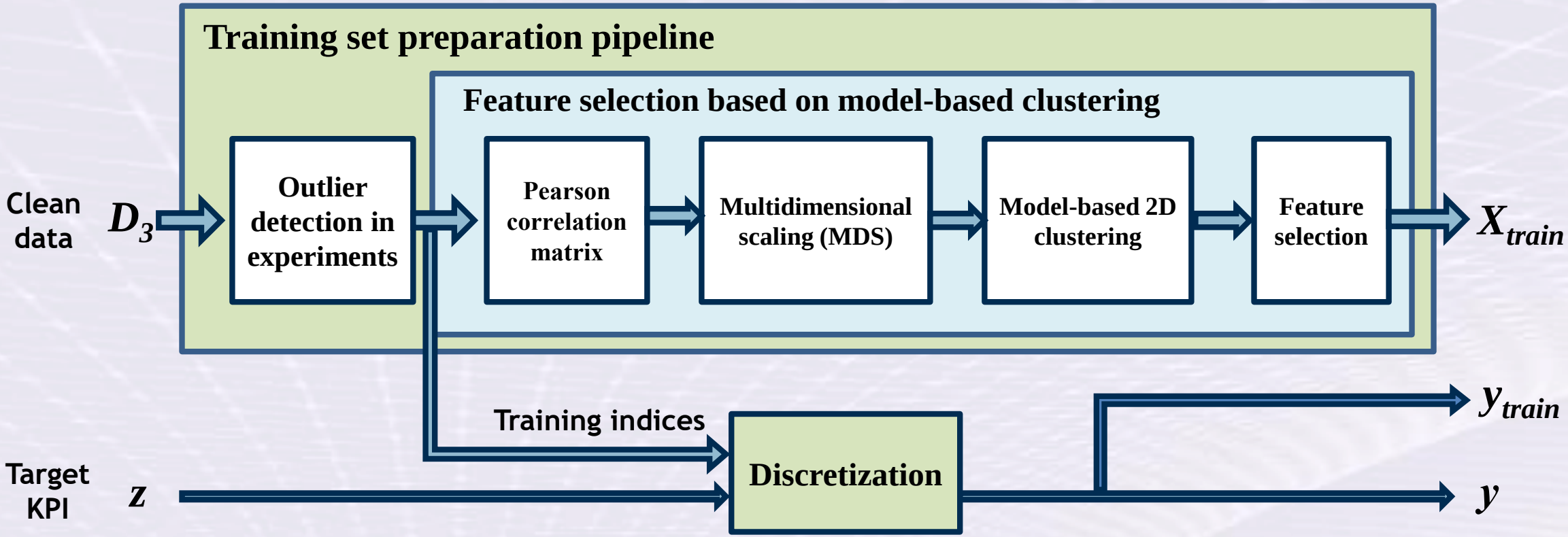
ack_b2a	End_SINR	Start_SINR	pure_ack	initial_cw	RTT
1	5.8	-4334330.1	23	2	100
1	3.7	-5.3	40	543466456	139
5	8.5	-0.3	9	9	201
4	5.8	-7332354.9	75	531231441	283
7	0.9	-3.8	53	3	139
2	1,6	0.1	42	5	195
6	2.9	-11.3	13	5	321

Outliers or gross errors

Remove columns

Preprocessing steps are performed in an unsupervised way using the MAD criterion [3]

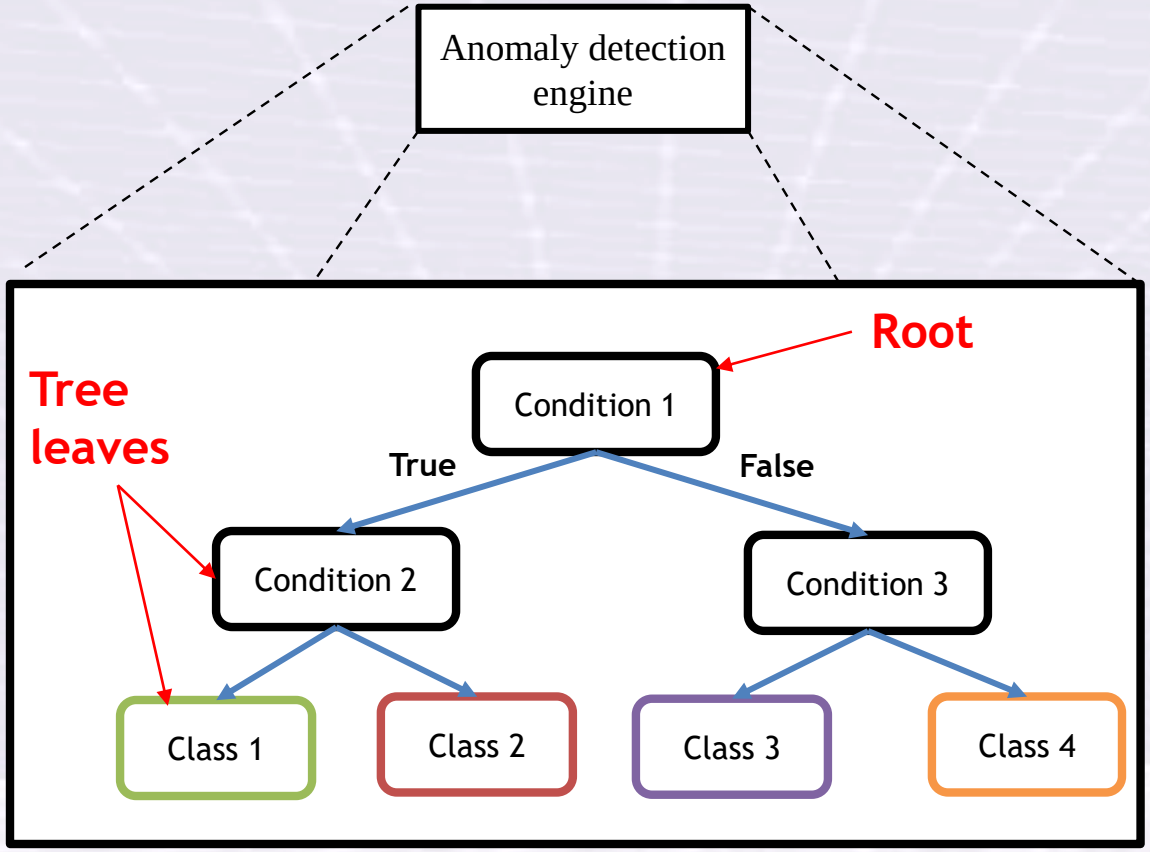
Training set preparation for the anomaly detection engine



- This stage extracts both the **training samples and the training labels** to optimize the anomaly detection engine.
  - The selected samples describe the normal behavior of the system.
  - Misclassified samples will be considered anomalous scenarios.

