Analysis of Preprocessing Steps for z-Anonymity in IoT Architectures

Mateusz Kaczynski
TU Dresden
mateusz.kaczynski@mailbox.tu-dresden.de

February 16, 2026

Abstract

This thesis investigates the impact of preprocessing steps on the effectiveness of z-anonymity in multi-layered IoT architectures. (To be continued)

Contents

List of Abbreviations

IoT	Internet of Things

1 Introduction

In the era where Internet of Things (IoT) continuously gathers importance [1], the collection of high-frequency data from smart meters offers unprecedented opportunities for operational efficiency [2]. Smart meters are a fundamental component of demand response systems, supplying them with the consumption data that is needed to operate effectively. These systems leverage this data to detect changes in the energy usage in response to electricity price fluctuations or incentive payments, which are designed to encourage lower electricity use at times of high wholesale market prices [3]. This knowledge can be used by the energy provider to balance the energy production levels, ultimately improving the system efficiency [4].

However, the collection of such detailed consumption data from smart meters poses significant privacy risks. Smart meter reading can reveal sensitive characteristics about marital status, employment status, income, or the number of residents [5]. Moreover, behavioral patterns, including residents' presence, sleep schedule, or meal times, can be inferred from this consumption data. When combined with information about specific appliances, this information may be exploited by adversaries for malicious purposes [6].

Consequently, protecting this sensitive information is not merely a technical necessity for user safety, but a strict legal requirement. Within the European Union, the General Data Protection Regulation (GDPR), enacted in 2018, establishes strict rules for processing, storing, managing personal data from individuals. Its primary goal is to give individuals more responsibility over their own data and ensure that companies handle data responsibly [7].

A naive approach might suggest removing user's private information in the published data to avoid those threats. Yet, it is well-established that simply removing direct identifiers is insufficient for protecting user privacy. Infamous real-world examples, such as the de-anonymization of the Netflix Prize dataset, and foundational academic research have proven that individuals can be easily re-identified by linking seemingly innocuous data points known as quasi-identifiers [8].

To counter this threat, formal privacy models like k-anonymity were developed for static datasets [9]. However, these traditional models are often too computationally intensive for the real-time, zero-delay requirements of modern data streams. This has led to the development of stream-specific algorithms, such as z-anonymity, which are designed to provide lightweight, continuous privacy protection [10].

While z-anonymity provides the core privacy, we argue that its effectiveness and the resulting data utility can be significantly influenced by data preprocessing techniques. These steps, applied before the main anonymization algorithm, can modify the data to influence the privacy-utility trade-off. For instance, generalization could possibly increase the amount of data that is ultimately published, while aggregation could drastically reduce communication overhead and system load.

The availability of these powerful tools raises a critical, underexplored architectural question: at which layer of the architecture should privacy-preserving preprocessing steps be deployed? The conventional Centralized model requires placing a high degree of trust in the central entity that collects all raw, sensitive data [11]. However, modern IoT networks are multi-layered [1], consisting of smart meters, gateways, and a central entity. This motivates our investigation into how shifting preprocessing steps, such as prefiltering, generalizing, and aggregating, affect the final privacy-utility trade-off when performed closer to the source data itself.

This research will systematically evaluate the impact of deploying these data preprocessing steps at different layers within a centralized smart meter architecture. Specifically, this work will establish a baseline scenario where raw data is processed only by the z-anonymity algorithm at the Central Entity. This baseline will then be systematically compared against variations where data preprocessing steps are first applied at the edge (on Smart Meters) or at the gateway level, allowing for a precise measurement of their impact on the final privacy-utility trade-off. By measuring the effects on data utility, information loss, and system performance, this research will provide a clear framework for understanding the trade-offs of moving computation closer to the user. All experiments will be grounded in a realistic context by using the SmartMeter Energy Consumption Data in London Households [12] dataset.

This thesis is structured as follows. Chapter 2 introduces the background knowledge required to understand this work. Chapter 3 reviews the related literature. Chapter 4 describes the research methodology, while Chapter 5 presents the experimental results. Chapter 6 discusses these results, and Chapter 7 concludes the thesis.

2 Background

2.1 The Smart Meters and IoT Architecture

Smart meters are digital energy measurement devices that automatically record electricity consumption and transmit these measurements to the utility provider at regular intervals [2]. Depending on the deployment and configuration, the reporting interval usually ranges from seconds to hours. This continuous data collection enables detailed monitoring of energy usage. While this data is essential for applications such as demand response and load forecasting [3], [4], it also introduces significant challenges related to data volume, communication overhead, and user privacy [5], [6].

Smart meters operate within the IoT architectures [1]. While early implementations often relied on direct device-to-cloud communication, modern systems are increasingly organized as multi-layered architectures consisting of the Edge Layer, the Fog Layer, and the Central Layer [13], [14]. The split into different layers brings its advantages.

The Edge Layer consists of the smart meters themselves. Those devices are deployed in close proximity to end users and generate raw consumption data. Due to constraints in computational power, memory, and energy consumption, edge devices are typically limited in the complexity of calculations they can execute [15]. However, performing lightweight preprocessing at the edge offers important advantages. By executing initial tasks such as data filtering or local aggregation, the system can scale effectively. This is particularly critical given that the amount of data produced by smart meters will continue to grow in the upcoming years, potentially exceeding the capacity of conventional cloud computing to handle raw streams [13]. Moreover, it also brings the advantage with respect to privacy, as sensitive data can be transformed before leaving the user's premises, thereby reducing the trust needed in the central entity, which is currently seen as a potential risk [11], [13].

The Gateway of Fog Layer represents an intermediate layer between the edge devices and the central entity. They work by analyzing the data sent by edge nodes and reducing it significantly by removing unimportant data or performing preprocessing tasks such as filtering, aggregation, or generalization. By reducing the volume of the data transmitted, this layer can decrease communication overhead and address the inherent cloud problems such as "unreliable latency, lack of mobility support and location-awareness" [14], while still retaining more computational resources than individual smart meters.

The Central Entity, often described as a cloud-based system, is responsible for long-term storage, large-scale analysis, and decision-making processes. While centralized processing simplifies system management and enables complex analytics [2], it requires the transmission of raw or minimally processed data, which increases both privacy risks and the level of trust placed in the data collector [6].

This layered architecture reflects a fundamental trade-off between efficiency, privacy, and system complexity. Moving computation closer to the data source can reduce latency and bandwidth requirements while limiting the exposure of sensitive information. At the same time, it constrains the available computational resources. These trade-offs are central to this thesis, which investigates how the placement of privacy-preserving preprocessing steps within this architecture affects the final privacy-utility balance.

