

PRESTO: A Python package for recommending privacy preservation algorithm based on user preferences.

Olivera Kotevska¹, A. Gilad Kusne², Prasanna Balaprakash¹, and Robert Patton¹

¹ Oak Ridge National Laboratory, United States ² National Institute of Standards and Technology, United States

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Open Journals](#)

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

PRESTO (Privacy REcommendation and SecuriTy Optimization) is a Python-based toolkit that automates the selection of differential-privacy mechanisms (Dwork & Roth, 2014) to balance data utility and privacy loss. By integrating descriptive and inferential statistics, Bayesian optimization, and data-similarity metrics, PRESTO analyzes arbitrary datasets—numerical, categorical, or structured—and recommends the optimal privacy algorithm and ϵ -parameter setting. Its modular design supports CPU/GPU execution, streaming and batch data, and extensibility for new algorithms and utility metrics. PRESTO's automated multi-objective optimization delivers application-specific, data-driven recommendations with quantified uncertainty, empowering both experts and non-experts to integrate privacy-preserving methods seamlessly into their workflows.

Statement of Need

As data collection proliferates across healthcare, finance, IoT, and beyond, safeguarding individual privacy without handicapping downstream analytics has become critical. Existing differential-privacy tools often require deep theoretical knowledge, manual tuning of privacy parameters, and trial-and-error to discover the right trade-off between noise injection and data utility. This steep adoption barrier impedes widespread deployment of privacy-preserving analytics in industrial and research settings. There is a pressing need for an intuitive, automated solution that can—given any dataset—identify the most suitable privacy mechanism and its optimal ϵ , quantify the remaining utility, and provide confidence intervals on its recommendations. PRESTO fills this gap, reducing the technical burden and accelerating safe, compliant data analysis.

State of the Field

A variety of packages from industry and academia—such as IBM's Diffprivlib (Holohan et al., 2019), Google's PyDP (Wilson et al., 2020), Facebook's Opacus (Yousefpour et al., 2021), LDP-Pure (Cormode et al., 2021), SmartNoise (Gaboradi et al., 2025), PETINA—offer implementations of noise-based DP mechanisms (Laplace, Gaussian, Exponential) (Dwork & Lei, 2009), local-DP protocols (Randomized Response, RAPPOR) (Erlingsson et al., 2014), and gradient perturbation for machine learning. However, they typically expose raw APIs, leaving users responsible for selecting and tuning algorithms, and provide limited guidance on choosing ϵ . Recent research has explored automatic hyperparameter tuning via cross-validation

39 or surrogate modeling, but these approaches rarely integrate multi-objective optimization or
40 deliver quantitative uncertainty measures.

41 PRESTO advances the state of the art by unifying statistical dataset analysis, Bayesian
42 optimization, and data-similarity metrics into a single recommendation engine. It implements
43 a broad suite of privacy mechanisms—including both batch and streaming algorithms—and
44 automates their selection based on data characteristics and user-specified privacy–utility trade-
45 offs, while providing 95% confidence intervals on its recommendations. Crucially, PRESTO is
46 built on a modular architecture, enabling users to plug in new privacy algorithms or custom
47 utility metrics at any time without modifying core logic. This extensibility ensures that
48 PRESTO can evolve alongside emerging research and domain-specific needs, making it uniquely
49 adaptable compared to existing static libraries.

50 Methodology

51 1. Dataset Profiling

- 52 ▪ Compute descriptive (mean, variance, skewness, kurtosis) and, for categorical data,
53 domain-size and frequency distributions.

54 2. Mechanism Library

- 55 ▪ Maintain a dictionary of privacy functions (`get_noise_generators()`), each map-
56 ping (`data, \varepsilon`) \rightarrow `privatized_data`.

57 3. Bayesian Optimization of ε

- 58 ▪ For each mechanism, define:

$$f(\varepsilon) = -\text{RMSE}(\text{data}, \text{mechanism}_\varepsilon(\text{data}))$$

- 59 ▪ Maximize this over:

$$\varepsilon \in [\varepsilon_{\min}, \varepsilon_{\max}]$$

60 using Gaussian-process Bayesian optimization.

