SKALD: Scalable K-Anonymisation for Large Datasets

Kailash Reddy, Novoneel Chakraborty, Amogh Dharmavaram, Anshoo Tandon

Email: kailash.nanaluri@gmail.com, cnovoneel@gmail.com, amogh.dharma1@gmail.com, anshoo.tandon@gmail.com

IUDX/CDPG (FSID), IISc Bengaluru

Abstract—Data privacy and anonymisation are critical concerns in today's data-driven society, particularly when handling personal and sensitive user data. Regulatory frameworks worldwide recommend privacy-preserving protocols such as k-anonymisation to de-identify releases of tabular data. Available hardware resources provide an upper bound on the maximum size of dataset that can be processed at a time. Large datasets with sizes exceeding this upper bound must be broken up into smaller data chunks for processing. In these cases, standard k-anonymisation tools such as ARX can only operate on a per-chunk basis. This paper proposes SKALD, a novel algorithm for performing k-anonymisation on large datasets with limited RAM. Our SKALD algorithm offers multi-fold performance improvement over standard k-anonymisation methods by extracting and combining sufficient statistics from each chunk during processing to ensure successful k-anonymisation while providing better utility.

Index Terms—k-anonymisation, data privacy, large datasets, data utility

I. INTRODUCTION

In the modern era, data is a pseudo-currency that underpins digital and social transactions. However, the data economy is fraught with significant privacy risks [1], [2] that threaten individual confidentiality. This makes safeguarding the privacy of data containing personal and sensitive information a critical concern if the data is to be shared, analysed or released.

Globally, regulatory pressures drive the data anonymisation imperative, and advisory guidelines are provided on techniques and execution of anonymisation [3]. In India, the Digital Personal Data Protection (DPDP) Act [4] is a precursor to more stringent and clearly outlined legislation around anonymisation. Canada, the EU and the UK stipulate that data should be rendered anonymous such that the data subject is no longer identifiable, while Singapore delineates acceptable privacy parameter ranges for the sharing of anonymised data through well-established privacy-preserving techniques such as k-anonymisation [5], [6].

The General Data Protection Regulation (GDPR), long considered the gold standard for comprehensive data protection rules, outlines stringent guidelines that emphasise the importance of data sovereignty [7]. The transfer of personal data to third parties or international organisations is strictly regulated. Local processing of data is encouraged to ensure an auditable chain of control and accountability which implies the need for dedicated hardware systems to carry out anonymisation tasks. Available hardware resources provide an upper bound on the maximum size of dataset that can be processed at a time.

Large datasets with sizes exceeding this upper bound must be broken up into smaller data chunks for processing. Although optimal k-anonymisation implementation requires knowledge of the entire dataset, standard k-anonymisation tools such as ARX can only operate on a per-chunk basis if the dataset is too large. This lack of a global view of the dataset can negatively affect the output information utility.

In this paper, we extend our work in [8] on data de-identification pipelines to propose a novel algorithm - SKALD (Scalable K-Anonymisation for Large Datasets), that operates efficiently when large datasets are broken into smaller chunks due to constraints on the main memory (RAM). We use the loss metrics proposed in [9] to demonstrate that SKALD offers multi-fold performance improvement over standard k-anonymisation methods, such as the open-source ARX software [10], by extracting and combining sufficient statistics from each chunk during processing to ensure successful k-anonymisation while providing better utility.

II. PRELIMINARIES AND PRIOR WORK

A. K-Anonymisation

K-Anonymisation is a well-established syntactic model of privacy that is popularly used to anonymise releases of tabular data [11]–[13]. To understand how k-anonymisation works, we need to understand the following terms:

1) Direct Identifiers: Direct identifiers are attributes that can be used to directly identify an individual in a dataset.