2.2 Data Privacy Fundamentals

The collection of smart meter data raises significant privacy concerns, as energy consumption patterns can reveal sensitive information about individuals and households [6]. In this matter, it is essential to distinguish between explicit identifiers and quasi-identifiers.

Explicit identifiers are attributes that uniquely identify an individual on their own, such as a name, address, or customer identification number. In the context of smart metering, examples include the physical address of a household or a unique meter identifier. These attributes are typically removed before data is shared or published [9].

Quasi-identifiers are attributes that are not identifying in isolation but can be used to re-identify individuals when combined with other data sources. In smart meter datasets, quasi-identifiers may include timestamps, energy consumption values, and location-related information. Although such attributes appear innocuous, their combination can form distinctive usage patterns that uniquely characterize households [9].

The risk posed by quasi-identifiers is demonstrated by linkage attacks, in which anonymized datasets are combined with external information, often publicly available, from an auxiliary dataset that shares common attributes to re-identify individuals [9].

A foundational example of this vulnerability is the work of Latanya Sweeney, who demonstrated that 87% of the U.S. population could be uniquely identified using only a combination of ZIP code, gender, and date of birth [16]. Sweeney proved the validity of this threat by successfully re-identifying the medical records of the Governor of Massachusetts, linking the "anonymized" hospital data (which contained diagnosis and demographics) with the public Voter Registration List (which contained names and demographics). Similarly, Narayanan and Shmatikov de-anonymized the Netflix Prize dataset [8]. Although the dataset had been stripped of user IDs, the researchers were able to re-identify specific users by linking the timestamped movie ratings in the dataset with public reviews posted on IMDb.

In the smart meter context, linkage attacks may exploit publicly available information, such as occupancy schedules or appliance usage statistics, to infer private attributes including presence patterns, daily routines, or socio-economic characteristics [5]. These risks illustrate that simply removing explicit identifiers is insufficient to guarantee privacy and motivate the use of formal privacy models that provide quantifiable protection against re-identification [9].

2.3 Formal Privacy Models

To mitigate the re-identification risks described in the previous section, several formal privacy models have been developed. These models transition privacy from a heuristic process to a mathematically provable state. This section outlines the foundational model of k-anonymity, the alternative paradigm of Differential Privacy, and the stream-specific derivation known as z-anonymity.

2.3.1 K-Anonymity and Equivalence Classes

The foundational concept in privacy-preserving data publishing is k-anonymity, proposed by Sweeney. Formally, a dataset satisfies k-anonymity if every record is indistinguishable from at least $k-1$ other records with respect to the set of quasi-identifiers [9].

The core mechanism relies on partitioning the dataset into equivalence classes. An equivalence class is a set of records that share identical values for their quasi-identifiers. To achieve a specified threshold k, algorithms employ generalization (reducing granularity, e.g., converting a specific timestamp to a 1-hour interval) or suppression (removing the value entirely) until the cardinality of every equivalence class is $\geq k$ [9].

However, k-anonymity is inherently ill-suited for the IoT data streams described in Section 2.1. Standard algorithms require a global view of a static dataset to calculate optimal generalization hierarchies. In a streaming context, data arrives sequentially and indefinitely [17]. To form an equivalence class of size k, the gateway would need to buffer incoming tuples until enough matching records arrive. This introduces "blocking," creating significant latency that contradicts the real-time requirements of smart grids [10]. Furthermore, the computational cost of recalculating equivalence classes as new data arrives is often prohibitive for resource-constrained edge devices [17].

2.3.2 Differential Privacy

Differential Privacy (DP) is widely regarded as the "gold standard" in modern privacy research. Unlike k-anonymity, which focuses on the structure of the output data, DP focuses on the algorithm itself. It guarantees that the output of a query is substantially similar whether or not any single individual's data is included in the input [18].

Mechanically, DP is achieved by injecting calibrated noise (typically via Laplace or Gaussian mechanisms) into the data or query results [18]. While robust against linkage attacks, this noise injection presents a fundamental conflict with the operational requirements of smart metering. As emphasized in [3], energy data must be reliable and precise to avoid substantial financial costs, with billing meters subject to rigorous accuracy standards (e.g., IEC 61036). Consequently, the probabilistic distortion introduced by DP is often considered unacceptable for raw energy data streams where precise measurements are mandatory.

Furthermore, in a continuous streaming context, the Basic Composition Theorem [18] dictates that privacy loss accumulates with every query ($\sum \epsilon_i$). As established in Section 3.5 of the foundational literature, this cumulative property makes it mathematically impossible to maintain a fixed privacy guarantee over an indefinite period without eventually stopping transmission or introducing infinite noise.

2.3.3 z-Anonymity

To address the latency constraints of k-anonymity while avoiding the noise injection of DP, z-anonymity was proposed as a lightweight, stream-centric alternative. Unlike k-anonymity, which enforces distinct groups of k current records, z-anonymity operates on the principle of historical frequency. It assumes that privacy risks are highest for "outliers"—values that appear rarely in the data stream [10].

The algorithm processes data in a single pass with O(1) complexity, making it ideal for the Edge and Fog layers. It functions through the following steps, based on the definition in [10]:

1. **History Maintenance**: The system maintains a sliding window or historical table of observed values for a specific quasi-identifier.

2. **Lookup**: For every incoming data tuple, the algorithm queries the historical count of that attribute value.

3. **Threshold Check**: If the count exceeds the threshold z, the value is deemed "safe" (common enough to not be identifying) and is transmitted immediately.

4. **Suppression**: If the count is below z, the value is considered a rare outlier and is suppressed or generalized.

This approach eliminates the need for buffering, allowing for zero-delay transmission [10]. Crucially, z-anonymity is probabilistically linked to k-anonymity. The authors demonstrate that if a value appears z times in a sufficiently large historical window, the probability that the record belongs to a crowd of size k approaches 1. Therefore, z-anonymity acts as a performant proxy for k-anonymity in high-velocity environments, balancing privacy protection with the strict low-latency requirements of IoT architectures [10].

2.4 Data Preprocessing Techniques for Privacy

While formal privacy models like k- or z-anonymity provide the criteria for protection, preprocessing techniques are the mechanisms used to transform raw data to meet these criteria. In the context of IoT architectures, these transformations are crucial for striking a balance between privacy (indistinguishability), utility (analytical precision), and system performance (bandwidth and latency). This section formally defines the three primary preprocessing steps investigated in this thesis: Data Generalization, Temporal Aggregation, and Local Prefiltering.

