University of Padova

Department of Mathematics
Master Thesis in Data Science

Adaptive Density-Aware Sampling For High-Dimensional Datasets

Supervisor
Professor Susto Gian Antonio
University of Padova

Co-Advisor
David Dandolo
Statwolf Data Science S.r.l.

Master Candidate
Devis Marzola

Academic Year 2025–2026

Acknowledgements

I feel truly fortunate to be surrounded by such caring people and meaningful relationships in my life. My biggest thanks go to my family, especially my parents, my sister, and my grandmother, whose constant presence, encouragement, and belief in me made this achievement possible. I am deeply grateful to my friends from my hometown, the ones I grew up with and shared so many moments with over the years. Knowing I could always count on you has meant more than words can say. Among them, a special thought goes to my best friend Chiara, who is no longer with us. She was a fundamental part of my life, and I carry her memory with me every day. She will always remain in my heart. A special thanks goes to the friends I met during my Master's degree in Padova, Andrea, Francesco, Mattia, Mattia, and Matteo for the laughs, the support, and all the moments that made this journey unforgettable. This achievement is not just mine, but belongs to all of you who have been part of this journey. I would also like to thank the company where I carried out my internship for the opportunity to work on a real-world project and for their support during this experience.

Abstract

In the era of big data, the efficient management and analysis of large and highly non-uniform datasets have become critical challenges in machine learning and data science. This thesis addresses the problem of data reduction by proposing a novel sampling framework designed to preserve the structural properties of the feature space while reducing redundancy. The work introduces the Cluster-Based Density (CBD) sampling framework, a method that leverages density-based clustering, specifically HDBSCAN, to guide the sampling process. Unlike traditional approaches that focus primarily on statistical representativeness, the proposed method explicitly accounts for the spatial distribution of data points. By selecting fewer samples from dense regions and more from sparse areas, the framework aims to maintain a balanced and informative representation of the dataset. The effectiveness of the approach is evaluated through its application to a real-world dataset provided by an industrial partner. The results demonstrate that the CBD method significantly reduces dataset size while preserving predictive performance, particularly in complex and heterogeneous feature spaces where standard sampling techniques tend to fail. Furthermore, the thesis extends the framework to dynamic environments by introducing an adaptive procedure capable of incrementally integrating new observations. This extension allows the model to update its internal structure over time, ensuring that the sampled representation remains consistent with the evolving data distribution. The results highlight the effectiveness and scalability of the proposed method, making it a practical solution for data reduction in both static and dynamic machine learning scenarios.

Contents

1 Introduction **1**
- 1.1 The Concept of Sampling . . . 1
- 1.2 Sampling Framework . . . 1
  - 1.2.1 Steps Involved in Sampling . . . 2
- 1.3 Sampling in Machine Learning . . . 2
- 1.4 Types of Sampling in Machine Learning . . . 3

2 Density-Aware Sampling Process **5**
- 2.1 Sampling Objective . . . 5
- 2.2 Limitations of Machine Learning Sampling Methods . . . 5
- 2.3 HDBSCAN . . . 6
  - 2.3.1 Algorithm Overview . . . 7
  - 2.3.2 Core Distance . . . 10
  - 2.3.3 Cluster Membership Probabilities . . . 11
- 2.4 Cluster-Based Density-Aware Sampling . . . 12
  - 2.4.1 Motivation and Objectives of the Sampling Strategy . . . 12
  - 2.4.2 Cluster Density Estimation . . . 12
  - 2.4.3 Sampling Weights . . . 13
  - 2.4.4 Sampling Probabilities . . . 14
  - 2.4.5 Density-Aware Sampling Procedure . . . 15
  - 2.4.6 CBD Sampling Algorithm . . . 16

3 Application to a Real-World Dataset **17**
- 3.1 Introduction . . . 17
- 3.2 The Hosting Company: Statwolf . . . 17
- 3.3 Dataset Description . . . 18
- 3.4 Category-Based Data Preparation . . . 18
- 3.5 Input Feature Densities . . . 19
  - 3.5.1 Simple Random Sampling . . . 20
- 3.6 CBD Sampling . . . 21
  - 3.6.1 Sampling Based on Core Distances . . . 23
  - 3.6.2 Sampling Based on Probabilities . . . 25
  - 3.6.3 Sampling Based on Cluster Sizes . . . 27
  - 3.6.4 Final Sampled Dataset . . . 29
- 3.7 Cluster Metadata . . . 31
- 3.8 Regression Performance and Sampling Evaluation . . . 32
  - 3.8.1 Experimental Setup . . . 32
  - 3.8.2 Pipeline Execution Time Analysis . . . 35
  - 3.8.3 Discussion of Results . . . 36

4 Adaptive Update of the Sampling Procedure **37**
- 4.1 Assignment of New Observations . . . 37
- 4.2 Formation of New Clusters from Noise . . . 41

4.3 Cluster Evolution under Seasonal Distribution Shift 42

5 Conclusions **46**
5.1 Research Objective 46
5.2 Main Findings 46
5.3 Limitations and Improvements 47
5.4 Future Directions 47
5.5 Final Remarks 48

A Supplementary Results for the Real-World Dataset **49**
A.1 Results for *active_circuite* = 0.0 50
A.1.1 Core Distance-based CBD sampling 51
A.1.2 Probabilities-based CBD sampling 52
A.1.3 Cluster Size-based CBD sampling 53
A.1.4 Cluster Metadata 54
A.2 Results for *active_circuite* = 2.0 55
A.2.1 Core Distance-based CBD sampling 56
A.2.2 Probabilities-based CBD sampling 57
A.2.3 Cluster Size-based CBD sampling 58
A.2.4 Cluster Metadata 59

List of Figures

1.1 Example of Sampling . . . 1
1.2 Main steps involved in the sampling process . . . 2
1.3 Overview of the main sampling techniques in machine learning . . . 4

2.1 Example of minimum spanning tree constructed using mutual reachability distances 7
2.2 Hierarchical clustering structure obtained from the minimum spanning tree . . . 8
2.3 Condensed cluster tree highlighting cluster persistence across density levels . . . 9
2.4 Final cluster selection based on stability across density levels . . . 10
2.5 Illustration of core distance for different points with $k = 5$ . . . 10
2.6 Effect of mutual reachability distance with $k = 5$ . . . 11

3.1 KDE of the input features for the subset *active_circuite* = 1.0. . . . 19
3.2 Comparison between the original dataset and the subset obtained via SRS. . . . 20
3.3 Distribution of the input features across the clusters identified by HDBSCAN. . . . 21
3.4 KDE comparison of the input features before and after core distance-based CBD sampling. . . . 23
3.5 Histogram comparison of the input features before and after core distance-based CBD sampling. . . . 24
3.6 KDE comparison of the input features before and after probabilities-based CBD sampling. . . . 25
3.7 Histogram comparison of the input features before and after probability-based CBD sampling. . . . 26
3.8 KDE comparison of the input features before and after cluster size-based CBD sampling. . . . 27
3.9 Histogram comparison of the input features before and after cluster size-based CBD sampling. . . . 28
3.10 KDE comparison of the input features before and after probability-based CBD sampling to the full dataset. . . . 29
3.11 Histogram comparison of the input features before and after probability-based CBD sampling to the full dataset. . . . 30
3.12 Cluster-level metadata based on membership probabilities. . . . 31

4.1 Assignment of a point consistent with the existing cluster structure. . . . 39
4.2 Assignment of a perturbed point classified as noise. . . . 40
4.3 Comparison between original cluster 0 points (January) and newly assigned points (first week of July) . . . 43
4.4 Comparison between original cluster 0 points (January) and newly assigned points (second and third week of July) . . . 44
4.5 Comparison between original cluster 0 points (January) and newly assigned points (full month of July) . . . 45

A.1 Distribution of input features across clusters identified by HDBSCAN. . . . 50
A.2 KDE and histogram comparison of the input features before and after core distance-based CBD sampling. . . . 51

A.3 KDE and histogram comparison of the input features before and after core distance-based CBD sampling. 52
A.4 KDE and histogram comparison of the input features before and after core distance-based CBD sampling. 53
A.5 Cluster-level metadata based on membership probabilities. 54
A.6 Distribution of input features across clusters identified by HDBSCAN. 55
A.7 KDE and histogram comparison of the input features before and after core distance-based CBD sampling. 56
A.8 KDE and histogram comparison of the input features before and after core distance-based CBD sampling. 57
A.9 KDE and histogram comparison of the input features before and after core distance-based CBD sampling. 58
A.10 Cluster-level metadata based on membership probabilities. 59

List of Tables

3.1 RMSE comparison using only model features 33
3.2 RMSE comparison using both model and target features 34
3.3 Pipeline execution time comparison (hours) 35

1

Introduction

1.1 The Concept of Sampling

Sampling is the process of selecting a subset of observations from a larger population in order to estimate population characteristics (see Figure 1.1). A sample is a collection of individuals, things, or things collected from a large population for measurement in research. So, sampling is performed to obtain accurate data. We employ sampling because examining an entire population would generally be extremely costly and time-consuming. For example, evaluating whether all the chips produced in a factory meet quality standards would be impractical. Instead, we select a subset of items, typically at random, and assess relevant characteristics such as flavour, shape, and size. In this context, sampling represents a fundamental methodological approach, particularly when dealing with large populations, as it enables us to obtain reliable insights while substantially reducing the required resources.

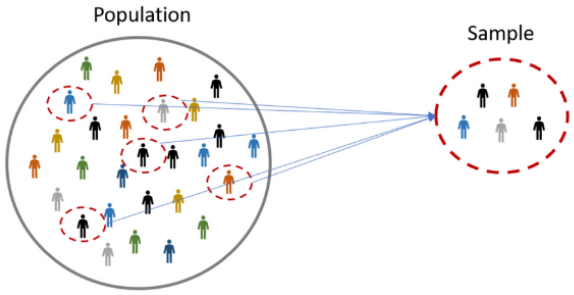

Figure 1.1: Example of Sampling

1.2 Sampling Framework

In order to perform sampling, it requires that we carefully define our population and the method by which we will select, and possibly reject, observations to be a part of our data sample. This may very well be defined by the population parameters that we wish to estimate using the sample. Some aspects to consider before collecting a data sample include:

- *sample goal*: the population property that we wish to estimate using the sample;
- *population*: the scope or domain from which observations could theoretically be made;

- *selection criteria*: the methodology that will be used to accept or reject observations in our sample;
- *sample size*: the number of observations that will constitute the sample.

1.2.1 Steps Involved in Sampling

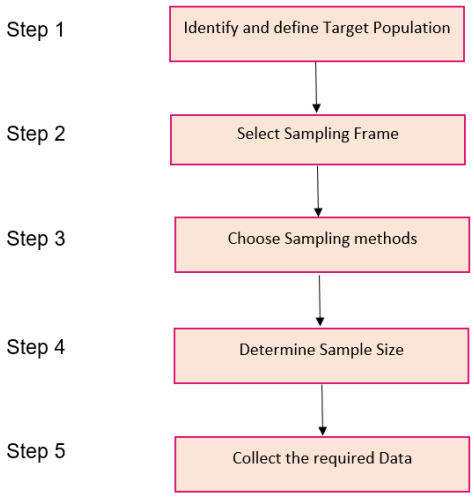

Figure 1.2: Main steps involved in the sampling process

The sampling process can be structured into a sequence of key steps, as illustrated in Figure 1.2:

1. Step 1: The first stage consists of clearly defining the target population.
2. Step 2: Selection of the sampling frame, i.e., the list of elements or individuals that make up the population from which the sample is drawn.
3. Step 3: Selection of the sampling method. In many applications, probability sampling methods are preferred, as they ensure that each unit has a known and non-zero probability of being selected.
4. Step 4: Determination of the sample size, namely the number of units to be included in the sample to achieve the desired level of accuracy and precision in inference. In general, larger sample sizes lead to more accurate estimates.
5. Step 5: Once the target population, sampling frame, sampling method, and sample size have been defined, data collection can be carried out from the selected sample.

1.3 Sampling in Machine Learning

A solid understanding of sampling techniques is essential in data science, as it enables us to manage data effectively, develop robust predictive models, ensure compliance with regulatory requirements, and support informed decision-making. These competencies are particularly relevant in practical applications, where datasets often present challenges such as large scale, structural complexity, class imbalance, or temporal dynamics. In this context, sampling methods play a crucial role in machine learning. They influence not only model performance, but also computational efficiency, scalability, and the ability to adapt to evolving data environments. The main motivations for employing sampling techniques can be summarized as follows:

- **Handling Real-World Data Complexity**: In real-world scenarios, data frequently exhibit significant variability in both scale and structure. By leveraging appropriate sampling techniques, we can manage this complexity more effectively, especially when dealing with large or heterogeneous datasets that cannot be fully processed due to computational or operational constraints.
- **Improving Model Performance**: Sampling strategies have a direct impact on the quality of machine learning models. By ensuring that training data are representative and properly balanced, we can enhance the accuracy and reliability of predictive models.
- **Mitigating Class Imbalance**: Many practical applications involve imbalanced datasets, where certain classes are underrepresented. Techniques such as stratified sampling, oversampling, and undersampling allow us to address this issue, improving the model's ability to learn from minority classes and thus enhancing overall predictive performance.
- **Increasing Efficiency and Reducing Costs**: Sampling can significantly reduce the resources required for data collection and processing. For example, cluster sampling is particularly useful when the population is geographically distributed or when data acquisition is costly, enabling a more efficient workflow.
- **Ensuring Scalability in Large-Scale Settings**: In big data contexts, sampling techniques such as reservoir sampling become essential. They allow us to handle data streams and extremely large datasets that cannot be entirely stored in memory, supporting the development of scalable machine learning systems.
- **Supporting Data-Driven Decision-Making**: Sampling is a fundamental component of exploratory data analysis. By carefully selecting and analyzing subsets of data, we are able to derive insights and make predictions that are representative of the underlying population.
- **Facilitating Continuous Learning**: In dynamic environments, models often need to be updated as new data become available. Through appropriate sampling strategies, we can efficiently incorporate new information into the learning process without retraining models on the entire dataset.

The increasing size and dimensionality of modern datasets pose significant computational challenges for machine learning algorithms. In addition, the curse of dimensionality and the degradation of distance-based methods in high-dimensional spaces further complicate data analysis. In this context, efficient sampling techniques become essential to reduce computational costs while preserving the structural properties of the data.

1.4 Types of Sampling in Machine Learning

In probability sampling, each element of the population is associated with a known and non-zero probability of being selected. This property ensures that the resulting sample can be considered representative of the underlying population, particularly when the population exhibits a relatively homogeneous structure. In the context of machine learning, probability-based sampling techniques are widely adopted, as they allow us to efficiently select representative subsets of data while reducing computational costs and preserving the original data distribution. A schematic overview of the main sampling techniques in machine learning is presented in Figure 1.3. The principal methods can be summarized as follows:

- **Simple Random Sampling**: Simple random sampling represents the most basic sampling strategy. It consists of selecting data points uniformly at random from a dataset, ensuring that each observation has the same probability of being included in the sample.

This approach reduces selection bias and minimizes sampling error, and is often used during exploratory data analysis to obtain an unbiased view of the data.

- **Systematic Sampling**: Systematic sampling involves selecting elements from a population according to a fixed and predetermined interval. Once the sampling interval is defined, typically as the ratio between the population size and the desired sample size, every n-th element is included in the sample. This method is particularly suitable for large datasets as it ensures a uniform coverage of the population, although it may be sensitive to periodic patterns in the data.
- **Stratified Random Sampling**: Stratified random sampling consists of partitioning the population into homogeneous subgroups, or strata, based on specific characteristics and then performing random sampling within each stratum. This approach improves representativeness and reduces estimation variance. It is especially important in machine learning applications involving imbalanced datasets, where preserving the proportion of minority classes is crucial.
- **Cluster Sampling**: Cluster sampling involves dividing the population into groups, or clusters, and randomly selecting a subset of these clusters from which observations are collected. Unlike stratified sampling, clusters are typically naturally occurring groups. This method is particularly useful in scenarios where data collection is costly or geographically distributed. Variants include single-stage, two-stage, and multi-stage cluster sampling.

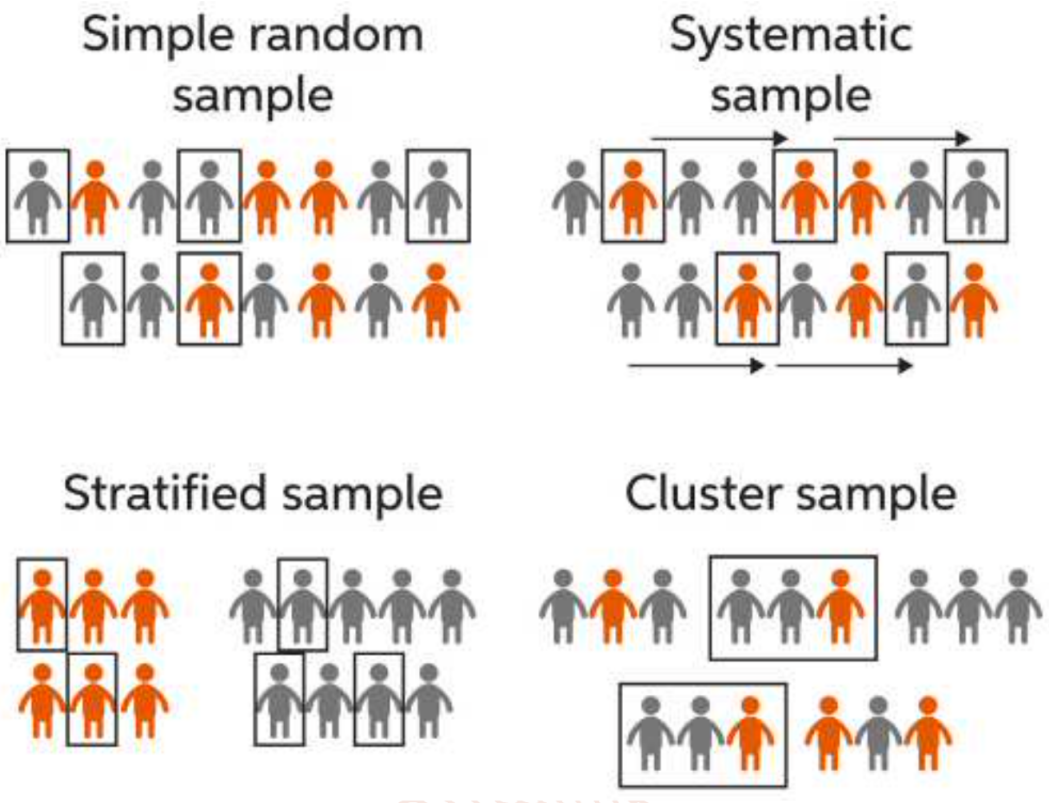

Figure 1.3: Overview of the main sampling techniques in machine learning

2

Density-Aware Sampling Process

2.1 Sampling Objective

Sampling is widely used in machine learning and data analysis as a means to reduce the size of large datasets while preserving their most relevant characteristics. Traditional sampling strategies are typically designed to ensure statistical representativeness, which means that the empirical distribution of the selected subset closely approximates that of the original data set. However, in many real-world applications, datasets are characterized by a large number of observations and a highly non-uniform distribution across the feature space. In such contexts, the objective of sampling extends beyond the preservation of the underlying distribution. Instead, the goal is to construct a reduced subset that retains the global geometric structure of the data while limiting redundancy in regions with high point density. From an operational perspective, this implies selecting fewer observations from densely populated areas and relatively more samples from sparsely populated regions, thereby improving the overall coverage of the feature space. This viewpoint emphasizes the importance of adopting sampling strategies that explicitly take into account the spatial organization of the data rather than relying exclusively on probabilistic notions of representativeness.

2.2 Limitations of Machine Learning Sampling Methods

In this section, we analyze a set of commonly adopted sampling techniques in machine learning, previously introduced, and examine their behavior when applied to datasets characterized by complex geometric structures. The focus is placed on how their underlying mechanisms influence the spatial distribution of the sampled points. Through this analysis, we identify the key limitations that prevent these methods from effectively addressing the sampling objective defined in this work.

Simple Random Sampling
Simple random sampling assigns an equal probability of selection to each observation in the dataset and is widely used due to its simplicity and well-established theoretical properties. However, when the data exhibit a non-uniform distribution, this method inherently preserves the original density pattern: regions with a high concentration of points remain densely populated in the sampled subset, while sparse regions continue to be underrepresented. As a result, simple random sampling neither reduces redundancy in dense areas nor improves coverage in low-density regions. Given that the objective of this work is to mitigate the dominance of dense regions, this approach does not provide a suitable solution.

Systematic Sampling
Systematic sampling involves selecting observations at regular intervals based on their position in an ordered dataset, typically by choosing every k-th element after a random starting point. This approach is independent of the feature values and therefore strongly depends on the ordering of the data. If the dataset is organized in such a way that similar observations are grouped together, systematic sampling may repeatedly select points from the same dense regions, leading to significant redundancy in the sampled subset. Since the ordering of the data is generally unrelated to the geometric structure of the feature space, this method does not ensure a balanced spatial distribution. Consequently, it cannot be used to control local density or to promote a more uniform coverage of the feature space.

Stratified Random Sampling
Stratified random sampling relies on partitioning the population into disjoint subgroups and performing sampling independently within each group. This approach is particularly effective when meaningful strata are naturally available, such as class labels or categorical attributes. In the scenario considered here, however, density variations occur in a continuous and multidimensional feature space, where no natural stratification is readily identifiable. Defining strata based on individual features would fail to capture the joint distribution of the data, reflecting only marginal properties. Furthermore, constructing strata in a high-dimensional space would require arbitrary discretization, introducing additional assumptions and potential distortions. For these reasons, stratified sampling does not adequately address the main challenge, namely the heterogeneous distribution of points across the feature space.

Cluster Sampling
Cluster sampling consists of partitioning the dataset into groups and randomly selecting a subset of these clusters, including all observations within the chosen groups. This method is often adopted when sampling individual elements is impractical. In the present context, clusters tend to correspond to regions of high density in the feature space. Selecting entire clusters would therefore preserve large sets of highly similar observations, while potentially excluding other regions altogether. Both outcomes lead to a distorted representation of the overall data structure. Moreover, cluster sampling operates at the level of groups rather than individual observations, and thus does not allow for a gradual reduction of density within clusters. This lack of flexibility makes it unsuitable for scenarios where the objective is to reduce redundancy in dense regions while preserving the global geometric structure of the dataset.

2.3 HDBSCAN

To address the limitations of traditional sampling approaches, this work adopts a clustering-based strategy aimed at capturing variations in point density across the feature space. In particular, the HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) algorithm is employed to identify regions characterized by different density levels and to provide a structured representation of the underlying data geometry. HDBSCAN extends classical density-based clustering methods, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), by introducing a hierarchical framework and removing the need to specify the number of clusters a priori. This property makes it particularly suitable for datasets that exhibit heterogeneous density patterns and complex spatial structures. By explicitly modeling local density variations, HDBSCAN enables a principled distinction between dense regions, sparse areas, and noise points. This capability is essential for guiding density-aware sampling strategies, as it provides a foundation for selectively reducing redundancy in high-density regions while preserving the overall geometric structure of the data.

2.3.1 Algorithm Overview

HDBSCAN operates by combining geometric proximity and local density information in order to identify meaningful structures in the data.

The first step of the algorithm consists of estimating the local density around each observation. This is achieved by computing the distance between each point and its k-th nearest neighbor, where k is determined by the parameter ***min_samples***. This quantity, referred to as the **core distance**, provides a point-wise measure of how densely populated the neighborhood of each observation is. Rather than relying directly on standard pairwise distances, HDBSCAN introduces a modified metric known as the **mutual reachability distance**, which integrates both geometric and density-related information:

$$d_{\text{reach}}(a, b) = \max\{\text{core}_k(a), \text{core}_k(b), \|a - b\|\} \tag{2.1}$$

This transformation effectively increases distances in low-density regions while preserving relatively small distances in dense areas. As a result, points located in dense regions remain strongly connected, whereas points in sparse regions become more isolated. Based on the mutual reachability distances defined in Equation 2.1, the algorithm constructs a weighted graph in which observations correspond to vertices and edge weights reflect their pairwise relationships. The minimum spanning tree of this graph provides a compact representation of the connectivity structure of the data, as illustrated in Figure 2.1.