Anomaly detection engine

- The decision tree classifier is selected to detect anomalous scenarios.
  - Decision tree is an **explainable model** and easy to understand.
  - Its structure can be evaluated to observe conditions describing the **network's normal behavior.**



1. The training stage is implemented using the CART algorithm.
2. The Gini impurity is selected as the cost function to be minimized.

$$G = 1 - \sum_{i=1}^{k} p_i^2$$

3. The number of samples per leaf is restricted to 5 and the tree depth is constrained to

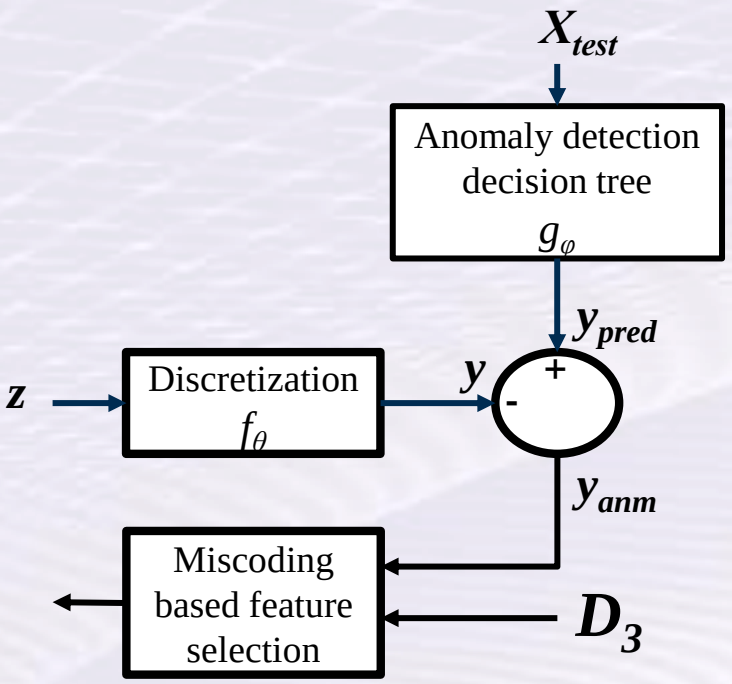
$$\text{depth} = \lfloor (\log_2 m_\ell)/2 \rfloor$$

4. A cost-complexity pruning is implemented to minimize the probability of overfitting.

Selection of relevant features

- **Vector of anomalous scenarios:**
  - Discretized target KPI: $\boldsymbol{y} = f_\phi(\boldsymbol{z})$
  - Predicted labels: $\boldsymbol{y}_{pred} = g_\phi(\boldsymbol{X}_{test})$
  - Vector of anomalous scenarios: $\boldsymbol{y}_{anm} = \boldsymbol{y}_{pred} - \boldsymbol{y}$
- **Feature selection based on Miscoding metric:** the miscoding [4] between every feature vector and the vector of anomalous scenarios is computed to assess the relevance of the available attributes.

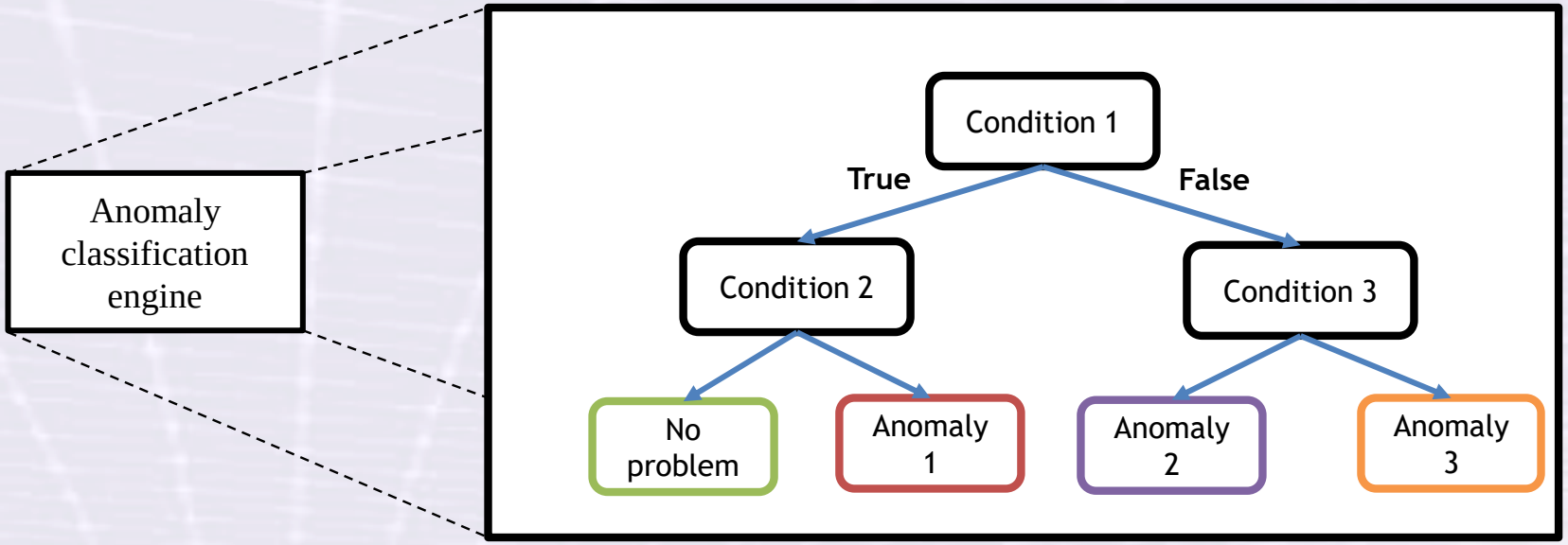
$$\mathrm{mscd}(\boldsymbol{X}(:,i),y) = \frac{1-\mathrm{NCD}(\boldsymbol{X}(:,i),y)}{\sum_{j=1}^{m}\mathrm{NCD}(\boldsymbol{X}(:,j),y)}$$



Identification of problematic attributes and samples

- This stage recognizes attributes and operation scenarios related to network performance inefficiencies.
  - **Model-based 1D clustering:** Applies a 1D clustering to every attribute vector whose optimal number of cluster is based on the BIC criterion.
  - **Identification of problematic features:** Identifies as problematic attributes those features that generates more than one cluster.
  - **Identification of problematic experiments:** Recognizes as problematic experiments those attribute clusters with a large concentration of misclassified samples.
  - **Generates the labels of the anomaly classification engine.**

