Pacific Northwest
NATIONAL LABORATORY

PNNL-37723

# Differential Privacy in Grid Kitchen

## Implementation & Software Documentation

March 2026

Kaustav Bhattacharjee
Kapil Duwadi
Alex A. Anderson
Alka Singh

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
Under contract DE-AC05-76RL01830

## DISCLAIMER



PACIFIC NORTHWEST NATIONAL LABORATORY
*operated by*
BATTELLE
*for the*
UNITED STATES DEPARTMENT OF ENERGY
*under Contract DE-AC05-76RL01830*

**Printed in the United States of America**

**Available to DOE and DOE contractors from**
**the Office of Scientific and Technical Information,**
**P.O. Box 62, Oak Ridge, TN 37831-0062**
**www.osti.gov**
**ph: (865) 576-8401**
**fox: (865) 576-5728**
**email: reports@osti.gov**

**Available to the public from the National Technical Information Service**
**5301 Shawnee Rd., Alexandria, VA 22312**
**ph: (800) 553-NTIS (6847)**
**or (703) 605-6000**
**email: info@ntis.gov**
**Online ordering: http://www.ntis.gov**

PNNL-37723

# Differential Privacy in Grid Kitchen

Implementation & Software Documentation

March 2026

Kaustav Bhattacharjee
Alex A. Anderson
Kapil Duwadi
Alka Singh

Prepared for
the U.S. Department of Energy
Under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

# Executive Summary

Sharing of power grid feeder models faces significant challenges due to the potential risk of exposing sensitive operational information. Traditional anonymization techniques have shown notable limitations in other sensitive domains, as evidenced by documented re-identification attacks that combine supposedly anonymized datasets with auxiliary information, raising concerns that similar vulnerabilities could affect power grid data. Consequently, there is a pressing need for a more rigorous privacy protection strategy that not only delivers formal mathematical guarantees but also preserves the analytical value of the shared models.

To address this challenge, we have enhanced the Grid Kitchen framework by implementing differential privacy mechanisms within the distribution model dehydration pipeline. This implementation carefully calibrates and applies noise to sensitive attributes in feeder models according to configurable privacy levels—low, moderate, and high—each offering different balances between data utility and privacy protection. Our approach uses established noise functions (Gaussian for continuous data and Discrete Laplace for integer values) with parameters carefully calibrated so that the impact of individual data points is effectively masked in the final output.

The integration leverages our Noise Catalog, which we developed to categorize feeder model properties by component type, data type, and sensitivity. This catalog guides the application of appropriate noise functions and privacy parameters ($\varepsilon$ and $\delta$) to each attribute, ensuring consistent privacy protection across the model while maintaining its structural integrity and analytical usefulness. This implementation also includes evaluation tools that allow model owners to assess the impact of privacy-preserving transformations before sharing data with external parties.

This report provides documentation for the differential privacy capabilities added to the Grid Kitchen project. It includes a primer on differential privacy concepts and their importance in modern data sharing, details the architecture of our implementation, explains the privacy modes and parameter configurations, and offers practical guidance on using the code for applying differential privacy to grid feeder models. Through examples and code snippets, we demonstrate the effective application of these privacy-enhancing technologies, enabling utility operators and researchers to confidently share grid data while protecting sensitive information.

# Acknowledgments

This project was supported by the Department of Energy, Office of Electricity, Advanced Grid Research and Development Program. The authors would like to thank Chris Irwin for his support and contributions to shaping the scope and direction of this work.

# Contents

# Figures



# Tables

# 1.0 Introduction

Differential privacy provides a robust mechanism for balancing data utility with privacy guarantees when sharing sensitive information like grid feeder models. In this section, we introduce the fundamental concepts of differential privacy, explain its advantages over traditional anonymization techniques, and examine the inherent trade-offs between privacy protection and analytical utility that inform our implementation choices.

## 1.1 What is Differential Privacy?

Differential privacy is a rigorous mathematical framework designed to provide strong privacy guarantees when analyzing and sharing information derived from sensitive datasets. It ensures that the removal or addition of a single individual's data does not significantly affect the outcome of any analysis, thereby limiting the risk of unintended disclosure. In formal terms, a randomized algorithm $A$ is said to satisfy ($\varepsilon$, $\delta$)-differential privacy if for all neighboring databases $D$ and $D'$ (which differ in one individual's data) and for every set $S$ of outputs, the following condition holds [1]:

$$\mathbb{P}[A(D) \in S] \leq e^{\varepsilon} \cdot \mathbb{P}[A(D') \in S] + \delta \tag{1}$$

This definition mathematically encapsulates the idea that the algorithm's output does not substantially change whether any one individual's data is included or not.

The importance of differential privacy lies in its ability to provide data analysts and institutions with a quantifiable, mathematically provable measure of privacy. With increasing concerns about data breaches, misuse of personal data, and the re-identification of anonymized datasets, differential privacy offers a robust solution that balances valuable insights from data with the protection of individual privacy. Its framework allows organizations to make informed decisions about the trade-off between the accuracy of the information released and the privacy of the individuals whose data is being analyzed.

Easy-to-understand examples include scenarios like publishing statistical summaries from large databases—such as the average income or the count of people with a specific characteristic. For instance, in a simple query that calculates the average salary, adding carefully calibrated noise ensures that an individual's contribution is masked. One common method for achieving this is the Laplace mechanism, which adds noise drawn from the Laplace distribution. Mathematically, if $f(D)$ is a query function with global sensitivity $\Delta f$ (the maximum change in $f$ caused by modifying one individual's data), the released result can be computed as:

$$f(D) + \text{Lap}\left(\frac{\Delta f}{\varepsilon}\right) \tag{2}$$

Here, Lap $\left(\frac{\Delta f}{\varepsilon}\right)$ represents the noise from the Laplace distribution scaled by $\frac{\Delta f}{\varepsilon}$, ensuring that the privacy guarantee holds. By controlling $\varepsilon$, practitioners can tune the trade-off between privacy (stronger privacy with a smaller $\varepsilon$) and accuracy of the result.

## 1.2 Why is it used?

Differential privacy is used primarily to ensure that organizations can extract valuable insights from large datasets while safeguarding individual privacy. In today's data-driven world, the risk of re-identification from aggregated or anonymized datasets is high due to the availability of

auxiliary information. Differential privacy provides a formal framework that quantifies the privacy risk by guaranteeing that the inclusion or exclusion of a single data point does not significantly affect the overall outcome. This is essential in contexts such as healthcare, finance, and social research, where sensitive data is frequently analyzed.

One notable application of differential privacy is within the U.S. Census Bureau's Census data calculations, where revealing detailed population statistics must be balanced against the risk of identifying individuals [2]. In this setting, differential privacy is employed to protect the census respondents while still enabling the release of high-quality population data. For example, randomized noise may be added to the counts of individuals in various geographic subdivisions, ensuring that small shifts in individual participation do not lead to predictable differences in published figures. This allows statistical agencies to fulfill their mandate of transparency and public accountability while rigorously safeguarding personal information.

The concept of differential privacy can be intuitively understood through the "coin flip" metaphor. Imagine a researcher who wants to ask individuals a sensitive question (e.g., "Have you ever falsified your tax return?"). Instead of answering directly, respondents flip a coin: if it's heads, they answer truthfully; if it's tails, they flip a second coin and answer "Yes" if heads or "No" if tails. This randomized response technique gives individuals "plausible deniability" since the researcher cannot determine whether any specific "Yes" response came from the coin flip or an honest answer. Differential privacy formalizes and extends this concept, providing mathematical guarantees of plausible deniability that protect individuals while allowing accurate aggregate statistics to be computed—precisely why it has become the gold standard for privacy protection in modern data analysis.

In the absence of differential privacy, several traditional techniques have been employed, such as $k$-anonymity [3], [4], $l$-diversity [5], and $t$-closeness [6]. For example, $k$-anonymity achieves privacy by grouping records so that each record is indistinguishable from at least $k-1$ others, yet it is vulnerable to homogeneity attacks and adversaries with external background knowledge. Similarly, $l$-diversity and $t$-closeness attempt to address these weaknesses by ensuring diversity in sensitive attributes and limiting the distance between the distribution of sensitive values in any group and the overall dataset, respectively; however, these methods can still be compromised when the underlying distributions are skewed or when attackers leverage subtle discrepancies in the data [7]. While synthetic data generation and secure multi-party computation have also been proposed to fortify privacy, they, too, come with inherent limitations in terms of robustness and lack the formal, quantifiable guarantees provided by differential privacy.