2.4.1 Data Generalization

Data generalization is a non-perturbative technique that replaces specific, precise values with broader, less specific categories or intervals. The fundamental goal is to increase the probability that a specific tuple belongs to a larger equivalence class (or meets the z-threshold) by reducing the granularity of the data. Formally, generalization relies on a Value Generalization Hierarchy (VGH). A VGH is a tree structure where leaf nodes represent raw values (e.g., 1.234 kWh) and root nodes represent the most general state (e.g., "Any Energy Value"). In between are intermediate levels of granularity [19].

There are two primary approaches to defining these hierarchies:

1. **Static Generalization (Fixed-Intervals)**: The domain of an attribute is divided into predefined intervals (e.g., 0-5 kWh, 5-10 kWh) or precision levels (e.g., rounding to the nearest integer). For categorical attributes, such as ZIP codes, this typically involves masking the least significant digits (e.g., replacing the last digit with *) [9]. This method is computationally lightweight (O(1)) and deterministic, making it highly suitable for the resource-constrained Edge layer of IoT devices [13].

2. **Dynamic Binning**: As noted in literature, static intervals typically rely on rigid hierarchies that may fail to capture the actual data distribution, resulting in unnecessary information loss. To address this, dynamic partitioning techniques such as Mondrian [20] were proposed to adapt the intervals based on the local density of the data. While dynamic binning often yields higher utility—measured, for instance, by the Normalized Certainty Penalty (NCP) defined in [21], it typically requires

sorting or multiple passes over the data, which introduces computational overhead that may not be viable for real-time edge processing.

For this thesis, generalization is modeled through precision reduction (rounding). This acts as a static VGH where moving up the hierarchy corresponds to reducing the number of decimal places, effectively grouping precise sensor readings into discrete "bins" to facilitate indistinguishability.

2.4.2 Temporal Aggregation

Temporal aggregation serves as a dimensionality reduction technique along the time axis. In high-frequency smart metering (e.g., reading every second), the timestamp itself acts as a quasi-identifier that can reveal minute-by-minute lifestyle patterns [6].

This technique replaces a sequence of fine-grained data points $d_1, d_2, \dots, d_n$ occurring within a time window W with a single aggregate value V_{agg} (typically the mean, sum, or max). The windowing strategy determines how data is grouped:

1. **Tumbling Windows**: These are fixed, non-overlapping windows (e.g., 10:00–10:15, 10:15–10:30). In this approach, every data point contributes to exactly one aggregate value.

2. **Sliding Windows**: These are overlapping intervals (e.g., 10:00–10:15, 10:05–10:20), where a single data point contributes to multiple aggregate values.

In this research, tumbling windows are employed to calculate average energy consumption. This approach significantly reduces the data volume transmitted to the central entity while retaining the coarse-grained load profile necessary for forecasting applications [3].

2.4.3 Local Prefiltering (Suppression)

Suppression is the most drastic form of data protection, where specific attribute values are entirely removed from the dataset [9]. In a centralized architecture, suppression typically occurs after the data has reached the central server (global suppression). However, Local Prefiltering shifts this decision to the Gateway or Edge layer [11].

This technique operates on a rule-based logic to identify "outliers" locally. In the context of z-anonymity, an outlier is a value that appears with insufficient frequency to be safe [10]. By implementing a local threshold (z_{local}), the Gateway can assess the rarity of a value before transmission.

Rule: If Frequency(value)$<z_{\text{local}}$, the value is suppressed locally.

Effect: This prevents the transmission of data that is statistically likely to fail the global privacy check at the Central Entity anyway.

While prefiltering significantly reduces Communication Overhead (message count), it introduces a risk to Data Utility (Publication Ratio). If the local view of the data at the Gateway does not perfectly reflect the global distribution at the Central Entity, the Gateway might suppress a value that—had it been sent—would have found matches from other Gateways. Investigating this trade-off between bandwidth savings and false-positive suppression is a key objective of this study.

3 Related work

This chapter contextualizes the research within the broader field of privacy-preserving technologies for IoT. It reviews existing stream-specific anonymization algorithms, analyzes architectural approaches to distributed privacy, and identifies the research gap regarding the placement of preprocessing steps in multi-layered architectures.

3.1 Privacy-Preserving Algorithms for Data Streams

While standard k-anonymity is the foundational model for privacy [9], its application to data streams is non-trivial. As noted in recent literature, finding an optimal k-anonymity scheme is NP-hard [22], and heuristic approximations typically require multiple scans of the dataset to minimize information loss. In a streaming context, where data is potentially infinite and high-velocity, such multi-pass algorithms introduce unacceptable latency and buffering requirements [17].

To address these constraints, researchers have developed stream-centric algorithms that typically rely on micro-aggregation or clustering.

CASTLE (Continuously Anonymizing STreaming data via adaptive cLustEring) is a stream-centric anonymization approach that models tuples as points in a similarity space defined by quasi-identifier attributes. Incoming tuples are incrementally clustered and released using a common generalization, supporting both numerical and categorical attributes. To ensure timely data publication, CASTLE enforces a delay constraint δ, which bounds the maximum time between a tuple's arrival and its anonymized release [23].

Subsequent improvements, such as FADS (Fast Anonymization of Data Streams) [17] or FAANST [24], attempt to optimize this process by maintaining dynamic or fixed groups to reduce computational overhead. These algorithms adhere to the strict principle that data streams should be scanned only once. However, despite their computational efficiency, they still fundamentally rely on buffering incoming data to form valid k-groups. This requirement inherently introduces variable latency, as the release of a tuple is blocked until sufficient tuples arrive, a critical drawback for real-time monitoring applications.

In contrast, z-anonymity, the algorithm central to this thesis, offers a different paradigm. Instead of clustering current records (which requires waiting for the arrival of similar tuples), it uses historical statistical frequency. By checking if a value has appeared z times in the past, it enables a zero-delay decision process with O(1) complexity [10]. This makes it distinct from the clustering-based family (CASTLE/FADS) and particularly suitable for low-latency smart metering.