2) Indirect Identifiers or Quasi-Identifiers: Quasi-identifiers (QIDs) can be used to identify an individual in a dataset when combined with other attributes, but not on their own [14]. A famous example of identification using QID is given by Sweeney [15] - 87% of people in the United States were identifiable by a tuple of their {5-digit ZIP, gender, date of birth} in the year 2000. QIDs are of two types - numerical and categorical.

3) Sensitive Attributes: A sensitive attribute refers to any piece of information that, if disclosed, could lead to privacy violations or harm to an individual.

4) Equivalence Classes: An equivalence class (EC) in this context is a group of records that share the same values for a set of QIDs.

In a k-anonymised release, each equivalence class must contain at least k records. This is achieved by generalising QIDs to a less granular representation until the k-anonymity

requirement is met, thus reducing the risk of identifying individuals uniquely.

5) Record Suppression: Record suppression in the context of k-anonymisation is a method used to remove records to improve the utility of a k-anonymisation solution.

B. The Generalisation Lattice

In general, finding an optimal k-anonymity solution is NP-hard [16], and previous works have presented solutions involving traversal of a generalisation lattice to find a k-anonymisation solution [9], [17]. A generalisation lattice is a structured representation of all the possible ways to generalise the QIDs of a dataset. Each node in the lattice represents a single transformation that defines a specific resolution for each of the QIDs. The resolution becomes coarser as the generalisation levels increase from bottom to top. An arrow or edge in the lattice indicates that a transformation is a direct generalisation, meaning that the node can be derived by incrementing the generalisation level of a single QID from the predecessor. The bottom-most node of the lattice represents the finest resolution, while the top node represents the coarsest resolution, or the maximally generalised dataset [10]. In general, the information loss increases from the bottom to the top of the lattice. This lattice represents the search space for finding the k-minimal node - the k-anonymous node with the lowest information loss in the lattice.
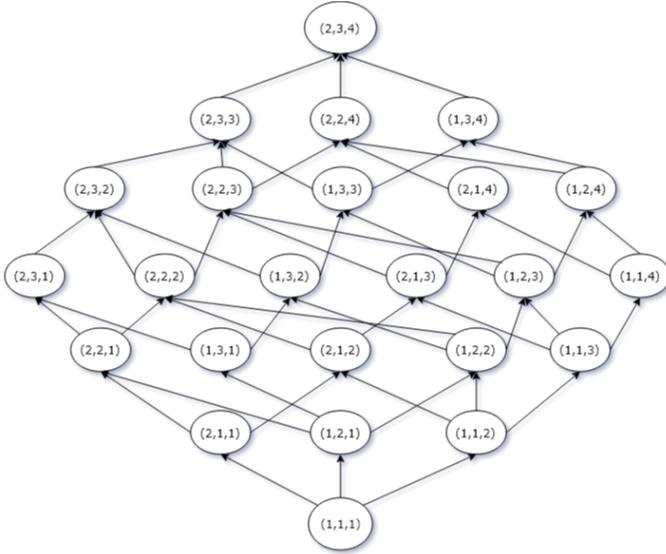

Fig. 1. An generalisation lattice with 3 QIDs generalised to 2,3 and 4 levels respectively.

The size of this search space can be quantified as the product of the number of levels for each QID's generalisation hierarchy, and grows with the number of QIDs for which these hierarchies are defined. For example in Fig.1 there are 3 QIDs that are generalised to 2, 3 and 4 levels respectively, and the total number of nodes in the generalisation lattice is 24.

C. Prior Work and Their Limitations

Flash [18]—an existing globally-optimal k-anonymity algorithm—relies on the ability to hold the entire dataset in "main-memory" (referring to the primary-volatile memory type RAM) to build the required memory layouts. Flash [18] mentions that in the case of limited main-memory or very large datasets, a disk-based implementation might be required, which has not yet been developed. A complementary strategy for high-dimensional datasets, Lightning [10], provides a heuristic, best-effort solution. However, dataset dimensionality is not a direct determinant of dataset volume, and algorithms such as Lightning, Flash, and Incognito [17], may still struggle when the dataset size far exceeds the available RAM. These algorithms can perform k-anonymisation only on the portion of the data that fits in the available physical memory. Thus, the solutions obtained are far from optimal. In comparison, we offer a solution that outperforms these algorithms with the same constraints.