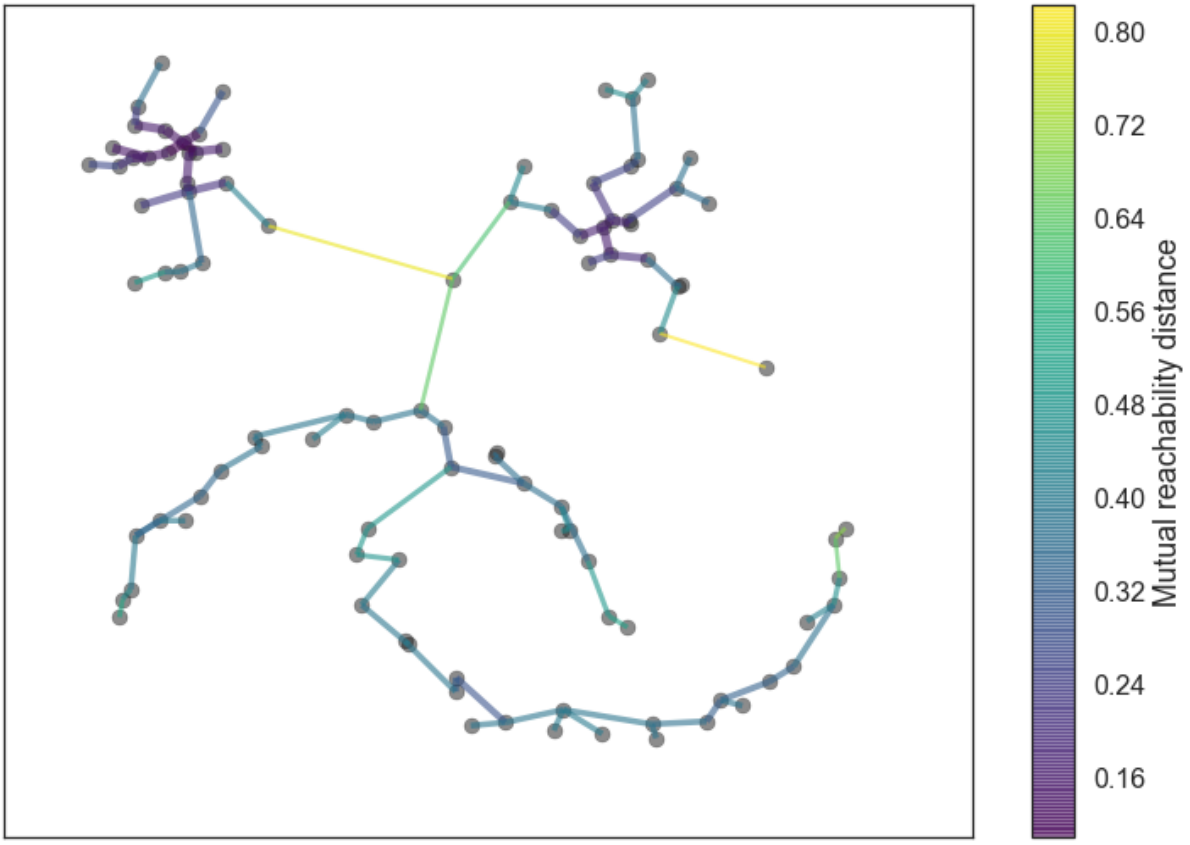

Figure 2.1: Example of minimum spanning tree constructed using mutual reachability distances

Starting from the structure of the minimum spanning tree, HDBSCAN builds a hierarchy of clusters by progressively removing edges in order of increasing distance. This procedure generates a sequence of nested partitions corresponding to different density levels, leading to a hierarchical organization of the data, as depicted in Figure 2.2.

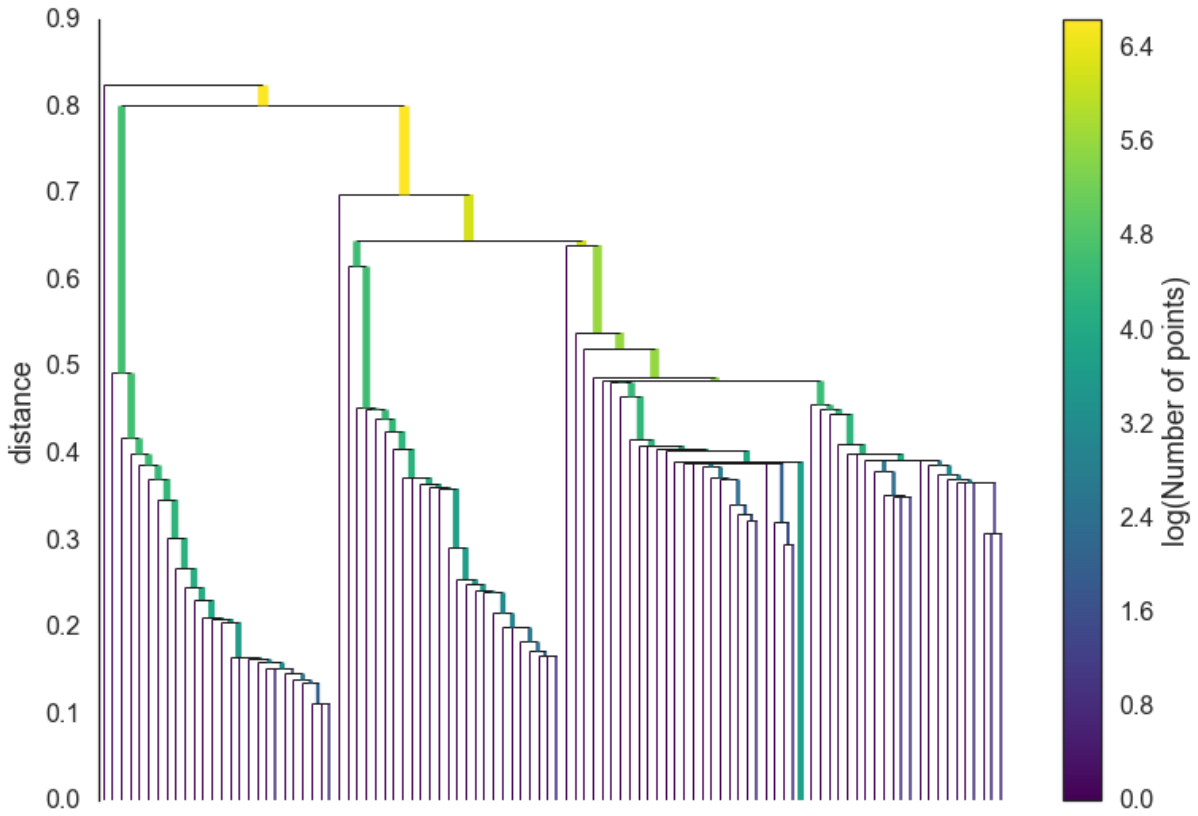

Figure 2.2: Hierarchical clustering structure obtained from the minimum spanning tree

To extract meaningful clusters from this hierarchy, the algorithm introduces the parameter ***minimum_cluster_size***, which plays a central role in the condensation process. This parameter defines the minimum number of points that a cluster must contain in order to be considered significant, effectively acting as a threshold for filtering out small and potentially spurious groupings. As the hierarchical structure is traversed from higher to lower density levels, the algorithm continuously monitors the evolution of clusters and evaluates each splitting event. In particular, when a cluster divides into two or more subclusters, the sizes of the resulting components are examined to determine whether the split reflects a meaningful structural separation or merely the detachment of a limited number of observations. If one of the resulting subclusters contains fewer points than the specified *minimum_cluster_size*, it is interpreted as a set of points that are no longer sufficiently supported by the surrounding density. In this case, these points are considered to be leaving the parent cluster and are treated as noise or weakly associated observations, while the parent cluster retains its identity and continues to persist within the hierarchy. Conversely, when all resulting subclusters exceed the minimum size threshold, the split is regarded as a genuine division into distinct clusters. This mechanism allows the algorithm to preserve only those structures that are sufficiently large and stable, while progressively eliminating small, short-lived branches that are likely to arise from local fluctuations in density. Through this process, the original hierarchical tree is simplified into a condensed representation that emphasizes the most relevant and persistent clusters, providing a more robust and interpretable description of the data.

The resulting condensed hierarchy is illustrated in Figure 2.3, where the long-lived clusters correspond to more stable and meaningful structures in the data.

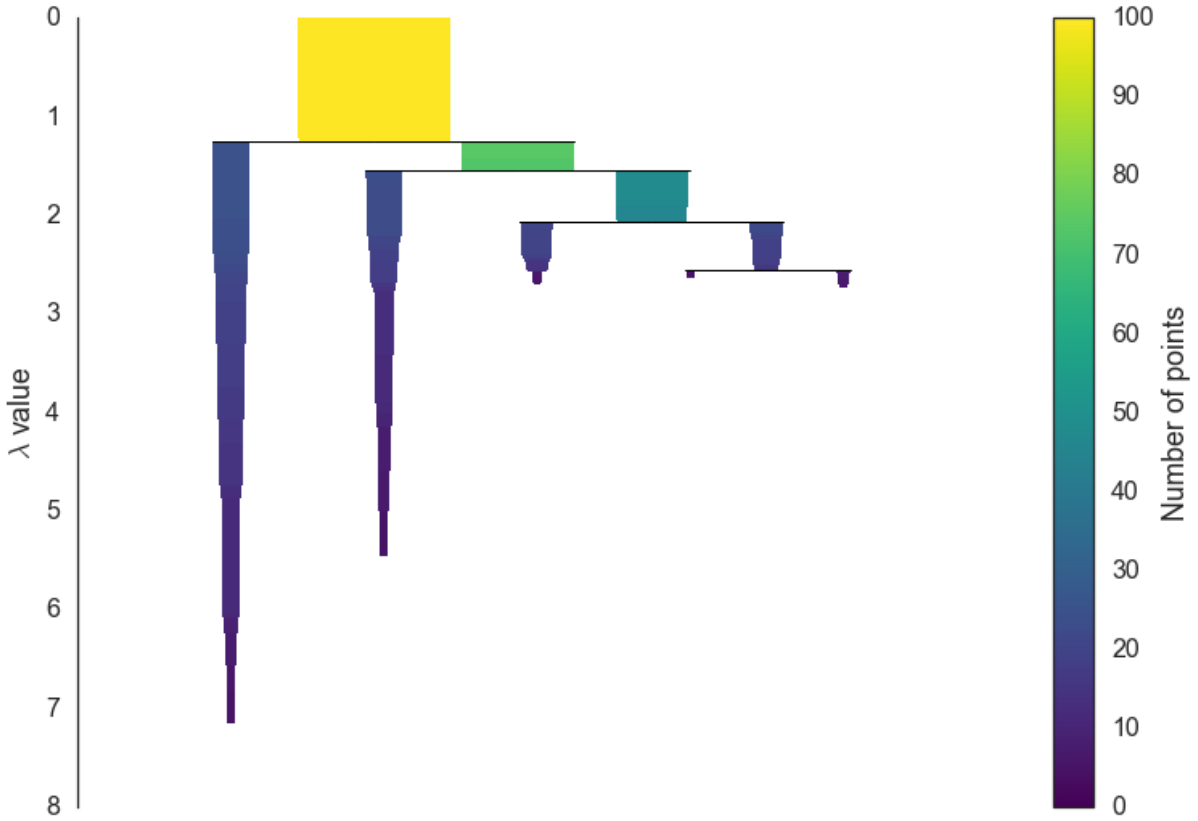

Figure 2.3: Condensed cluster tree highlighting cluster persistence across density levels

Cluster selection is then based on their stability under varying density thresholds. To formalize this, HDBSCAN defines a measure of local density as follows:

$$\lambda = \frac{1}{\text{distance}} \tag{2.2}$$

For each group, λ_{birth} denotes the density level at which the group appears, while λ_{death} corresponds to the level at which it splits. Each point p is associated with a value λ_p, which represents the density level at which it leaves the cluster.

The stability of a cluster is defined as

$$\text{stability} = \sum_{p \in \text{cluster}} (\lambda_p - \lambda_{\text{birth}}) \tag{2.3}$$

Clusters that persist across a wide range of density levels are considered stable and are selected for the final partition. This selection process respects the hierarchical structure: if a cluster is selected, its descendants are not chosen separately, thereby avoiding redundancy.

The final result of this process is illustrated in Figure 2.4, where only the most stable clusters are retained, while points that do not belong consistently to any cluster are labeled as noise.

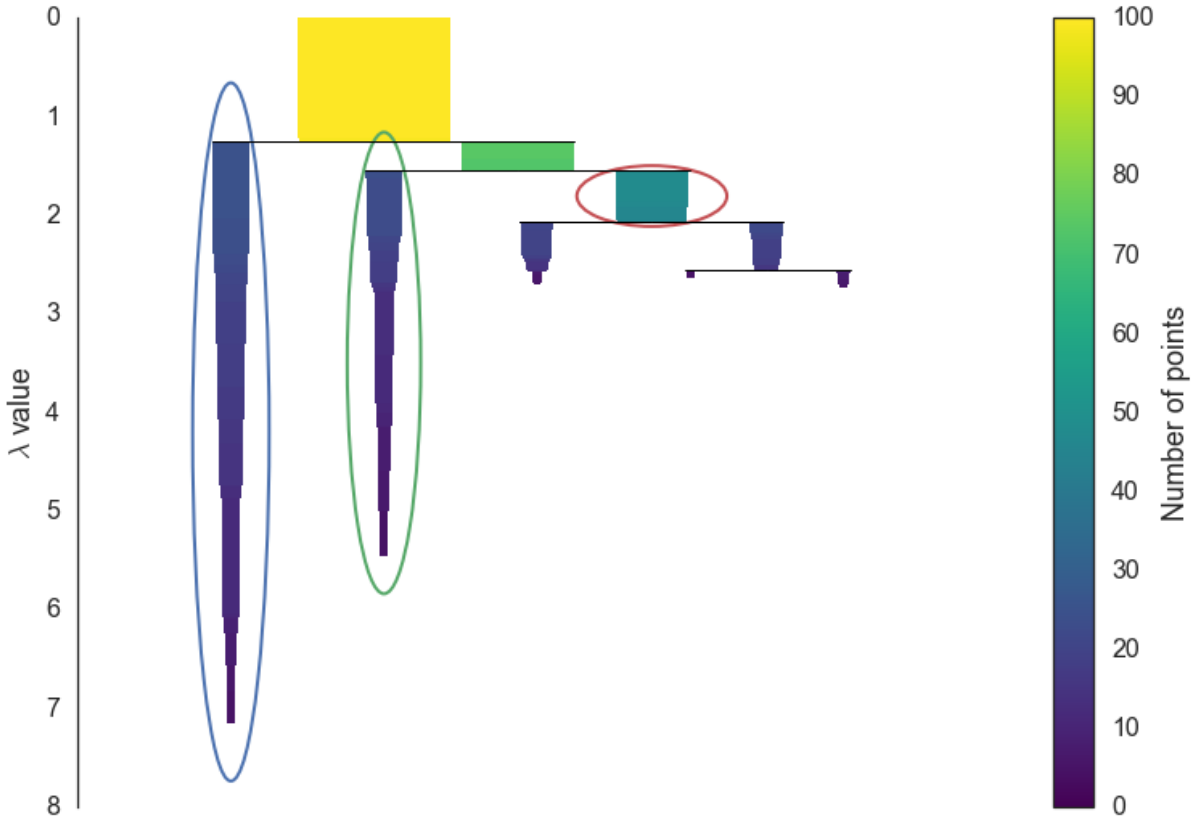

Figure 2.4: Final cluster selection based on stability across density levels

2.3.2 Core Distance

The notion of **core distance** plays a fundamental role in HDBSCAN, as it provides a local and adaptive characterization of point density in the feature space. For each observation x_i, the core distance is defined as the distance to its k-th nearest neighbor, where k is determined by the parameter ***min_samples***:

$$\text{core\_dist}(x_i) = \text{core}_k(x_i) \tag{2.4}$$

An illustrative example of this concept is shown in Figure 2.5, where the core distance corresponds to the radius of a neighborhood containing the k nearest points.

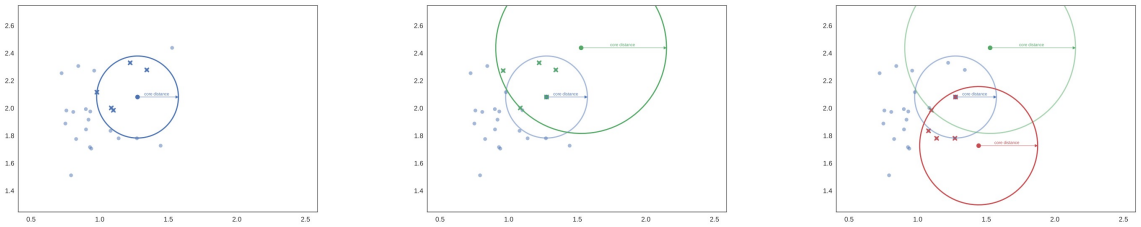

Figure 2.5: Illustration of core distance for different points with $k = 5$

From a geometric perspective, the core distance can be interpreted as the radius of the smallest hypersphere centered at x_i that contains at least k neighboring points. Consequently, it provides a direct indication of how concentrated the data are around a given observation.

Points located in densely populated regions are associated with small core distances, since only a limited neighborhood is required to include the prescribed number of neighbors. Conversely, observations in sparse regions exhibit larger core distances, reflecting the need to consider a wider neighborhood to reach the same number of surrounding points. Unlike global density measures, which impose a uniform notion of density across the entire dataset, the core distance adapts to local variations in the data distribution. This property is particularly important in settings where density is heterogeneous across the feature space. By assigning a density-related quantity to each observation, HDBSCAN is able to capture these variations without relying on a single global threshold.

The core distance is not used in isolation, but rather contributes to the definition of the **mutual reachability distance** between pairs of points. In this context, the core distances of both observations are combined with their original pairwise distance, typically the standard Euclidean distance, as defined in Equation 2.1. The effect of this transformation is illustrated in Figure 2.6, where the distances are expanded in low-density regions while remaining relatively unchanged in dense areas.

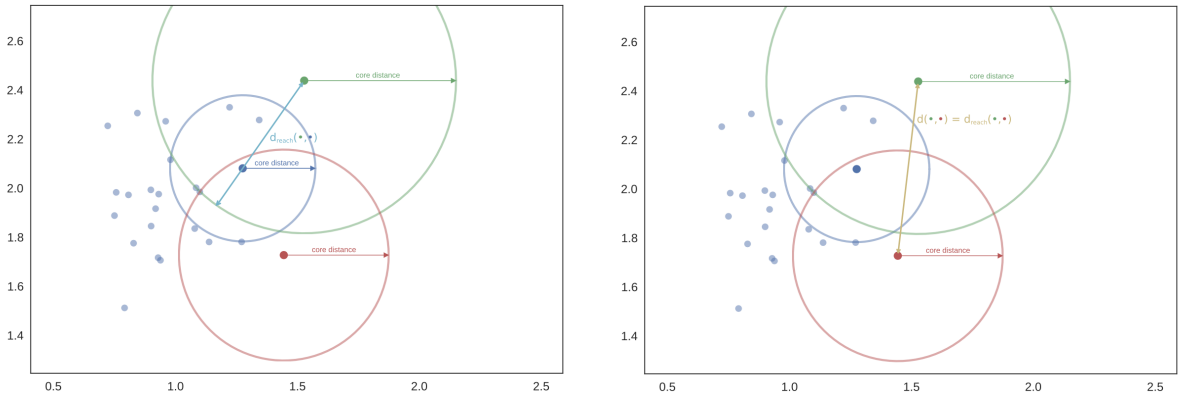

Figure 2.6: Effect of mutual reachability distance with $k = 5$

This construction inflates distances in regions where local density is low, making points in sparse areas less likely to be grouped together. At the same time, distances in dense regions remain relatively small, preserving strong connections between nearby observations. In this way, the core distance acts as a mechanism for embedding local density information directly into the geometry used for clustering.

2.3.3 Cluster Membership Probabilities

In addition to providing a hard assignment of observations to clusters, HDBSCAN associates each point with a membership probability, commonly denoted as ***probabilities***. This quantity, taking values in the interval $[0, 1]$, expresses the degree of confidence with which a point can be considered part of its assigned cluster and is derived from the hierarchical structure underlying the clustering process. These probabilities are closely related to the notion of cluster stability. As discussed in the previous subsection, clusters emerge and disappear as the density threshold varies, giving rise to a hierarchical organization of the data. Clusters that persist over a wide range of density levels are interpreted as robust and well-defined structures, whereas short-lived clusters are considered less reliable. The membership probability of a point reflects both the persistence of the cluster to which it belongs and the strength of its association with that structure. Points located in the core of dense and stable clusters are typically assigned probability values close to one. These observations remain consistently associated with the same cluster across multiple density levels and can therefore be regarded as highly representative of the underlying structure. In contrast, points located near cluster boundaries or in regions charac-

terized by rapid density variations tend to exhibit lower probability values. In such cases, the assignment is inherently less stable, as the point may change the cluster membership or become weakly associated with any cluster as the hierarchy evolves. From a geometric perspective, the membership probability can be interpreted as a soft measure of the strength of association between a point and a dense region of the feature space. Unlike a binary cluster label, it provides additional information about the uncertainty of the assignment, which is particularly relevant in the presence of ambiguous or transitional regions.

In the context of density-aware sampling, these probabilities offer a principled mechanism to differentiate between highly representative observations within dense clusters and points whose association is less certain. This allows for more refined sampling strategies, where decisions are guided not only by the presence of clusters but also by their internal structure and the relative importance of individual observations.

2.4 Cluster-Based Density-Aware Sampling

2.4.1 Motivation and Objectives of the Sampling Strategy

Following the clustering phase performed with HDBSCAN, a density-aware sampling strategy, referred to as **Cluster-Based Density (CBD) sampling**, is applied within the identified clusters. This approach is motivated by the need to effectively handle datasets characterized by a highly non-uniform distribution of observations across the feature space, where certain regions exhibit a high concentration of points while others remain relatively sparse. In such settings, uniform sampling strategies tend to over-represent dense areas while potentially discarding valuable information from less populated regions. The proposed CBD sampling method addresses this limitation by explicitly leveraging the cluster structure obtained through HDBSCAN and incorporating local density information into the sampling process. Rather than treating all clusters uniformly, the approach assigns different sampling rates based on their estimated density. As a result, clusters corresponding to dense regions are sampled more aggressively, whereas clusters associated with lower-density regions are preserved to a greater extent. The objective of this strategy is twofold. First, it aims to reduce the number of observations in highly populated regions, thereby limiting their influence in subsequent analyses or learning tasks. Second, it seeks to retain a representative subset of points across clusters with varying density levels, ensuring that both dense and sparse structures are adequately captured. This balance allows the resulting subsampled dataset to preserve the essential geometric and statistical properties of the original data distribution, while achieving a significant reduction in size.

2.4.2 Cluster Density Estimation

The first step of the proposed approach consists in estimating the density associated with each cluster. Let C_j denote the j-th cluster identified by HDBSCAN, and let $|C_j|$ represent its cardinality, i.e., the number of observations assigned to that cluster.

A first density estimate can be derived from the notion of core distance. As discussed previously, the core distance corresponds to the radius of the smallest neighborhood containing a fixed number of points, and therefore provides an inverse indication of local density: smaller core distances are associated with denser regions of the feature space. Based on this observation, the inverse of the core distance can be used as a proxy for the point-wise density. By averaging this quantity over all observations within a cluster, we obtain an aggregate measure of the typical density of that cluster. This estimate reflects how concentrated the observations are on average and enables a direct comparison across clusters.