3.2 Architectural Solutions for Privacy in IoT

As detailed in the background regarding IoT architectures, modern smart grid systems have evolved from cloud-centric models to multi-layered Edge/Fog topologies. This architectural shift has profound implications for privacy implementation. Conventional smart metering approaches rely on a centralized "Trusted Third Party" model, where the utility provider collects raw high-frequency data before applying any anonymization measures. However, this creates a single point of failure and a significant privacy risk, as highlighted by Lisovich et al., who demonstrated that detailed household activities can be inferred from such centralized raw streams [6].

To mitigate this trust requirement, recent paradigms advocate for Distributed Privacy, where anonymization occurs before data leaves the user's control.

Approaches at the Edge layer leverage the local processing power of smart meters to sanitize data at the source. By performing tasks such as noise addition or local suppression directly on the device, these systems ensure that no raw data ever traverses the network [13]. However, implementing complex privacy protocols on end devices is non-trivial. As noted by Zhou et al., the resource constraints of current smart meter hardware, specifically limited memory, often restrict the complexity of the algorithms that can be executed locally [15].

Intermediate gateways (Fog nodes) offer a compromise, possessing significantly more computational power than individual meters while remaining geographically closer to the user than the cloud [14]. Several distributed privacy systems rely on this layer to balance performance and security. For instance, the deZent model [11] utilizes gateways to coordinate data aggregation horizontally. This allows the system to identify and suppress rare values locally without a central entity seeing the raw inputs. Similarly, distributed k-anonymity protocols [25] use encryption across distributed nodes to compute global privacy parameters without revealing raw data.

These approaches demonstrate a clear trend in the literature: shifting the "privacy workload" from the Cloud to the Edge and Fog enhances security by minimizing data exposure, though it introduces new challenges regarding distributed coordination and resource management.

3.3 Evaluation Metrics

To evaluate the effectiveness of the proposed architectures, it is necessary to quantify the cost paid to achieve privacy. In privacy-preserving data publishing, this is typically measured in terms of Information Loss (General Utility) and Task-Specific Utility.

Information loss metrics quantify how much the data has been distorted or generalized relative to the raw input. These metrics are independent of the specific application (e.g., billing or forecasting) and provide a general measure of data precision.

Discernibility Metric (DM): In static data contexts, the Discernibility Metric [26] is a standard measure of information loss. It assigns a penalty to each tuple based on the size of the equivalence class to which it belongs; larger classes imply higher indistinguishability and thus higher information loss [27]. However, by definition, DM calculates information loss solely based on group size, meaning it does not account for the magnitude of the generalization range. It assigns the same penalty to a compact cluster as it does to a widely dispersed one, provided they contain the same number of tuples.

Normalized Certainty Penalty (NCP): For streaming and numerical contexts where preserving the precision of values is critical, the Normalized Certainty Penalty (NCP) is preferred [21]. Unlike DM, NCP quantifies the precision lost by measuring the width of the generalization interval relative to the total domain range. This makes it particularly suitable for smart meter data, where the "spread" of the aggregated values directly impacts the granularity of the reading.

3.4 Summary and Research Gap

The literature review highlights that while privacy-preserving algorithms for data streams (such as z-anonymity) and distributed IoT architectures (Edge/Fog) are well-established individually, there remains a gap in understanding their integration.

Most existing studies focus on optimizing the algorithms themselves (e.g., reducing the complexity of clustering) or designing secure architectures. However, there is a lack of systematic analysis regarding the placement of specific preprocessing steps, such as prefiltering, generalization, or temporal aggregation, within a multi-layered architecture. Specifically, the trade-off between the privacy gained and the system utility lost when shifting these tasks from the Cloud to the Edge or Fog layers has not been comprehensively evaluated in the context of high-frequency smart metering. This thesis addresses this gap by experimentally evaluating these preprocessing strategies to determine the optimal architectural configuration.

4 Methodology

4.1 Dataset and Preprocessing

The experimental evaluation in this thesis utilizes the "SmartMeter Energy Consumption Data in London Households" dataset, provided by UK Power Networks as part of the Low Carbon London project [12]. The original dataset covers the period from November 2011 to February 2014 and contains approximately 167 million records from 5,567 representative households in the Greater London area.

4.1.1 Data Selection

Due to the computational volume of the full dataset, a representative time window was selected for the simulations. The experiments utilize a one-week period from November 5, 2012 (00:00:00) to November 11, 2012 (23:30:00). This subset allows for the analysis of both weekday and weekend consumption patterns while maintaining a manageable data volume for iterative testing. The selected subset contains 1,854,193 tuples. It includes readings from 5,529 unique households. Although the project lists 5,567 total participants, the discrepancy of 38 individuals is attributed to missing reports or device downtime during this specific week. As this represents less than 0.7% of the data population, it does not statistically impact the validity of the privacy analysis.

4.1.2 Preprocessing and Schema

Raw data contained tariff information (such as the stdorToU column indicating Standard or Dynamic Time-of-Use tariffs). As the focus of this research is on the privacy of consumption values rather than billing mechanisms, this column was removed during preprocessing. The final data schema used for the experiments is defined as a tuple $T=\langle ID,t,v\rangle$, where:

- **ID (LCLid):** A unique anonymized identifier for the smart meter (household).
- **t (DateTime):** The timestamp of the reading (30-minute granularity).
- **v (KWH/hh):** The energy consumption value in kilowatt-hours per half hour.

Prior to the experiments, the dataset was checked for missing values. No null entries were found in the selected one-week subset; therefore, no data cleaning regarding missing values was required, and the full sample of 1,854,193 tuples was used for the simulations.

4.2 System Model and IoT Architecture

To evaluate the impact of preprocessing on privacy, we model a hierarchical IoT architecture consisting of three distinct layers: the **Edge** (Smart Meters), the **Fog** (Gateways), and the **Central Entity** (Cloud). This model simulates the flow of energy consumption data from the household to the utility provider.

The Fog layer functions as an intermediate tier between the household and the utility provider. To simulate this topology, the total user population (N=5,529) is partitioned into 56 logical neighborhoods, with each subset reporting to a simulated Fog node (gateway). The role of this layer varies depending on the experimental scenario. In most cases, the gateways serve as passive relays, facilitating the transmission of data to the Central Entity. However, in the *Local Prefiltering* scenario, the gateways act as active processing points. In this specific configuration, each gateway executes a local privacy check to suppress unique values within its neighborhood, thereby reducing the volume of unique outliers transmitted upstream.