61 4. Confidence & Reliability

- 62 ▪ Compute a 95% confidence interval on RMSE at the optimal ε^* , then define:

$$\text{Reliability} = \frac{1}{\text{Mean RMSE} \times \text{CI Width}}.$$

63 5. Similarity Assessment

- 64 ▪ Measure distributional similarity via Kolmogorov–Smirnov, Jensen–Shannon, Pear-
65 son correlation.

66 6. Multi-Objective Ranking

- 67 ▪ Recommend top mechanisms on **max similarity**, **max reliability**, and **max privacy**
68 axes.

69 Experiments

70 We conducted experiments to evaluate the effectiveness of our approach.

71 Energy Compumtion with Bayesian Optimization (Dataset: Hourly Consumption (Min))

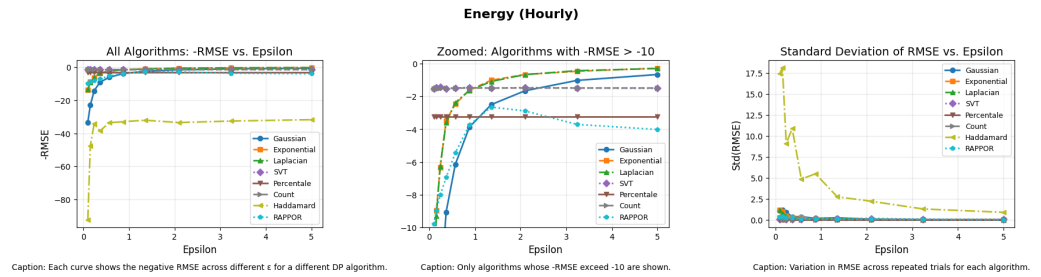


Figure 1: Privacy loss (epsilon) vs utility (RMSE) for selected/preferred privacy algorithms

72 Top-3 Recommendations:

- 73 ■ **DP_Laplace:** $\epsilon = 3.6277$, mean_rmse=0.3817, ci_width=0.0279, reliability=93.90
- 74 ■ **DP_Exponential:** $\epsilon = 3.6300$, mean_rmse=0.3835, ci_width=0.0416, reliability=62.68
- 75 ■ **DP_Gaussian:** $\epsilon = 4.1687$, mean_rmse=0.8326, ci_width=0.0525, reliability=22.88

76 Medical Measuments with Bayesian Optimization (Dataset: Heart Rate (Min))

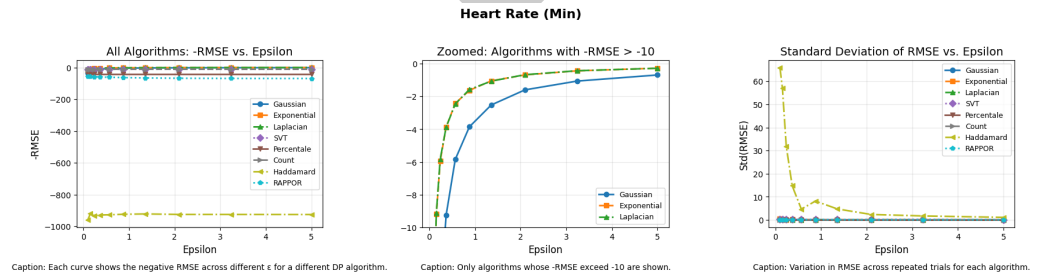


Figure 2: Privacy loss (epsilon) vs utility (RMSE) for selected/preferred privacy algorithms

77 Top-3 Recommendations:

- 78 ■ **DP_Laplace:** $\epsilon = 3.6254$, mean_rmse=0.3901, ci_width=0.0054, reliability=474.71
- 79 ■ **DP_Exponential:** $\epsilon = 3.6319$, mean_rmse=0.3916, ci_width=0.0051, reliabil-
- 80 ity=500.71
- 81 ■ **DP_Gaussian:** $\epsilon = 5.0000$, mean_rmse=0.6824, ci_width=0.0047, reliability=311.79