Formally, the density of the cluster C_j can be defined as:

$$\rho_j^{(\text{core})} = \frac{1}{|C_j|} \sum_{x_i \in C_j} \frac{1}{\text{core\_dist}(x_i)}. \tag{2.5}$$

An alternative formulation is based on the membership probabilities provided by HDBSCAN. As discussed in the previous subsection, these probabilities quantify the degree of confidence with which each observation is associated with its assigned cluster and are derived from the stability of clusters within the hierarchical structure. In this case, the density of a cluster is computed as the average of the membership probabilities of its constituent points. This definition reflects the intuition that clusters composed of strongly associated observations tend to exhibit higher average probabilities, whereas clusters containing more peripheral or unstable points are characterized by lower values. Consequently, this measure can be interpreted as an indicator of how compact and well-defined a cluster is within the feature space. Formally, we define:

$$\rho_j^{(\text{prob})} = \frac{1}{|C_j|} \sum_{x_i \in C_j} \text{prob}(x_i). \tag{2.6}$$

Both definitions provide a cluster-level summary of density, capturing complementary aspects of the underlying structure. In particular, higher values of ρ_j correspond to clusters that are both more concentrated and more stable, and therefore more representative of dense regions of the feature space.

2.4.3 Sampling Weights

Once the cluster densities have been estimated, a sampling weight is assigned to each cluster in order to regulate the proportion of observations to be retained during the subsampling process. Specifically, the weight associated with cluster C_j is defined as:

$$w_j = \frac{1}{(\rho_j)^\alpha}, \tag{2.7}$$

where ρ_j denotes the chosen estimate of cluster density and $\alpha > 0$ is a tunable parameter that controls the influence of density on the sampling procedure.
The parameter α plays a key role in determining how strongly density differences between clusters affect the resulting weights. When α approaches zero, the dependence of w_j on ρ_j becomes negligible, and the weights tend to be approximately uniform across clusters. In this regime, the sampling strategy behaves similarly to a uniform allocation scheme. As α increases, the influence of the density of the cluster becomes progressively more pronounced. In particular, clusters characterized by higher density are assigned smaller weights, whereas clusters associated with lower density receive relatively larger weights. This mechanism results in a more aggressive reduction of observations in densely populated regions and a stronger preservation of points in sparse areas of the feature space.

Overall, this formulation establishes an inverse relationship between sampling weights and cluster density, while allowing its intensity to be controlled through a single interpretable parameter. As a consequence, the proposed weighting scheme provides a flexible framework for balancing two competing objectives: reducing the dominance of highly concentrated regions and maintaining an adequate representation of less dense, yet potentially informative, structures.

2.4.4 Sampling Probabilities

The sampling weights defined at the cluster level are subsequently transformed into sampling probabilities through a normalization step. This operation ensures that the weights define a valid probability distribution over the set of clusters. In particular, the sampling probability associated with the cluster C_j is given by:

$$p_j = \frac{w_j}{\sum_i w_i}, \tag{2.8}$$

where the denominator represents the sum of the weights across all clusters.
This normalization guaranties two key properties. First, all sampling probabilities are non-negative and sum to one, allowing them to be interpreted as proportions of the total sample to be allocated to each cluster. Second, the relative relationships induced by the weighting scheme are preserved: clusters with larger weights—corresponding to lower density—are assigned higher probabilities, while denser clusters receive smaller probabilities.

As an alternative, sampling probabilities can be defined based on the relative sizes of the clusters rather than their estimated densities. In this case, the probability of selecting observations from the cluster C_j can be expressed as:

$$p_j = \frac{\log(|C_j|)}{\sum_i \log(|C_i|)}, \tag{2.9}$$

The use of a logarithmic transformation is particularly suitable in this context, as it reduces the impact of large disparities in cluster cardinalities. In many practical scenarios, cluster sizes can differ by several orders of magnitude, leading to highly unbalanced allocations if probabilities were defined directly in proportion to $|C_j|$. In such cases, very large clusters would dominate the sampling process, resulting in a significant loss of information from smaller clusters. By applying the logarithmic function, the relative differences between cluster sizes are compressed: large clusters are down-weighted, while smaller clusters remain adequately represented in the sampling process. This allows the sampling procedure to maintain a degree of proportionality with respect to cluster size, without allowing dominant clusters to overwhelm the allocation. From a practical perspective, this transformation supports a more balanced trade-off between representativeness and diversity. While larger clusters are still assigned higher probabilities, ensuring that major structures in the data are adequately captured, smaller clusters remain sufficiently represented, preserving potentially informative but less frequent patterns.

The resulting probabilities, whether derived from density-based weights or from cluster cardinalities, determine the fraction of observations to be selected from each cluster during the sampling phase. This provides flexibility in adapting the allocation strategy to local density variations or to the relative sizes of clusters, depending on the objectives of the analysis.

2.4.5 Density-Aware Sampling Procedure

Once the sampling probabilities have been determined, the final step consists of selecting a subset of observations from each cluster according to the previously defined allocation scheme. Two alternative sampling settings can be considered depending on whether the desired total sample size is specified.

If a target sample size n is provided, the number of observations to be drawn from each cluster C_j is calculated as:

$$n_j = \lfloor p_j \cdot n \rfloor, \tag{2.10}$$

where p_j denotes the sampling probability associated with the cluster C_j and $\lfloor \cdot \rfloor$ denotes the floor operator (i.e., rounding down to the nearest integer). This formulation determines how the overall sampling budget is distributed between clusters.

Alternatively, if no global sample size is specified, the sampling procedure is performed independently within each cluster. In this case, the number of observations selected from the cluster C_j is given by:

$$n_j = \lfloor p_j \cdot |C_j| \rfloor, \tag{2.11}$$

where $|C_j|$ denotes the cardinality of the cluster C_j. This corresponds to selecting a fraction p_j of the observations in the cluster through uniform random sampling.

In both settings, the selection within each cluster is carried out uniformly at random. This ensures that the internal variability and structural characteristics of each cluster are preserved while avoiding biases that could arise from deterministic selection mechanisms. From a global perspective, the allocation induced by p_j introduces an adaptive mechanism that accounts for differences in the density or size of the cluster. Clusters corresponding to dense regions of the feature space are associated with smaller sampling fractions, leading to a stronger reduction in redundancy. In contrast, clusters located in sparse regions are preserved to a greater extent.

The resulting subsampled dataset achieves a balance between efficiency and representativeness. Dense regions are selectively thinned, whereas less populated areas remain adequately represented, enabling the reduced data set to preserve the essential geometric and statistical properties of the original distribution.

2.4.6 CBD Sampling Algorithm

The CBD sampling procedure can be formalized as an algorithmic framework that integrates clustering information with density-aware allocation.

Starting from the set of clusters $\{C_j\}_{j=1}^{N}$ obtained via HDBSCAN, the sampling probabilities are calculated based either on the densities of the clusters or on the sizes of the cluster. These probabilities are then used to determine the number of observations to be selected from each cluster, followed by a random selection step within each cluster.

Algorithm Cluster-Based Density (CBD) Sampling

Require: Dataset X, clusters $\{C_j\}_{j=1}^{N}$, total sample size n (optional)
Ensure: Subsampled dataset X'

Obtain clusters $\{C_j\}_{j=1}^{N}$ using HDBSCAN
Select sampling strategy
if density-based approach **then**
    Estimate cluster densities ρ_j
    Compute weights $w_j = 1/(\rho_j)^{\alpha}$
    Normalize weights to obtain probabilities p_j
else
    Compute probabilities p_j based on cluster sizes
end if
Compute number of samples per cluster
if n is provided **then**
    $n_j = \lfloor p_j \cdot n \rfloor$
else
    $n_j = \lfloor p_j \cdot |C_j| \rfloor$
end if
for each cluster C_j **do**
    Randomly select n_j observations from C_j
end for
$X' \leftarrow \bigcup_{j=1}^{N}$ sampled points from C_j
return X'

3

Application to a Real-World Dataset

3.1 Introduction

In this chapter, the Cluster-Based Density (CBD) sampling procedure introduced in the previous chapter is applied to a real-world dataset. The objective of this analysis is to assess the behavior of the method in practical scenarios, where data are typically large in scale and exhibit a highly non-uniform distribution across the feature space. The implementation of the proposed sampling strategy is described step by step, highlighting the key choices involved in adapting the method to the dataset under consideration. The resulting subsampled data were then analyzed to evaluate the effectiveness of the approach in reducing the influence of densely populated regions while preserving representative observations from less populated areas. This analysis provides insight into the practical applicability of the method in real-world and industrial settings and serves as a basis for the subsequent discussion of the results.

3.2 The Hosting Company: Statwolf

Statwolf is a data science and analytics company that hosted the internship associated with this thesis. It focuses on the development of advanced solutions for handling and analyzing complex data. The company provides a range of tools and methodologies for data integration, exploratory data analysis, and machine learning–based modeling, with the objective of transforming heterogeneous data sources into actionable insights. By combining data engineering, statistical analysis, and artificial intelligence techniques, Statwolf supports organizations to address data-intensive challenges. In particular, its solutions are designed to improve decision-making processes and improve operational efficiency across a variety of application domains.

STATWOLF

3.3 Dataset Description

The dataset used for the experimental analysis was provided by a Statwolf client company operating in the production of heat pumps. It consists of multivariate time-stamped measurements collected from different machines through on-board sensors, where each observation corresponds to an aggregated record with a temporal resolution of 15 seconds. A dedicated timestamp column specifies the acquisition time of each measurement, leading to a rapid growth in the number of observations even over short time horizons. Data are organized at the device level, and each machine belongs to a specific product family. In the broader application context, models are trained at the family level, allowing devices within the same group to contribute jointly to the learning process. This structure highlights the importance of variability between operating conditions, providing the information necessary to learn reliable relationships between inputs and outputs.

For the purposes of this thesis, the analysis focuses on data collected during **January 2025** for a single machine. The complete dataset consists of approximately **4 million observations and 46 variables**.

In the initial phase of the analysis, a subset of five input variables was selected, leading to a reduced dataset of **76013 observations and 5 variables**:

- *generic_signal_1*, numerical variable;
- *power_request*, numerical variable;
- *inlet_common_user*, numerical variable;
- *oulet_common_user*, numerical variable;
- *active_circuite*, categorical variable with three levels (0.0, 1.0, 2.0).

These variables represent the input features of the predictive model and define the feature space in which the clustering and sampling procedures are applied. Their distribution is therefore central to the analysis, as it directly influences the behavior of the proposed method.

The goal of the sampling procedure in this context is to reduce the size of the dataset while preserving the most informative patterns in the data. In particular, the approach aims to maintain representative observations across different operating conditions, ensuring that the resulting dataset retains sufficient variability for subsequent modeling tasks.

3.4 Category-Based Data Preparation

The first step of the analysis consists of partitioning the dataset according to the categorical variable *active_circuite*. This operation allows us to isolate observations corresponding to different operating modes and to analyze their distributions independently.

Starting from the dataset of size $(76013, 5)$, three subsets are obtained, each corresponding to one level of *active_circuite*:

- $(32577, 4)$ observations for $active\_circuit = 0.0$,
- $(21249, 4)$ observations for $active\_circuit = 1.0$,
- $(22187, 4)$ observations for $active\_circuit = 2.0$.

The CBD sampling procedure is then applied independently to each subset. This choice allows the method to capture the density structure specific to each category, avoiding distortions that could arise from mixing observations characterized by different operating conditions.

After sampling, the subsets can be recombined to form a unified dataset that preserves the contribution of each category while reducing redundancy within each group. This approach ensures that the resulting data retain the heterogeneity of the original feature space, providing a balanced basis for subsequent analyses.

For the remainder of this chapter, the analysis focuses on the subset corresponding to *active_circuite* = 1.0, which is used as a representative case to examine the effects of the sampling procedure and the distribution of the selected features. The results obtained for the other two subsets are reported in the Appendix A.

3.5 Input Feature Densities

Before applying the CBD sampling procedure, we examine the distribution of the input features using a Kernel Density Estimation (KDE) approach. This non-parametric method provides a smooth approximation of the underlying probability density functions without requiring assumptions on their functional form.

The estimated densities for the selected features are shown in Figure 3.1.

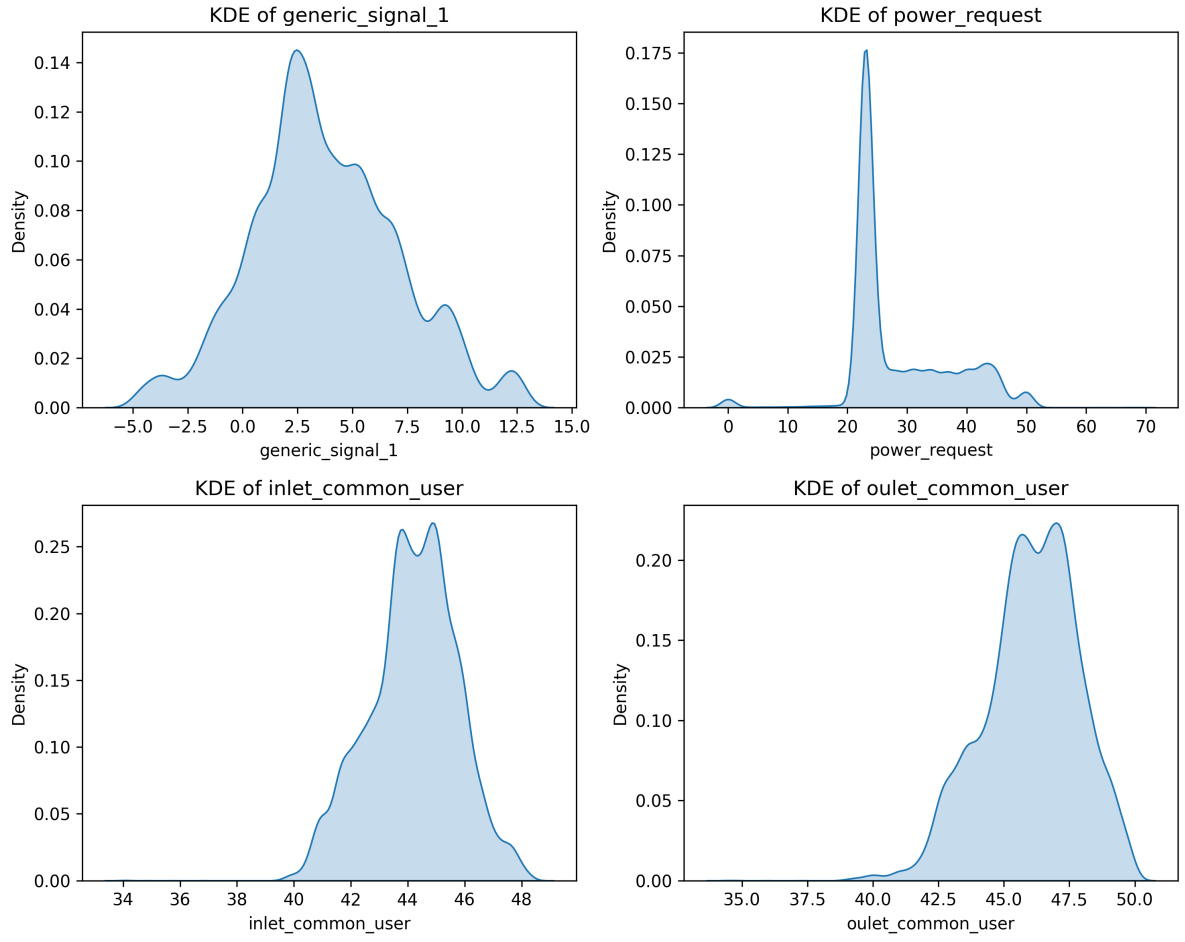

Figure 3.1: KDE of the input features for the subset *active_circuite* = 1.0.

The plots highlight a pronounced non-uniformity in the distribution of the data, with observations concentrated in narrow high-density regions and other areas sparsely populated. All features exhibit distinct peaks together with extended low-density tails, indicating a significant imbalance in sample density. This structure provides empirical motivation for the use of

a density-aware sampling strategy, as it suggests the need to reduce the dominance of highly concentrated regions while preserving information from less populated areas of the feature space.

3.5.1 Simple Random Sampling

To assess the suitability of standard sampling strategies for the objectives of this work, Simple Random Sampling (SRS) is considered as a baseline approach. The comparison between the original dataset and the SRS sample is illustrated in Figure 3.2.

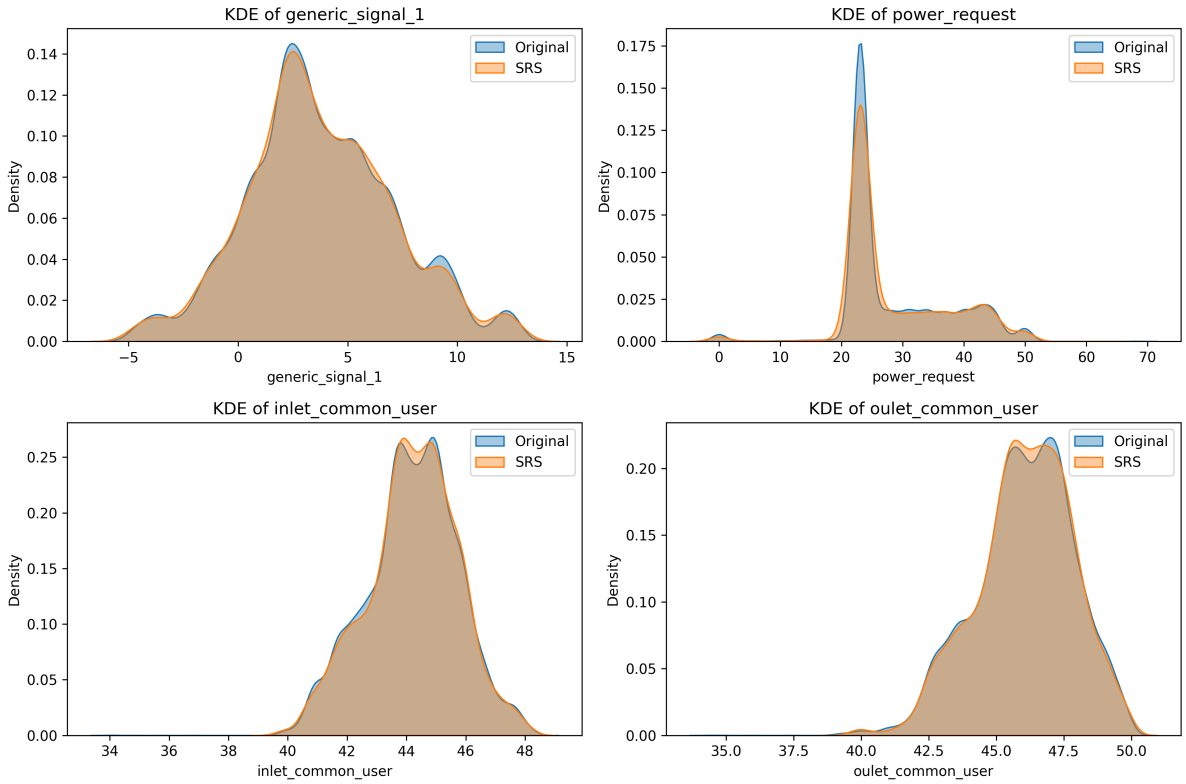

Figure 3.2: Comparison between the original dataset and the subset obtained via SRS.

The results show that the density distributions after random sampling closely resemble those of the original dataset, with the primary difference being the reduced number of observations. This behavior is expected, as SRS selects points uniformly at random and therefore preserves the empirical feature distributions. Consequently, dense regions remain dominant, while sparsely populated areas continue to be under-represented. This characteristic highlights a limitation of SRS in the present context. While it is effective for reducing dataset size, it does not alter the underlying density structure, and therefore does not address the objective of balancing the representation of dense and sparse regions.

Despite this limitation, SRS remains a **valuable tool** in large-scale scenarios. When the dataset is too large to allow the direct application of HDBSCAN, a preliminary SRS step can be used to extract a smaller and more representative subset. Thanks to its distribution-preserving property, this subset retains the main structural characteristics of the full dataset, enabling the application of clustering methods at a reduced computational cost. The cluster structure identified on the sampled subset can then be extended to the remaining observations by assigning each point to the closest cluster in the feature space. This two-step procedure provides a scalable approximation of the clustering that would be obtained on the full dataset, making the overall approach applicable to large-scale settings.

3.6 CBD Sampling

We now apply the CBD sampling procedure to the dataset.

As a preliminary step, the input features are standardized so that they are expressed on a comparable scale. This transformation is performed before splitting the data according to the categorical variable *active_circuite*, ensuring that the distances are computed within a consistent metric space. As a result, the clustering outcomes obtained across the different subsets are directly comparable. Clustering is then performed using the HDBSCAN algorithm. In particular, the parameters $min\_cluster\_size = 60$ and $min\_samples = 6$ are selected to balance the level of detail of the clustering structure and its robustness to noise. These values yield a sufficient number of clusters while avoiding excessive fragmentation and at the same time allow a moderate proportion of observations to be classified as noise. This configuration provides a meaningful representation of the density structure, which is essential to guide the subsequent sampling procedure. The CBD sampling strategy is then applied independently to each subset. In the case of the subset corresponding to $active\_circuite = 1.0$, which contains 21,249 observations, the clustering step identifies 12 clusters, labeled from 0 to 11, while approximately 15% of the points are classified as noise, labeled as -1. The effect of the sampling procedure on the feature distributions is illustrated in Figure 3.3.

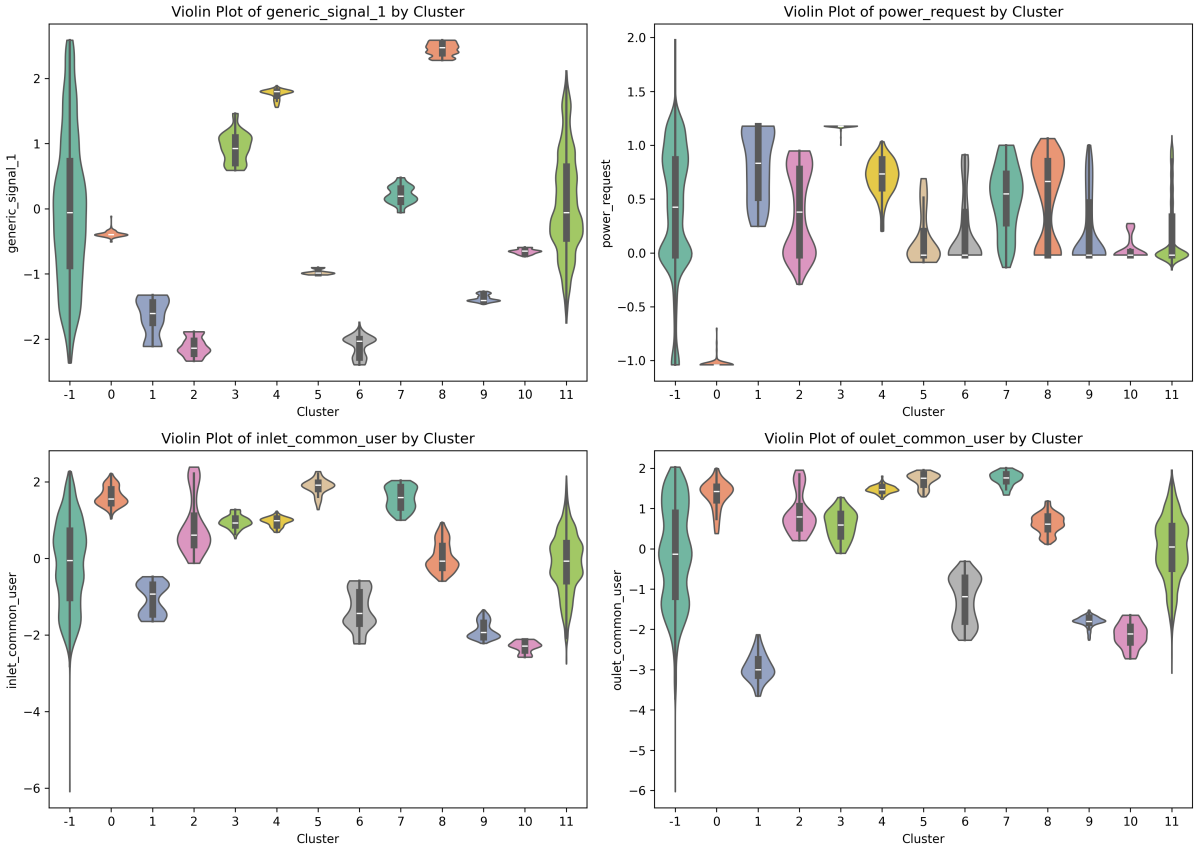

Figure 3.3: Distribution of the input features across the clusters identified by HDBSCAN.