4.2.1 Layer 1: The Edge (Smart Meters)

The Edge layer forms the foundation of the architecture, serving as the primary data source. In this simulation, each unique household identifier (LCLid) functions as an independent Edge Node. Primarily responsible for generating timestamped energy readings, these nodes are modeled as resource-constrained devices. However, in scenarios involving distributed preprocessing, we assume the Edge possesses sufficient computational capability to perform lightweight scalar transformations. Specifically, in the *Data Generalization*, *Temporal Aggregation*, and *Local Prefiltering* experiments, the Edge node performs the initial data transformation, rounding values to a specific precision or computing temporal averages, before transmission. Crucially, the Edge defines the system's Trust Boundary. It is the only fully trusted zone in the architecture; data residing on the device is considered private, but once it crosses the network interface, it is treated as exposed unless anonymized.

4.2.2 Layer 2: The Fog (Gateways)

Situated between the household and the utility provider, the Fog layer functions as an intermediate aggregation point. To simulate this topology, the total user population

(N=5,529) is partitioned into 56 logical neighborhoods, with each subset reporting to a simulated Gateway.

The role of this layer varies depending on the experimental scenario. In the baseline and purely edge-based experiments, the gateways serve as passive relays. However, in the *Local Prefiltering* scenario, the Fog layer acts as an active privacy enforcer. In this hybrid configuration, the Gateway receives already generalized (rounded) data from its local Edge nodes and executes a local z-anonymity check. It suppresses any values that appear fewer than z_{local} times within that specific neighborhood, ensuring that only locally frequent values are forwarded to the Central Entity.

4.2.3 Layer 3: The Central Entity (Cloud)

The Central Entity represents the utility provider's backend infrastructure and serves as the final destination for all data streams. Its primary function is to collect incoming tuples from lower layers to construct a global snapshot of the energy grid at any given timestamp t. Within this privacy framework, the Central Entity is designated as the executor of the Global Snapshot z-Anonymity algorithm. It aggregates frequencies across the entire population and suppresses any tuple failing the condition $Count(v) \geq z$. From a security perspective, this entity is modeled as "honest-but-curious": while it is trusted to follow the protocol correctly, it is simultaneously the adversary from whom individual raw values must be concealed.

4.3 The Privacy Model: Snapshot z-Anonymity

The core privacy mechanism employed in this research is an adaptation of the z-anonymity algorithm proposed by Jha et al. [10].

4.3.1 Adaptation for Synchronous Streams

The original algorithm utilizes a sliding time window (Δt) to accumulate frequency counts from sporadic data streams. However, smart metering infrastructure operates in a synchronous manner, in which the entire population of devices reports consumption data at aligned intervals (e.g., every 30 minutes). To address this, this thesis implements a Spatial (or Snapshot) z-Anonymity model. Instead of maintaining a historical buffer for individual users, the algorithm evaluates the privacy of a tuple based on the frequency of its value across the entire reporting population at a specific timestamp t.

4.3.2 Formal Definition

Let P_t be the set of all tuples received by the Central Entity at timestamp t. A consumption value v is eligible for publication if and only if it satisfies the z-anonymity threshold:

$$Count(v, P_t) \geq z \tag{4.1}$$

where $Count(v, P_t)$ represents the number of households reporting the exact value v at time t.

Based on the privacy guarantees defined by Jha et al. [10], this implementation represents the z-anonymity requirement using a surplus publication mechanism. In a snapshot context, this ensures that the first $z - 1$ occurrences of a value are treated as a privacy buffer. If the threshold is met, the number of tuples actually published for value v, denoted as $N_{pub}(v, t)$, is calculated as:

$$N_{pub}(v, t) = Count(v, P_t) - (z - 1) \tag{4.2}$$

If $Count(v, P_t) < z$, then $N_{pub}(v, t) = 0$. This logic ensures that even in the presence of an adversary with partial knowledge of $z - 1$ individuals, the published data represents a "crowd" where no single individual's contribution can be definitively isolated.

4.4 Implementation of Experimental Scenarios

To verify the hypotheses, the simulation framework was developed using the **Python 3.13** programming language. The implementation leverages the scientific Python ecosystem to handle the large volume of smart meter data efficiently. Specifically, the **Pandas** library is utilized for high-performance data manipulation, employing vectorized operations to process the 1.8 million tuples without the performance overhead of iterative loops. Visualization of the resulting data distributions and metrics is handled by **Matplotlib** and **Seaborn**.

The codebase follows a modular object-oriented design pattern to ensure reproducibility. Rather than a monolithic script, each experimental scenario is encapsulated in a distinct, standalone Python class. While these classes do not share a common inheritance parent, they adhere to a standardized structural contract, implementing identical methods for data loading, processing, and metric calculation. This design ensures that all strategies are evaluated against the same criteria and produce a standardized dictionary of results. The simulations are orchestrated by a central runner script (`main.py`), which iterates through the parameter ranges and consolidates the outputs into a single CSV file for comparative analysis.

4.4.1 Scenario A: Baseline (Cloud-Only)

The first scenario serves as the control group, representing the traditional centralized "trusted third party" model. In this configuration, no preprocessing or data transformation occurs at either the Edge (Smart Meter) or the Fog (Gateway) layers. Implemented in the `BaselineExperiment` class, this simulation models Smart Meters transmitting raw, high-precision energy readings (three decimal places) directly upon generation. The intermediate Gateways at the Fog layer perform no filtering or local analysis, acting strictly as passive relays that forward the complete data stream to the Central Entity. Consequently, the global z-anonymity algorithm is executed entirely at the Cloud layer. This scenario establishes the benchmark metrics for publication ratio and bandwidth consumption. By identifying how much data is suppressed when raw, high-entropy values from the entire population are subjected to strict anonymity constraints, it provides a "worst-case" reference point against which the effectiveness of distributed preprocessing is measured.

4.4.2 Scenario B: Data Generalization (Edge)

In the second scenario, the privacy-preserving workload is shifted to the Edge layer, simulating a model where the Smart Meter sanitizes data before transmission. The primary objective is to increase the probability of satisfying the z-anonymity threshold by reducing the granularity of the consumption values. When the raw readings are less precise, the likelihood of multiple users reporting identical values at the same timestamp increases, thereby reducing data suppression.