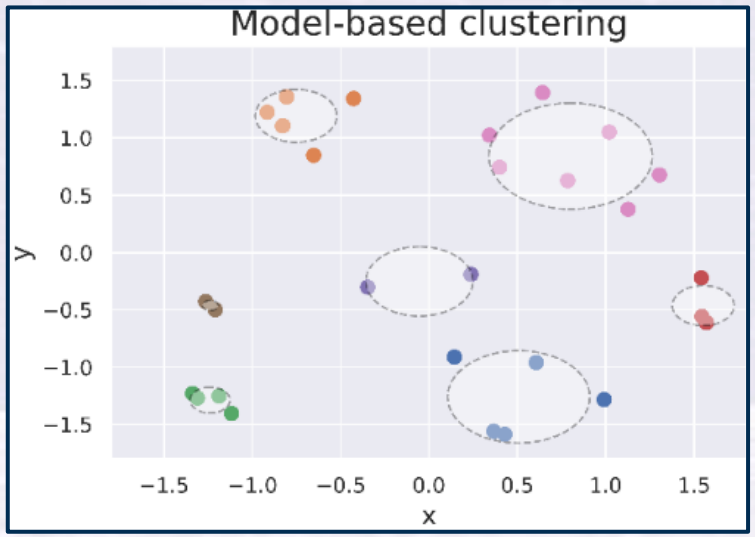
Anomaly classification engine

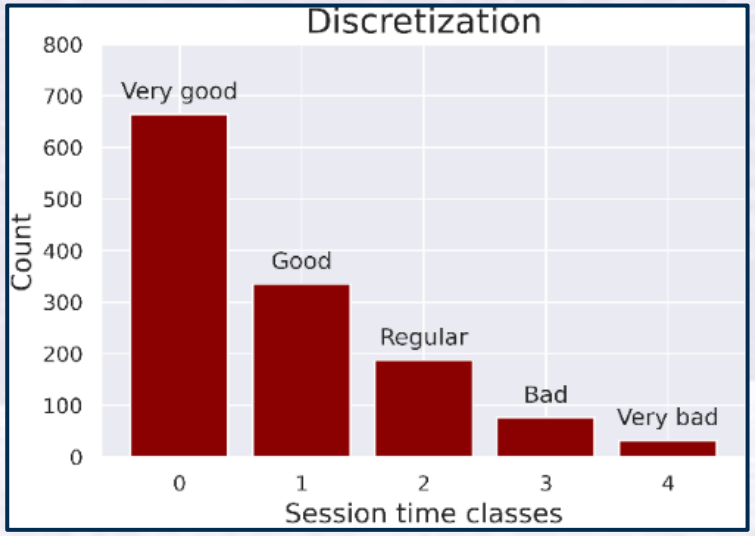


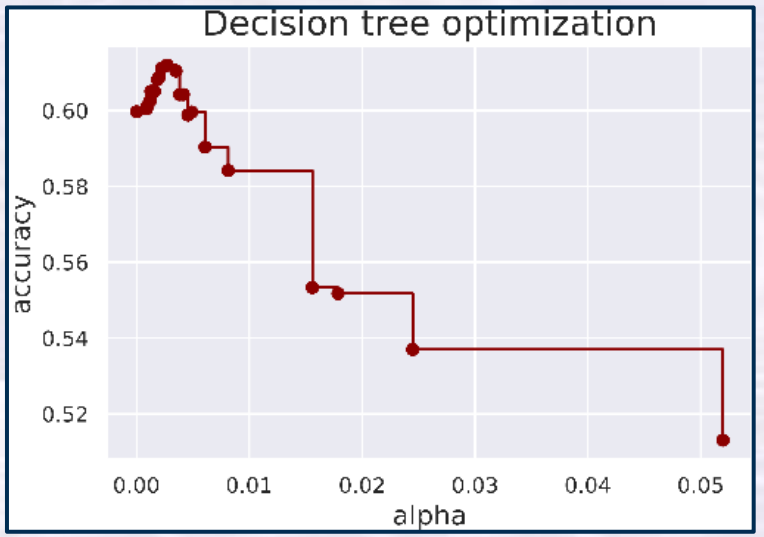
- We also select the decision tree model to classify performance anomalies.
  - Decision tree flowchart can be analyzed to explain which parameters and thresholds are closely related to **network's anomalies.**
  - Its **interpretability** allows to identify the network components that are deviated from their normal behavior and take the corrective actions.

Experiments

- **The Nokia dataset.**
  - This dataset was collected in 2019 to test the performance of 4G mobile networks in various European countries [5].
  - TCP profiles and radio variables measurements when a user device downloads a 3 MB file (HTTP_FILE_DL).
  - 1326 attributes (columns) and 1730 experiments (rows).