Once the clustering structure has been obtained, different sampling strategies can be defined by exploiting the information provided by HDBSCAN. In particular, three alternative approaches are considered:

- **Core distance-based sampling**, where the selection mechanism is driven by the core distance of each observation, providing a direct measure of local density;
- **Probabilities-based sampling**, where the selection is guided by the membership probabilities, reflecting the strength of association between each point and its assigned cluster;
- **Cluster size-based sampling**, where the sampling rate is defined at the cluster level, with larger clusters being more aggressively reduced than smaller ones.

These strategies provide complementary mechanisms to control the representation of the data, allowing dense regions to be selectively reduced while preserving observations from less populated areas.
The choice among these approaches depends on the specific objective of the analysis, as each strategy emphasizes different aspects of the data structure, such as local density, cluster stability, or cluster size. In the following sections, their impact on the resulting feature distributions is examined.

3.6.1 Sampling Based on Core Distances

The first sampling strategy exploits the core distances estimated by HDBSCAN, which provide a point-wise indication of local density. In this setting, the parameter *min_samples* is fixed to 6, and therefore the core distance used corresponds to the 6-th nearest neighbor, as defined in Equation 2.4. Cluster densities are then estimated by aggregating the inverse core distances within each cluster, following the formulation introduced in Equation 2.5. In this experiment, a small regularization term $\varepsilon = 10^{-6}$ is added for numerical stability. The resulting density estimates are used to compute sampling weights according to Equation 2.7, with $\alpha = 1.5$. This choice amplifies the differences between dense and sparse clusters, leading to a stronger reduction of highly populated regions. The weights are subsequently normalized, as described in Equation 2.8, to obtain the sampling probabilities p_j, which are used to guide the selection of observations within each cluster.

The effect of the sampling procedure is illustrated in Figures 3.4 and 3.5, which compare the original dataset (21249 observations) with the sampled dataset (**4025 observations**).

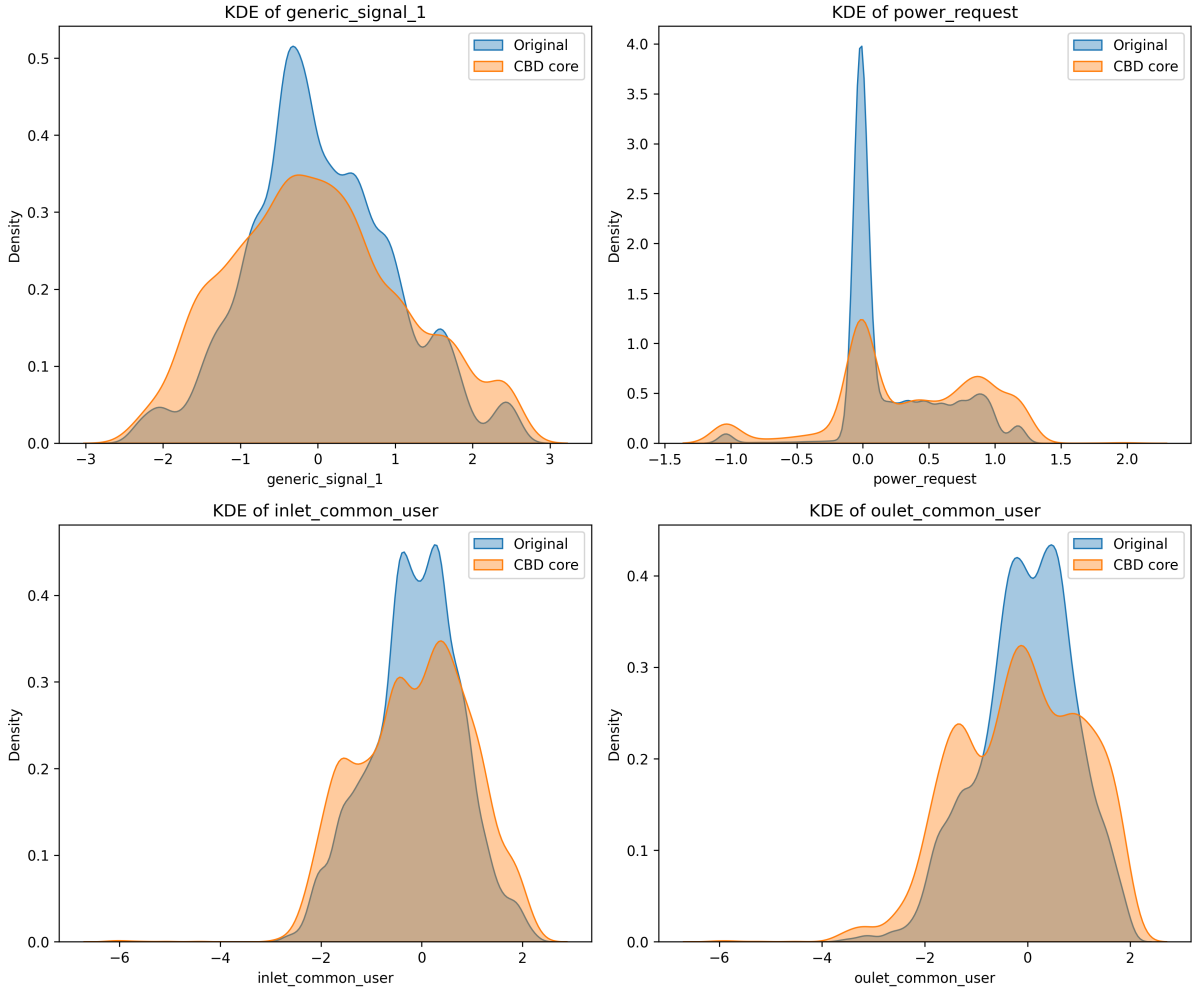

Figure 3.4: KDE comparison of the input features before and after core distance-based CBD sampling.

The kernel density estimates show that the sampling procedure reduces the prominence of peaks associated with high-density regions, while preserving the structure of lower-density areas. This results in a more balanced representation of the feature space, without altering the overall shape of the distributions.

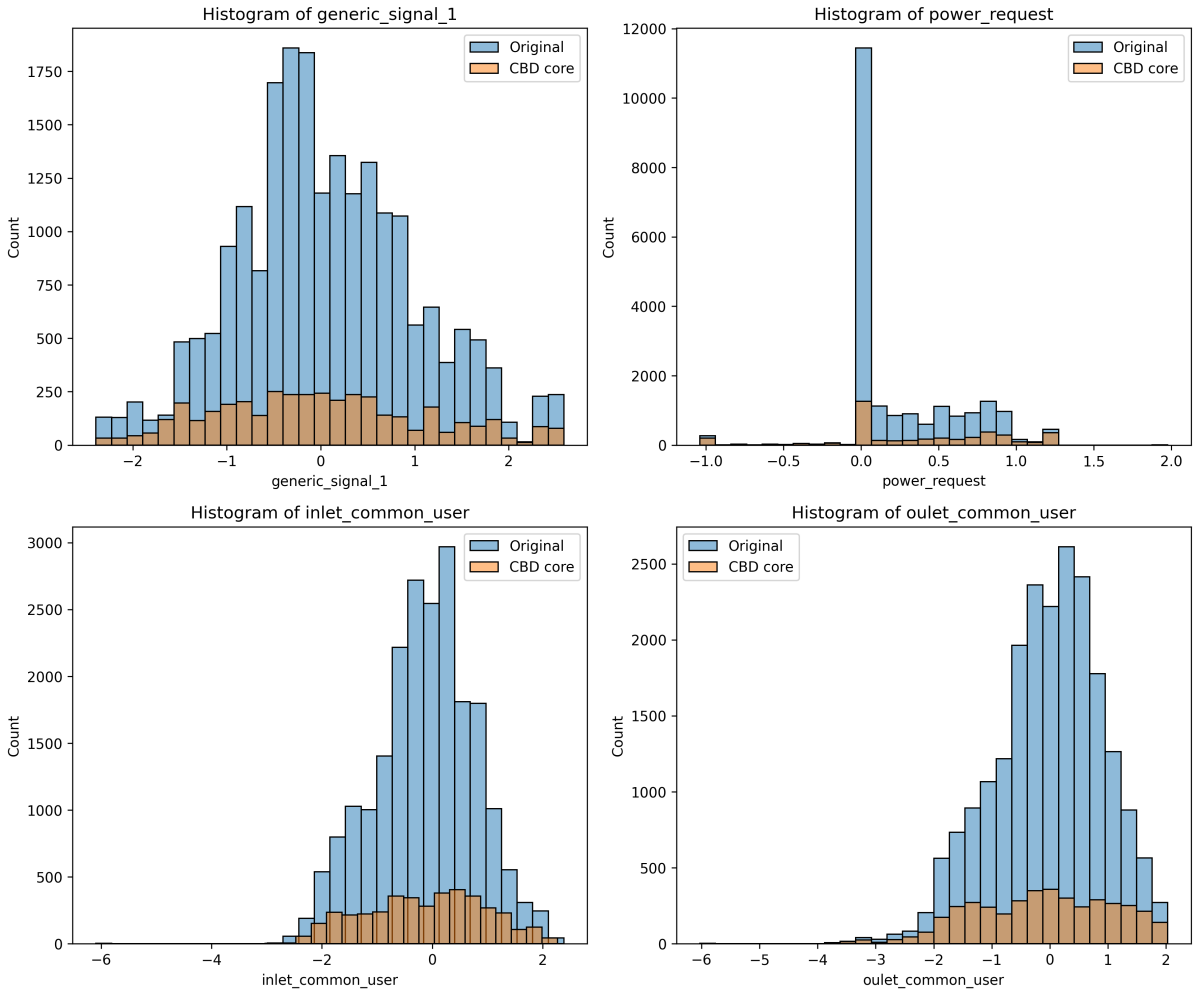

Figure 3.5: Histogram comparison of the input features before and after core distance-based CBD sampling.

The histograms provide a complementary perspective, showing how the number of observations is redistributed across the feature space. In particular, dense regions are visibly thinned, while less populated areas retain a meaningful number of observations, confirming the effectiveness of the density-aware sampling strategy.

3.6.2 Sampling Based on Probabilities

The second sampling strategy is based on the membership probabilities provided by HDBSCAN, which quantify the degree of confidence with which each observation is associated with its assigned cluster. Cluster densities are estimated by aggregating these probabilities at the cluster level, following the formulation introduced in Equation 2.6. This approach captures not only the concentration of points but also the stability of the clusters within the hierarchical structure. The resulting density estimates are then used to compute sampling weights according to Equation 2.7, with $\alpha = 1.5$. As in the previous strategy, this choice emphasizes differences between dense and less dense clusters, leading to a stronger reduction of highly populated regions. The weights are then normalized, as described in Equation 2.8, to obtain the sampling probabilities p_j, which guide the selection of observations within each cluster.

The effect of this sampling strategy is illustrated in Figures 3.6 and 3.7, which compare the original dataset (21249 observations) with the sampled dataset (**4613 observations**).

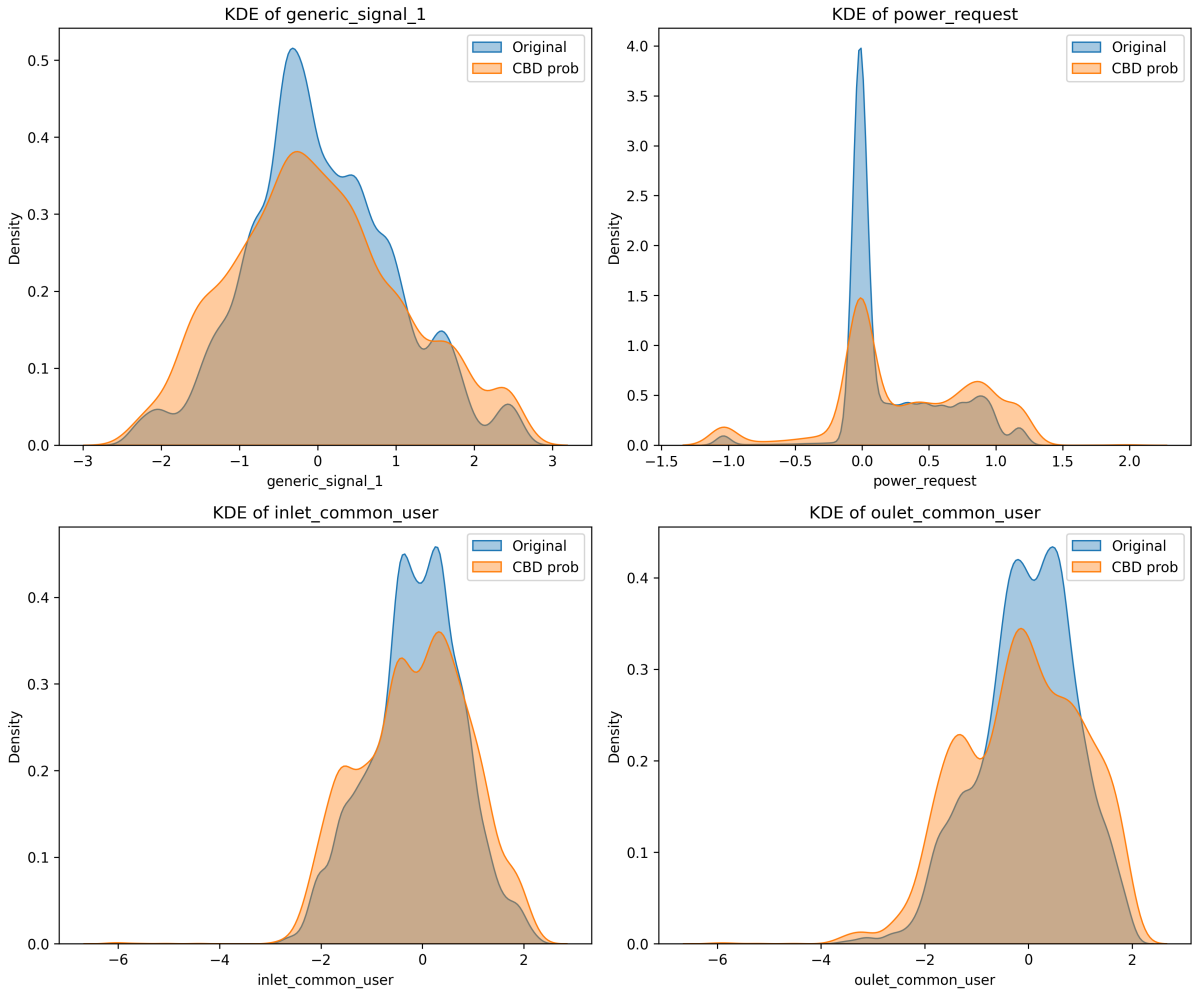

Figure 3.6: KDE comparison of the input features before and after probabilities-based CBD sampling.

The kernel density estimates show a reduction in the peaks associated with high-density regions while preserving the overall structure of the feature distributions. This behavior is consistent with the objective of the sampling procedure, which aims to balance the representation of dense and sparse areas.

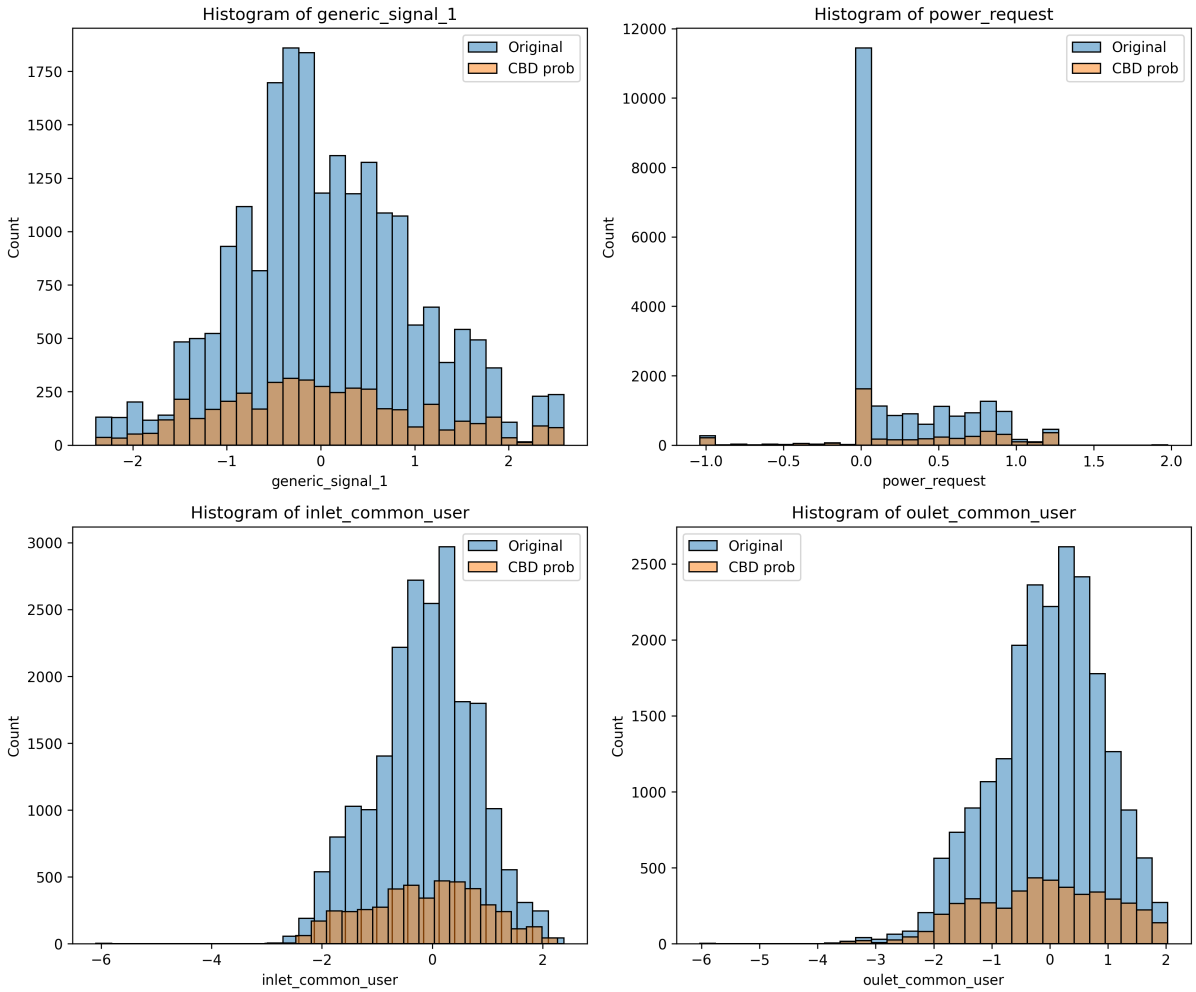

Figure 3.7: Histogram comparison of the input features before and after probability-based CBD sampling.

The histograms provide a complementary view, highlighting how the number of observations is redistributed across the feature space after sampling. In particular, the most frequent value ranges are reduced, resulting in a more balanced distribution.

Overall, the results are consistent with those obtained using the core distance-based strategy. Both approaches produce a comparable attenuation of high-density regions while preserving the general structure of the data, indicating that the CBD sampling framework is robust with respect to the choice of the density estimation criterion.

3.6.3 Sampling Based on Cluster Sizes

The third sampling strategy is based on the sizes of the clusters identified by HDBSCAN. In contrast to the previous approaches, this method does not rely on density-related quantities, but directly exploits the number of observations contained in each cluster. The sampling probabilities are computed according to Equation 2.9, where cluster sizes are transformed using a logarithmic function. This transformation reduces the impact of large disparities in cluster cardinality, preventing very large clusters from dominating the sampling process while still assigning them a higher probability than smaller ones. As a result, larger clusters are downsampled more aggressively, while smaller clusters retain a meaningful contribution. This leads to a balanced redistribution of observations without introducing excessively strong differences across clusters.

The effect of this strategy is illustrated in Figures 3.8 and 3.9, which compare the original dataset (21249 observations) with the sampled dataset (**5951 observations**).

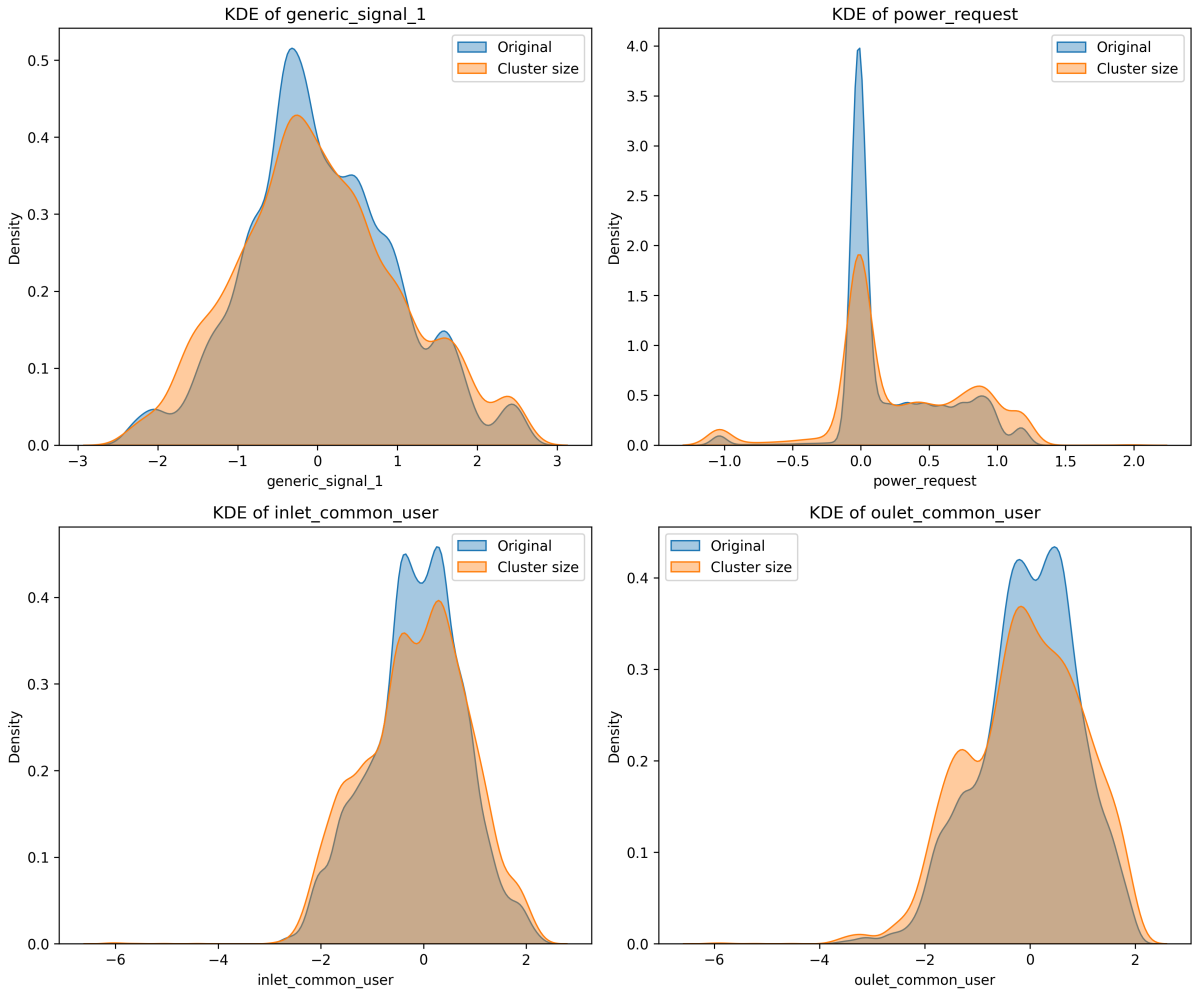

Figure 3.8: KDE comparison of the input features before and after cluster size-based CBD sampling.