This logic is implemented in the `GeneralizationExperiment` class. Before the privacy check, a scalar rounding transformation is applied to every energy reading v. To ensure that the data is mapped to its mathematically nearest representative value, the implementation uses the standard round-half-up formula:

$$v' = \frac{\lfloor v \cdot 10^p + 0.5 \rfloor}{10^p} \tag{4.3}$$

In this equation, p represents the target precision level (the number of decimal places). The inclusion of the $+0.5$ offset within the floor function $\lfloor \dots \rfloor$ is a deliberate architectural choice to perform rounding rather than simple truncation. Without this offset, the data would always be rounded down, leading to a significant systematic bias and higher information loss. By rounding to the nearest neighbor, the algorithm minimizes the error introduced during generalization, preserving as much utility as possible for the utility provider.

The simulation systematically evaluates three levels of generalization ($p \in \{0, 1, 2\}$) and compares them against the raw dataset resolution ($p = 3$). For $p = 0$, values are rounded to the nearest integer (e.g., 0.158 kWh becomes 0.0 kWh), whereas $p = 2$ retains a higher degree of granularity. The resulting quantization error and its impact on data utility are further analyzed using both the standard Normalized Certainty Penalty (NCP) and the refined Effective NCP metric, both of which are described in detail in Section 4.5.

4.4.3 Scenario C: Temporal Aggregation (Edge)

The third scenario evaluates the impact of reducing the temporal granularity of the data stream. Implemented in the `TemporalAggregationExperiment` class, this strategy simulates an Edge node that buffers multiple readings and transmits a single representative average over a fixed time window W. This approach contrasts with the baseline 30-minute reporting interval and is specifically designed to evaluate the trade-off between real-time monitoring and network efficiency.

The implementation utilizes the Pandas `resample` function to apply non-overlapping "tumbling" windows. For a set of n readings $\{v_1, v_2, \ldots, v_n\}$ captured within a window W, the Edge node calculates and transmits the arithmetic mean $\bar{v}$:

$$\bar{v} = \frac{1}{n} \sum_{i=1}^{n} v_i \tag{4.4}$$

The simulation evaluates window sizes of $W \in \{1H, 2H, 4H\}$, corresponding to $n \in \{2, 4, 8\}$ readings per transmission. This preprocessing step has a dual impact on the dataset:

1. **Data Volume:** It directly reduces the total row count of the transmitted dataset by a factor of n. This leads to significant *Bandwidth Savings* (as defined in Section 4.5), which is a critical performance objective in resource-constrained IoT environments.

2. **Value Transformation:** The process of averaging values over a time window aims to reduce the distinctness of usage peaks, theoretically making individual profiles less unique and more similar to the broader population. This research investigates whether this smoothing increases the publication ratio by making users appear more similar, or if the generation of new, high-precision arithmetic means actually increases the sparsity of the dataset, thereby harming data availability.

Unlike the Generalization scenario, which only reduces the precision of a single reading, Temporal Aggregation fundamentally alters the temporal resolution of the data, masking fine-grained behavioral patterns while providing a significant performance boost for the IoT architecture.

4.4.4 Scenario D: Local Prefiltering (Fog)

The final scenario evaluates a distributed privacy-preserving architecture that leverages the intermediate Fog layer. This logic is implemented in the `LocalPrefilteringWithGeneralization` class and represents a multi-stage hybrid approach to data protection.

To simulate a realistic network topology, the user population ($N = 5,529$) is logically partitioned into 56 neighborhoods. To ensure statistical consistency across the majority of the simulation, 55 of these neighborhoods are modeled as standard clusters of 100 households each. The final neighborhood consists of the remaining 29 households. This configuration allows the research to evaluate the preprocessing strategies across a near-uniform distribution while maintaining the integrity of the full real-world dataset. The processing pipeline consists of three distinct stages:

1. **Edge Level Generalization:** Prior to transmission to the gateway, each Smart Meter rounds its energy reading v to a fixed precision of $p = 2$ decimal places. This initial transformation is necessary to ensure that enough identical values exist within a small local neighborhood to satisfy a privacy threshold.

2. **Fog Level Prefiltering:** Each Gateway executes a local z-anonymity check on the generalized data received from its neighborhood. Following the surplus publication model, the Gateway identifies the frequency of each value v within its local subset U_i. A tuple is forwarded to the Central Entity only if it satisfies $Count(v, U_i) \geq z_{local}$. The number of tuples forwarded for a given value, N_{fwd}, is calculated as:

$$N_{fwd}(v, U_i) = \max(0, Count(v, U_i) - (z_{local} - 1)) \tag{4.5}$$

3. **Global Anonymity Check:** The Central Entity collects the surplus tuples from all gateways and performs the final global z-anonymity check (as defined in Section 4.3) before the data is considered fully anonymized and ready for publication.

This scenario evaluates the hypothesis that performing a preliminary privacy check at the Fog layer can significantly reduce the volume of "unique outliers" transmitted over the core network. By suppressing rare values close to the source, the architecture aims to maximize *Bandwidth Savings* while maintaining high global data availability for the Central Entity.

4.5 Evaluation Metrics

To quantify the trade-offs among privacy, utility, and performance across different architectural scenarios, three primary metrics are used.

4.5.1 Publication Ratio

The Publication Ratio (PR) measures the availability of data after the z-anonymity constraints have been applied. It is defined as the percentage of tuples that satisfy the privacy threshold and are released to the end-user or application:

$$PR = \frac{N_{published}}{N_{total}} \times 100\% \tag{4.6}$$

where $N_{published}$ is the count of tuples satisfying $Count(v, P_t) \geq z$, and N_{total} is the total number of tuples in the original dataset.

4.5.2 Bandwidth Savings

To evaluate the performance benefits of the Fog and Edge layers, we measure the reduction in data transmission volume. Bandwidth Savings (BS) is calculated as the percentage of messages suppressed or aggregated before reaching the Central Entity:

$$BS = \left(1 - \frac{N_{transmitted}}{N_{total}}\right) \times 100\% \tag{4.7}$$

In the baseline, BS is 0%, whereas in temporal aggregation and local prefiltering, this value represents the network load reduction.

4.5.3 Information Loss and Effective NCP

Information loss is measured using the Normalized Certainty Penalty (NCP). The standard NCP normalizes the width of a generalization interval $[L, U]$ against the global range of the attribute:

$$NCP_{std} = \frac{U - L}{max(v) - min(v)} \times 100\% \tag{4.8}$$

However, smart meter data is characterized by high skewness and extreme outliers. To prevent these rare high-consumption values from artificially inflating the denominator and "diluting" the perceived information loss, this research introduces a refined **Effective NCP (NCP_{eff})**. This metric evaluates utility loss relative to the range of the typical population. The "Effective Range" is determined through a robust statistical process:

1. The median consumption $\tilde{v}$ of the dataset is calculated.

2. For every reading v_i, the absolute deviation from the median is computed: $|v_i - \tilde{v}|$.

3. A cutoff distance D is identified as the 95th percentile of these deviations.

4. The dataset is filtered to include only the 95% of observations where $|v_i - \tilde{v}| \leq D$.

The NCP_{eff} uses the range of this filtered subset as the denominator. This ensures that the metric accurately reflects the impact of rounding on the vast majority of households, providing a more rigorous assessment of data utility than the standard global normalization.