Before the advent of differential privacy, privacy protection methods largely relied on statistical disclosure controls and ad hoc anonymization techniques. While these methods provided a first line of defense, they did not offer a consistent measure of risk and could not satisfactorily address the privacy challenges posed by modern data analytics. The balance between maintaining data utility and enforcing privacy restrictions is delicate, and traditional approaches often fall short in striking this balance in a rigorous, mathematical manner.

The seminal work that laid the foundation for differential privacy can be traced back to the influential paper by Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith published in 2006 [8]. Their work, often regarded as the starting point of differential privacy research, introduced the critical concept of "privacy loss" and provided the mathematical framework now standard in the field. Additionally, Cynthia Dwork's earlier research further refined these ideas, opening up an entirely new approach to privacy-preserving data analysis [9]. These pioneering papers have significantly influenced both academic research and practical applications, setting new standards for data privacy in an increasingly interconnected world.

## 1.3 On the balance between privacy and utility

Differential privacy inherently involves a delicate balance between privacy protection and the utility of the data being analyzed. Several foundational works in data privacy and statistical disclosure limitation have investigated this trade-off, emphasizing the need for any privacy-preserving mechanism to maintain a level of statistical accuracy that renders the data useful for analysis [1], [8], [10]. In traditional data protection approaches, researchers have explored methods that optimize this balance, often comparing the loss in utility against potential risks of individual re-identification. For example, Sweeney (2002) demonstrated a dramatic re-identification risk when she successfully linked ”anonymized” medical records to the Governor of Massachusetts by combining them with publicly available voter registration data, highlighting the inadequacy of simple anonymization techniques [3]. Subsequent research has advanced various approaches to optimize the privacy-utility trade-off, including synthetic data generation methods that preserve statistical properties [11], practical algorithms for privacy-aware machine learning [12], and techniques for data-dependent calibration of noise based on dataset characteristics [13], all aiming to minimize utility loss while maintaining strong privacy guarantees. Although early research in the statistical community did not provide a formal framework like differential privacy, it underscored the importance of maintaining acceptable levels of utility in published data.

In the context of differential privacy, the privacy-utility trade-off is more explicitly managed by carefully tuning parameters such as $\varepsilon$ (epsilon) and $\delta$ (delta). A smaller $\varepsilon$ represents a stronger privacy guarantee, but it also necessitates the addition of more noise to the data, which in turn decreases utility. Similarly, when $\delta$ is set to a nonzero value—as in the approximate differential privacy scenario—a certain relaxation of privacy guarantees is allowed, which can sometimes enable higher utility compared to the pure differential privacy setting (where $\delta = 0$). Researchers have extensively explored these trade-offs, offering guidelines and theoretical bounds that help practitioners determine acceptable compromises based on the requirements of their particular applications.

The impact of this trade-off can be illustrated with a few examples. For instance, consider a query that counts the number of individuals meeting a certain criterion in a census dataset. Applying strict differential privacy with a very low $\varepsilon$ would require injecting substantial noise into the result, potentially rendering the count less reliable for policy analysis. On the other hand, relaxing the privacy constraint by opting for a larger $\varepsilon$ (or introducing a small $\delta$) could yield a much more accurate count, but at the cost of slightly increased privacy risk. In another example, a healthcare provider releasing summary statistics on disease prevalence might face similar choices: higher utility is critical for informed clinical research, yet patient privacy must be preserved, demanding a carefully chosen balance.

These trade-offs are not driven solely by a theoretical interest; they often reflect real-world policy decisions. For example, the U.S. Census Bureau’s adoption of differential privacy in the 2020 Census sparked considerable debate [14], [15]. Some stakeholders argued that the noise introduced to protect individual privacy reduced data utility, particularly for small geographic areas where accurate counts are vital. In scenarios like these, some may opt for lower privacy guarantees to achieve higher utility, especially when the benefits of precise data for public policy or targeted interventions can be very significant.

Within differential privacy mechanisms, the choice between the Laplace and Gaussian methods underscores these trade-offs further. The Laplace mechanism, generally used in pure differential privacy ($\delta = 0$), adds noise proportionate to the global sensitivity of the function divided by $\varepsilon$. This approach ensures robust privacy guarantees but may introduce more significant perturbations when high accuracy is needed. Alternatively, the Gaussian mechanism is typically employed under an approximate differential privacy model ($\delta > 0$), allowing for a

more flexible balance in cases where multiple queries or compositions are involved. The Gaussian approach often results in tighter overall error bounds and improved utility in practical settings, although it requires accepting a slight relaxation in privacy.

Ultimately, the interplay between privacy and utility is central to the practical implementation of differential privacy. Decision makers must evaluate whether the benefits of higher data accuracy outweigh the risks associated with reduced privacy; in many cases, the context of the application will dictate this trade-off [16]. For instance, a policymaker might accept somewhat lower privacy guarantees if it means obtaining data that can more accurately inform critical decisions, while organizations handling highly sensitive personal data might prioritize privacy at the expense of some utility. Understanding and managing these trade-offs through mechanisms like the Laplace and Gaussian methods is essential for developing systems that responsibly balance individual privacy with the actionable insights derived from data in contexts such as power grid feeder models. This balance is especially critical in environments where operational precision is paramount, and it sets the stage for the next section, which details the implementation of differential privacy in Grid Kitchen.

## 1.4 What is Grid Kitchen?

The increasing complexity of modern power networks has led to a critical challenge: the lack of accessible, representative distribution system models. Many power utility companies (also referred to as “utilities” in this text, to avoid confusion with data utility in privacy contexts) maintain their operational and customer data in isolated silos out of concern for privacy breaches, which restricts collaborative research and robust analysis [17]. This problem is further compounded by the diversity of modeling practices among over $2,000$ electric distribution utilities. Moreover, the inability to access comprehensive models severely hampers critical simulation and analysis tasks, such as resilience assessments, hosting capacity evaluations, and coordinated transmission and distribution planning. Thus, it has become increasingly apparent that a unified platform capable of aggregating representative models while concurrently safeguarding sensitive information is required.

Grid Kitchen is envisioned as a solution to bridge the gap between data availability and the practical requirements of robust power system analysis. By creating a centralized repository of both prototypical and synthetic distribution models, Grid Kitchen aims to serve researchers, industry professionals, and utilities alike. This initiative is designed to capture the heterogeneity found in utility systems, spanning a wide range of operational characteristics and network topologies. In doing so, the Kitchen provides a critical resource for varied analyses—from resilience assessments to operational strategy evaluations. Moreover, it harnesses both real-world utility data and algorithmically generated models to deliver comprehensive coverage of diverse distribution scenarios. This dual approach not only fills the existing data void but also fosters the development of innovative analysis techniques that can inform future planning and operational improvements.

Our plan leverages different methodologies to collect, synthesize, and securely share utility data. The process involves targeted outreach to utilities to acquire reference distribution feeder models, which are then transformed using state-of-the-art model obfuscation techniques. One key tool in this transformation is the Python package currently being developed to enable utilities to safely modify and share their network models without the need for additional Non-Disclosure Agreements (NDAs). This approach ensures that each model remains both useful for large-scale analysis and sufficiently masked to protect sensitive operational and customer data. By carefully preserving critical network characteristics while anonymizing sensitive details, our methodology maintains the analytical integrity of the models for simulation and planning

initiatives. Moreover, the process incorporates standardization protocols that facilitate seamless integration of diverse utility representations into broader analytical frameworks.

Differential privacy stands at the core of our strategy to address privacy concerns while maximizing data utility. By adding carefully calibrated noise to critical metrics, differential privacy provides a formal mathematical guarantee against the exposure of sensitive details. This rigorous approach not only preserves the analytical value of power system parameters but also instills confidence among utilities regarding the safety of their data. The precision with which differential privacy can quantify privacy risk offers transparency in the privacy-utility trade-off, making it particularly suitable for sensitive grid data. As we delve into the next sub-section, readers will discover why this approach has been chosen, highlighting its capacity to balance robust privacy protections with the essential needs of modern grid analysis.