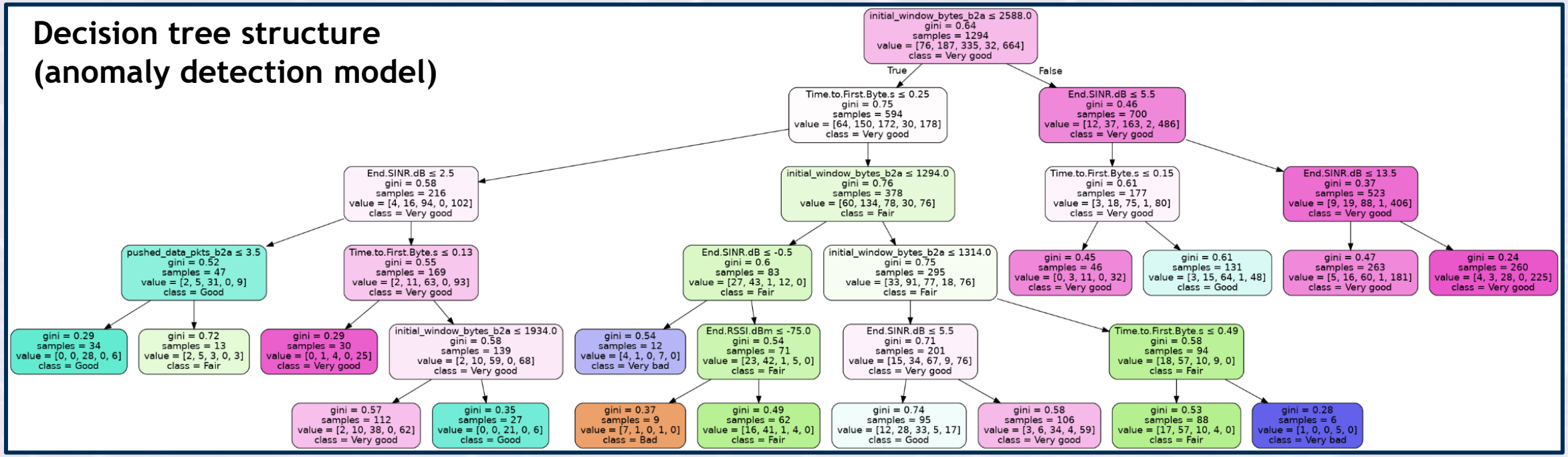






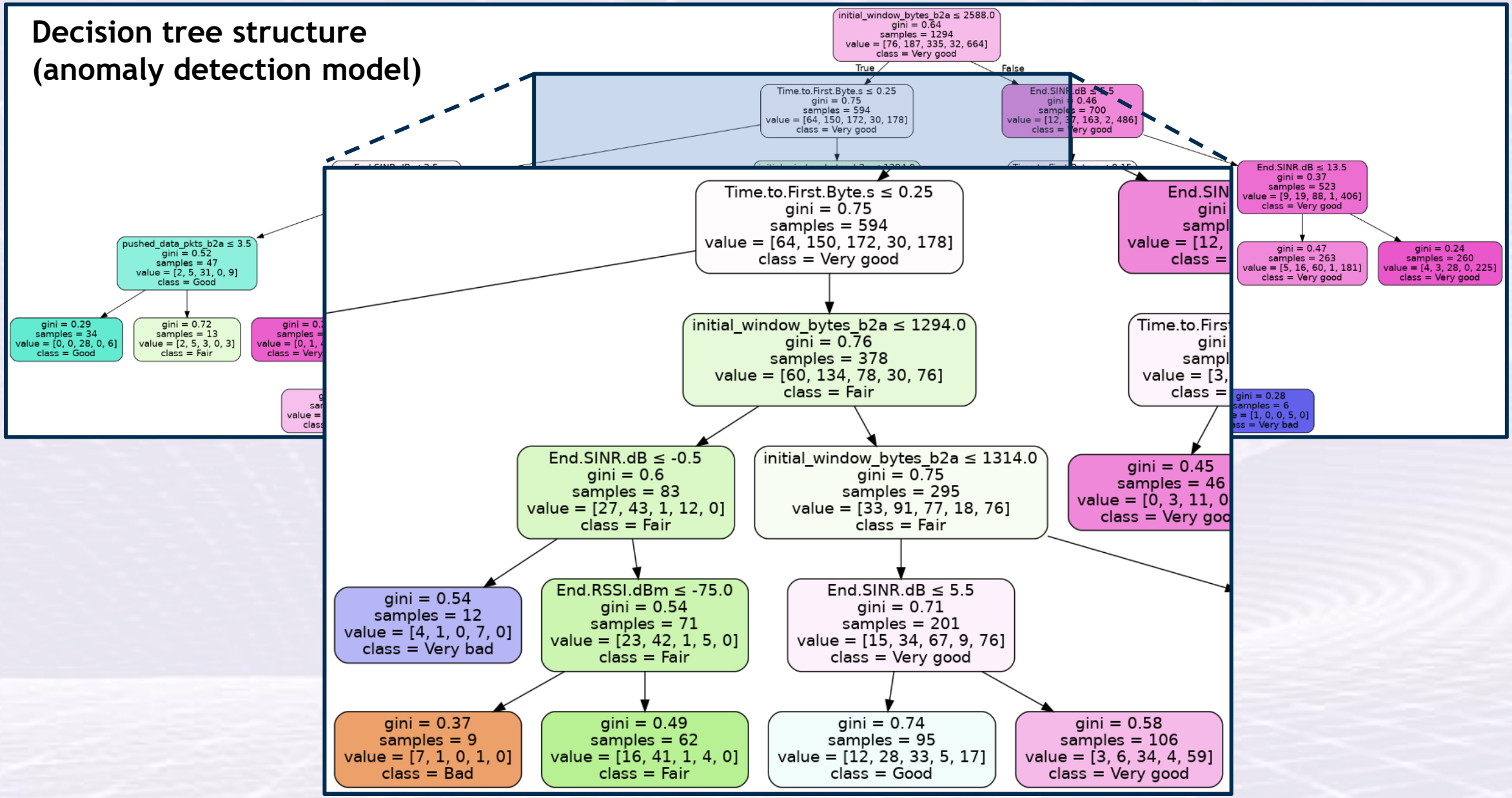
The Nokia dataset

Decision tree structure (anomaly detection model)

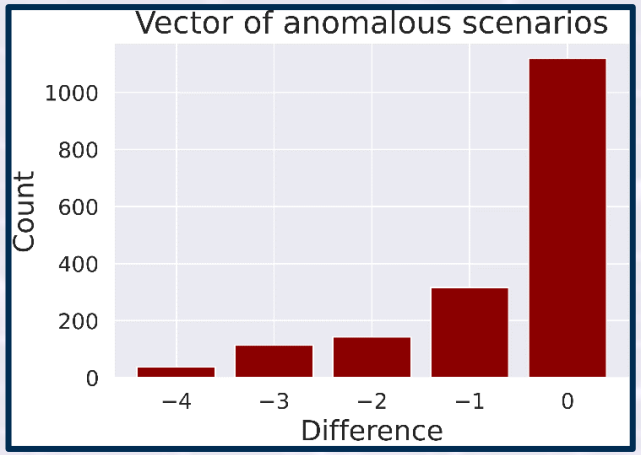


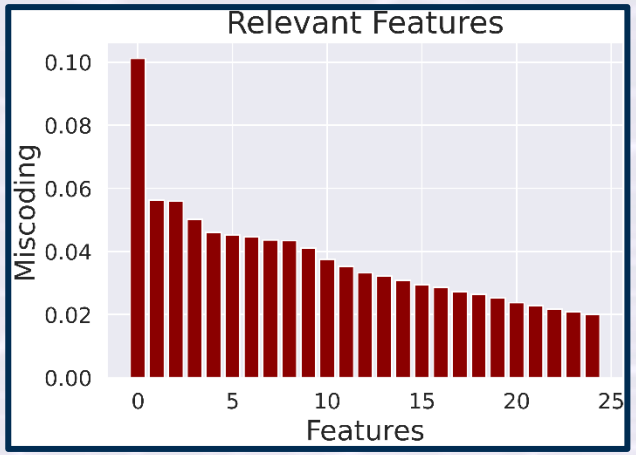
The Nokia dataset

Decision tree structure (anomaly detection model)