82 Finance Transactions with Bayesian Optimization (Dataset: Payment Transactions (Min))

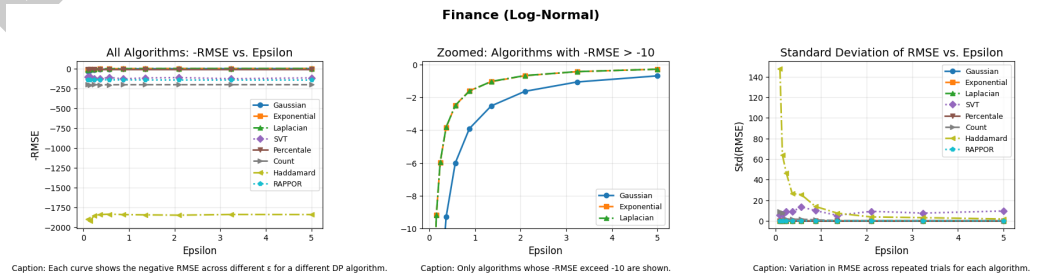


Figure 3: Privacy loss (epsilon) vs utility (RMSE) for selected/preferred privacy algorithms

83 Top-3 Recommendations:

- 84 ■ **DP_Laplace:** $\varepsilon = 4.1687$, mean_rmse=0.3461, ci_width=0.0340, reliability=84.98
- 85 ■ **DP_Exponential:** $\varepsilon = 3.6296$, mean_rmse=0.3864, ci_width=0.0453, reliability=57.13
- 86 ■ **DP_Gaussian:** $\varepsilon = 4.1690$, mean_rmse=0.8270, ci_width=0.0560, reliability=21.59

87 Sensor Temperature Time-Series with Bayesian Optimization (Dataset: Payment Transactions 88 (Min))

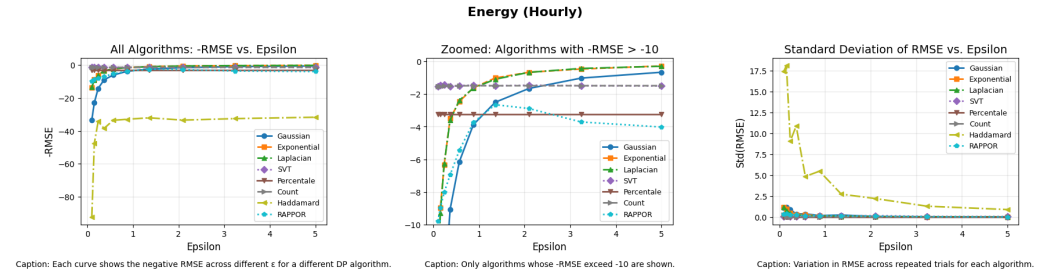


Figure 4: Privacy loss (epsilon) vs utility (RMSE) for selected/preferred privacy algorithms

89 Top-3 Recommendations:

- 90 ■ **DP_Laplace:** $\varepsilon = 3.6296$, mean_rmse=0.3846, ci_width=0.0126, reliability=206.36
- 91 ■ **DP_Exponential:** $\varepsilon = 3.6296$, mean_rmse=0.3883, ci_width=0.0187, reliability=137.72
- 92 ■ **DP_Gaussian:** $\varepsilon = 3.6296$, mean_rmse=0.9459, ci_width=0.0334, reliability=31.65

94 Energy Consumption with Fixed epsilon = 1

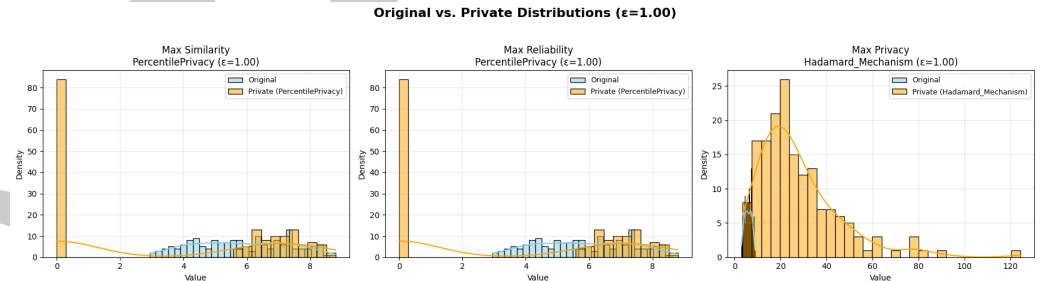


Figure 5: The best algorithm for a given epsilon

- 95 ■ Best by Similarity: {'algorithm': 'PercentilePrivacy', 'score': np.float32(0.9841)}
- 96 ■ Best by Reliability: {'algorithm': 'PercentilePrivacy', 'score': inf}
- 97 ■ Best by Privacy: {'algorithm': 'Hadamard_Mechanism', 'score': 71.6581}