## 1.5 Why this approach?

A significant challenge in harnessing utility data is the well-founded reluctance of utilities to share their distribution models. Many utilities keep their data securely within silos because of the fear that any data release could lead to unintended privacy breaches, potentially exposing sensitive or operational details [17], [18]. This reluctance naturally impedes collaboration and limits the breadth of analysis that can be conducted on real-world network behaviors. The siloed nature of these data repositories particularly hampers critical resilience analysis studies, which require diverse, real-world grid configurations to develop and validate robust methodologies against extreme weather events and cyber-physical threats [19], [20]. Furthermore, research on optimal energy storage deployment—crucial for managing increasing renewable penetration—suffers from inadequate access to representative distribution network models, resulting in suboptimal planning decisions and missed opportunities for system-wide benefits [21], [22]. These limitations extend to numerous other areas, including hosting capacity analysis for distributed energy resources and advanced control strategy development, where the absence of diverse, realistic test cases forces researchers to rely on oversimplified or synthetic models that may not capture critical real-world complexities [23]–[25]. Addressing these concerns is critical; without a method to protect proprietary and customer-related information, utilities are understandably cautious about sharing even de-identified data.

Differential privacy has been adopted as an essential component in our approach precisely to mitigate these concerns. By design, it ensures that the output from data analysis is statistically robust while preventing the leakage of any single utility's specific details. In this framework, even if an adversary were to access the published or shared models, they would be unable to reverse-engineer the sensitive aspects of the original datasets. The strength of differential privacy lies in its formal mathematical foundations, offering quantifiable privacy guarantees rather than the qualitative assurances of traditional anonymization techniques. At its core, differential privacy operates through the carefully calibrated addition of noise to query results, with the magnitude of this noise precisely determined by both the sensitivity of the query and a predetermined privacy budget ($\varepsilon$). Unlike traditional anonymization methods such as $k$-anonymity or data masking—which can be vulnerable to linkage attacks or mosaic effects when auxiliary information is available—differential privacy offers provable bounds on the probability of information leakage. This principled noise addition transforms deterministic outputs into probabilistic ones, ensuring that the presence or absence of any individual entity's data has a mathematically limited impact on analysis results. In the context of power systems, this means that critical network characteristics—such as load profiles, equipment specifications, and topological features—can be shared while maintaining a quantifiable level of indistinguishability between the actual and adjacent possible network configurations. The ability

to tune this privacy-utility trade-off through explicit parameters provides grid stakeholders with unprecedented control over how their sensitive operational data contributes to collaborative research initiatives. We believe that this mathematical robustness, inherent in differential privacy's framework, will instill trust among utilities, reassuring them that their data, when processed through our Grid Kitchen pipeline, will remain protected against the exposure of individual and operational details while still enabling valuable collaborative analysis.

The implementation of differential privacy in Grid Kitchen also opens up new avenues for collaborative data sharing. Utilities can be invited to use our tool to safely transform their existing feeder models, modify data where appropriate, and share obfuscated versions without the need for extensive non-disclosure agreements. This process not only streamlines data contribution but further enhances the repository by ensuring that every data point meets our stringent privacy criteria. Ultimately, this approach will pave the way for broader engagement among utilities, researchers, and other stakeholders, fostering an ecosystem of trust and innovation in grid analysis and planning.

# 2.0 Differential Privacy Implementation in Grid Kitchen

Differential privacy has been methodically incorporated into the Grid Kitchen framework, strategically positioning privacy-preserving mechanisms within the existing feeder model processing pipeline. This design provides flexible, configurable levels of protection while maintaining appropriate data utility for various analytical needs. In this section, we discuss our approach from architectural design through privacy mode configurations and systematic noise application via the Noise Catalog, culminating in practical code implementation guidelines that enable users to effectively deploy and evaluate differential privacy in their grid data workflows.

## 2.1 Overview of Implementation Strategy

In the Grid Kitchen project, the integration of differential privacy is strategically positioned as the final stage of the pipeline (Figure 1). This decision was made after extensive evaluation of where the privacy layer would have the most impact without disrupting earlier data transformation processes, ensuring that all raw attribute metrics are fully refined by the time noise is added. At the heart of our differential privacy implementation, dehydrated models generated from prior processes are ingested and prepared for the privacy transformation, thereby leveraging the mature output and avoiding processing redundancy. Figure 1 illustrates the entire flow from raw data input through to dehydrated model output, helping stakeholders understand how differential privacy fits into the broader architecture. By placing the differential privacy step at the end, the system capitalizes on the existing modular architecture for easier maintenance and future enhancements. This design choice also minimizes redundant handling of data while ensuring that the subsequent noise addition complies with privacy policies and requirements.

Upon ingesting the dehydrated models, the system enumerates all attribute metrics contained within the dataset. This initial step is essential for identifying the specific data points that will undergo differential privacy noise injection. By exhaustively listing these metrics, the process achieves a high level of granularity, enabling the assignment of customized privacy parameters to individual or grouped attributes. The resulting inventory, encapsulated in the Noise Catalog, supports a rational decision-making procedure about which data elements warrant stricter privacy measures and which may tolerate lower settings. Moreover, this evaluation helps to verify that no sensitive information remains unprotected prior to the application of noise.

Once all metrics are recorded, the subsequent phase involves determining the appropriate privacy level to be applied. Four distinct options—low, moderate, high, and a custom mode—offer the flexibility necessary to accommodate various regulatory and analytical requirements. This selection process allows system operators to balance data utility with privacy needs in a context-specific manner, addressing a common shortcoming of one-size-fits-all differential privacy approaches. The structured categorization of privacy modes also paves the way for more detailed discussions on their individual configurations in the subsequent sections.

Following the determination of privacy level, the system applies differential privacy noise to the dataset according to the chosen parameters. Noise injection is calibrated using established metrics such as sensitivity, epsilon ($\varepsilon$), and delta ($\delta$) to ensure robust privacy guarantees while maintaining data utility. We developed the Noise Catalog that provides a direct mapping from selected privacy levels to corresponding noise distributions, thereby ensuring consistent and repeatable modifications across all metrics. Alternative approaches, such as dynamic real-time noise adjustment, were evaluated; however, employing the Noise Catalog offers a robust, standardizable, and auditable framework. This structured approach aligns with differential

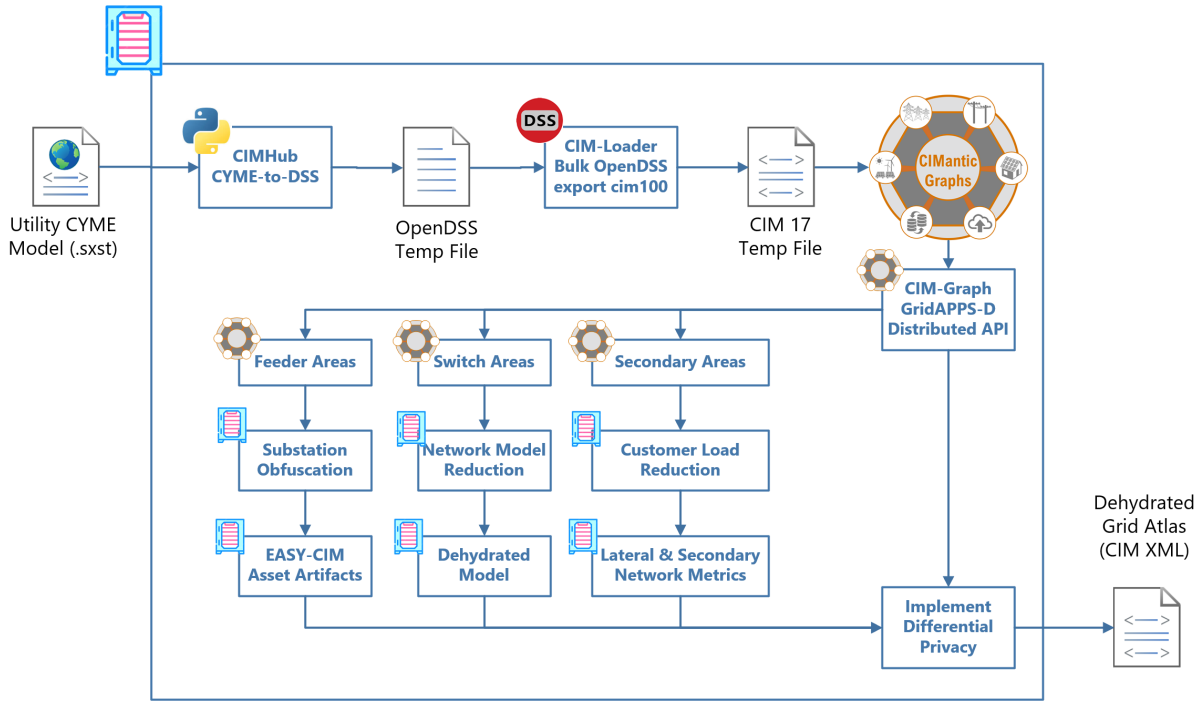


Figure 1: Schematic overview of the pipeline, illustrating differential privacy as the final processing stage before the generation of the dehydrated Grid Kitchen model.

privacy implementation guidelines that emphasize systematic parameter selection and documentation [1].