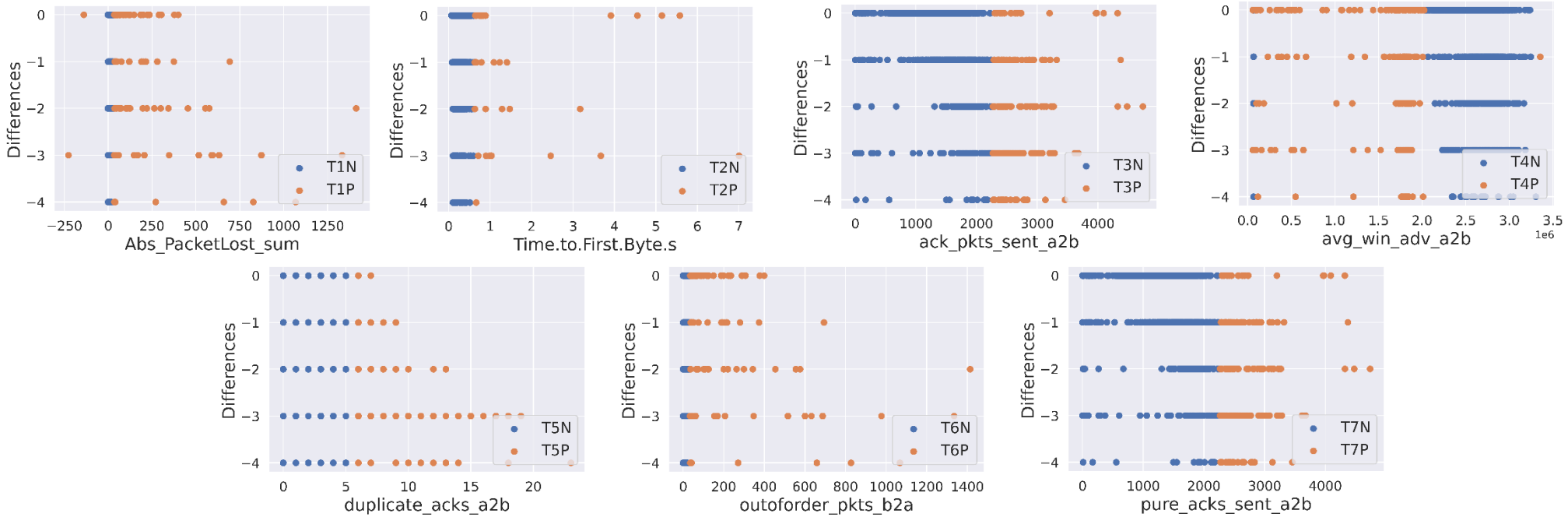


The Nokia dataset



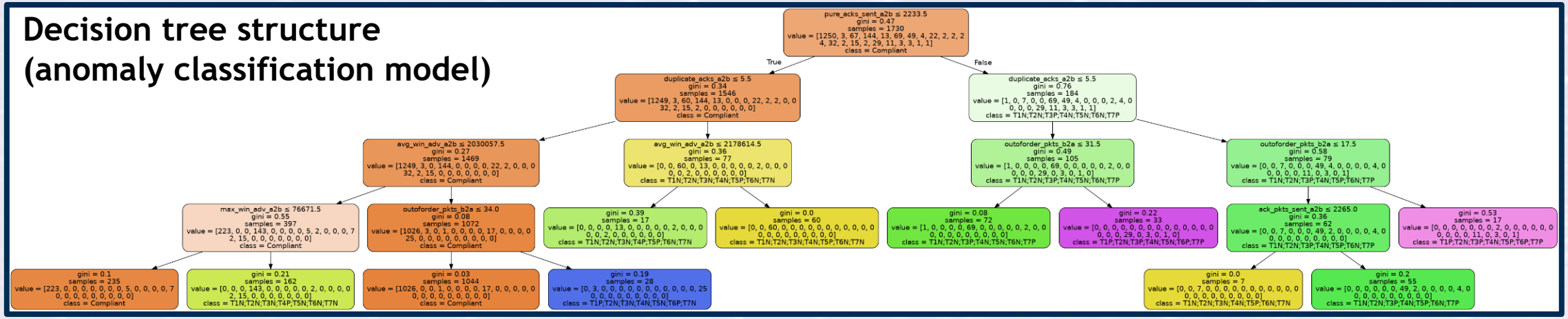


Identification of problematic attributes and experiments



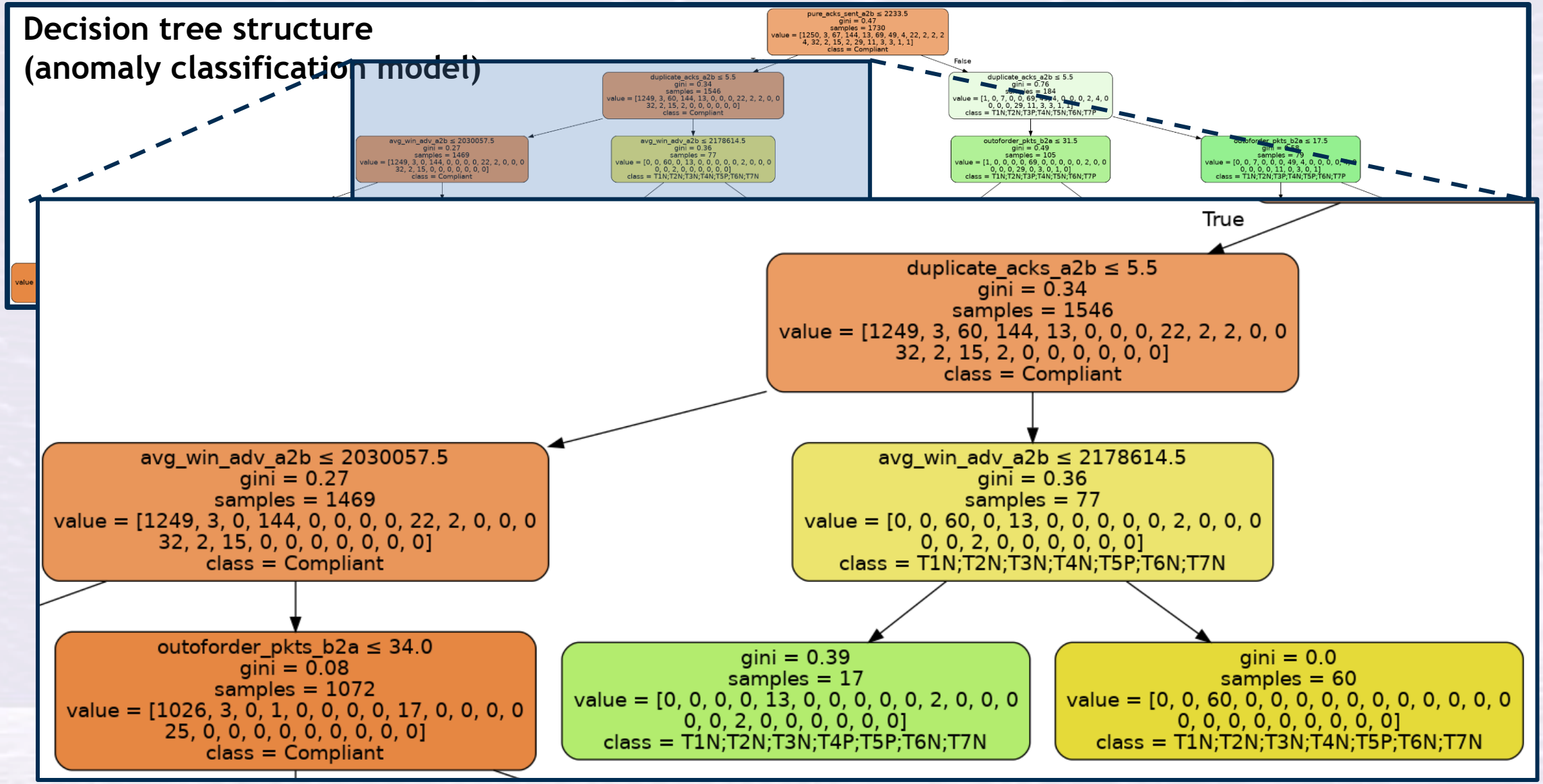
The Nokia dataset

Decision tree structure (anomaly classification model)