III. SKALD: OUR PROPOSED SOLUTION

In this paper, we focus on a real-world scenario where the dataset size exceeds the amount of RAM available. The data thus needs to be parsed in chunks to be anonymised. Although each chunk only gives us a limited view of the dataset, we require a global view of all the chunks for efficient k-anonymisation. To solve this problem, we present the SKALD algorithm that is capable of providing a significantly improved solution when presented with batched datasets. Our algorithm is able to parse through chunks of data to retain and combine sufficient statistics from each chunk to k-anonymise the entire dataset without needing to hold the entire dataset in main memory at once. In order to do this, we make use of the fundamentals of the Optimal Lattice Anonymisation (OLA) developed in [9] for navigation of the generalisation lattice, while modifying it to suit our methodology.

A. Bounding the Number of Histogram Bins

The amount of available physical memory (RAM) constrains the amount of space available for a histogram of a dataset. Equivalently, this also imposes a bound on the maximum number of bins in the histogram, which we denote by N_{RAM}. Each bin in our histogram represents a specific combination of multi-dimensional values across the QIDs.

Consider a dataset D with L categorical and M numerical QIDs. For the ith categorical QID with resolution R_i, let $C_i[R_i]$ denote the number of unique categorical values. For numerical QIDs, we assume the existence of metadata that provides us with the maximum and minimum values. For the jth numerical QID, let $Q_{j,max}$, $Q_{j,min}$ and W_j denote the maximum value, minimum value, and bin width respectively. The number of bins in the histogram are given by

$$N_{data} = \left[(\prod_{i=1}^{L} C_i[R_i])(\prod_{j=1}^{M} (Q_{j,max} - Q_{j,min} + 1)/W_j) \right] + 1$$

(1)

where,

- $\prod_{i=1}^{L} C_i[R_i]$ denotes the number of unique combinations of values across categorical QIDs in the dataset for a given resolution
- $\prod_{j=1}^{M} (Q_{j,max} - Q_{j,min} + 1)/W_j$ denotes the number of unique combinations of values across numerical QIDs with respectively chosen bin widths
- The addition of the one to the product of both the above terms represents the suppressed records, which constitute their own bin in the histogram

The upper bound on the number of bins of a histogram due to the RAM constraint is given by

$$N_{data} \leq N_{RAM} \qquad (2)$$

The resolution R_i for categorical QIDs and the bin widths W_j for numerical QIDs should be chosen to retain as much fine-grained information as possible, while satisfying (2). The exact procedure for selecting these parameters is described below.

B. The Three Phases of the SKALD Algorithm

The SKALD algorithm can be broken down into three execution phases explained below.

1) Phase 1: Encoding Numerical QIDs and Choosing (R_i, W_j): There are two operations in the first phase. The first is an encoding procedure, wherein sparse numerical QIDs are encoded to reduce the total number of histogram bins. If there is a large discrepancy between the maximum number of possible values and the number of values present in the dataset, then there is a chance that successful k-anonymisation will require a larger bin width than desired for that particular QID. This increases the amount of information loss in the anonymised dataset and necessitates the use of encoding to reduce the sparsity. To perform this encoding, we pass the chunks of data one at a time, taking the unique values of the numerical QIDs to be encoded in the dataset, sorting the values, and then assigning a new identifier to that value. A key-value map is created during the encoding process and is later used for decoding.

For the second operation, we need to determine appropriate values for (R_i, W_j) to retain as much fine-grained information as possible while satisfying (2). We begin by defining the finest possible resolution $R_{i,\text{init}}$ for categorical QIDs and finest bin width $W_{j,\text{init}}$ for numerical QIDs.