After noise is applied to attribute metrics, the resultant data passes through a robust verification layer designed to check the accuracy of both privacy guarantees and data utility. This verification phase is emphasized during testing and validation, where the modified data is compared against established statistical baselines to detect deviations caused by excessive noise or misconfigured privacy parameters. It functions as an error-checking mechanism that can trigger alerts or corrective actions when data fidelity is compromised. In doing so, this layer enforces a vital system of checks and balances, aligning with modern data governance practices [26]. While currently integral to quality assurance, it is anticipated that this verification layer may eventually be phased out in future releases of `Gridmeta` once the underlying processes have been fully validated and alternative methods are incorporated.

Once data verification is successfully completed, the system generates the final output: a dehydrated and differentially private feeder model dataset. This final dataset is essential for downstream analysis within the Grid Kitchen environment, ensuring that privacy persists through all subsequent processing and reporting stages. Figure 2 illustrates the full data transformation—from attribute metrics categorization to the production of a verified, privacy-enhanced feeder model. The modular architecture, with distinct stages for model dehydration, noise application, and verification, not only promotes clarity but also enables simpler troubleshooting and iterative improvements. In this manner, the final output encapsulates the dual benefits of rigorous data preparation alongside robust privacy protection.

Throughout the design and implementation of this process, multiple architectural choices

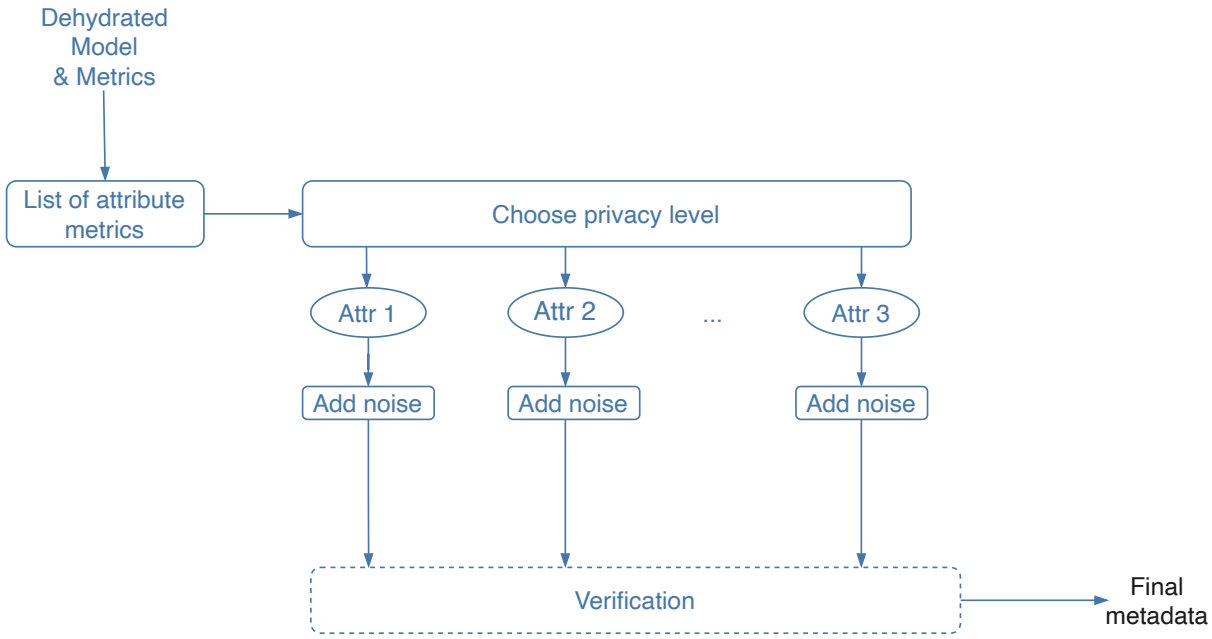


Figure 2: Overview of the Differential Privacy task flow in Grid Kitchen.

were carefully evaluated. One alternative approach considered was integrating differential privacy earlier in the pipeline, perhaps during the raw data processing phase. However, this alternative was set aside due to the potential for increased complexity and the risk of propagating noise through subsequent processing layers. Another option considered was the use of a unified privacy management module that handled side-by-side operation with data transformation steps, but this was found to reduce the modularity and traceability of the process. By isolating the differential privacy implementation at the end of the pipeline, the system benefits from clarity of purpose, ease of updates, and a strong audit trail—a design philosophy that conforms with recommendations in the literature [1].

In summary, this section describes the integration of differential privacy within the final stage of the pipeline in the Grid Kitchen project. The architecture ingests dehydrated models, catalogs all relevant attribute metrics, and allows for flexible privacy level selection. Differential privacy noise is then methodically applied, verified, and ultimately encapsulated in a final, privacy-preserving output model. The decision to maintain a modular, end-stage approach not only enhances maintainability and auditability but also provides a clear pathway for future modifications and scalability. While more detailed discussion on the privacy modes—low, moderate, high, and custom—will be provided in the next subsection, this overview sets the stage for understanding the essential architectural choices and trade-offs that informed our strategy.

## 2.2 Privacy Modes and Parameter Configuration

In this subsection, we present the privacy modes and parameter configuration for our differential privacy implementation in Grid Kitchen. Our system incorporates three predefined privacy modes—low, moderate, and high—as well as a custom mode that will allow users to supply

their own parameters through a JSON configuration. Each of these modes is designed to balance the trade-off between data utility and privacy protection. For each mode, differential privacy is achieved through the careful calibration of the key parameters epsilon ($\varepsilon$) and delta ($\delta$). Figure 3 shows an infographic summarizing the configuration of different privacy modes along with their respective trade-offs, providing a clear visualization of our approach.

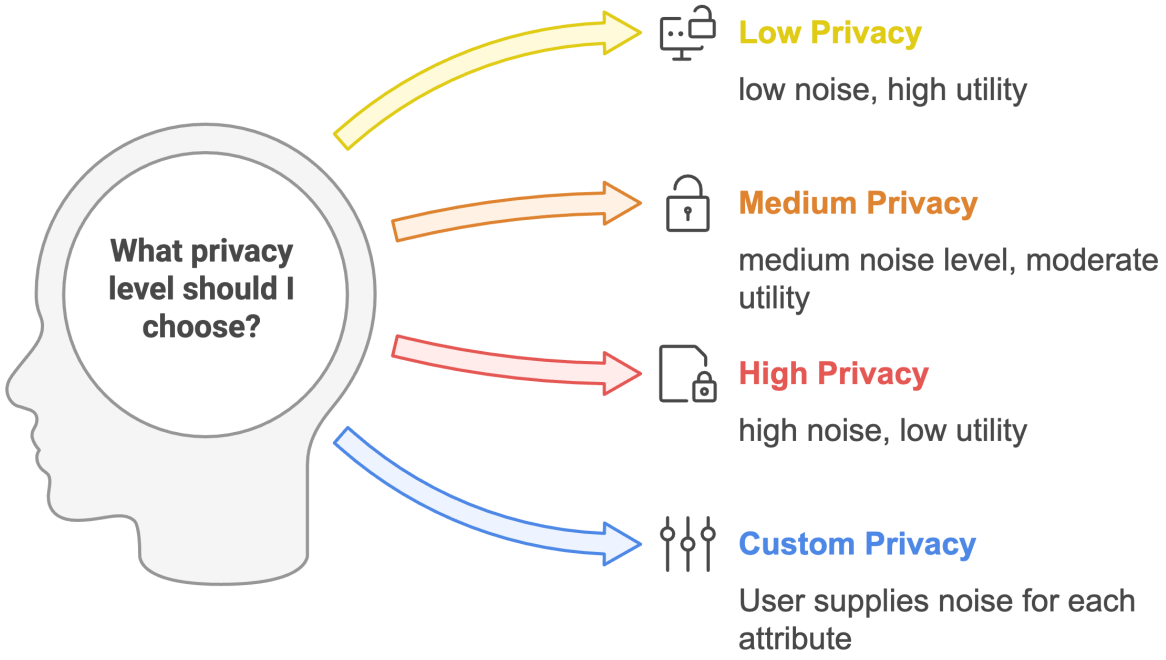


Figure 3: Different privacy modes and the corresponding trade-offs.