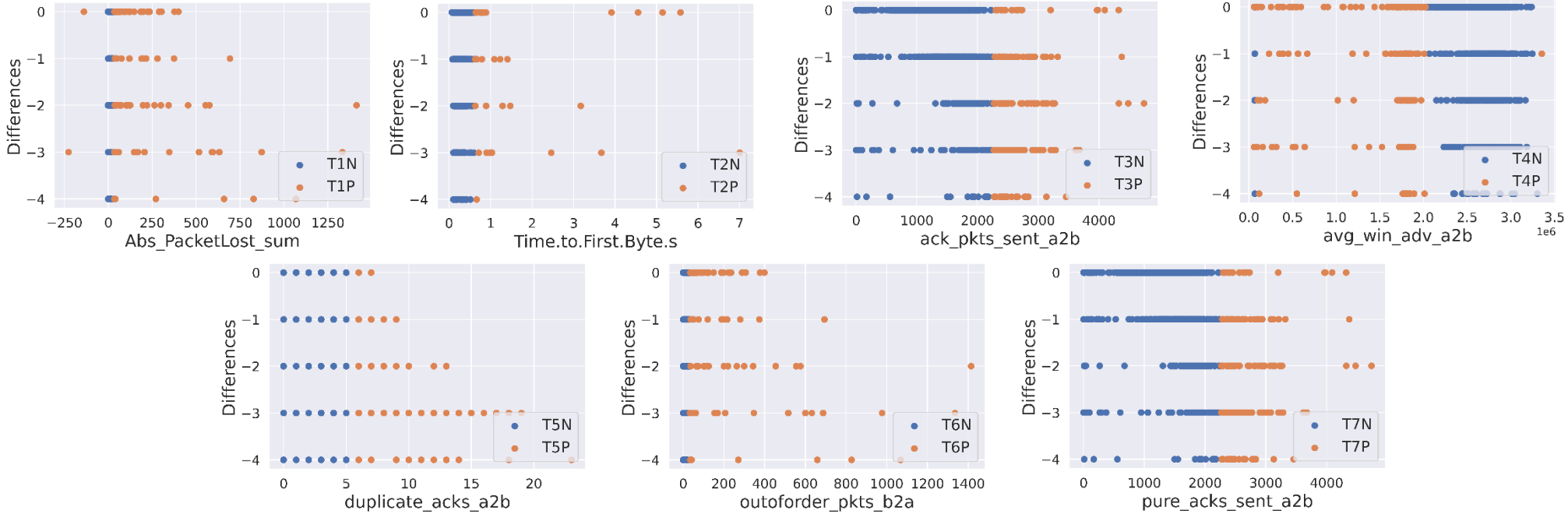


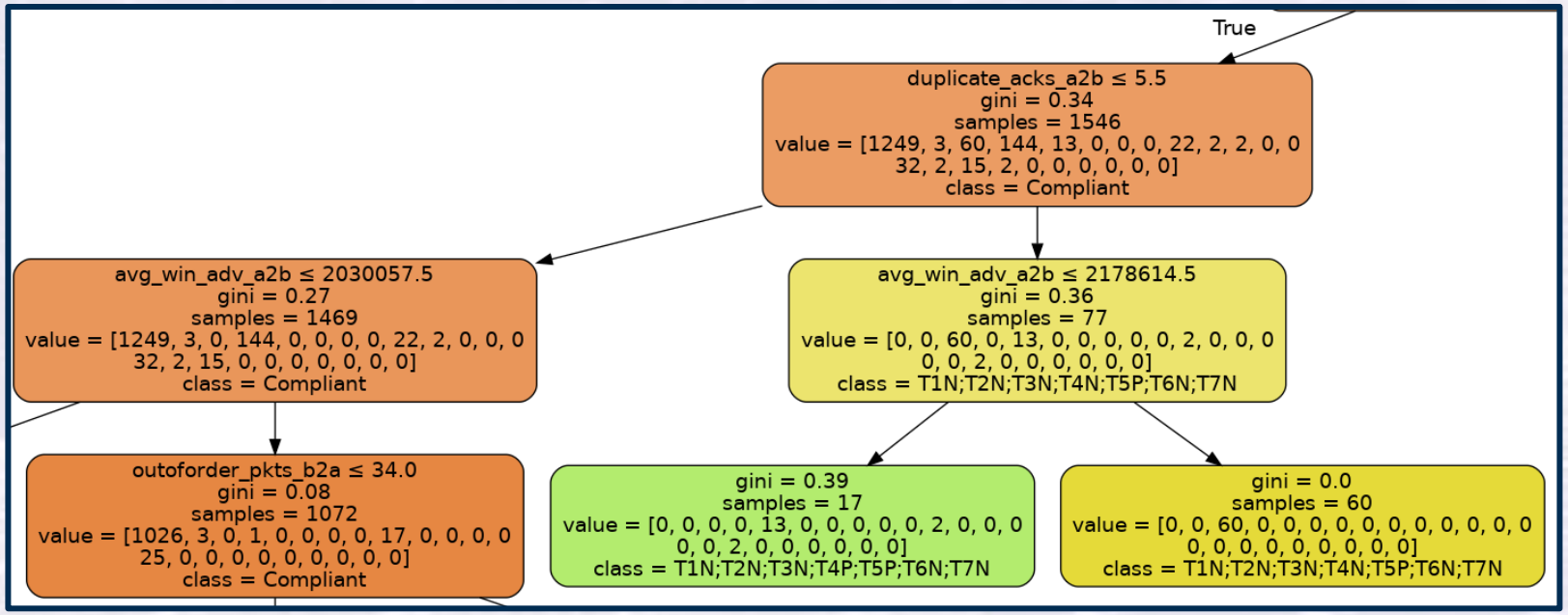
The Nokia dataset

Decision tree structure (anomaly classification model)



Identification of problematic attributes and experiments





Experiments

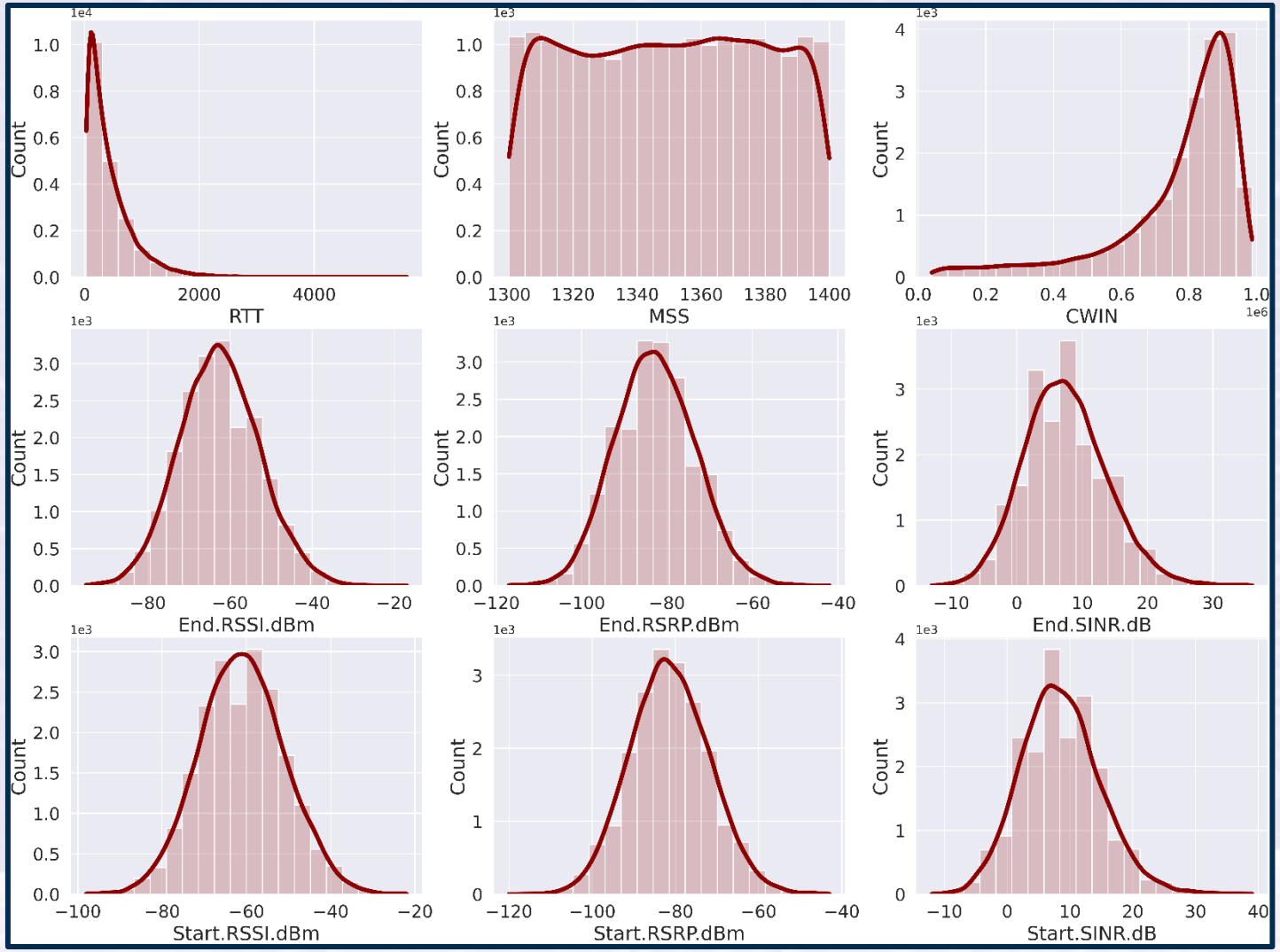
- **The synthetic dataset:**
  - We build a testbed that randomly generates every attribute whose components obey a particular statistical model.
  - The probability density functions were characterized such that their statistical models fit the attribute distributions exhibited by the Nokia dataset.
  - The TCP throughput is estimated using the Mathis model [6].