For categorical attributes, a generalisation hierarchy is used to progressively abstract information at different levels. For example, a specific profession such as *Mixed Media Artist* can be generalised to *Artist* at the first generalisation level, *Creative* at the second, and a broader category such as *Non-Technical Sector* at the third. Each level of abstraction reduces the granularity of information, ensuring that categorical attributes can be generalised in a structured manner. For these attributes, $R_{i,\text{init}}$ is chosen as the finest resolution that retains all the categorical value options in the dataset.

For numerical attributes, the finest bin width is determined by the semantic meaning of the unit of measurement. For instance, attributes such as *Age*, typically recorded in whole *years*, or *Weight*, expressed in *kilograms* or *pounds*, naturally have a finest bin width of 1 unit, as further granularity is not meaningful in most analytical contexts. Similarly, for a QID such as *Latitude*, which is measured in *degrees* with up to six decimal places of precision, the finest bin width $W_{j,\text{init}}$ would correspond to 10^{-6} degrees, aligning with the smallest meaningful distinction in spatial coordinates.

Using this $(R_{i,\text{init}}, W_{j,\text{init}})$ pair, we can construct a generalisation lattice where the amount of generalisation increases with every level. The lattice is then traversed, starting from the middle node of the middle level. We select this node as it allows us to eliminate the maximum possible nodes using predictive tagging [9]. Having labelled all nodes in the lattice as compliant or non-compliant based on (2), we then utilise the precision metric defined in [11] to select the node with the best value of precision among all the passing nodes, labelling this node $(R_{i,\text{RAM}}, W_{j,\text{RAM}})$. In case two nodes have the same value of precision, we can utilise a pre-determined hierarchy of semantic importance for the QIDs to break the tie.

2) Phase 2: Histogram and Lattice Construction: In the second phase, the data is passed chunk-wise through the algorithm for a second time. A histogram is created to store the counts of users that fall into the bins defined by $(R_{i,\text{RAM}}, W_{j,\text{RAM}})$. This histogram is updated for each chunk, so the final histogram created after passing all the chunks has the aggregate counts from all the chunks, i.e. the entire dataset. A new generalisation lattice is then constructed, using $(R_{i,\text{RAM}}, W_{j,\text{RAM}})$ as the root node. For all nodes that pass the k-anonymity condition within the desired suppression limit, we compute the monotonic Discernibility Metric (DM*) proposed in [9]. This enables us to select the node with the lowest (best) DM* value, corresponding to minimal information loss, among all the nodes that satisfy k-anonymity. We label this node $(R_{i,\text{final}}, W_{j,\text{final}})$.

3) Phase 3: Decoding and Generalising the Dataset: In the third phase the data chunks are passed a final time. In this pass, the encoded QIDs are decoded and the chunks are generalised according to the bin widths defined in $(R_{i,\text{final}}, W_{j,\text{final}})$. This gives us the final k-anonymised dataset.

The operations within each of the three phases of the SKALD algorithm are compactly summarized in Fig. 2.