5 Results

This chapter presents the experimental results of the proposed preprocessing strategies. The primary objective is to quantify the trade-off between privacy (Publication Ratio), data utility (NCP/Precision), and system performance (Bandwidth Savings) across the four architectural scenarios.

5.1 Analysis of Input Data Distribution

Before evaluating the privacy algorithms, it is essential to characterize the underlying distribution of the energy consumption data. The performance of z-anonymity is directly dependent on the frequency of duplicate values within a snapshot. Therefore, identifying where the "crowd" of users resides is critical for understanding the baseline suppression rates.

Figure 5.1 illustrates the distribution of energy readings for the selected week on a logarithmic scale. The data exhibits a heavy right-skewed distribution. While the dataset contains extreme outliers reaching up to 9.26 kWh, these represent a statistically small portion of the total observations.

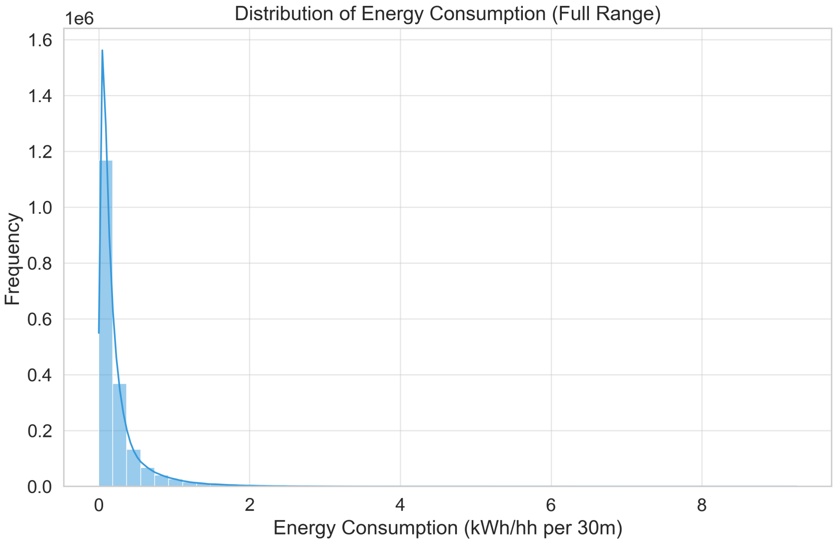

Figure 5.1: Distribution of raw energy consumption (Log Scale).

To better understand the behavior of the majority of the population, Figure 5.2 provides a focused view of the 0.0 to 1.0 kWh range. This interval is highly significant as it contains approximately 96.5% of all recorded tuples.

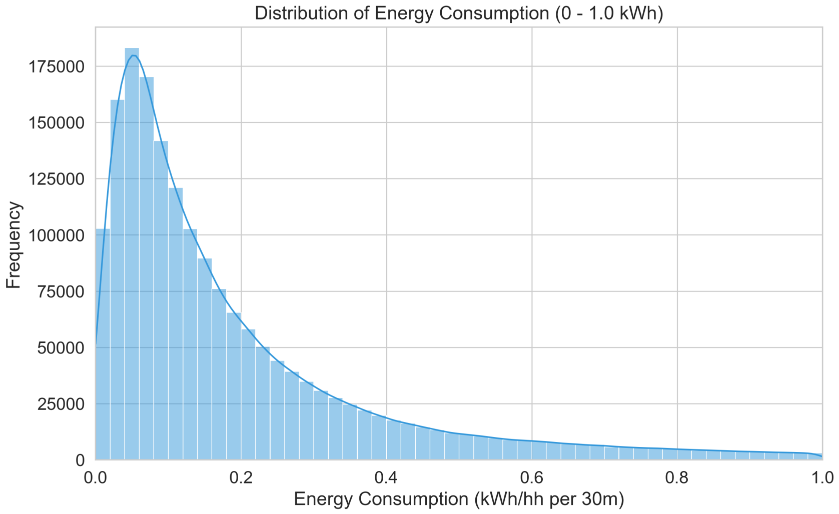

Figure 5.2: Detailed view of the 0-1.0 kWh range, where 96.5% of the population data resides.

The distribution reveals two primary characteristics:

1. **Low-Load Concentration:** A significant majority of readings are clustered between 0.05 kWh and 0.2 kWh. In this range, the high density of users suggests a higher probability of satisfying z-anonymity thresholds.

2. **High-Precision Sparsity:** Despite the concentration of users in the low-load range, the readings are recorded with three-decimal-point precision. Consequently, even small variations (e.g., 0.101 vs. 0.102 kWh) result in tuples being treated as unique values. Beyond the 0.5 kWh mark, the frequency of any specific raw value drops to near zero, indicating that most active usage records are mathematically unique within a 30-minute snapshot.

Note: I will update this section later with additional graphs showing daily usage patterns and the frequency of unique measurements at different times of the day.

5.2 Baseline Performance (Cloud-Only)

The Baseline scenario represents the centralized approach where raw data is transmitted directly to the Cloud before the privacy check is applied. This serves as the control group for the experiments.

Figure 5.3 depicts the Publication Ratio as a function of the privacy threshold z. The results indicate a rapid degradation of data availability as the privacy requirement increases.