$$B_{tcp} = \min\left(\frac{MSS}{RTT\sqrt{p}}, \frac{CWND}{RTT}\right)$$

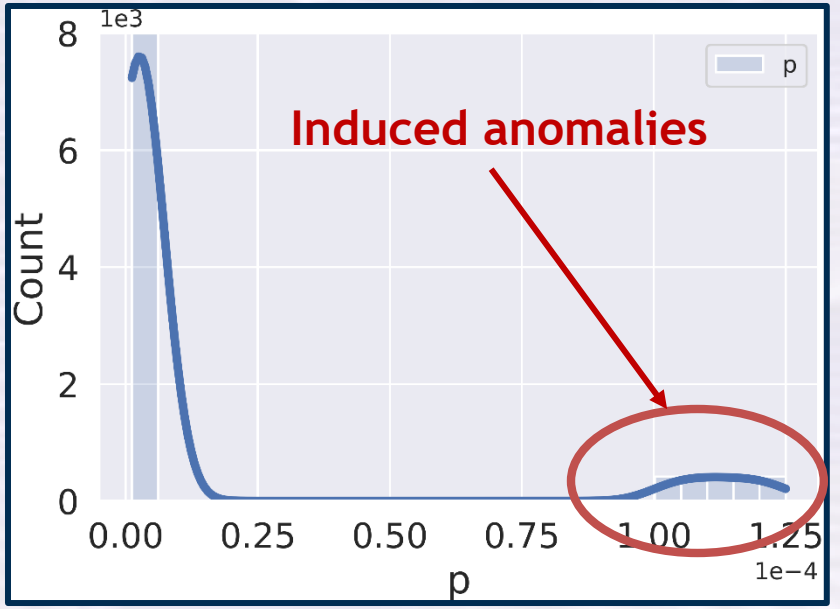
- **Advantages:**
  - This dataset enable the evaluation of the methodology at different stages.
  - It can be induced anomalous behaviors to test the accuracy of the anomaly detection and classification models.