IV. EXPERIMENTAL RESULTS

A. Synthetic Data

We create a synthetic dataset generated using a well-known library for creating pseudo-random but realistic data - *pyfaker*. This dataset is modelled on a real-world medical dataset with attributes that one would expect to find in a sample of data collected from patients by a medical centre or hospital. A snippet of this dataset is shown in Table I. In this format, we generate 125 chunks of data, each chunk with 1 million rows, where each row corresponds to a medical record of a unique user. This amounts to a total of 125 million records
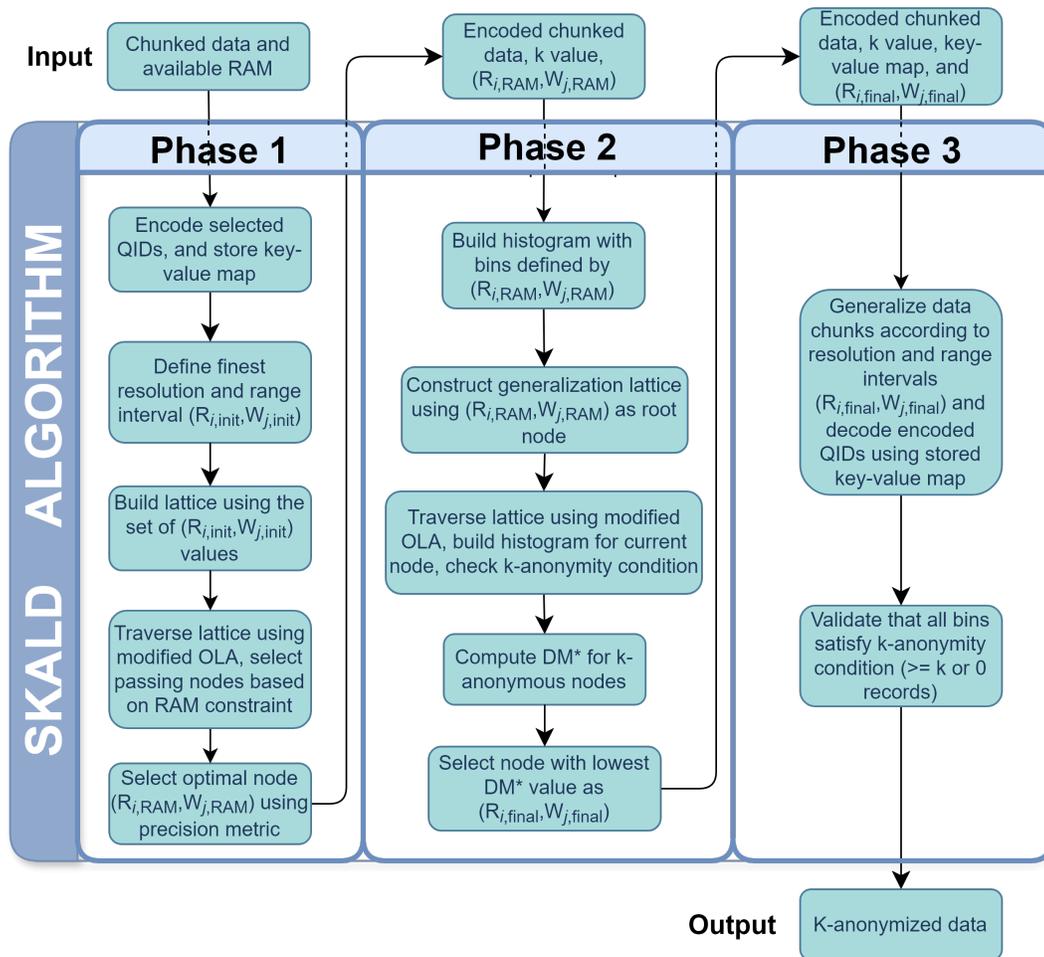
Fig. 2. The SKALD algorithm

in the entire dataset. The direct identifiers in this dataset are the "Patient ID", "Name" and "Address" attributes, which can be used to identify an individual without any external information, and are therefore completely suppressed in the anonymised dataset. The dataset has 2 categorical QIDs: ("Blood Group", "Profession"), 3 numerical QIDs: ("Age", "BMI", "PIN Code"), and the sensitive attribute is "Health Condition". If a hospital were to publicly release this dataset for analysis, then the attribute being released would most likely be the "Health Condition", and it would be important to anonymise the dataset to ensure that no individual could be tied to a particular health condition in order to preserve user privacy.

B. Generalisation Hierarchy

We construct generalisation hierarchies for each QID. These hierarchies define how raw data values can be systematically transformed into more generalised forms across multiple levels. The generalisation levels for both categorical and numerical attributes form the basis for building the anonymisation lattice used in our algorithm. The section below shows how the generalisation hierarchies are built for each QID in our synthetic dataset.