The low privacy mode is engineered to maintain higher data utility by imposing minimal noise on the feeder model outputs. In this mode, $\varepsilon$ is chosen to be 1, and $\delta$ is assigned a value of $10^{-5}$. The larger value of $\varepsilon$ in this mode means that less noise is added, thereby preserving more of the data's original characteristics. For instance, consider a running example with common parameters such as the transformer *count* and the average percent peak loading (*avg_pct_peak_loading*). For count parameters, the sensitivity is standardized at $1$, while for percentage measures like *avg_pct_peak_loading*, the sensitivity is set to $0.1$. Suppose the transformer count is initially **3** and *avg_pct_peak_loading* is **72.20**%. For the *count* value using the Discrete Laplace mechanism with $\varepsilon = 1$, the probability parameter is $p = e^{-1/1} \approx 0.368$; assuming a small noise sample of $+1$ is obtained, the final noisy *count* would be $3 + 1 =$ **4**. For *avg_pct_peak_loading*, which uses the Gaussian mechanism, the necessary standard deviation is calculated as $\sigma = (0.1 \times \sqrt{2 \cdot ln(1.25/10^{-5})})/1$; this evaluates to $\sigma \approx 0.4845$. If the Gaussian noise sampled is approximately $0.30$, then the final value becomes $72.20 + 0.30 =$ **72.50%**. These modest adjustments clearly demonstrate how low privacy mode introduces minimal perturbations, thereby preserving the overall integrity of the original data while ensuring differential privacy. This choice is particularly useful in scenarios where high accuracy is

required for operational decisions, and where the risk of privacy breach is deemed to be low or acceptable.

Moving to the moderate privacy mode, a balance is struck between data utility and stronger privacy guarantees. Under this mode, the same example parameters—like transformer *count* and *avg_pct_peak_loading*—are still used, but with a more conservative configuration. Here, $\varepsilon$ is set to $0.5$ and $\delta$ remains at $10^{-5}$, indicating that the applied noise is slightly higher than in the low mode. The reduction in $\varepsilon$ increases privacy protection by deviating the output data further from the true values while still retaining a reasonable level of accuracy. Continuing with our running example where the transformer *count* is **3** and *avg_pct_peak_loading* is **72.20%**, the effects of moderate privacy become apparent. For the *count* parameter, the Discrete Laplace mechanism with $\varepsilon = 0.5$ yields a probability parameter of $p = e^{-0.5/1} \approx 0.607$, which increases the likelihood of larger noise. If a noise value of $+2$ were generated, the resulting count would be **5**. Similarly, for *avg_pct_peak_loading*, the Gaussian mechanism would use $\sigma \approx 0.969$ (doubling the standard deviation compared to low privacy mode), possibly generating noise of $+0.85$ and producing a final value of **73.05%**. These more substantial adjustments demonstrate how moderate privacy mode provides enhanced protection while still maintaining usable data.

In contrast, the high privacy mode is designed for scenarios demanding stringent confidentiality. In this mode, the addition of noise is maximized by reducing $\varepsilon$ to 0.1 and more aggressively lowering $\delta$ to $10^{-12}$. The tight configuration in high privacy mode ensures that the outputs are significantly obscured, thereby offering robust privacy guarantees even under rigorous scrutiny. With our continuing example where the transformer *count* is **3** and *avg_pct_peak_loading* is **72.20%**, the impact of high privacy mode becomes quite pronounced. For the transformer *count*, the Discrete Laplace mechanism with $\varepsilon = 0.1$ results in a probability parameter of $p = e^{-0.1/1} \approx 0.905$, substantially increasing the likelihood of larger noise values. This might generate noise of $+5$, resulting in a *count* of **8**. For *avg_pct_peak_loading*, the Gaussian mechanism now uses $\sigma \approx 6.76$ (calculated with the more stringent $\delta$ value), potentially generating noise of $+7.32$ and yielding a final value of **79.52%**. These substantial modifications clearly illustrate how high privacy mode prioritizes privacy over utility, significantly altering the original values to provide comprehensive protection against data de-anonymization. These choices are consistent with leading practices in differential privacy, where reduced epsilon values are employed to mitigate potential privacy breaches [1], [27]–[29]. Table 1 provides an overview of the differential privacy modes and their respective parameter settings and effects.

In addition to the predefined modes, we envision a custom privacy mode for advanced users with specific needs or regulatory requirements. In this custom mode, the user can select one of the available base modes while also passing a JSON configuration that allows for a tailored specification of $\varepsilon$ and $\delta$ values. For example, a user may opt for the structural integrity of the moderate mode but decide to adjust epsilon and delta based on emerging insights or legal standards. This flexibility enables users to experiment with different trade-offs between data utility and privacy levels. Although the custom mode is slated for future releases, its planned implementation assures that our system remains adaptable to evolving requirements.

The numerical choices for $\varepsilon$ and $\delta$ across the modes are based on both theoretical foundations and practical requirements. Sensitivity is set to $1$ for count parameters, which is a standard assumption to simplify the calibration process without sacrificing the precision of privacy measurements. Additionally, selecting $\delta = 10^{-5}$ for both low and moderate modes provides a consistent error allowance, while the significant reduction to $\delta = 10^{-12}$ in high mode ensures an extra layer of privacy protection where needed. The $\varepsilon$ values of $1$, $0.5$, and $0.1$ for low, moderate, and high modes, respectively, reflect a graduated scale of privacy, where lower values correspond to higher levels of privacy at the cost of increased noise. These parameters have been chosen in keeping with guidelines in the literature and are subject to review as our implementation is evaluated further.

Table 1: Comparison of Different Privacy Modes

| **Privacy Mode** | **Low** | **Moderate** | **High** |
|---|---|---|---|
| Parameters | $\varepsilon = 1.0$<br>$\delta = 10^{-5}$ | $\varepsilon = 0.5$<br>$\delta = 10^{-5}$ | $\varepsilon = 0.1$<br>$\delta = 10^{-12}$ |
| **Transformer Count** (Sensitivity = 1, Discrete Laplace Mechanism) | | | |
| Probability Parameter | $p = e^{-1} \approx 0.368$ | $p = e^{-0.5} \approx 0.607$ | $p = e^{-0.1} \approx 0.905$ |
| Original Value | 3 | 3 | 3 |
| Example Noise | +1 | +2 | +5 |
| Noisy Output | **4** | **5** | **8** |
| **Avg. Percent Peak Loading** (Sensitivity = 0.1, Gaussian Mechanism) | | | |
| Std. Deviation | $\sigma \approx 0.4845$ | $\sigma \approx 0.969$ | $\sigma \approx 6.76$ |
| Original Value | 72.20% | 72.20% | 72.20% |
| Example Noise | +0.30 | +0.85 | +7.32 |
| Noisy Output | **72.50%** | **73.05%** | **79.52%** |
| **Key Takeaway:** As privacy protection increases from low to high mode, the magnitude of added noise increases as well. Low privacy mode preserves high data utility with minimal perturbations, moderate mode balances utility and privacy, while high privacy mode prioritizes confidentiality by introducing substantial noise, significantly altering original values to prevent de-anonymization. | | | |

In summary, this subsection describes a carefully structured approach to differential privacy in Grid Kitchen. The three predefined modes—low, moderate, and high—offer users clear, standardized options, with the low mode prioritizing higher utility, the moderate mode achieving a balanced scope, and the high mode enforcing rigorous privacy safeguards. Our implementation also anticipates a custom mode to accommodate specific user configurations via a JSON interface, enhancing the system's flexibility. Each mode is supported by well-defined $\varepsilon$ and $\delta$ values, where reductions in $\varepsilon$ and $\delta$ strengthen privacy guarantees at the expense of data accuracy. In the following subsection on Noise Catalog Development, we will explore how these privacy parameters are systematically applied across various feeder model components and attributes.