The kernel density estimates show a reduction in the prominence of high-density regions, although less pronounced compared to the density-based strategies. The overall structure of the feature distributions is preserved, reflecting the fact that this approach is driven by cluster cardinality rather than local density.

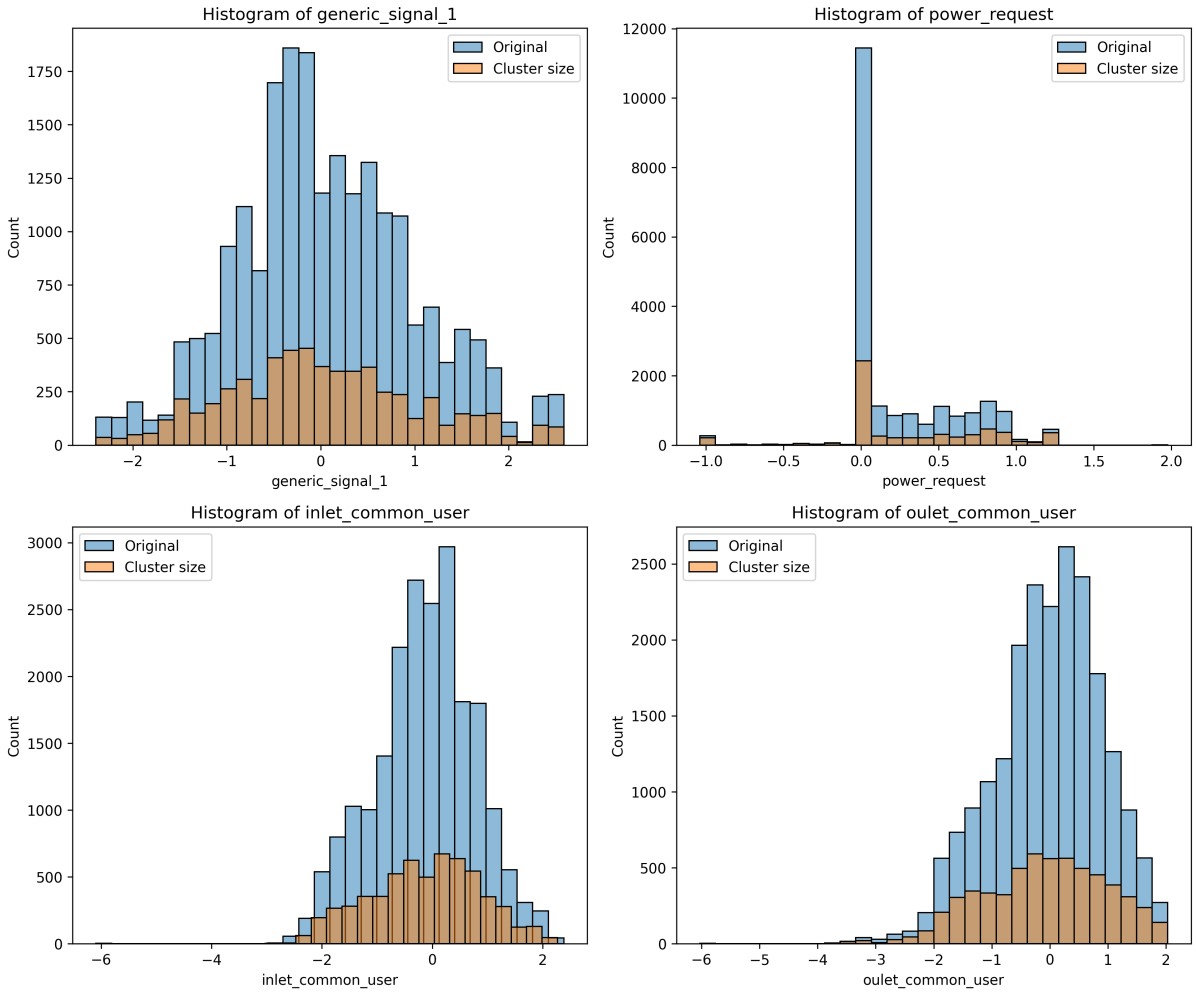

Figure 3.9: Histogram comparison of the input features before and after cluster size-based CBD sampling.

The histograms confirm this behavior, showing a moderate reduction in the number of observations in the most populated regions of the feature space. Compared to the previous strategies, the redistribution effect is less aggressive, resulting in a more conservative adjustment of the original data distribution.

Overall, this strategy provides a simpler alternative to density-based approaches, offering a compromise between computational simplicity and effective reduction of highly populated clusters.

3.6.4 Final Sampled Dataset

After applying the CBD sampling procedure independently to each subset, the resulting datasets are combined to reconstruct the final sampled dataset. This dataset represents the union of the samples obtained for the different levels of *active_circuite*. The final dataset contains **17972 observations**, compared to 76013 in the original data. This reduction is achieved through the probability-based CBD sampling strategy, using the same clustering configuration described previously.

The results for the individual subsets are summarized as follows:

- *active_circuite* = 0.0: 3 clusters, **8937** sampled observations out of 32577;
- *active_circuite* = 1.0: 12 clusters, **4613** sampled observations out of 21249;
- *active_circuite* = 2.0: 10 clusters, **4422** sampled observations out of 22187.

The distributions of the input features before and after sampling are shown in Figures 3.10 and 3.11.

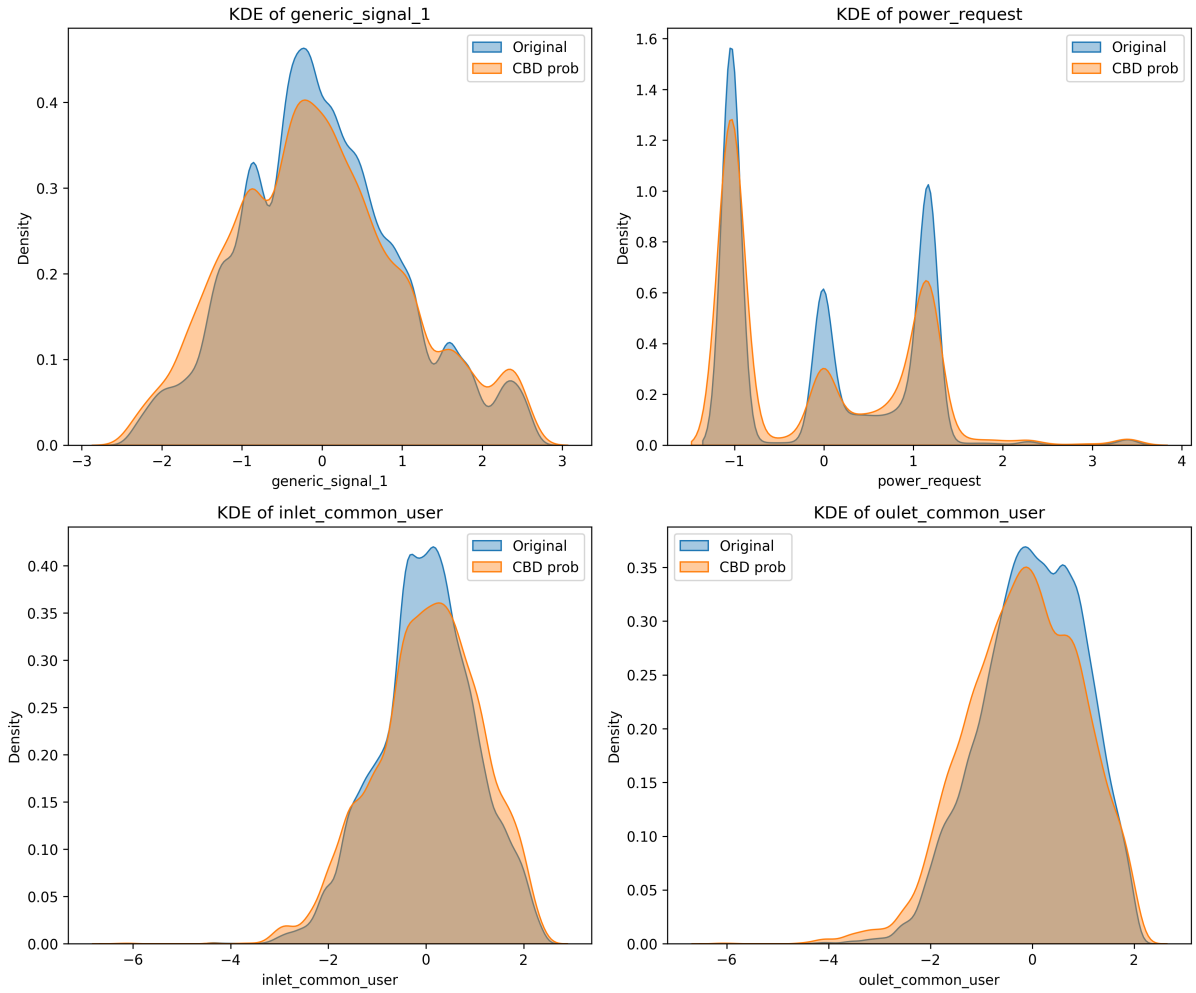

Figure 3.10: KDE comparison of the input features before and after probability-based CBD sampling to the full dataset.

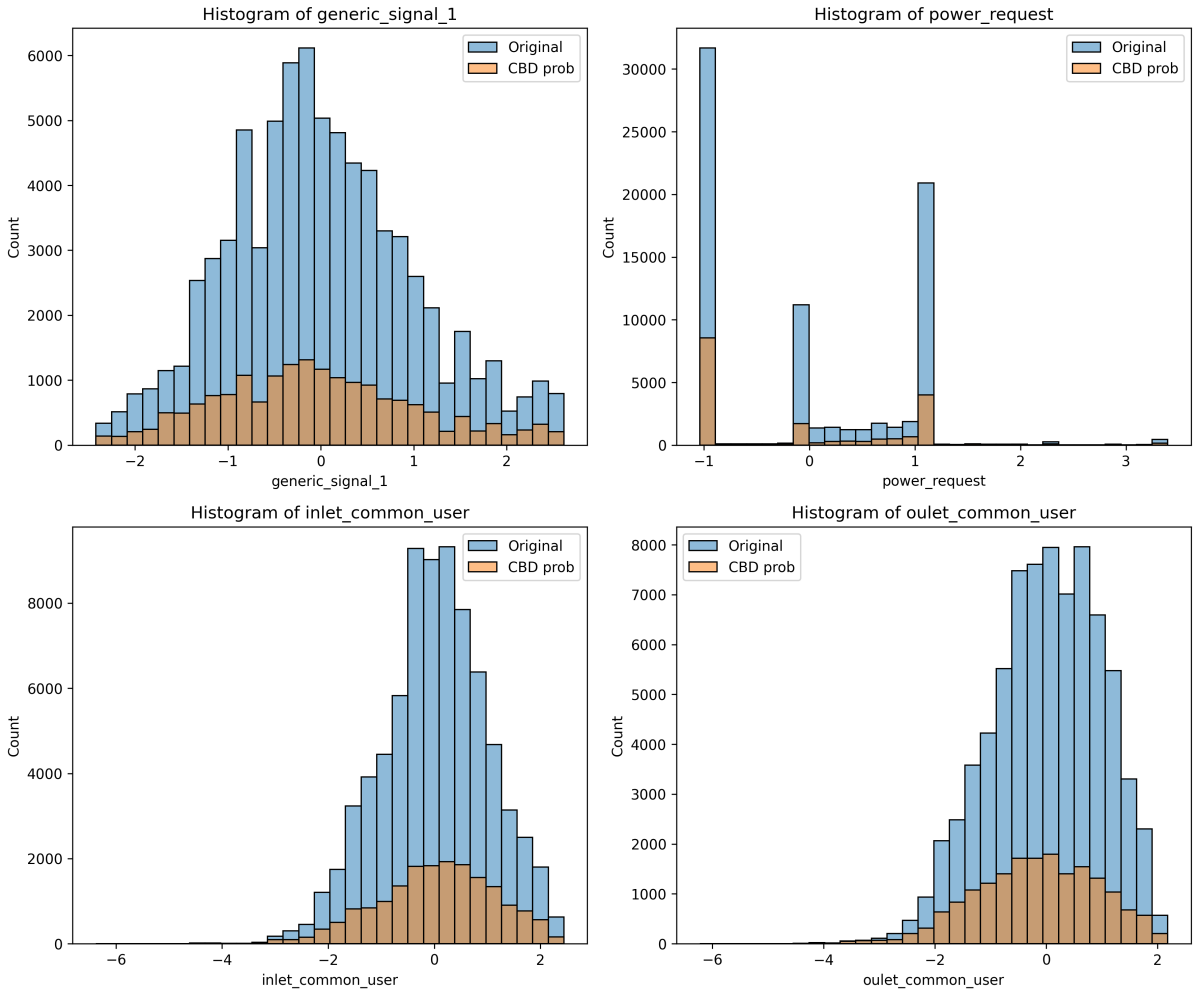

Figure 3.11: Histogram comparison of the input features before and after probability-based CBD sampling to the full dataset.

The results confirm that the sampling procedure effectively reduces the dominance of highly populated regions while preserving the overall structure of the feature space. In particular, the density peaks are attenuated, and the distribution of observations becomes more balanced across different value ranges. At the same time, the general shape of the original distributions is retained, indicating that the essential characteristics of the data are preserved.

3.7 Cluster Metadata

After the clustering step, it is useful to derive cluster-level metadata to facilitate the inspection and comparison of the resulting groups. In this context, metadata refers to summary statistics that describe the main characteristics of each cluster, enabling a concise representation of the clustering structure. The use of cluster metadata serves several purposes. First, it allows a direct comparison of clusters within the same subset, highlighting differences in density and internal composition. Second, it provides a compact representation of the clustering outcome, which is particularly valuable when dealing with large datasets. Finally, it enables the evaluation of how the clustering structure evolves under different processing steps or configurations.

In this analysis, cluster metadata are computed using the membership probabilities provided by HDBSCAN. These values quantify the strength of association between each observation and its assigned cluster, and therefore reflect the local density structure captured by the algorithm. For each cluster, descriptive statistics are computed by aggregating these probabilities, including the mean, median, interquartile range, and extreme values.

The resulting metadata for the subset corresponding to *active_circuite* = 1.0 are shown in Figure 3.12, providing an overview of the characteristics of the identified clusters.

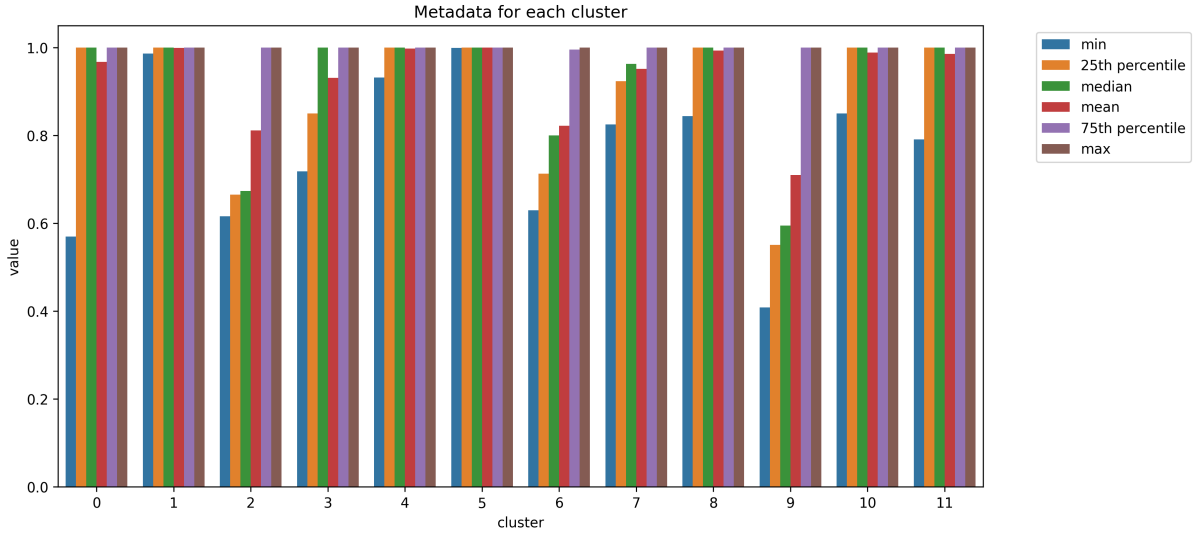

Figure 3.12: Cluster-level metadata based on membership probabilities.

3.8 Regression Performance and Sampling Evaluation

In this section, the impact of the proposed sampling strategies is evaluated on a downstream regression task. A CatBoostRegressor model is trained to predict the following eight numerical target variables:

- *evaporating_pressure*
- *evaporating_pressure_conv*
- *discharge_temperature*
- *condensing_pressure*
- *condensing_pressure_conv*
- *super_heating*
- *eev_perc*
- *speed_inverter*

Model performance is evaluated using the Root Mean Squared Error (RMSE).

The analysis compares four different datasets:

1. the original full dataset,
2. the dataset obtained using CBD sampling based on probabilities,
3. the dataset obtained using CBD sampling based on cluster sizes,
4. the dataset obtained via Simple Random Sampling (SRS).

The sampling strategy based on core distances is not included in the final comparison. Preliminary experiments showed that this approach consistently resulted in higher RMSE values, leading to a deterioration of predictive performance. For this reason, the analysis focuses on the most effective strategies, namely probability-based and cluster size-based sampling, together with the SRS baseline.

3.8.1 Experimental Setup

For each of the eight target variables, a separate CatBoostRegressor model is trained.

The dataset is split into training, validation, and test sets using an 80% / 10% / 10% partition. The test set is extracted from the original dataset and kept fixed across all experiments. Each model is then trained and validated on the corresponding dataset (either full or sampled), while evaluation is consistently performed on the same test set. This ensures a fair comparison of the RMSE across different sampling strategies.

The evaluation is conducted under two distinct settings, designed to assess the robustness of the sampling procedure under different input configurations.
In the first setting, the model is trained using only the selected input features, allowing us to isolate the effect of the sampling procedure on the feature space.
In the second setting, the input space is extended by including the target variables, excluding the one currently being predicted, resulting in a richer representation of the data and allowing for a more realistic evaluation.

Model Features Only

In the first setting, the model is trained using only the five input features previously introduced:

- *generic_signal_1*, numerical variable;
- *power_request*, numerical variable;
- *inlet_common_user*, numerical variable;
- *oulet_common_user*, numerical variable;
- *active_circuite*, categorical variable.

Each of the eight target variables is predicted independently using the same set of input features. Since the feature space used for clustering is identical across all prediction tasks, the same HDBSCAN configuration is adopted, ensuring consistency in the clustering structure and enabling a fair comparison between sampling strategies.

For each target variable, the model is trained on the four datasets introduced previously, and the corresponding RMSE values are computed. The results are summarized in Table 3.1.

Table 3.1: RMSE comparison using only model features

Target Variable	RMSE Full Dataset	RMSE CBD Prob	RMSE Cluster Size	RMSE SRS
evaporating_pressure	**0.1454**	0.1621	0.1589	0.1532
evaporating_pressure_conv	**1.1796**	1.3093	1.2848	1.2449
discharge_temperature	**1.2421**	1.3697	1.3476	1.3233
condensing_pressure	**0.1085**	0.1136	0.1091	0.1161
condensing_pressure_conv	**0.3164**	0.3328	0.3225	0.3281
super_heating	**1.2990**	1.5541	1.3734	1.3440
eev_perc	**3.9687**	4.5921	4.4668	4.1767
speed_inverter	**2.4154**	3.9292	3.9647	2.9737

The results show that the best performance is consistently achieved using the entire dataset. Nevertheless, the sampled datasets, which are approximately **four times smaller**, exhibit only a moderate increase in RMSE for most target variables. In this setting, the SRS approach generally provides results that are closer to those of the full dataset. This can be attributed to the relatively limited size of the dataset and the reduced dimensionality of the feature space. Under these conditions, preserving the original data distribution is sufficient to maintain good predictive performance. By contrast, the CBD sampling strategies introduce controlled modifications to the data distribution, which may lead to a slight degradation in performance in this scenario. However, these approaches are expected to become more effective in larger-scale settings, where reducing redundancy in dense regions can improve both computational efficiency and the quality of the learned representations.
Overall, the results indicate that a substantial reduction in dataset size can be achieved while maintaining a comparable level of predictive performance.

Model and Target Features

In the second setting, the input space is extended by including both the model features and the target variables, excluding the one currently being predicted. This results in a richer and task-dependent feature space, which varies across the eight prediction problems. Due to this variability, the HDBSCAN parameters are selected separately for each target variable, based on unsupervised criteria such as the number of clusters, the proportion of noise points, and the interpretability of the resulting clustering structure.

As in the previous setting, for each target variable a CatBoostRegressor model is trained on the four datasets, and the corresponding RMSE values are computed.

Table 3.2: RMSE comparison using both model and target features

Target Variable	RMSE Full Dataset	RMSE CBD Prob	RMSE Cluster Size	RMSE SRS
evaporating_pressure	0.0379	0.0342	**0.0338**	0.0407
evaporating_pressure_conv	0.2582	0.2558	**0.2415**	0.2991
discharge_temperature	**1.8641**	2.3352	2.1767	2.1529
condensing_pressure	0.0725	0.0655	**0.0640**	0.0752
condensing_pressure_conv	0.2698	0.2524	**0.2371**	0.3150
super_heating	**0.9534**	1.2158	1.1552	1.0844
eev_perc	**1.8052**	2.3133	2.1906	2.0719
speed_inverter	**1.7498**	2.6307	2.1870	2.1311

Table 3.2 reports the RMSE values obtained in this setting. Compared to the previous configuration, the overall errors are significantly lower, confirming that the inclusion of additional variables enhances the predictive capability of the model. Despite the sampled datasets being approximately **six times smaller** than the full dataset, the CBD sampling strategies, particularly the cluster size-based approach, often achieve comparable or improved performance. For several target variables, such as *evaporating_pressure*, *condensing_pressure*, and their corresponding converted versions, the sampled datasets yield lower RMSE values. This suggests that, in a higher-dimensional feature space, reducing redundancy in dense regions can improve the quality of the training data and lead to better generalization. For other targets, including *discharge_temperature*, *super_heating*, *eev_perc*, and *speed_inverter*, the full dataset still provides the best performance. This indicates that, for these variables, the availability of a larger number of observations remains important to capture more complex relationships. The SRS approach generally performs worse than the CBD strategies in this setting, as it does not exploit the structure of the data and may discard informative observations, particularly in less represented regions of the feature space. This limitation becomes more evident as the dimensionality of the input space increases.
Overall, these results highlight a trade-off between dataset size and predictive performance. While the full dataset provides strong baseline results, the CBD sampling strategies are able to achieve comparable, and in some cases better, performance using significantly fewer observations, confirming their effectiveness in reducing redundancy while preserving informative patterns.

3.8.2 Pipeline Execution Time Analysis

In addition to predictive performance, the proposed approach is evaluated in terms of computational efficiency. In particular, we consider the end-to-end pipeline execution time, which includes data pre-processing, the optional sampling step, and model training. All experiments are conducted on a single machine to ensure a consistent comparison. The model is trained on approximately one year of data, while the pre-processing stage, comprising data cleaning and feature preparation, requires about one hour and remains constant across all runs.

Table 3.3 reports the total execution times observed over multiple runs for both the full dataset and the sampled dataset.

Table 3.3: Pipeline execution time comparison (hours)

Run	Full Dataset	Sampled Dataset
1	9.2	4.3
2	9.5	4.6
3	8.9	4.4
4	9.3	4.7
5	9.1	4.5
6	9.4	4.6
7	9.0	4.4
8	9.3	4.5
9	9.2	4.6
10	9.1	4.3
Average	**9.2**	**4.5**

The results show that the full pipeline execution requires approximately 9–9.5 hours, whereas the sampled dataset reduces this time to about 4.3–4.7 hours.
On average, the execution time decreases from 9.2 to 4.5 hours, corresponding to a reduction of over 50%. The low variability across runs indicates a stable execution process in both settings. This reduction reflects the decrease in dataset size achieved through the sampling procedure and highlights the strong dependence of computational cost on data volume.