1) Categorical Data:

- **Blood Group:**
  - Level 1 (Resolution = 1):
    Original blood groups (8 categories):
    `A+, A-, B+, B-, AB+, AB-, O+, O-`
  - Level 2 (Resolution = 2):
    Generalised by type (4 categories):
    `A, B, AB, O`
  - Level 3 (Resolution = 3):
    Maximally generalised (1 category):
    `*` (representing a suppressed value)

- **Profession:**
  - Level 1 (Resolution = 1)
    Original professions (16 categories):
    `Medical Specialists, Allied Health, Nursing, Healthcare Support,...`
  - Level 2 (Resolution = 2)
    Generalised into broader domains (4 categories):
    `Healthcare, Education, Creative, Engineering`

TABLE I

A SAMPLE OF THE SYNTHETICALLY GENERATED DATASET.

Patient ID	Name	Address	Blood Group	Profession	Age	BMI	PIN Code	Health Condition
P0001-9112-1	Ekaraj Parikh	156, Kala Road	O+	Data Scientist	33	23.8	560044	Dementia
P0002-8603-2	Jack Shankar	15/402, Ganesh Chowk	A-	Project Manager	38	25.3	560008	Dementia
P0003-6406-1	Nakul Prabhakar	91/335, Sule Circle	B-	Software Engineer	60	25.9	560018	Gout
P0004-8305-3	Chandani Lanka	104, Bahl Marg	B-	Product Manager	41	23.9	561164	Asthma
P0005-9512-2	Alka Kumar	052, Nadig Zila	AB-	Business Analyst	29	22.2	560059	Asthma

TABLE II

A SAMPLE OF A 50-ANONYMISED DATASET WITH (2,1),(1, 24, 32)

Blood Group	Profession	Age	BMI	PIN Code	Health Condition
O	Data Scientist	[33 - 33]	[12 - 36)	[560032 - 560063]	Dementia
A	Project Manager	[38 - 38]	[12 - 36)	[560000 - 560031]	Dementia
B	Software Engineer	[60 - 60]	[12 - 36)	[560000 - 560031]	Gout
B	Product Manager	[41 - 41]	[12 - 36)	[561164 - 561195]	Asthma
AB	Business Analyst	[29 - 29]	[12 - 36)	[560032 - 560063]	Asthma

- Level 3 (Resolution = 3)
  Grouped by sector (2 categories):
  `Service Sector, Non-Service`
- Level 4 (Resolution = 4)
  Maximally generalised (1 category):
  `*` (representing a suppressed value)

2) Numerical Data:

- **Age:**
  - Level 1 (Bin width = 1): Raw values
    `33 → [33 - 33]`
  - Level 2 (Bin width = 2):
    `33 → [32 - 33]`
  - Level 3 (Bin width = 4):
    `33 → [32 - 35]`
    . . . and so on. The bin width doubles at each subsequent level until level 7. The final level which spans the entire domain is the maximally generalised level:
  - Level 8 (Bin width = 67):
    `33 → [19 - 85]`

- **BMI:**
  - Level 1 (Bin width = 0.1): Raw values with original precision
    `22.5, 23.8, 27.6`
  - Level 2 (Bin width = 1): Values are rounded or grouped into bins of width 1
    `23.8 → [23 - 24)`
  - Level 3 (Bin width = 2):
    `23.8 → [23 - 25)`
    . . . and so on. The bin width doubles at each subsequent level until level 5. The final level which spans the entire domain is the maximally generalised level:
  - Level 6 (Bin width = 24):
    `23.8 → [12 - 36)`

- **PIN Code:** The synthetic dataset has 1347 unique PIN Codes. Due to sparsity, PIN Codes are first encoded into integers, where the smallest numerical value of the PIN Code in the dataset is mapped to 1 and the largest is

mapped to 1347.