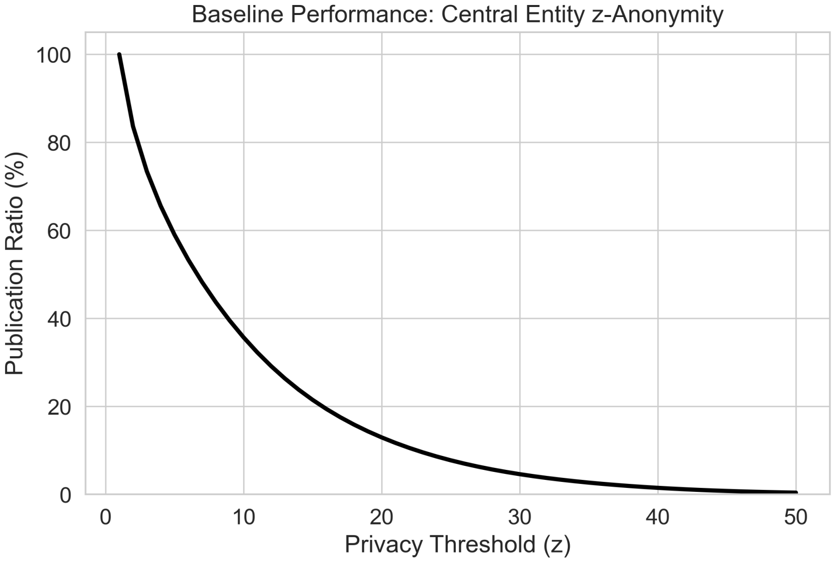

Figure 5.3: Baseline Performance: Publication Ratio vs. z.

Consistent with the sparsity observed in the data distribution analysis, the system struggles to maintain availability:

- **At $z = 2$:** The system publishes 83.6% of the tuples.
- **At $z = 10$:** The publication ratio drops to approximately 35.7%.
- **At $z = 50$:** The publication ratio falls below 0.4%.

Given the total population of 5,529 households, finding at least two matching consumption readings at a single timestamp is statistically probable, especially within the low-load ranges identified in Section 5.1. This explains the relatively high data availability at z=2. However, as the threshold increases toward z=50, the probability of finding such a large number of matching users in a single snapshot becomes nearly zero. The data resolution is too fine for such a dense crowd to form by coincidence, leading to the near-total suppression of the dataset.

Bibliography

[1] M. Farooq, M. Waseem, S. Mazhar, A. Khairi, and T. Kamal, "A Review on Internet of Things (IoT)," *International Journal of Computer Applications*, vol. 113, 2015. DOI: 10.5120/19787-1571

[2] M. Rasoulnia, E. Yaghoubi, E. Yaghoubi, A. Hussain, and I. Kamwa, "A comprehensive systematic and bibliometric review of technologies and measurement tools for power quality events detection, classification, and fault location in smart grids," *Renewable and Sustainable Energy Reviews*, vol. 226, 2026. DOI: 10.1016/j.rser.2025.116302

[3] P. Siano, "Demand response and smart grids—A survey," *Renewable and Sustainable Energy Reviews*, vol. 30, 2014. DOI: 10.1016/j.rser.2013.10.022

[4] G. Strbac, "Demand side management: Benefits and challenges," *Energy Policy*, vol. 36, 2008. DOI: 10.1016/j.enpol.2008.09.030

[5] C. Beckel, L. Sadamori, T. Staake, and S. Santini, "Revealing household characteristics from smart meter data," *Energy*, vol. 78, 2014. DOI: 10.1016/j.energy.2014.10.025

[6] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring Personal Information from Demand-Response Systems," *IEEE Security & Privacy*, vol. 8, 2010. DOI: 10.1109/MSP.2010.40

[7] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/ec (General Data Protection Regulation) (Text with EEA relevance)*, 2016.

[8] A. Narayanan and V. Shmatikov, "Robust De-anonymization of Large Sparse Datasets," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, 2008. DOI: 10.1109/SP.2008.33

[9] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, 2002. DOI: 10.1142/S0218488502001648

[10] N. Jha, T. Favale, L. Vassio, M. Trevisan, and M. Mellia, "Z-anonymity: Zero-Delay Anonymization for Data Streams," 2020. DOI: 10.1109/BigData50022.2020.9378422

[11] C. Brunn, "Dezent: Decentralized z-Anonymity with Privacy-Preserving Coordination," 2025.

[12] U. P. Networks, *SmartMeter Energy Consumption Data in London Households*, 2014.

[13] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, 2016. DOI: 10.1109/JIOT.2016.2579198

[14] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, Hangzhou, China: Association for Computing Machinery, 2015. DOI: 10.1145/2757384.2757397

[15] Y. Wang, Y. Li, D. Qin, and H. V. Poor, "Introducing Edge Intelligence to Smart Meters via Federated Split Learning," *Nature Communications*, 2024.

[16] L. Sweeney, "Simple Demographics Often Identify People Uniquely," 2018. DOI: 10.1184/R1/6625769.v1

[17] K. Guo and Q. Zhang, "Fast clustering-based anonymization approaches with time constraints for data streams," *Knowledge-Based Systems*, vol. 46, 2013. DOI: 10.1016/j.knosys.2013.03.007

[18] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, 2014. DOI: 10.1561/0400000042

[19] B. C. M. Fung, K. Wang, and P. S. Yu, "Top-Down Specialization for Information and Privacy Preservation," in *Proceedings of the 21st International Conference on Data Engineering*, USA: IEEE Computer Society, 2005. DOI: 10.1109/ICDE.2005.143

[20] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional K-Anonymity," in *22nd International Conference on Data Engineering (ICDE'06)*, 2006. DOI: 10.1109/ICDE.2006.101

[21] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, "Utility-based anonymization for privacy preservation with less information loss," *SIGKDD Explor.*, vol. 8, 2006.

[22] A. Meyerson and R. Williams, "On the complexity of optimal k-anonymity," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2004.

[23] J. Cao, B. Carminati, E. Ferrari, and K.-L. Tan, “Castle: Continuously Anonymizing Data Streams,” *IEEE Transactions on Dependable and Secure Computing*, vol. 8, 2011. DOI: `10.1109/TDSC.2009.47`

[24] H. Zakerzadeh and S. L. Osborn, “FAANST: Fast anonymizing algorithm for numerical streaming data,” in *Proceedings of the 5th International Workshop on Data Privacy Management, and 3rd International Conference on Autonomous Spontaneous Security*, Athens, Greece: Springer-Verlag, 2010.

[25] G. Zhong and U. Hengartner, “A distributed k-anonymity protocol for location privacy,” *2009 IEEE International Conference on Pervasive Computing and Communications*, 2008.

[26] R. Bayardo and R. Agrawal, “Data privacy through optimal k-anonymization,” in *21st International Conference on Data Engineering (ICDE’05)*, 2005. DOI: `10.1109/ICDE.2005.42`

[27] C. Ni, L. S. Cang, P. Gope, and G. Min, “Data anonymization evaluation for big data and IoT environment,” *Information Sciences*, vol. 605, 2022. DOI: `10.1016/j.ins.2022.05.040`