From a long-term perspective, this advantage becomes increasingly relevant. As new data are continuously collected, the size of the full dataset grows steadily, while the sampled dataset expands at a slower rate due to the density-aware selection mechanism.
As a result, the gap in computational cost between the two approaches is expected to widen over time, making the proposed sampling strategy particularly suitable for large-scale and long-term applications.

3.8.3 Discussion of Results

The results presented in the previous sections highlight the trade-offs introduced by the proposed sampling strategies in terms of predictive performance and computational efficiency. The primary objective of this work was to construct a reduced dataset capable of preserving generalization performance while maintaining an accuracy level comparable to that of the full dataset. The experimental findings show that this objective is largely achieved, as the sampled datasets consistently exhibit performance close to the full dataset despite a substantial reduction in size.

A key aspect emerging from the analysis is the role of feature space complexity. When only model features are used, the benefits of density-aware sampling are limited. In this scenario, the relatively low dimensionality allows even simple random sampling to preserve the data distribution effectively. In contrast, when the input space is extended with additional variables, the structure of the data becomes more complex. Under these conditions, the CBD-based strategies, particularly the cluster size-based approach, demonstrate improved effectiveness, in some cases outperforming the full dataset. This suggests that reducing redundancy in dense regions can enhance the quality of the training data and improve generalization in higher-dimensional settings. From a quantitative perspective, the results indicate that a significant reduction in dataset size can be achieved with only a limited impact on RMSE, and in some cases even a slight improvement. This confirms that the proposed method is able to retain the most informative observations while discarding redundant data.

Beyond predictive performance, the analysis of pipeline execution time provides further insight into the practical advantages of the approach. The reduction in dataset size translates into a substantial decrease in computational cost, with execution time reduced by more than 50% and stable behavior across runs. This advantage becomes increasingly relevant in large-scale and long-term scenarios. As data volume grows over time, the full dataset leads to progressively higher computational requirements, whereas the sampling procedure limits this growth by controlling redundancy in dense regions. As a result, the proposed approach enables a more scalable pipeline.

The proposed method achieves a favorable trade-off between accuracy and efficiency. Although the full dataset may still provide the best performance in certain cases, the observed loss in accuracy remains limited and is outweighed by significant improvements in computational efficiency. These results confirm that CBD sampling represents an effective approach for reducing dataset size in regression tasks, especially in large-scale and high-dimensional scenarios.

4

Adaptive Update of the Sampling Procedure

A key objective of this work is not only to define an effective sampling strategy for static datasets, but also to design a procedure that can be updated over time as new data become available. In many real-world applications, the underlying processes are inherently dynamic, with data continuously generated and evolving. As a result, a static representation of the dataset may quickly become outdated and fail to capture emerging patterns. Motivated by this consideration, we aim to develop an incremental and adaptive methodology capable of updating its internal structure as new observations are introduced. The goal is to maintain a representation of the data that remains both accurate and informative over time.

This chapter presents an extension of the proposed approach that enables the assignment of new observations to existing clusters, as well as the dynamic formation of new clusters in previously unstructured regions of the feature space. In this way, the method adapts to the growth of the dataset while preserving a coherent and consistent representation of its structure.

4.1 Assignment of New Observations

After performing the initial clustering with HDBSCAN and applying the CBD sampling procedure, the dataset is organized into a set of clusters and a set of observations classified as noise. In dynamic scenarios, new data points are continuously generated and must be integrated into this existing structure without recomputing the clustering from scratch. The objective of this section is therefore to define a consistent and efficient rule for assigning new observations either to one of the existing clusters or to the noise component.

The proposed approach is based on the local density structure of the clusters. Each cluster is characterized through the distribution of distances between its points and their nearest neighbors, which provides a compact representation of its internal density and spread. For each cluster C_j, we compute a set of intra-cluster distance values. In particular, for each point $x_i \in C_j$, we consider the average distance to its k nearest neighbors within the same cluster, where k corresponds to the parameter *min_samples*:

$$d_i^{(j)} = \frac{1}{k} \sum_{x \in \mathcal{N}_k(x_i)} \|x_i - x\| \tag{4.1}$$

where $\mathcal{N}_k(x_i)$ denotes the set of the k nearest neighbors of x_i in C_j, and $\|\cdot\|$ represents the

Euclidean distance in the scaled feature space. The collection $\{d_i^{(j)}\}_{x_i \in C_j}$ defines a distribution that captures the local density characteristics of the cluster C_j. In practice, this distribution is summarized through histograms, providing an interpretable description of cluster compactness.

Given a new observation x_{new}, the same preprocessing steps are first applied in order to ensure consistency with the feature space used during the initial clustering phase. In particular, the same scaling transformation is adopted, so that all distance computations remain comparable. Once projected into the same feature space, we compute a local distance-based quantity that reflects the position of the new observation with respect to the existing data. Specifically, we evaluate its average distance to its k nearest neighbors:

$$d_{\text{new}} = \frac{1}{k} \sum_{x \in \mathcal{N}_k(x_{\text{new}})} \|x_{\text{new}} - x\| \tag{4.2}$$

This quantity can be interpreted as a proxy for the local density around x_{new}: smaller values indicate that the point lies in a dense region, while larger values suggest that it is located in a sparse or previously unexplored area of the feature space. The computed value d_{new} is then compared with the intra-cluster distance distributions previously estimated for each cluster. This comparison allows us to assess whether the local density characteristics of the new observation are compatible with those of an existing cluster. The assignment is performed through a threshold-based criterion. For each cluster C_j, we consider a reference percentile (e.g., the 95th percentile) of its distance distribution. If d_{new} falls below this threshold, the observation is considered consistent with the density structure of that cluster. When multiple clusters satisfy this condition, the assignment is refined by selecting the cluster that minimizes the relative distance, thus associating the point with the most compatible local structure. Conversely, if no cluster satisfies the threshold criterion, the observation is classified as noise, indicating that its local characteristics do not align with any of the existing clusters.

This procedure provides a principled and interpretable mechanism for integrating new observations into the existing cluster–noise framework. By relying on local density comparisons rather than global distance measures, the method preserves the structural properties learned during the initial clustering phase and adapts naturally to the geometry of the feature space.

To illustrate the behavior of the proposed method, two representative examples are reported for the subset corresponding to *active_circuite* = 1.0.

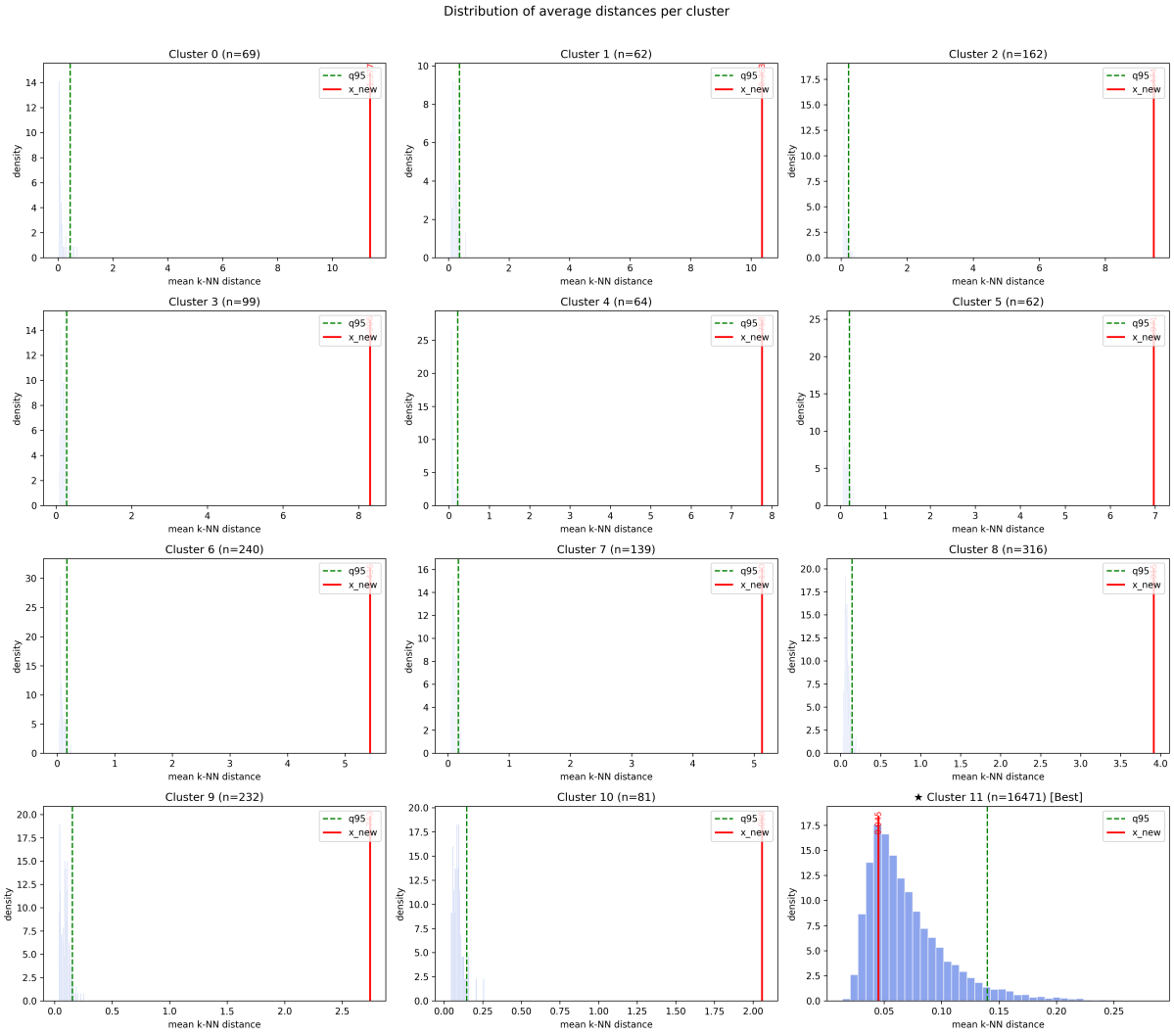

Figure 4.1: Assignment of a point consistent with the existing cluster structure.

As shown in Figure 4.1, a point whose local distance characteristics are coherent with the cluster structure is correctly assigned to its corresponding cluster. In this example, the observation belongs to the original dataset, and the assignment procedure reproduces the initial clustering result obtained during the HDBSCAN phase.

This behavior highlights the consistency of the proposed method, as points that conform to the learned density structure are reliably associated with the same cluster. Moreover, it confirms that the assignment rule preserves the geometric and density-based properties of the feature space, ensuring stability between the initial clustering and the subsequent integration of observations.

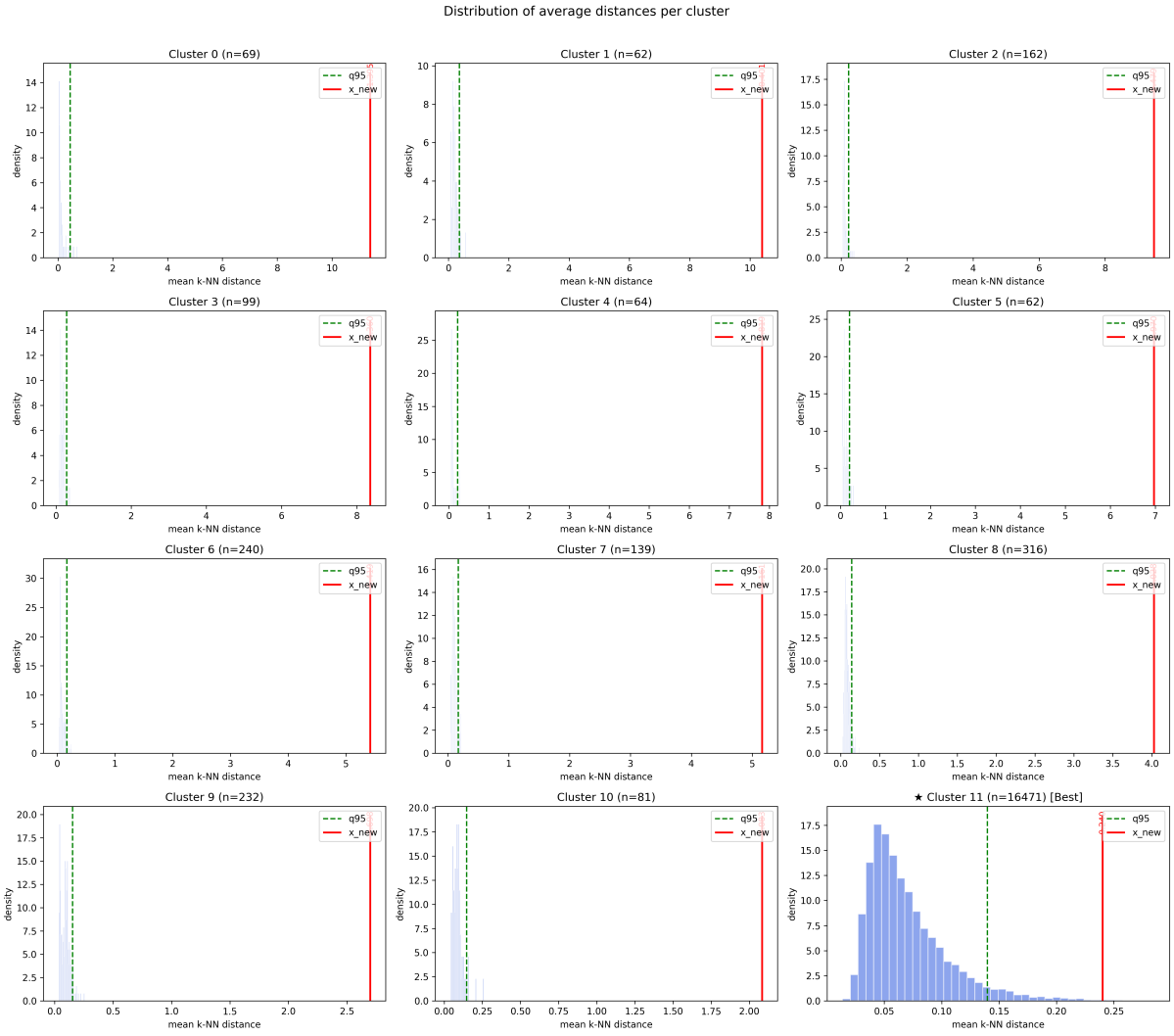

Figure 4.2: Assignment of a perturbed point classified as noise.

Figure 4.2 shows the evaluation of a perturbed version of the same observation. The introduction of noise alters its local distance structure, making it inconsistent with all existing clusters. As a result, the procedure correctly classifies the point as noise. This behavior demonstrates the sensitivity of the method to variations in local density, as even relatively small perturbations can lead to a loss of compatibility with the cluster structure. More importantly, it confirms that the assignment rule does not force observations into clusters when their characteristics are not aligned with the learned patterns, thereby preventing incorrect associations.

Taken together, these examples highlight the robustness of the proposed method, which is able to distinguish between observations that conform to the learned structure and those that deviate from it.

4.2 Formation of New Clusters from Noise

In dynamic scenarios, assigning new observations to existing clusters is not sufficient to fully capture the evolution of the data. As new observations are introduced over time, previously unseen patterns may emerge, making the initial clustering structure no longer adequate to describe the dataset. For this reason, the proposed framework is designed to be adaptive, allowing not only the integration of new observations, but also the identification of new clusters. Observations that are not compatible with any existing cluster are classified as noise. While isolated noise points may correspond to outliers, a persistent increase in the number of such observations can indicate the presence of previously unobserved structures in the feature space.

The key idea is to monitor the evolution of the noise component and to trigger a re-clustering step when this component becomes sufficiently large. In this way, the method is able to detect emerging patterns without recomputing the clustering over the entire dataset.

Let η_0 denote the initial proportion of noise obtained after the first application of HDBSCAN. As new observations are progressively added and evaluated through the assignment procedure, we compute the updated noise proportion η_t. The formation of new clusters is triggered when the following condition is satisfied:

$$\eta_t \geq (1 + \alpha)\, \eta_0 \tag{4.3}$$

where $\alpha > 0$ is a parameter controlling the sensitivity of the procedure to the growth of the noise component (in our experiments, $\alpha = 0.5$).

When the condition in Equation 4.3 is met, HDBSCAN is applied exclusively to the subset of observations currently classified as noise. This step aims to identify potential structures that were not present, or not detectable, during the initial clustering phase. If new clusters are identified, they are incorporated into the existing clustering structure and treated as standard clusters in subsequent steps. In particular, their internal distance distributions are computed, allowing them to participate in the assignment of future observations.

This mechanism enables the clustering structure to evolve in a consistent and data-driven manner. Rather than treating noise as a static by-product, it is leveraged as a source of information to detect emerging patterns in the data. As a result, the proposed approach operates within a continuous and adaptive framework, where both the assignment of new observations and the formation of new clusters contribute to maintaining an up-to-date and representative description of the feature space as the dataset grows.

4.3 Cluster Evolution under Seasonal Distribution Shift

The proposed dynamic procedure is evaluated on the real dataset introduced in Chapter 3, with the objective of analyzing how the cluster–noise structure evolves as new data are progressively incorporated over time.

The initial clustering and sampling are performed on data corresponding to January 2025, which serves as the baseline dataset. This dataset defines the reference cluster structure, together with the associated noise component.

New observations are then introduced from July 2025, following a temporal expansion scenario:

- first, one week of data (first week of July),
- then, two weeks of data (second and third week),
- finally, the entire month of July.

At each step, the new observations are processed using the assignment procedure described in the previous sections. Each point is either assigned to one of the existing clusters or classified as noise, depending on its compatibility with the learned density structure.

It is important to note that the data used to build the initial clustering correspond to a winter month (January), while the new data come from a summer period (July). Due to the significant seasonal and environmental differences between these two time frames, it is reasonable to expect that a large portion of the newly introduced observations will not be consistent with the original cluster structure. As a consequence, a substantial fraction of the new points is expected to be classified as noise.

First Week of July

The first week of July 2025 consists of **6690 new observations**. After integrating these points into the cluster–noise structure obtained from January 2025, the assignment procedure yields the following results:

- **6499 points** are classified as **noise**,
- **191 points** are assigned to **cluster 0**.

As expected, the majority of the newly introduced observations are classified as noise. This is consistent with the significant seasonal differences between January and July, which lead to a noticeable shift in the underlying data distribution. The points assigned to cluster 0 represent only a small fraction of the new observations. This cluster is the largest among the original ones, containing 31797 points, and therefore it is more likely to capture a limited number of new observations that still exhibit characteristics compatible with its structure.

To further validate the coherence of the assignment, we compare the distributions of the input features for:

- the original points belonging to cluster 0 (January),
- the new points from July assigned to the same cluster.

This comparison is performed using violin plots, as shown in Figure 4.3. The results indicate that, despite the temporal and seasonal differences, the newly assigned observations remain consistent with the distribution of the original cluster, thus supporting the correctness of the assignment procedure.

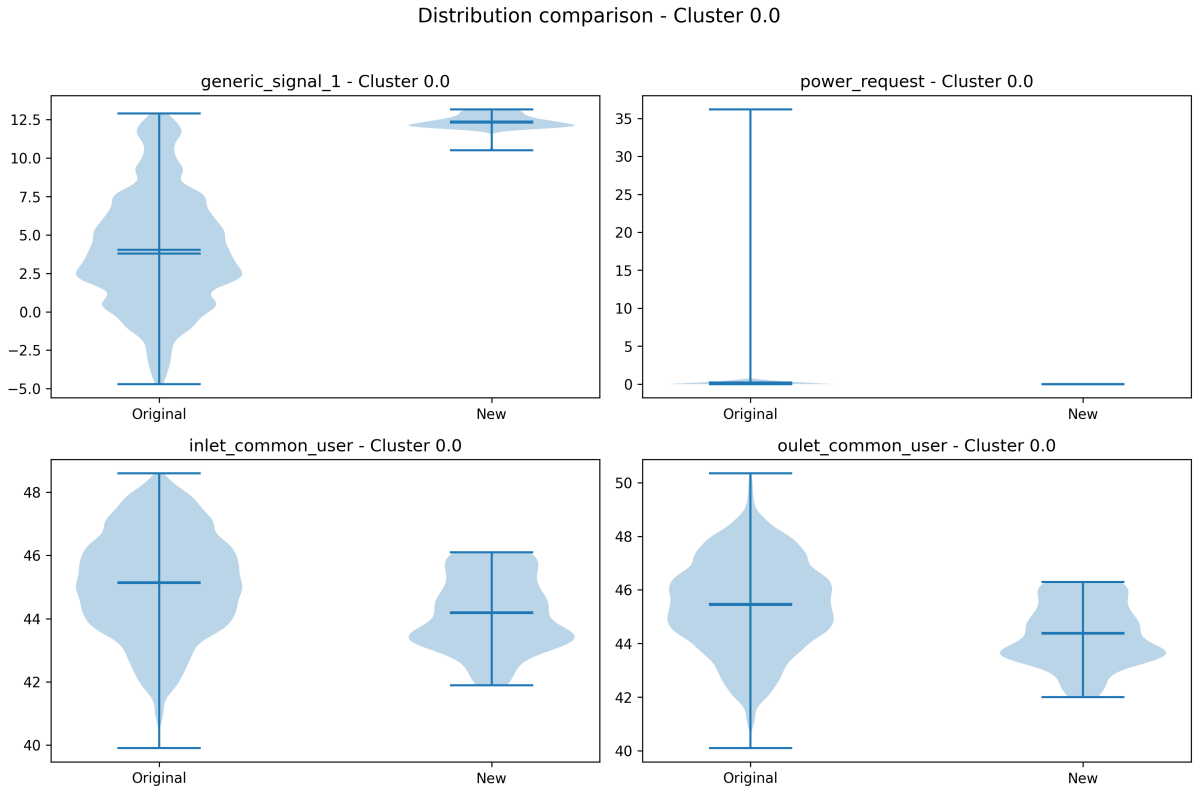

Figure 4.3: Comparison between original cluster 0 points (January) and newly assigned points (first week of July)

Second and Third Week of July

The second and third weeks of July 2025 introduce an additional set of observations, further extending the dataset beyond the first incremental step. After integrating these **new 8903 points** into the existing cluster–noise structure, the assignment procedure produces the following results:

- **8712 points** are classified as **noise**,
- **191** points are assigned to **cluster 0**.

As more data are incorporated, the proportion of points classified as noise remains dominant, confirming the presence of a persistent distributional shift between January and July. At the same time, the number of points assigned to cluster 0 remains comparable to that observed in the previous step, indicating that the same subset of observations continues to exhibit characteristics compatible with this cluster. This suggests that the cluster captures a stable region of the feature space that is only marginally affected by the newly introduced data.

To assess the consistency of these assignments, we again compare the feature distributions of the original cluster points with those of the newly assigned observations using violin plots. The comparison, reported in Figure 4.4, shows that the newly integrated points maintain a distributional structure similar to that of the original cluster, supporting the validity of the assignment mechanism.