- Level 1 (Bin width = 1):
  `570025 (encoded to 221) → [221 - 221]`
- Level 2 (Bin width = 2):
  `221 → [221 - 222]`
- Level 3 (Bin width = 4):
  `221 → [221 - 224]`
  . . . and so on. The bin width doubles at each subsequent level until level 11. The final level which spans the entire domain is the maximally generalised level:
- Level 12 (Bin width = 1347):
  `221 → [0 - 1346]`

For example, consider a node in the generalisation lattice, where the (R_i, W_j) pair is represented as ((2, 1),(1, 24, 32)). The first tuple (2, 1) corresponds to resolution of the categorical QIDs, indicating that Blood Group is generalised to Level 2 and Profession to Level 1. The second tuple (1, 24, 32) corresponds to the numerical QIDs, implying that Age is generalised with a bin width of 1, BMI with a bin width of 24, and encoded PIN Code with a bin width of 32.

V. RESULTS

To conduct our experiments we use a commodity-class server machine, equipped with a 16-thread (8-core, 16-thread) Intel Core i7-10700 CPU @ 2.90 GHz (up to 4.8 GHz), with a 16 MB L3 cache and 64 GB RAM.

To evaluate the effectiveness of SKALD, we compare it against ARX, a widely adopted k-anonymisation framework. ARX supports both the globally optimal Flash algorithm [18] as well as the heuristic-based Lightning algorithm [10]. Lightning is tailored for high-dimensional data and trades optimality for speed, making it unsuitable for our evaluation, which prioritizes data utility. Since our primary goal is to minimize information loss under memory constraints rather than optimize for runtime, we restrict the comparison to Flash.

A representative snippet of the output generated by SKALD for $k = 50$ using 125 chunks is shown in Table II. This

configuration resulted in the $(R_{i,\text{final}}, W_{j,\text{final}})$ pair: ((2, 1), (1, 24, 32)).

No. of Chunks	SKALD Generalisation Node	ARX Generalisation Node
1	$((3, 1), (4, 24, 1347))$	$((3, 3), (67, 16, 4))$
5	$((3, 1), (1, 24, 1347))$	$((3, 3), (67, 16, 4))$
25	$((3, 1), (1, 24, 256))$	$((3, 3), (67, 16, 4))$
125	$((3, 1), (1, 24, 128))$	$((3, 3), (67, 16, 4))$

Table III compares the generalisation nodes of SKALD and ARX for different numbers of chunks when $k = 1000$. SKALD algorithm extracts and combines sufficient statistics from each chunk to provide reduced bin widths with an increase in the number of chunks, where each data chunk contains 1 million records. This reduced bin width retains more fine-grained information. ARX, on the other hand, anonymises the data on a per-chunk basis and shows no improvement in the generalisation node with an increasing number of chunks.

To quantify information loss in the anonymised outputs, we use the *monotonic variant of the Discernibility Metric*, denoted as DM*, and first proposed in [9]. The metric is defined as

$$\text{DM}^* = \sum_{E \in \mathcal{E}} |E|^2 + s^2 \tag{3}$$

where,

- $\mathcal{E}$ is the set of equivalence classes,
- $|E|$ is the size of equivalence class E,
- s is the number of suppressed records.

A lower value of DM* implies lower information loss. To facilitate comparison, we define the DM^* Ratio as

$$\text{DM}^* \text{ Ratio} = \frac{\text{DM}^*_{\text{ARX}}}{\text{DM}^*_{\text{SKALD}}} \tag{4}$$

A DM* ratio greater than 1 indicates that SKALD achieved lower information loss than ARX for the same parameters.

For our experiments, we varied the anonymity parameter $k \in (10, 50, 250, 1000)$ with a varying number of chunks belonging to the set $\{1, 5, 25, 125\}$. Fig. 3 presents the results as a bar chart, and visualises the improved performance of SKALD against ARX with an increase in the number of chunks. In particular, the performance improvement using SKALD for 25 chunks is at least *four-fold* and for 125 chunks is at least *nine-fold* across different values of k. We can infer from these results that the performance gap between SKALD and ARX in terms of DM* increases as the number of chunks and thus, the dataset size, increases.