## 2.3 Noise Catalog Development

The Noise Catalog serves as an organized repository for all properties within the feeder model that require systematic application of differential privacy noise. The catalog is maintained in an Excel file consisting of multiple sheets, each representing a distinct component of the feeder model such as transformers, feeder_sections, capacitors, regulators, switches, and substations. For each sheet, the catalog records properties including the property name, data type (integer, string, or number), sensitivity, privacy parameters ($\varepsilon$ and $\delta$), and the assigned privacy level. In addition, there are notes that provide additional considerations regarding the choice of noise function, for instance, whether the Discrete Laplace or Gaussian function should be used for the

corresponding property. This detailed categorization ensures that every attribute is treated according to its statistical sensitivity and intended use, thereby facilitating a consistent application of differential privacy across the entire dataset.

Within the transformers module, a range of properties is documented, including *kva*, *count*, *high_kv*, *low_kv*, *avg_customers_served*, *min_customers_served*, *max_customers_served*, *std_customers_served*, *min_pct_peak_loading*, *avg_pct_peak_loading*, *max_pct_peak_loading*, and *std_pct_peak_loading*. Some properties such as *is_substation_transformer* and *num_phase* are explicitly marked to receive no noise. The specification of sensitivity is set to 1 for most count-related properties, and the corresponding $\varepsilon$ and $\delta$ values are assigned based on the chosen privacy level.

The feeder_sections module documents properties related to the physical and operational attributes of feeder lines. Key parameters include *kv*, *count*, *avg_feeder_miles*, *min_feeder_miles*, *max_feeder_miles*, *std_feeder_miles*, *min_ampacity*, *avg_ampacity*, *max_ampacity*, *std_ampacity*, as well as customer and peak loading metrics. For this sheet, certain properties such as *num_phase* and *construction_type* are excluded from noise application, as indicated in the catalog. The sensitivity for numerical attributes is defined and aligned with corresponding ε and δ parameters suitable for differential privacy.

The capacitors module is dedicated to properties related to capacitor characteristics within the feeder model, including *kvar*, *kv,* and *count*. Similar to the other components, the catalog specifies sensitivity and assigns privacy parameters accordingly. Properties like *num_phase* and *install_type* are deliberately excluded from noise application because they are either categorical or have low variability. For example, the number of phases generally takes on a limited set of values (e.g., 1 or 3), and introducing noise to this property could significantly alter its meaning and adversely affect downstream computations, thereby undermining the integrity of the feeder model. The Noise Catalog notes offer explicit recommendations on the use of noise functions, ensuring that only the properties with quantifiable sensitivity are modified. By clearly distinguishing between properties that require differential privacy noise and those that do not, the capacitor sheet supports precise calibration of the privacy mechanism.

For the regulators module, the catalog includes properties such as *kva*, *kv*, *count*, and *num_phase*. In this context, the *num_phase* property is again not subjected to noise injection due to its low variability. The other numerical properties, however, are assigned sensitivities and corresponding privacy parameters that align with the overall privacy strategy. The catalog further suggests the appropriate noise function—typically the Gaussian mechanism for properties with continuous numerical values.

The switches and substations modules are similarly organized to capture all necessary attributes for privacy processing. For switches, the Noise Catalog records properties such as *num_phase*, *kv*, *count*, *avg_ampacity*, *min_ampacity*, *max_ampacity*, and *std_ampacity*, while explicitly excluding *num_phase* and *is_normally_open* from noise addition. In the substations module, properties like *kva*, *high_kv*, *feeder_count*, and various *feeder_miles* statistics are documented with their corresponding sensitivity and privacy parameters. This structured approach guarantees consistency across different components of the feeder model and supports the reliable application of differential privacy measures.

In summary, the Noise Catalog development plays a critical role in the systematic application of differential privacy for the feeder model. By maintaining an Excel file with separate sheets for transformers, feeder_sections, capacitors, regulators, switches, and substations, the catalog ensures that each property is processed in line with its sensitivity and intended use. The specification of property attributes, data types, sensitivity, and the privacy parameters $\varepsilon$ and $\delta$, together with notes on the recommended noise functions, establishes a transparent framework for privacy enforcement. Properties that are considered non-sensitive or categorical are clearly marked for exclusion from noise addition, thereby preventing unnecessary data distortion. This

catalog not only facilitates consistent privacy measures across various components of the feeder model but also provides a mechanism for incorporating feedback from domain experts in future revisions. We anticipate releasing an enhanced version of the Noise Catalog in the future, incorporating additional refinements and expanded coverage based on collaboration with industry stakeholders and privacy researchers.

## 2.4 Code Architecture and Usage Guidelines

In this subsection, we describe the code architecture and usage guidelines for applying differential privacy in the `Gridmeta` repository. This repository is currently available at https://github.com/grid-kitchen/grid-meta. This documentation focuses on the core code components, including the privacy settings definition, the functions for adding noise to continuous and discrete data, and the usage examples for invoking different privacy modes. This section is intended to provide a concise, step-by-step guide on how to use the code to process dehydrated feeder models with appropriate noise, evaluate the noise injection, and generate comparison outputs.

The codebase for `Gridmeta` starts with the definition of the PrivacySetting dataclass, which serves as the foundation for applying differential privacy to feeder model attributes. This data class encapsulates key parameters such as the JSON path to the property, its data type (discrete or continuous), sensitivity, epsilon, delta, and a flag to enforce always-positive outputs. More importantly, its structure facilitates the rapid integration of the noise catalog data by aligning with the catalog's format, thereby streamlining the application of differential privacy measures across various attributes. The following code snippet defines the PrivacySetting class:

[I]:
```
@dataclass
class PrivacySetting:
    path: str
    type: Literal["discrete", "continuous"]
    sensitivity: int
    epsilon: float
    is_sensitivity_in_percentage: bool
    delta: float = 1e-5
    always_positive: bool = False,
```

Each field in this dataclass serves a specific purpose:

- *path*: Specifies the JSON path to the property in the feeder model. Follows the `Gridmeta`'s metadata schema.
- *type*: Determines whether the property should be treated as "discrete" or "continuous". Different noise functions are added based on the property type.
- *sensitivity*: Defines the maximum possible impact of a single record change.
- *epsilon*: The privacy budget parameter; lower values provide stronger privacy guarantees.
- *is_sensitivity_in_percentage*: Indicates whether sensitivity is a percentage of the original value.
- *delta*: The probability of privacy failure allowed (default: $10^{-5}$).

- *always_positive*: Flag to ensure the noisy value remains positive (default: $False$).

Next, we discuss the two differential privacy functions implemented to handle continuous and discrete values separately. For continuous properties, the Gaussian mechanism is employed due to its mathematical properties that provide a smooth noise distribution appropriate for real-valued data while maintaining strong privacy guarantees. The `add_differential_privacy_gaussian` function calculates the required standard deviation (sigma) based on the sensitivity, epsilon, and delta values, and adds calibrated Gaussian noise to the original input. Special handling is provided to preserve "NaN" entries, which may exist in the original feeder model where certain properties are not populated, thereby maintaining data consistency. Additionally, for properties that are inherently non-negative (such as average feeder miles), this function implements a post-processing step that takes the absolute value of any negative results, ensuring that the physical constraints of the model are respected while still maintaining differential privacy guarantees. This approach aligns with established differential privacy literature, which acknowledges that post-processing of differentially private outputs does not compromise the privacy properties of the mechanism [1], [8].