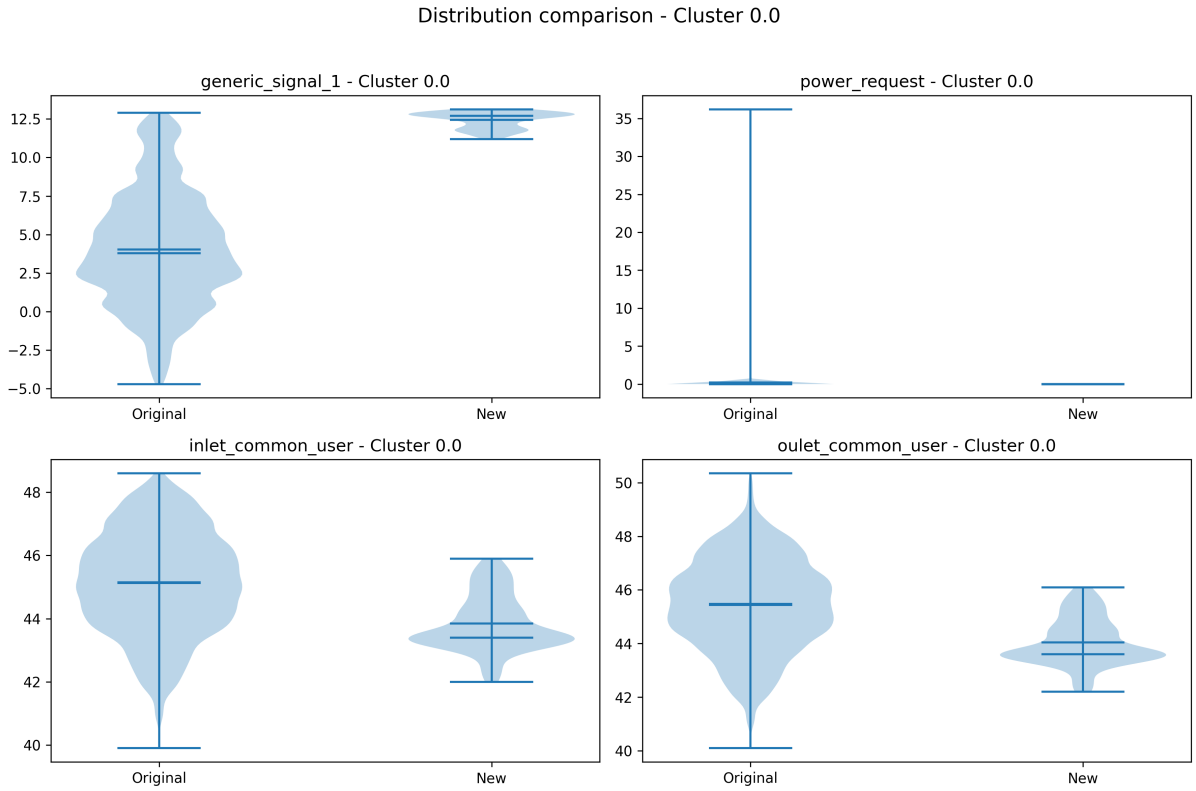

Figure 4.4: Comparison between original cluster 0 points (January) and newly assigned points (second and third week of July)

Full Month of July

Finally, the entire month of July 2025 is considered, leading to the largest incremental update of the dataset. After integrating all **17151 new observations**, the assignment procedure yields the following results:

- **16704 points** are classified as **noise**,
- **447 points** are assigned to **cluster 0**.

At this stage, the effect of the distributional shift becomes particularly evident. The large majority of the newly introduced points is classified as noise, accounting for approximately 97% of the new observations. This behavior is fully consistent with the significant differences between winter (January) and summer (July) data, and confirms the ability of the procedure to detect observations that deviate from the original cluster structure.

Regarding the points assigned to cluster 0, their total number is greater than the sum of those observed in the previous incremental steps (first week, second and third weeks). This indicates that additional points from the last week of July are also classified within this cluster, further supporting the interpretation of cluster 0 as a stable and persistent structure in the feature space.

The violin plots shown in Figure 4.5 provide additional insight into this behavior. Despite the increased variability introduced by the larger dataset, the feature distributions of the newly assigned points remain consistent with those of the original cluster.

Given the substantial increase in the noise component, the threshold condition defined in Equation 4.3 is exceeded. As a consequence, HDBSCAN is applied to the subset of points classified

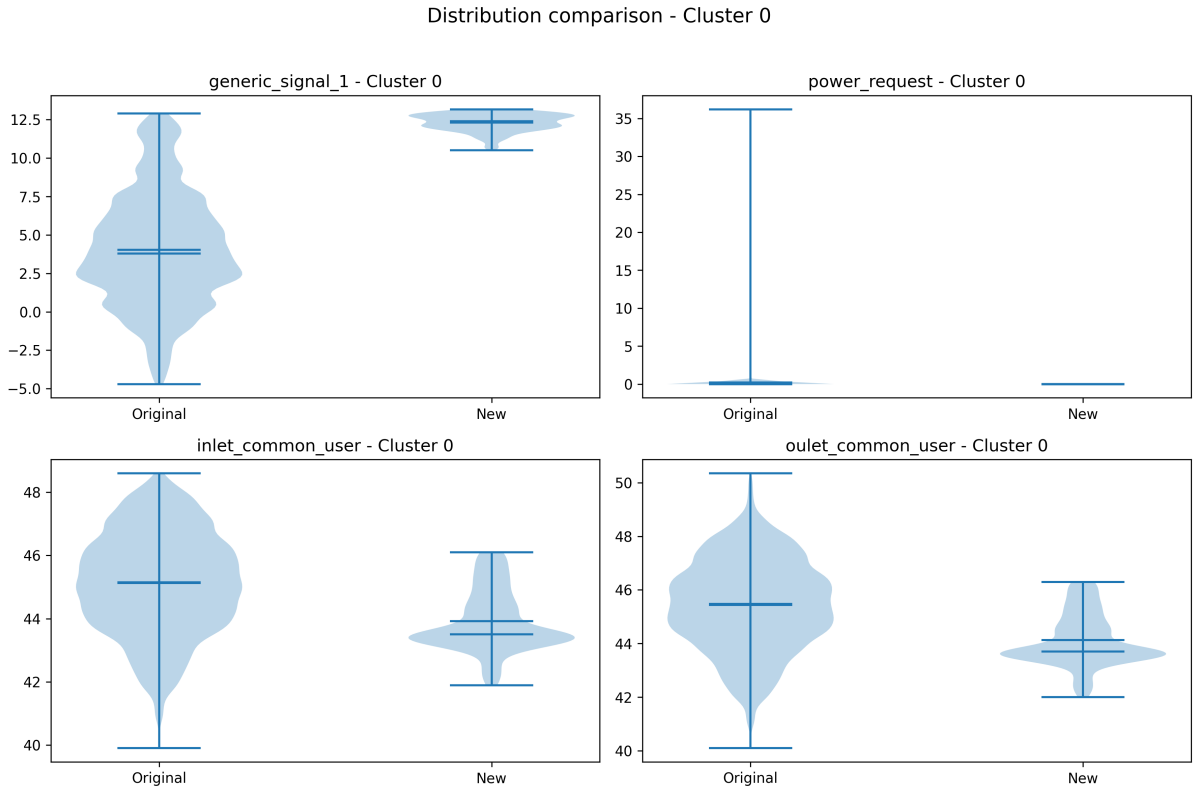

Figure 4.5: Comparison between original cluster 0 points (January) and newly assigned points (full month of July)

as noise in order to identify potential new cluster structures.

This step leads to the identification of **52 new clusters**, with labels ranging from **25 to 77**. After this re-clustering phase, the proportion of noise is reduced and stabilizes at approximately **37%** of the total July observations.

Finally, the CBD sampling procedure is applied to the updated cluster structure. As a result, the dataset corresponding to July is reduced from **17151** to **6945 observations**, preserving the most informative points while significantly reducing redundancy.

5

Conclusions

5.1 Research Objective

The objective of this thesis was to develop an effective strategy for reducing the size of large and non-uniform datasets while preserving the most informative structure of the feature space. To address this problem, a clustering-based approach was proposed, leading to the definition of the **Cluster-Based Density (CBD) sampling framework**. By leveraging HDBSCAN, the method captures local density variations and enables a sampling process that reduces redundancy in dense regions while preserving diversity in less populated areas. Beyond static data reduction, the work also extends the framework to dynamic scenarios, introducing mechanisms for the incremental integration of new observations and the evolution of the cluster structure over time.

5.2 Main Findings

The experimental results show that it is possible to substantially reduce dataset size while maintaining predictive performance. In the considered setting, the sampled datasets, despite being several times smaller than the original, consistently achieve performance levels close to those obtained using the full dataset. The effectiveness of the approach depends on the complexity of the feature space. When the input space is relatively simple, standard techniques such as Simple Random Sampling (SRS) can still provide competitive results. However, as the dimensionality and heterogeneity of the data increase, the advantages of the proposed method become more evident. In these cases, CBD sampling provides a more balanced representation of the feature space by explicitly accounting for local density variations. A key benefit of the approach is the reduction of redundancy in highly populated regions, which can lead to improved generalization in more complex settings. At the same time, the method ensures that less frequent but informative regions are adequately represented.

From a computational perspective, the reduction in dataset size translates into a significant decrease in pipeline execution time. The experiments show a reduction of more than 50% in end-to-end execution time, highlighting the strong relationship between data volume and computational cost.

Finally, the extension to dynamic scenarios demonstrates that the framework can adapt to evolving data distributions. The assignment mechanism allows new observations to be consistently integrated into the existing structure, while the monitoring of the noise component enables the detection of emerging patterns and the formation of new clusters. This adaptive

behavior makes the approach suitable for real-world applications in which data are continuously generated and updated over time.

5.3 Limitations and Improvements

Despite the encouraging results, several limitations should be considered.

A first limitation concerns the dependence on the clustering step. The effectiveness of the sampling procedure relies on the quality of the cluster structure identified by HDBSCAN, which in turn depends on the choice of its hyperparameters. Selecting appropriate values may require domain knowledge and can limit the robustness of the method when applied to new datasets. A second limitation is related to computational cost. While the sampling step improves the efficiency of downstream tasks, the initial clustering phase may become expensive for very large datasets. In such cases, HDBSCAN can represent a bottleneck, partially offsetting the benefits of the overall approach. Moreover, the method does not uniformly outperform simpler alternatives in all scenarios. In settings characterized by low dimensionality or limited complexity, standard random sampling may still provide competitive results. This suggests that the proposed framework is particularly advantageous in cases where the data exhibit strong heterogeneity and non-uniform density.

These limitations highlight several opportunities for improvement. A promising direction is the development of more automated and data-driven mechanisms for parameter selection, reducing the reliance on manual tuning.

Another important aspect concerns the adaptive component of the framework. In the current formulation, the decision to generate new clusters from noise is based on predefined thresholds. This mechanism could be refined by incorporating additional structural information, such as cluster sizes or assignment dynamics, allowing the procedure to react more flexibly to changes in the data distribution. Improving the level of automation is particularly relevant in industrial settings, where scalability and limited human intervention are key requirements.

5.4 Future Directions

The proposed framework can be extended in several directions, both methodological and applicative.

From a methodological perspective, future work may focus on improving the scalability of the clustering phase, for instance through approximate or distributed variants of HDBSCAN. Another direction concerns the development of more robust adaptive mechanisms for cluster evolution, enabling a more accurate detection of distributional shifts over time.

From an application perspective, the CBD sampling framework can be extended beyond tabular data. One promising direction is its application to large-scale textual datasets, particularly in the context of fine-tuning Large Language Models. In such scenarios, documents can be represented through embeddings, and clustering can be performed in the embedding space. This would allow the identification of semantically coherent groups and enable density-aware sampling that preserves diversity while reducing redundancy, leading to more efficient fine-tuning pipelines.

Another potential application concerns drift detection in MLOps systems. The cluster–noise structure can be used to monitor how new data relate to the existing distribution. An increasing proportion of noise or systematic deviations from established clusters may indicate distributional changes, providing a signal for model retraining or adaptation.

These directions suggest that the proposed framework can evolve from a data reduction technique into a more general tool for managing and monitoring complex data systems.

5.5 Final Remarks

This thesis shows that effective data reduction is not simply a matter of reducing the number of observations, but of preserving the underlying structure of the data. By leveraging density and clustering information, the proposed CBD framework enables the construction of smaller yet highly informative datasets, improving both efficiency and scalability. At the same time, the extension to dynamic scenarios demonstrates that the method can adapt to evolving data, maintaining a consistent and up-to-date representation of the feature space. Overall, the proposed approach provides a practical and flexible solution for handling large, non-uniform datasets, with strong potential for real-world applications.

A

Supplementary Results for the Real-World Dataset

This appendix reports additional results for the subsets corresponding to the remaining levels of the categorical variable *active_circuite*, complementing the analysis presented in Chapter 3.

For each subset, the same experimental pipeline is applied. In particular, clustering is performed using HDBSCAN with the same parameters adopted in the main analysis ($min\_cluster\_size = 60$ and $min\_samples = 6$), followed by the CBD sampling procedure, applied with the same configuration and parameters used in the main experiments.

For each subset, the following elements are reported:

- the distribution of input features across clusters, visualized through violin plots;
- the comparison between original and sampled distributions using kernel density estimates and histograms, for the three sampling strategies:
  - sampling based on core distances,
  - sampling based on probabilities,
  - sampling based on cluster sizes;
- the cluster-level metadata derived from HDBSCAN membership probabilities.

These results provide a global view of the proposed methodology across the whole dataset.

A.1 Results for *active_circuite* = 0.0

This section reports the results obtained for the subset corresponding to *active_circuite* = 0.0, which consists of 32,577 observations. The same analysis pipeline described in Chapter 3 for the subset *active_circuite* = 1.0 is applied here. For brevity, only the corresponding visual results are reported, as the interpretation remains consistent across subsets.

The HDBSCAN clustering algorithm identifies three main clusters, with a relatively small proportion of noise points, approximately 1%.

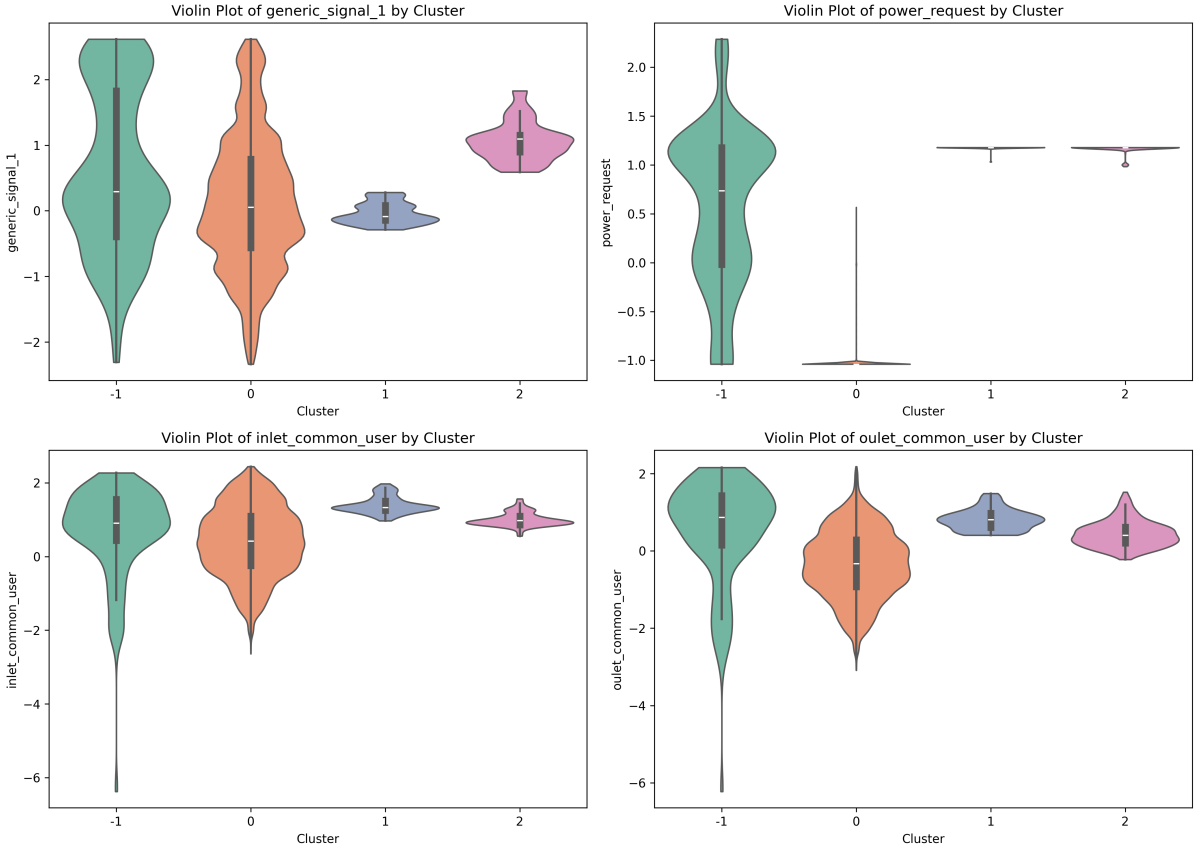

Figure A.1: Distribution of input features across clusters identified by HDBSCAN.

A.1.1 Core Distance-based CBD sampling

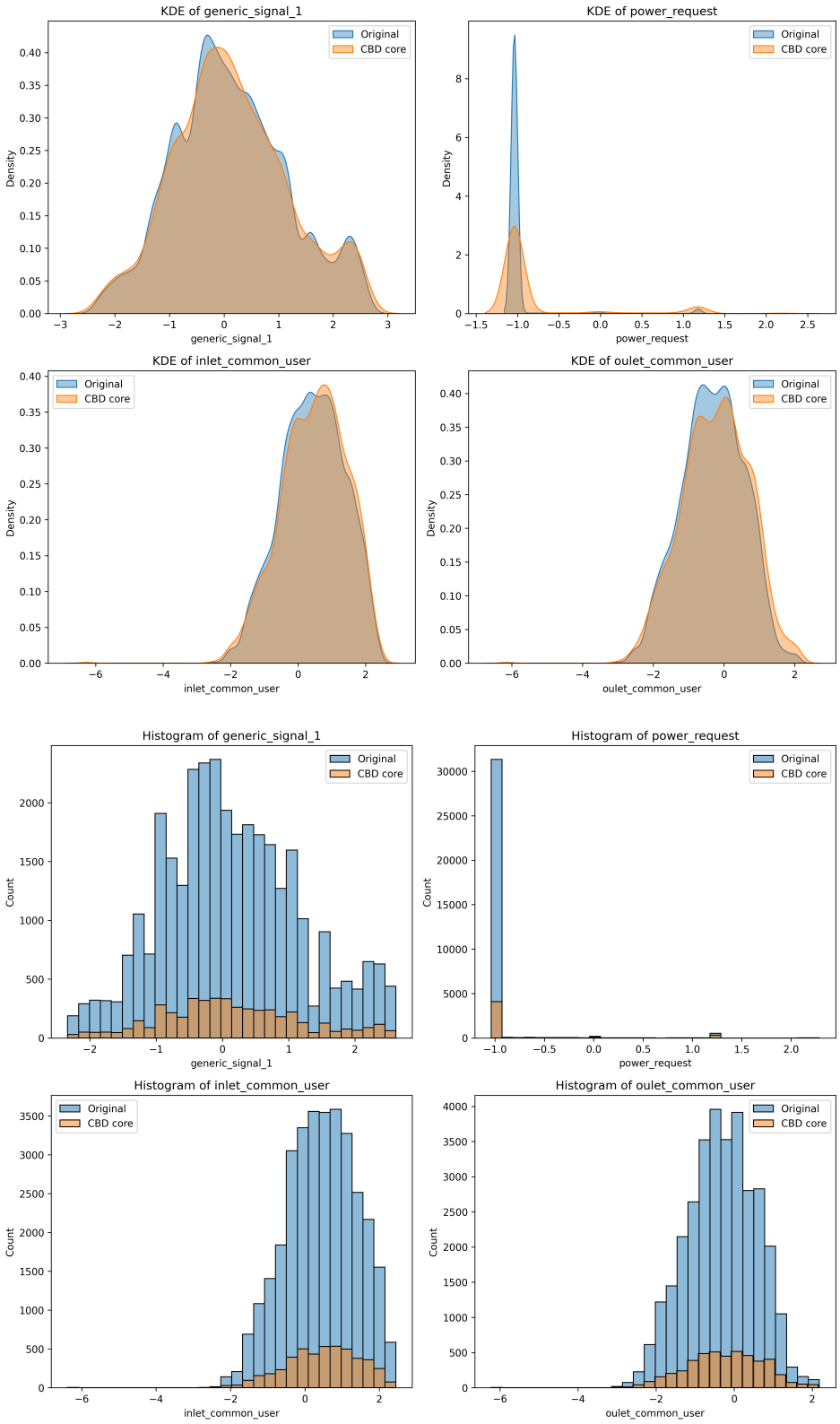

Figure A.2: KDE and histogram comparison of the input features before and after core distance-based CBD sampling.

A.1.2 Probabilities-based CBD sampling

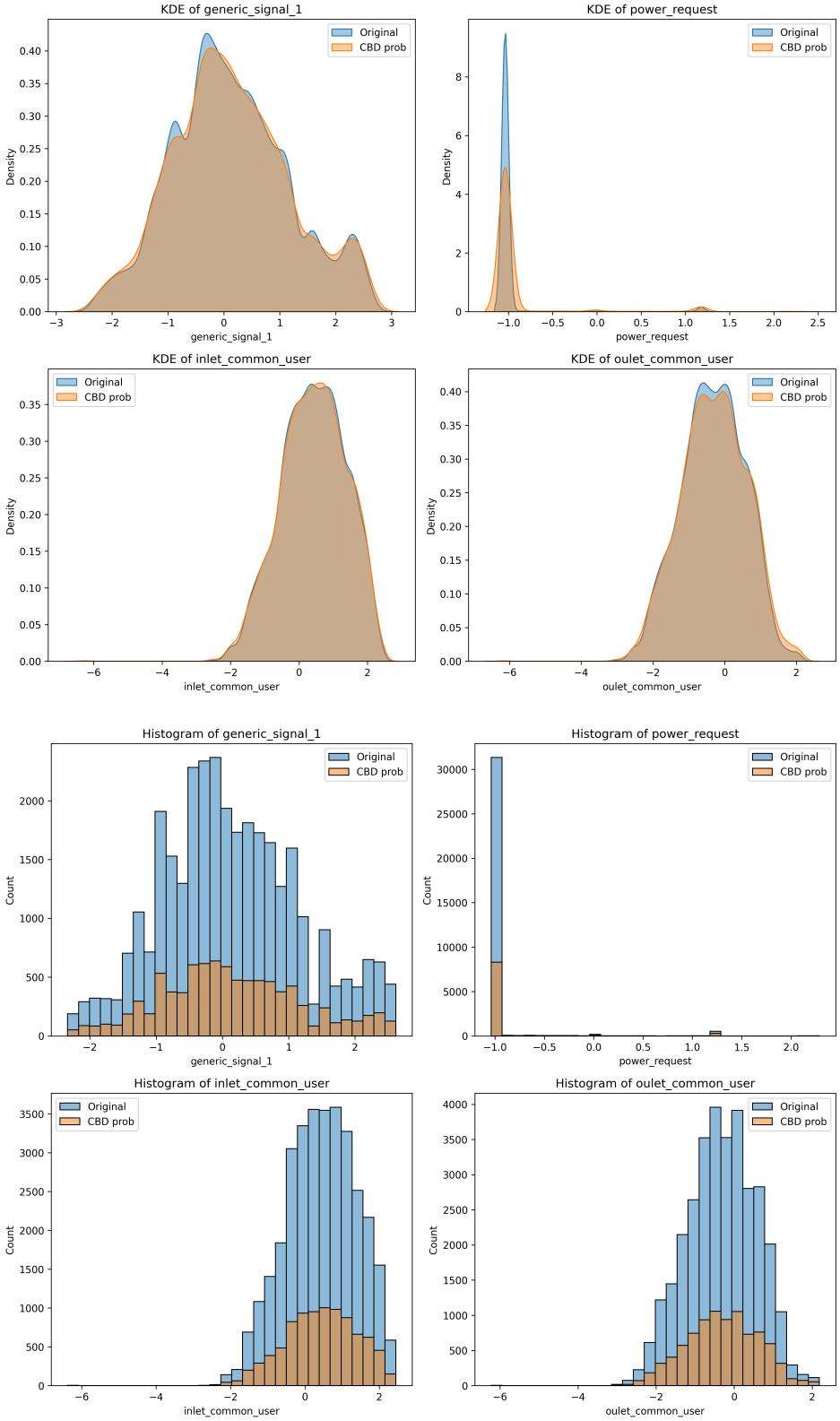

Figure A.3: KDE and histogram comparison of the input features before and after core distance-based CBD sampling.