Synthetic data distributions

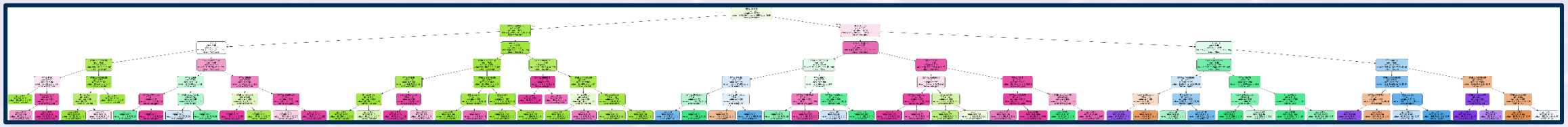
Distributions of synthetic features



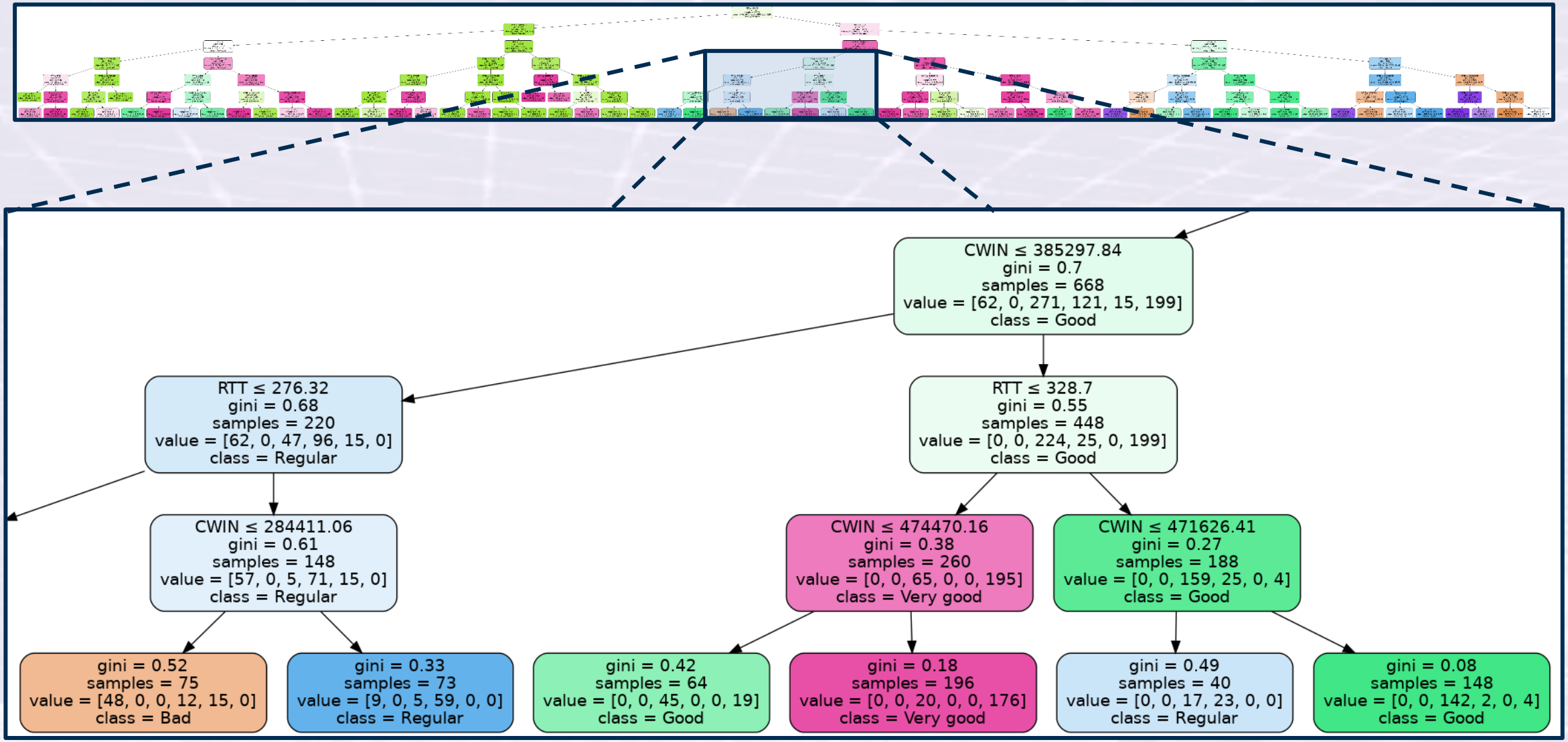
Packet loss probability



Anomaly detection engine for the synthetic dataset

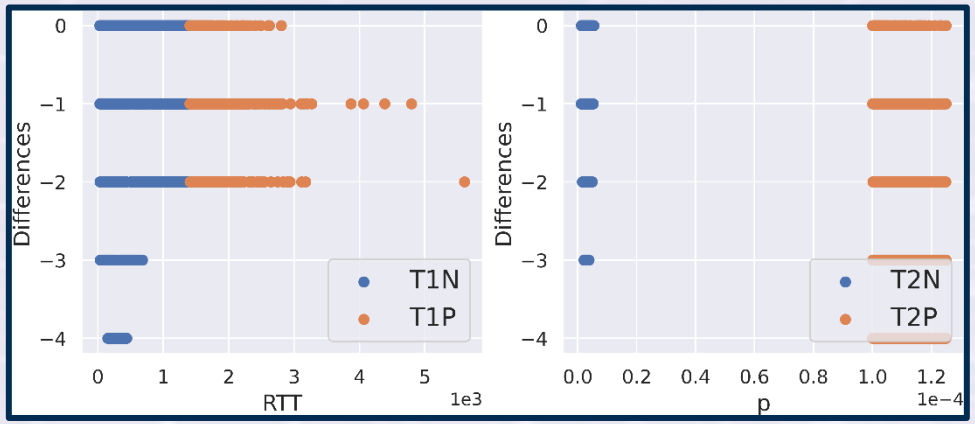


Anomaly detection engine for the synthetic dataset

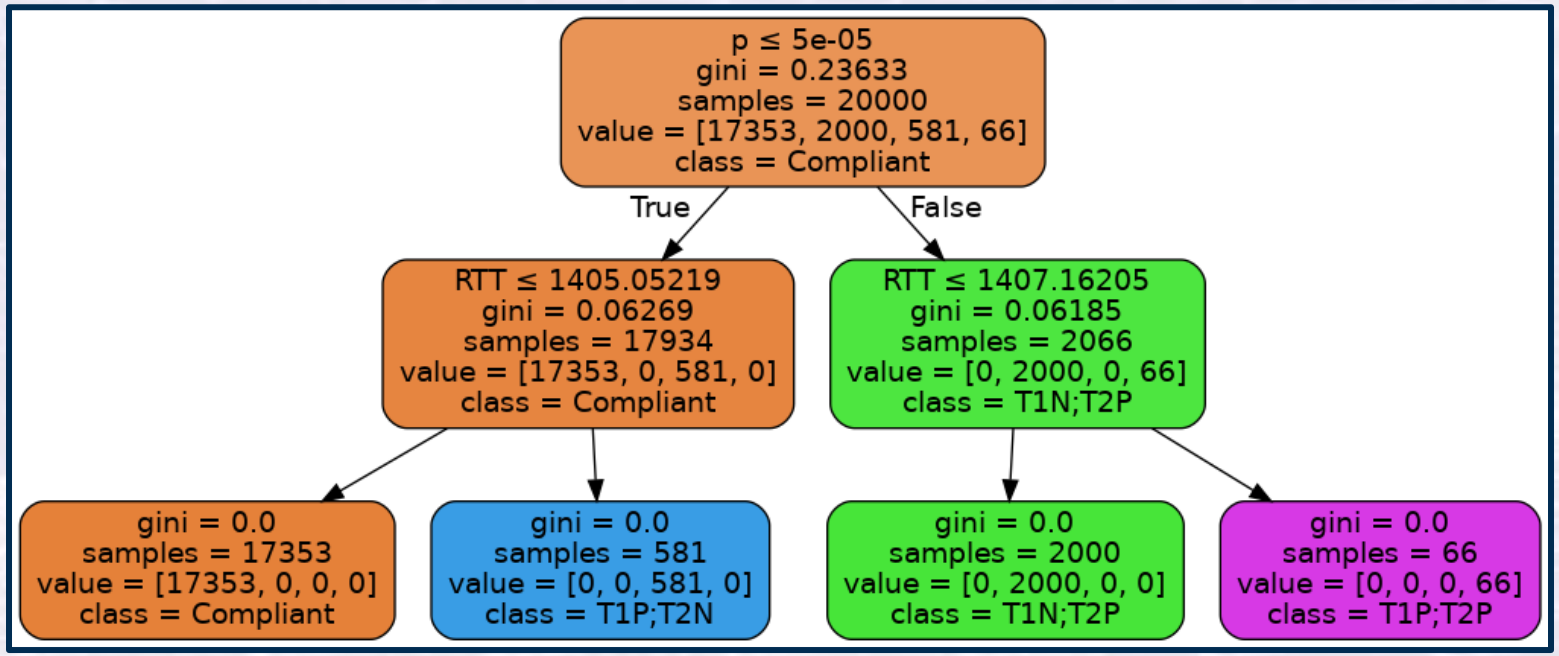


Anomaly classification engine for the synthetic dataset

Identification of problematic attributes and experiments



Decision tree of the anomaly classification stage



Concluding remarks

- The proposed methodology addressed the performance anomaly detection and classification in mobile networks through explainable machine learning engines.
- Two decision trees were included in the methodology as interpretable learning models to observe the rules describing both the network's normal performance and the anomaly classification stage.
- The generation of synthetic datasets was a useful tool to evaluate the methodology at different processing stages as well as to analyze the response in the presence of induced anomalies.

References

[1] Mohamed Moulay, Rafael A. García Leiva, Pablo J. Rojo Maroni, Javier Lazaro, Vincenzo Mancuso, and Antonio Fernández Anta, "A novel methodology for the automated detection and classification of networking anomalies," in 39th IEEE Conference on Computer Communications, INFOCOM Workshops 2020, Toronto, ON, Canada, July 6-9, 2020. 2020, pp. 780–786, IEEE.

[2] Guang Yu, Zhiping Cai, Siqi Wang, Haiwen Chen, Fang Liu, and Anfeng Liu, "Unsupervised online anomaly detection with parameter adaptation for KPI abrupt changes," IEEE Transactions on Network and Service Management, vol. 17, no. 3, pp. 1294–1308, 2020.

[3] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," Journal of Experimental Social Psychology, vol. 49, no. 4, pp. 764–766, 2013.

[4] Mohamed Moulay, Rafael Garcia Leiva, Vincenzo Mancuso, Pablo J. Rojo Maroni, and Antonio Fern´andez Anta, "Ttrees: Automated classification of causes of network anomalies with little data," in 2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2021, pp.199–208.

[5] Luca Scrucca, Michael Fop, T Brendan Murphy, and Adrian E Raftery, "mclust5: clustering, classification and density estimation using gaussian finite mixture,models," The R journal, vol. 8, no. 1, pp. 289, 2016.

[6] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, and Teunis Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," SIGCOMM Comput. Commun. Rev., vol. 27, no. 3, pp. 67–82, jul 1997