[II]:
```
def add_differential_privacy_gaussian(value: float, epsilon: float, sensitivity:
float, alwaysPositive: bool = False, delta: float = 1e-5) -> float:
    """
    Apply Gaussian noise for differential privacy.
    Parameters:
        value (int or float): The original value.
        epsilon (float): Privacy budget (smaller = more privacy).
        sensitivity (float): The maximum change in output for one data change.
        alwaysPositive (bool, optional): If True, ensure the noisy value is not
negative.
        delta (float, optional): Probability of failing to achieve differential
privacy.
    Returns:
        float: Privacy-protected value.
    """
    # Calculate the necessary standard deviation for Gaussian noise
    sigma = (sensitivity * np.sqrt(2 * np.log(1.25 / delta))) / epsilon

    # Generate Gaussian noise
    noise = np.random.normal(loc=0, scale=sigma)

    # Handle NaN values
    if value == "NaN":
        return value

    # Calculate noisy value
    noisy_value = value + noise

    # Ensure non-negative output if alwaysPositive is True
    if alwaysPositive and noisy_value < 0:
        noisy_value = abs(noisy_value)
```

```
    return noisy_value
```

The complementary function for discrete properties, named `add_discrete_differential_privacy`, implements the Discrete Laplace mechanism, which is particularly well-suited for integer-valued data as it preserves the discrete nature of the original values while providing rigorous differential privacy guarantees. This mechanism, which relies on geometric sampling to generate appropriate noise, ensures that the resulting values maintain their integer characteristics—an essential property for count-based metrics such as the number of customers served by a transformer. In this function, the probability parameter is computed based on epsilon and sensitivity values, and the noise is drawn from a discrete distribution that mirrors the shape of the continuous Laplace distribution but over the integer domain. The function incorporates careful handling of special cases, preserving ”NaN” entries that may exist in the original data and maintaining zero values where appropriate. Furthermore, similar to its continuous counterpart, this function enforces non-negativity constraints when the *alwaysPositive* flag is set, ensuring that inherently non-negative properties remain so after noise addition without compromising the differential privacy guarantees.

[III]:
```
def add_discrete_differential_privacy(value: int, epsilon: float, sensitivity:
float,alwaysPositive: bool = False, delta: float = 1e-5) -> int:
    """
    Apply Discrete Laplace noise for differential privacy.

    Parameters:
        value (int): The original integer value.
        epsilon (float): Privacy budget (smaller = more privacy).
        sensitivity (int): The maximum change in output for one data change.
        alwaysPositive (bool, optional): If True, ensure the noisy value is not
negative.
        delta (float, optional): Probability of failing to achieve differential
privacy. (not used in this function; but kept in order to sync with other
function declarations)

    Returns:
        int: Privacy-protected integer value.
    """
    # Compute probability parameter for discrete Laplace
    p = np.exp(-epsilon / sensitivity)

    # Sample from the discrete Laplace distribution
    u = np.random.uniform(-0.5, 0.5)
    sign = 1 if u > 0 else -1
    geom_sample = np.random.geometric(1 - p) - 1  # Geometric noise
    discrete_laplace_noise = sign * geom_sample
    # If value is already 0, don't return negative value
    if value == 0: return 0
    # Handle NaN values (string)
```

```
    if value == "NaN": return value
    # Calculate noisy value
    noisy_value = value + discrete_laplace_noise

    # Ensure non-negative output if alwaysPositive is True
    if alwaysPositive and noisy_value < 0:
        noisy_value = abs(noisy_value)

    return int(noisy_value)
```

Next, we demonstrate how to invoke the `Gridmeta` command-line tool to apply the differential privacy transformations on a dehydrated feeder model. In this example, the low privacy mode is utilized, which offers higher data utility by incorporating minimal noise. The low privacy mode is specifically designed for scenarios where preserving the accuracy of the extracted feeder model is paramount, with a corresponding trade-off of lower privacy guarantees.

The interface is invoked using the ”extract-opendss-dehydrated-dataset” command, which requires the input DSS file representing the feeder model. A flag (-pm) is used to specify the privacy mode; here, ”low” is selected to ensure that only minimal noise is added, thereby preserving the essential characteristics of the data. Additionally, the -o flag facilitates the designation of the output file name where the privacy-enhanced model is saved. This structured approach allows users to easily integrate differential privacy into their data extraction workflows.

The following code snippet illustrates the exact command used to perform this operation:

[IV]:
```
gridmeta extract-opendss-dehydrated-dataset -f tests\data\opendss\ieee13\master.
↪dss -pm "low" -o test_lowpm.json
```

For the moderate privacy mode, we present an intermediate configuration that balances data utility with enhanced privacy protection. This mode introduces a more substantial level of noise compared to the low privacy setting, thereby increasing the privacy guarantees while still maintaining reasonable data utility. The moderate privacy mode is appropriate for scenarios where data sensitivity is elevated, yet certain analytical tasks still require a degree of accuracy in the underlying data. The command structure remains consistent with the previous example, with only the privacy mode parameter being adjusted to reflect the desired level of protection.

[V]:
```
gridmeta extract-opendss-dehydrated-dataset -f tests\data\opendss\ieee13\master.
↪dss -pm "moderate" -o test_moderatepm.json
```

For applications demanding the highest level of privacy protection, the high privacy mode offers the most stringent safeguards by applying substantial noise to the feeder model data. This configuration significantly enhances privacy guarantees at the cost of reduced data utility. The high privacy mode is particularly relevant for scenarios involving highly sensitive grid data or when regulatory compliance mandates robust anonymization measures. When implementing this mode, users should anticipate more pronounced deviations from the original values, reflecting the inherent trade-off between privacy and utility. The command syntax maintains consistency with the previously demonstrated examples, with the privacy mode parameter adjusted accordingly.

[VI]:
```
gridmeta extract-opendss-dehydrated-dataset -f tests\data\opendss\ieee13\master.
↪dss -pm "high" -o test_highpm.json
```

These graduated privacy modes exemplify the systematic approach taken in the Grid Kitchen project to provide users with configurable privacy-utility trade-offs, accommodating diverse requirements within the domain of grid feeder model analysis and distribution.

To facilitate understanding of differential privacy's impact on feeder model data, the `Gridmeta` repository includes evaluation capabilities. Power utility operators and analysts need to quantify how differential privacy transformations affect the original data values before sharing privacy-enhanced models with external parties for research purposes and other uses. This evaluation step is crucial for maintaining operational confidence while implementing privacy safeguards.

The evaluation functionality serves multiple purposes in the workflow of a utility operator. First, it allows users to generate a baseline model without privacy protections, followed by creating versions with various privacy modes enabled. Through systematic comparison, operators can observe exactly how differential privacy alters specific properties—such as transformer counts, average percent peak loading, or feeder miles—under different privacy configurations. This insight enables informed decision-making regarding the appropriate privacy level for their particular use case, balancing regulatory compliance with operational requirements.

It is important to note that this evaluation capability does not compromise the privacy guarantees inherent in differential privacy. Once a privacy-enhanced model is shared externally, the mathematical properties of differential privacy ensure that observers cannot reliably determine the original values, even with knowledge of the noise distribution parameters. This property—that evaluation is possible internally but reconstruction is infeasible externally—exemplifies the strength of differential privacy as a privacy protection mechanism for sensitive grid data.

The `Gridmeta` repository facilitates this evaluation through a dedicated command that generates a CSV report documenting the differences between an original model and its privacy-enhanced counterpart. This report includes absolute differences across all relevant properties, enabling detailed analysis of the privacy-utility trade-off at the attribute level. The evaluation command can be invoked using this command:

[VII]:
```
gridmeta evaluate -f test.json -f test_lowpm.json -o differences.csv
```

In this example, the evaluate command takes two JSON files via the -f flag—the original model (test.json) and its privacy-enhanced counterpart (test_lowpm.json) generated using the low privacy mode—and outputs a detailed comparison to the file specified by the -o flag (differences.csv).

This evaluation framework empowers utility operators to make evidence-based decisions about privacy implementation while maintaining confidence that their shared data remains protected against re-identification or inference attacks.

# 3.0 Future Work

The implementation of differential privacy in Grid Kitchen represents a significant step forward in balancing data utility with privacy guarantees for grid feeder models. However, as with any complex system, there are several areas where additional refinement and expansion could further enhance the functionality and effectiveness of our approach.