A.1.3 Cluster Size-based CBD sampling

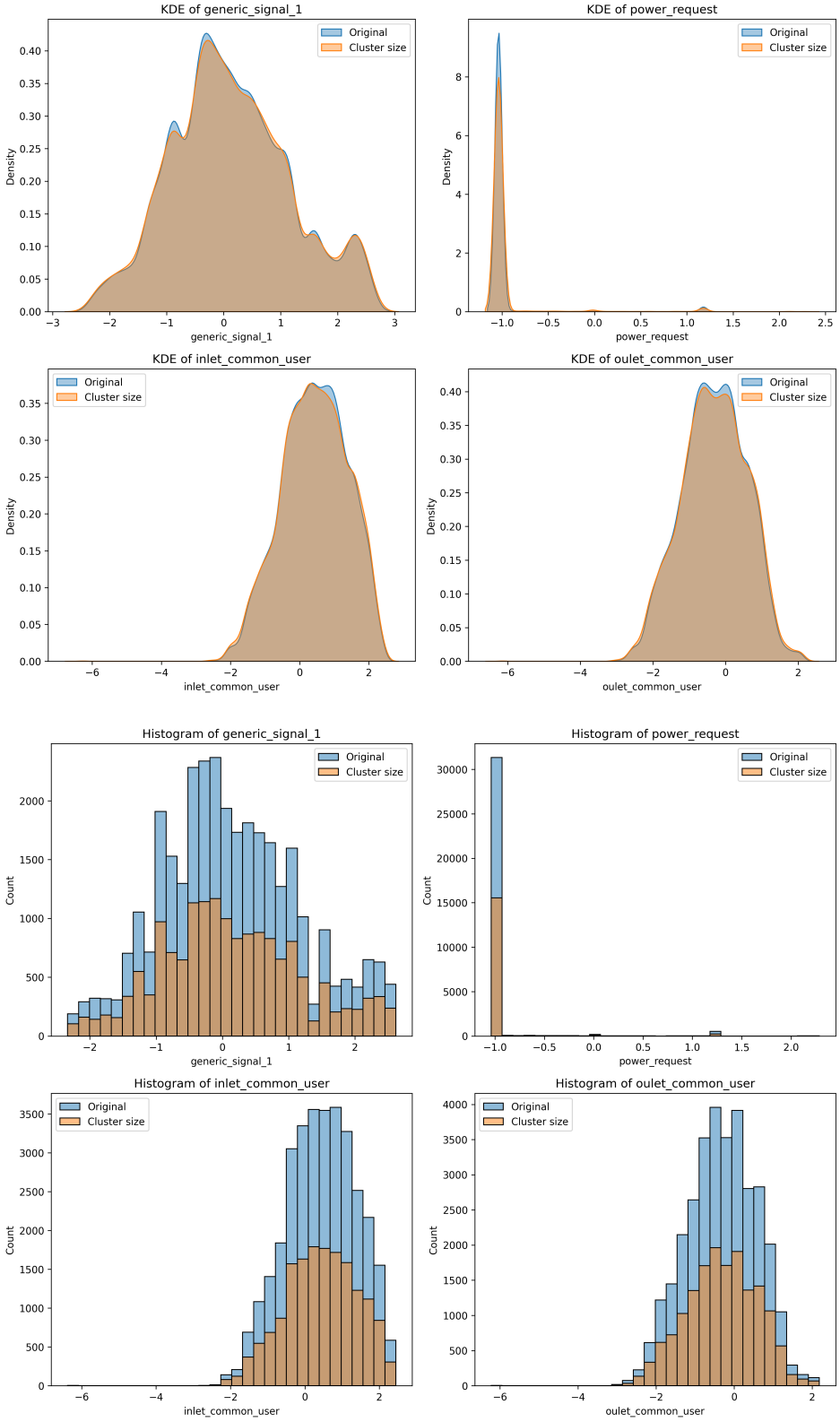

Figure A.4: KDE and histogram comparison of the input features before and after core distance-based CBD sampling.

A.1.4 Cluster Metadata

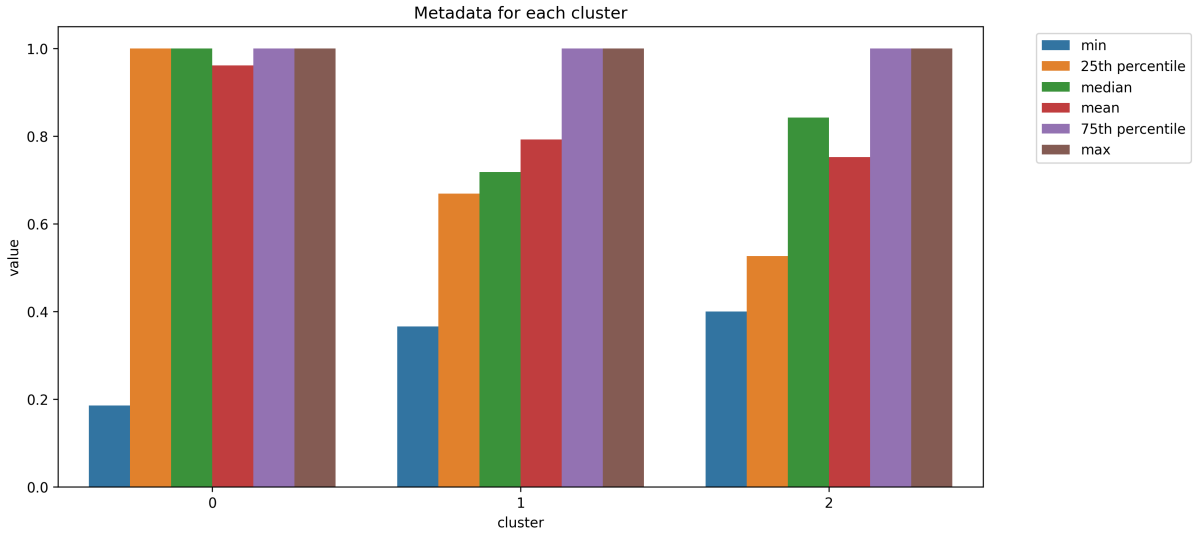

Figure A.5: Cluster-level metadata based on membership probabilities.

A.2 Results for *active_circuite* = 2.0

This section reports the results obtained for the subset corresponding to *active_circuite* = 2.0, which consists of 22,187 observations. The same analysis pipeline described in Chapter 3 for the subset *active_circuite* = 1.0 is applied here. For brevity, only the corresponding visual results are reported, as the interpretation remains consistent across subsets.

The HDBSCAN clustering algorithm identifies ten main clusters, with a moderate proportion of noise points, approximately 12%.

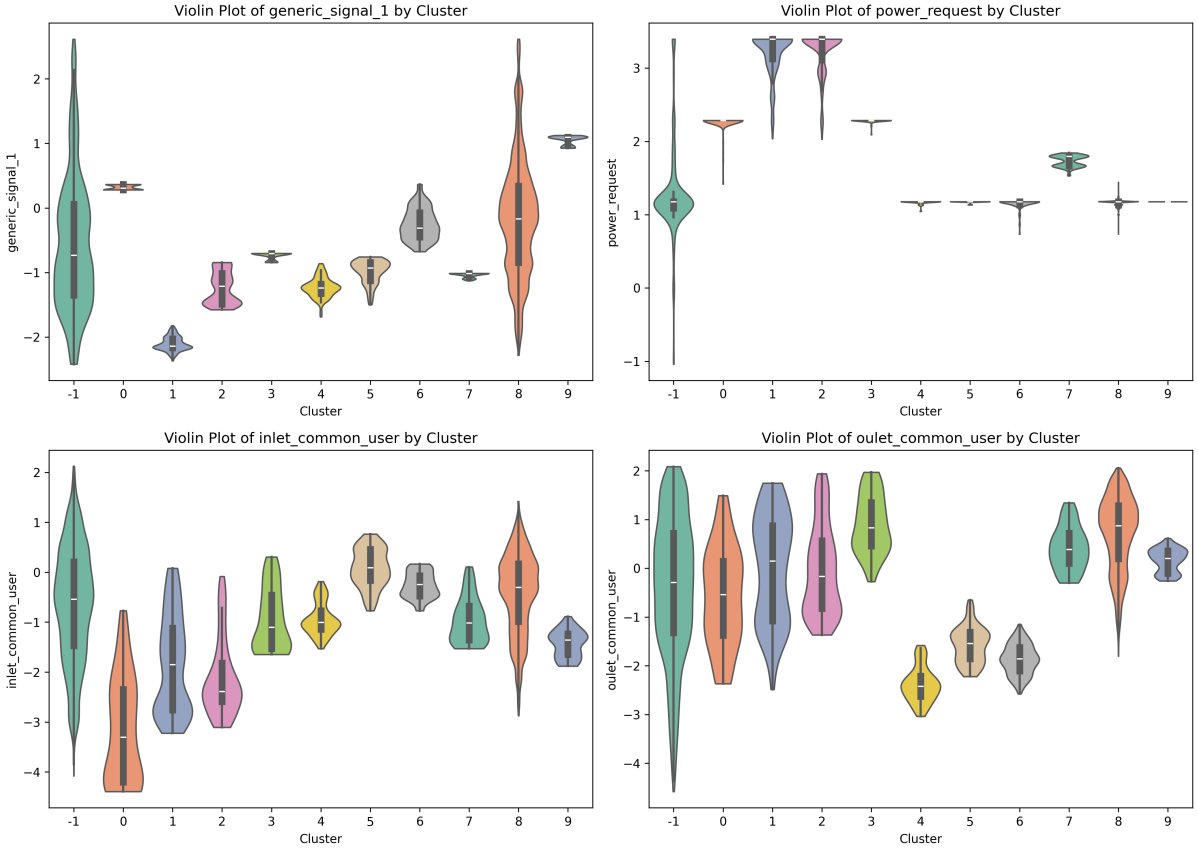

Figure A.6: Distribution of input features across clusters identified by HDBSCAN.

A.2.1 Core Distance-based CBD sampling

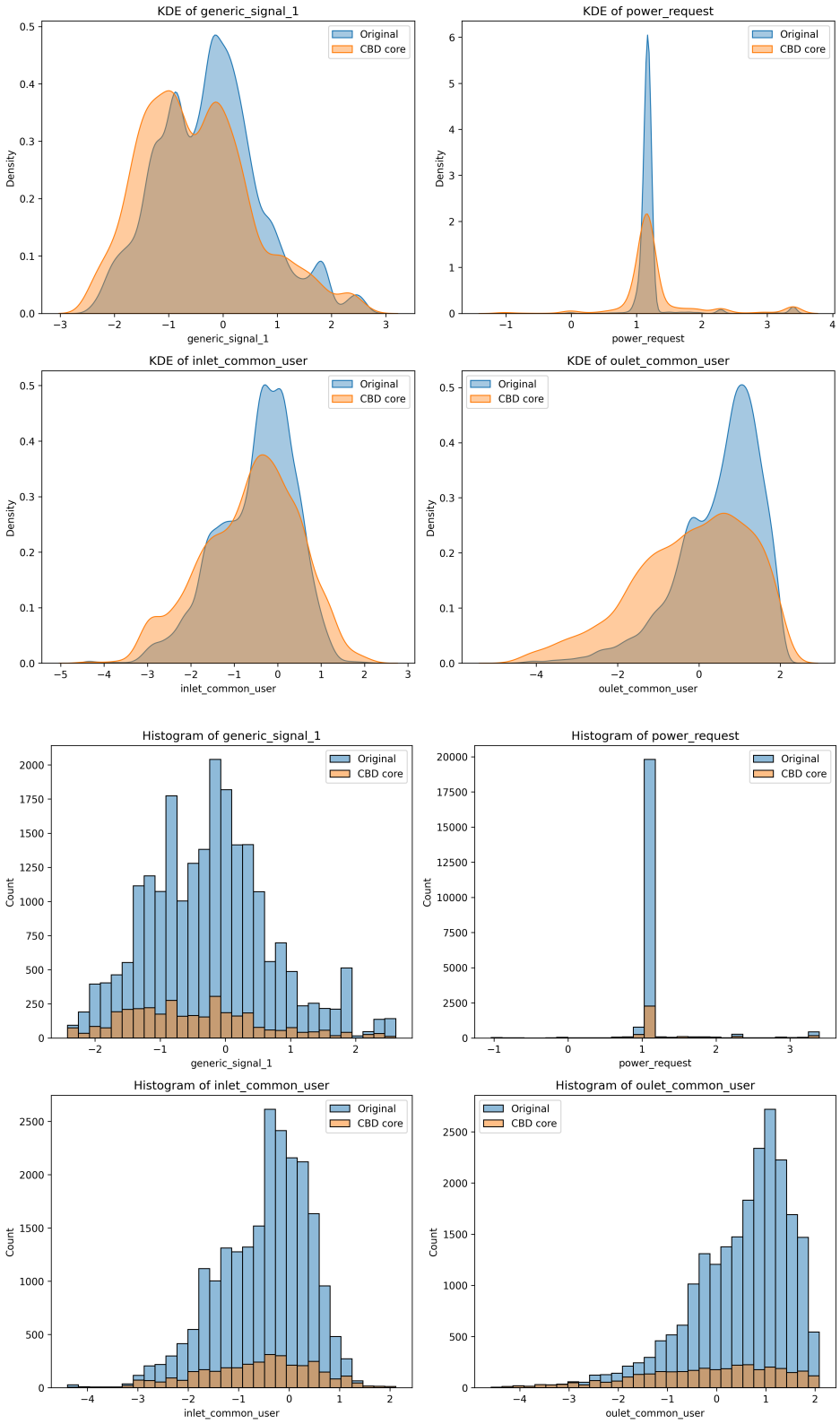

Figure A.7: KDE and histogram comparison of the input features before and after core distance-based CBD sampling.

A.2.2 Probabilities-based CBD sampling

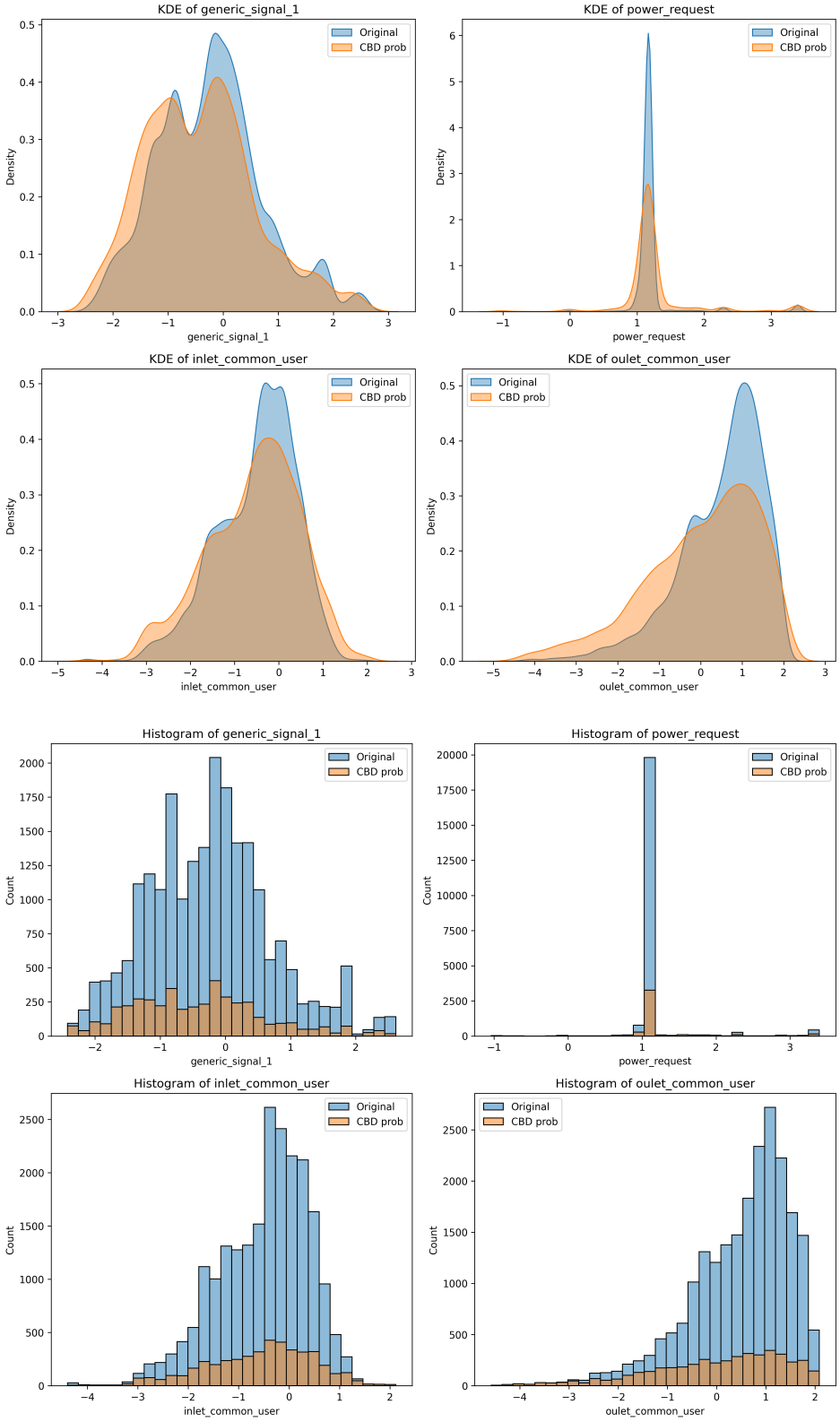

Figure A.8: KDE and histogram comparison of the input features before and after core distance-based CBD sampling.

A.2.3 Cluster Size-based CBD sampling

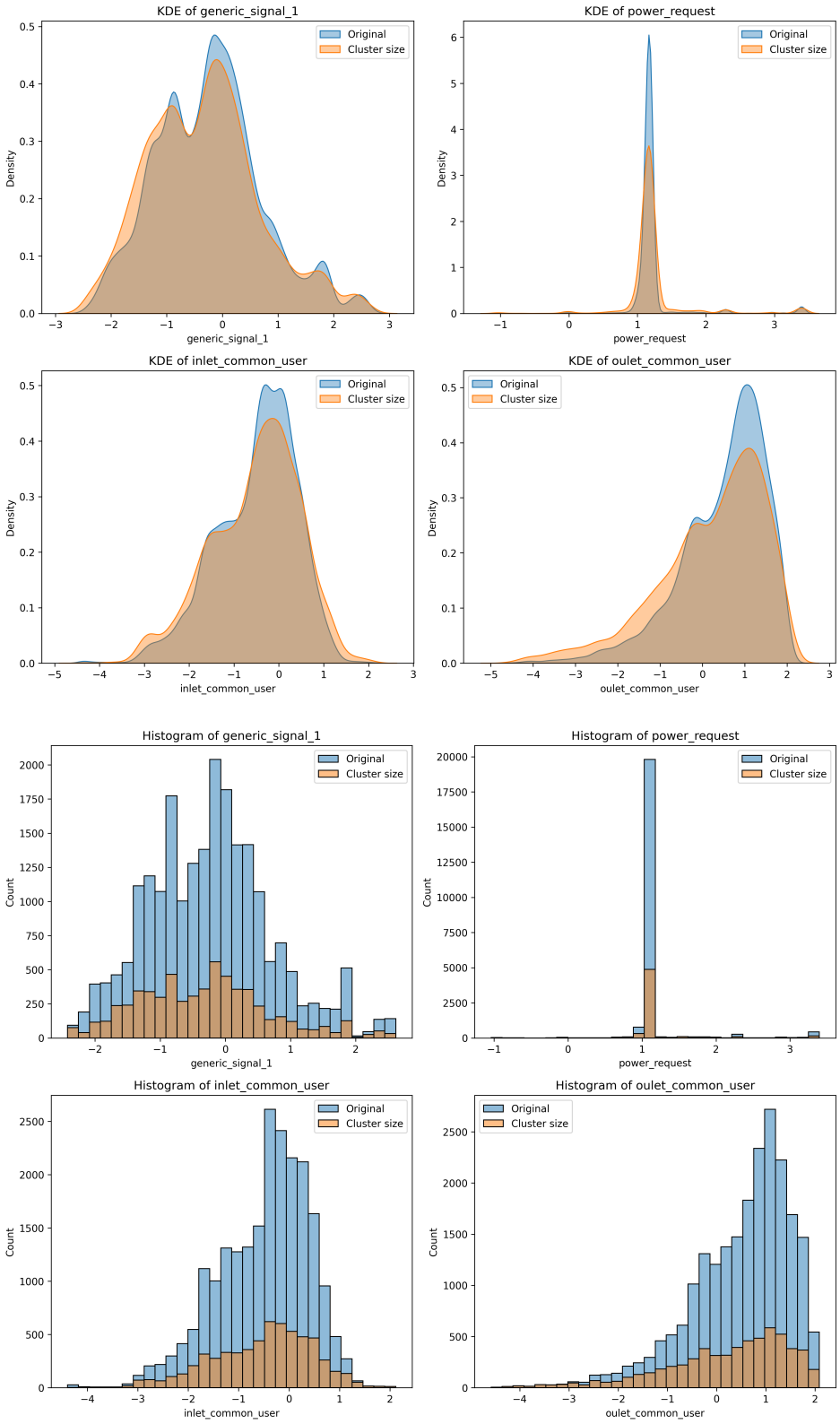

Figure A.9: KDE and histogram comparison of the input features before and after core distance-based CBD sampling.

A.2.4 Cluster Metadata

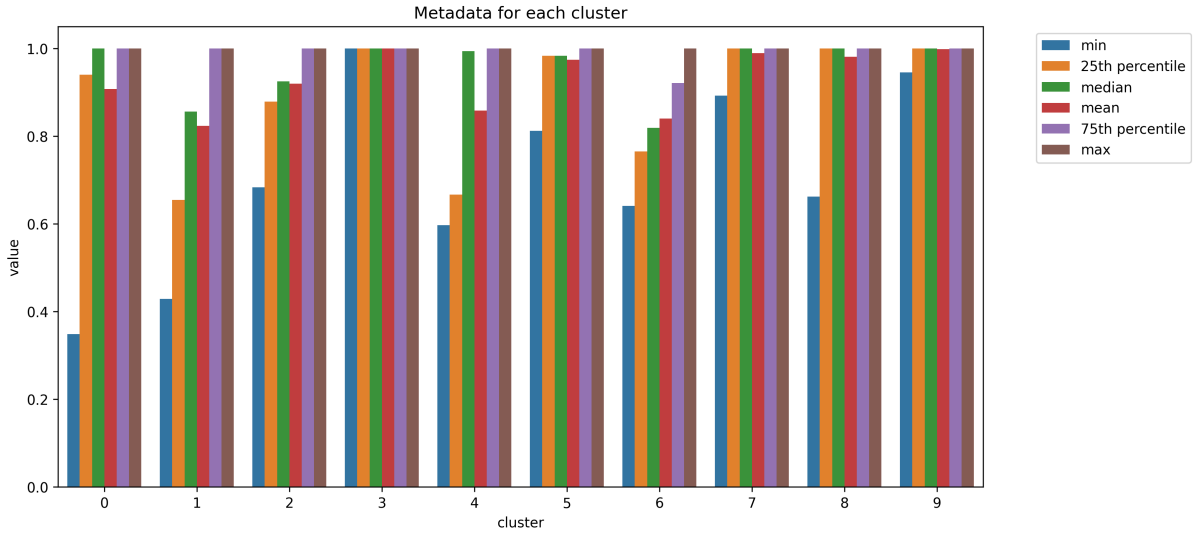

Figure A.10: Cluster-level metadata based on membership probabilities.

Bibliography

[1] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172. Springer, 2013.

[2] Ricardo J. G. B. Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data*, 10(1):1–51, 2015.

[3] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[4] Leland McInnes and John Healy. Accelerated hierarchical density clustering. *arXiv preprint arXiv:1705.07321*, 2017.

[5] HDBSCAN Documentation. How hdbscan works. `https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html`, 2023. Accessed: 2026-04-15.

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009.

[8] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] Dhaval Makwana et al. Sampling methods in research: A review. *International Journal of Trend in Scientific Research and Development*, 2023. Available on ResearchGate.

[10] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[11] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, 2010.

[12] Charu C. Aggarwal. *Outlier Analysis*. Springer, 2015.

[13] Charu C. Aggarwal. Mining big data. In *Proceedings of the 2012 IEEE International Conference on Management of Data*, 2012.

[14] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2011.

[15] Richard Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961. Introduces the curse of dimensionality.

[16] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International Conference on Database Theory*, pages 217–235, 1999.