98 ML Classification with Private Gradients

- 99 ■ Baseline Accuracy (no privacy): 93.00%
- 100 ■ DP Accuracy with 'PercentilePrivacy': 94.00%

101 ML Classification with Private Gradients

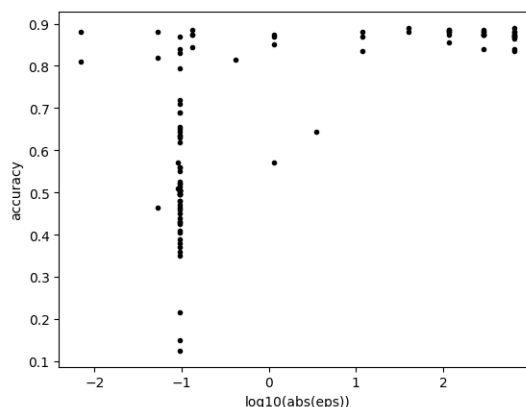


Figure 6: Pareto front for privacy budget vs accuracy

102 Conclusion

103 PRESTO delivers a data-driven, automated, and extensible framework for differential-privacy
104 mechanism selection and tuning. By profiling statistical properties, optimizing ϵ via Bayesian
105 methods, and quantifying both utility and uncertainty, PRESTO guides users to the privacy
106 solution best suited for their data. Its modular design allows seamless integration of new
107 algorithms and metrics, positioning PRESTO as a flexible platform for both practitioners and
108 researchers aiming to embed privacy guarantees in diverse analytical workflows.

109 Acknowledgements

110 This manuscript has been co-authored by UT-Battelle, LLC under Contract No. DE-AC05-
111 00OR22725 with the U.S. Department of Energy. The United States Government retains
112 and the publisher, by accepting the article for publication, acknowledges that the United
113 States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish
114 or reproduce the published form of this manuscript, or allow others to do so, for United
115 States Government purposes. The Department of Energy will provide public access to these
116 results of federally sponsored research in accordance with the DOE Public Access Plan
117 (<http://energy.gov/downloads/doe-public-access-plan>). This material is based upon work
118 supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific
119 Computing Research under Contract No. DE-AC05-00OR22725. This research is sponsored by
120 the Artificial Intelligence Initiative as part of the LDRD-SEED Program, at ORNL, managed
121 by UT-Battelle, LLC and DOE ASCR Program.

122 References

- 123 Cormode, G., Maddock, S., & Maple, C. (2021). Frequency estimation under local differential
124 privacy [experiments, analysis and benchmarks]. *Proceedings of the VLDB Endowment*,
125 14, 2046–2058.
- 126 Dwork, C., & Lei, J. (2009). Differential Privacy and Robust Statistics. *Proceedings of the*
127 *41st Annual ACM Symposium on Theory of Computing*, 371–380.
- 128 Dwork, C., & Roth, A. (2014). The Algorithmic Foundations of Differential Privacy. *Founda-*
129 *tions and Trends in Theoretical Computer Science*, 9(3–4), 211–407. <https://doi.org/10.1561/04000000042>

- 131 Erlingsson, Úlfar, Pihur, V., & Korolova, A. (2014). RAPPOR: Randomized Aggregatable
132 Privacy-Preserving Ordinal Response. *Proceedings of the 2014 ACM SIGSAC Conference*
133 *on Computer and Communications Security*, 1054–1067.
- 134 Gaboardi, M., Hay, M., & Vadhan, S. (2025). *OpenDP: The OpenDP Library* (Version 0.13.0).
135 <https://github.com/opendp/opendp>
- 136 Holohan, N., Braghin, S., Aonghusa, P. M., & Levacher, K. (2019). Diffprivlib: The IBM
137 Differential Privacy Library. *arXiv Preprint*. <https://arxiv.org/abs/1907.02444>
- 138 Wilson, R. J., Zhang, C. Y., Lam, W., Desfontaines, D., Simmons-Marengo, D., & Gipson, B.
139 (2020). *Google Differential Privacy Library*. <https://github.com/google/differential-privacy>
- 140 Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen,
141 J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., & Mironov, I. (2021). Opacus:
142 User-friendly differential privacy library in PyTorch. *arXiv Preprint arXiv:2109.12298*.

DRAFT