One of the primary areas for future work involves fine-tuning the core differential privacy parameters. While our current implementation provides reasonable defaults for epsilon, delta, and sensitivity values across the low, moderate, and high privacy modes, these parameters would benefit from systematic calibration based on empirical evaluation. Specifically, we plan to conduct comprehensive sensitivity analyses to determine how varying these parameters affects both the privacy guarantees and the utility of the resulting data. This calibration process will help establish more precise recommendations for parameter selection based on specific use cases and risk profiles.

The sensitivity parameter, in particular, requires careful consideration as it directly impacts the scale of noise added to each attribute. Currently, we use a sensitivity value of $1$ for most count parameters, which is a conservative approach. Future work will include developing more nuanced sensitivity calculations that account for the specific characteristics of each attribute and its distribution within typical feeder models. This could involve statistical analysis of historical data to better understand the potential impact of individual records on aggregate statistics, leading to more accurate sensitivity estimates and, consequently, more efficient noise addition.

Beyond parameter refinement, gathering feedback from domain experts and stakeholders will be crucial during the testing phase. Electric utility engineers, data scientists, and privacy experts bring different perspectives to the table, and their insights can help identify both technical improvements and practical considerations that might not be immediately apparent. We plan to establish a structured feedback collection process where alpha testers can report on the usability of the differential privacy features, the appropriateness of the privacy-utility trade-offs, and any unexpected behaviors or limitations they encounter. This feedback will directly inform subsequent iterations of the system.

The current noise catalog covers a comprehensive set of feeder model properties, but there remain opportunities to expand this coverage to additional attributes or derived metrics that may be relevant for specific analytical tasks. Future work will include a detailed review of additional properties that could benefit from privacy protection, particularly those that may emerge as important in new analytical workflows or regulatory contexts. This expansion may also involve developing specialized noise mechanisms tailored to the unique statistical characteristics of these new properties, ensuring that the privacy-utility balance is maintained even as the scope of protected attributes grows.

The integration of adaptive privacy mechanisms is another area for future exploration. Such mechanisms could dynamically adjust the privacy parameters based on the specific characteristics of the data being processed or the nature of the analysis being performed. For example, the system could apply more stringent privacy protection to attributes that show higher variability or are more likely to contain identifying information. Similarly, it could relax privacy constraints for analyses that are known to be less sensitive or where higher accuracy is critical. This adaptive approach could optimize the privacy-utility trade-off on a case-by-case basis, providing more flexibility while maintaining strong privacy guarantees.

As the field of differential privacy continues to evolve, new theoretical developments and practical implementations emerge that could benefit our system. Keeping abreast of these advancements and incorporating relevant innovations will be an ongoing effort. This includes monitoring new privacy mechanisms, composition theorems, and verification techniques that could enhance our implementation. Additionally, as the Grid Kitchen platform grows and

evolves, ensuring that the differential privacy components remain compatible with new features and workflows will be essential for maintaining a cohesive and effective system.

Finally, as the differential privacy module moves from alpha testing to wider deployment, developing comprehensive documentation, training materials, and best practice guidelines will be crucial for ensuring adoption and correct usage. This includes creating detailed user guides that explain the privacy modes and their implications in non-technical terms, providing examples of how to select appropriate privacy parameters for new use cases, and offering troubleshooting assistance for common issues. Making differential privacy accessible and understandable to a broader audience will help ensure that its benefits are fully realized across the ecosystem.

In conclusion, while our current implementation provides a solid foundation for differential privacy in grid feeder models, these future work directions represent opportunities to refine, extend, and strengthen the system. By pursuing these enhancements, we aim to create a differential privacy implementation that not only provides strong theoretical guarantees but also meets the practical needs of utility operators, researchers, and other stakeholders working with sensitive grid data.

# References


[1] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013.

[2] J. M. Abowd, "The us census bureau adopts differential privacy," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2867–2867.

[3] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[4] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *21st International conference on data engineering (ICDE'05)*, IEEE, Tokyo, Japan: IEEE, 2005, pp. 217–228.

[5] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, 3–es, 2007.

[6] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*, IEEE, Istanbul, Turkey: IEEE, 2007, pp. 106–115.

[7] K. Bhattacharjee, M. Chen, and A. Dasgupta, "Privacy-Preserving Data Visualization: Reflections on the State of the Art and Research Opportunities," in *Computer Graphics Forum*, vol. 39, Norrköping, Sweden: Wiley Online Library, 2020, pp. 675–692.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, Springer, 2006, pp. 265–284.

[9] C. Dwork, "Differential privacy," in *International colloquium on automata, languages, and programming*, Springer, 2006, pp. 1–12.

[10] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2003, pp. 202–210.

[11] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to noninteractive database privacy," *Journal of the ACM (JACM)*, vol. 60, no. 2, pp. 1–25, 2013.

[12] M. Hardt, K. Ligett, and F. McSherry, "A simple and practical algorithm for differentially private data release," *Advances in neural information processing systems*, vol. 25, 2012.

[13] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, 2007, pp. 75–84.

[14] S. Ruggles, "When privacy protection goes wrong: How and why the 2020 census confidentiality program failed," *Journal of Economic Perspectives*, vol. 38, no. 2, pp. 201–226, 2024.

[15] S. Ruggles, C. Fitch, D. Magnuson, and J. Schroeder, “Differential privacy and census data: Implications for social and economic research,” in *AEA papers and proceedings*, American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203, vol. 109, 2019, pp. 403–408.

[16] K. Bhattacharjee, “Interactive visualization workflows for mitigating analytical uncertainty,” Ph.D. dissertation, New Jersey Institute of Technology, 2024.

[17] P. McDaniel and S. McLaughlin, “Security and privacy challenges in the smart grid,” *IEEE security & privacy*, vol. 7, no. 3, pp. 75–77, 2009.

[18] E. McKenna, I. Richardson, and M. Thomson, “Smart meter data: Balancing consumer privacy concerns with legitimate applications,” *Energy Policy*, vol. 41, pp. 807–814, 2012.

[19] M. Panteli and P. Mancarella, “Influence of extreme weather and climate change on the resilience of power systems: Impacts and possible mitigation strategies,” *Electric Power Systems Research*, vol. 127, pp. 259–270, 2015.

[20] M. Ouyang, “Review on modeling and simulation of interdependent critical infrastructure systems,” *Reliability engineering & System safety*, vol. 121, pp. 43–60, 2014.

[21] C. K. Das, O. Bass, T. S. Mahmoud, G. Kothapalli, M. A. Masoum, and N. Mousavi, “An optimal allocation and sizing strategy of distributed energy storage systems to improve performance of distribution networks,” *Journal of Energy Storage*, vol. 26, p. 100 847, 2019.

[22] M. E. Ölmez, I. Ari, and G. Tuzkaya, “A comprehensive review of the impacts of energy storage on power markets,” *Journal of Energy Storage*, vol. 91, p. 111 935, 2024.

[23] B. Palmintier, R. J. Broderick, B. Mather, *et al.*, “On the path to sunshot: Emerging issues and challenges in integrating solar with the distribution system,” 2016.

[24] R. Seguin, J. Woyak, D. Costyk, J. Hambrick, and B. Mather, “High-penetration pv integration handbook for distribution engineers,” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2016.

[25] F. Ding, A. Nagarajan, S. Chakraborty, *et al.*, “Photovoltaic impact assessment of smart inverter volt-var control on distribution system conservation voltage reduction and power quality,” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2016.

[26] D. International, *DAMA-DMBOK: Data management body of knowledge*. Technics Publications, LLC, 2017.

[27] J. Hsu, M. Gaboardi, A. Haeberlen, *et al.*, “Differential privacy: An economic method for choosing epsilon,” in *2014 IEEE 27th Computer Security Foundations Symposium*, IEEE, 2014, pp. 398–410.

[28] R. Sarathy and K. Muralidhar, “Evaluating laplace noise addition to satisfy differential privacy for numeric data.,” *Trans. Data Priv.*, vol. 4, no. 1, pp. 1–17, 2011.

[29] J. Lee and C. Clifton, “How much is enough? choosing $\varepsilon$ for differential privacy,” in *Information Security: 14th International Conference, ISC 2011, Xi’an, China, October 26-29, 2011. Proceedings 14*, Springer, 2011, pp. 325–340.

# Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7675)

***www.pnnl.gov***