VI. CONCLUSIONS

Local hardware systems are preferred to carry out anonymisation tasks to ensure an auditable chain of control and accountability. The available hardware resources provide an upper bound on the maximum size of data that can be processed at a time. Large datasets thus need to be broken up into smaller chunks. In this work we proposed SKALD, a chunk-based k-anonymisation algorithm that is performant within
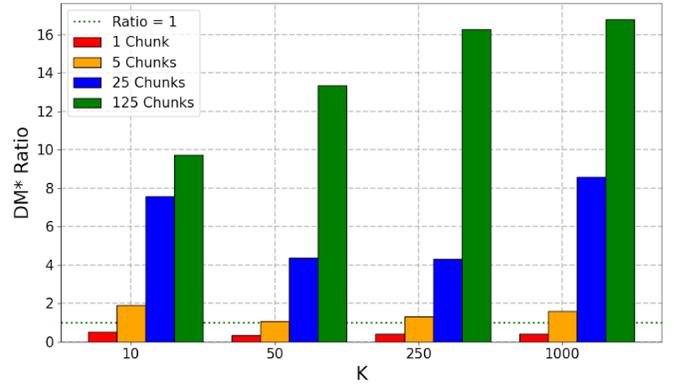

Fig. 3. DM* Ratio vs. k for different numbers of chunks

the memory bounds imposed by local hardware. SKALD significantly outperforms ARX's Flash algorithm (in terms of reducing information loss) with an increase in the number of data chunks. SKALD extracts and combines sufficient statistics from each chunk to maintain a global data perspective to provide better output utility. The SKALD algorithm's performance is independent of the size of the data chunks. To perform the comparison, we generated a synthetic dataset modelled on real-world data. Compared to ARX, the results of the experiments on this dataset showed at least a *nine-fold* reduction in the monotonic Discernibility Metric (DM*) value of SKALD for 125 chunks, across different values of k - where each data chunk was designed to contain 1 million records.

Future work includes extending SKALD to support *l*-diversity and *t*-closeness, as well as optimising the computational requirements of the SKALD algorithm.

REFERENCES

[1] C. Dwork, A. Smith, T. Steinke, and J. Ullman, "Exposed! A survey of attacks on private data," *Annual Review of Statistics and Its Application*, vol. 4, pp. 61–84, 2017, ISSN: 2326831X. DOI: 10.1146/annurev-statistics-060116-054123.

[2] J. Abowd, T. Adams, R. Ashmead, *et al.*, "The 2010 Census Confidentiality Protections Failed, Here's How and Why," *SSRN Electronic Journal*, 2023. DOI: 10.2139/ssrn.4675183.

[3] A. Gadotti, L. Rocher, F. Houssiau, A. M. Creţu, and Y. A. de Montjoye, "Anonymization: The imperfect science of using data while preserving privacy," *Science advances*, vol. 10, no. 29, 2024, ISSN: 23752548. DOI: 10.1126/sciadv.adn7053.

[4] Government of India, *Digital Personal Data Protection Act 2023*, New Delhi, 2023. [Online]. Available: https://www.meity.gov.in/writereaddata/files/Digital%20Personal%20Data%20Protection%20Act%202023.pdf.

[5] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002, ISSN: 02184885. DOI: 10.1142/S0218488502001648.

[6] DSCI, "The Future of Data Protection in India - Part 1: A Roadmap for Regulators," Data Security Council of India, Tech. Rep., 2023. [Online]. Available: https://www.dsci.in/resource/content/future-data-protection-india.

[7] European Parliament and Council of the European Union, *Regulation (EU) 2016/679 of the European Parliament and of the Council*, OJ L 119, 4.5.2016, p. 1–88, May 4, 2016. [Online]. Available: https://data.europa.eu/eli/reg/2016/679/oj.

[8] N. Chakraborty, A. Tandon, K. Reddy, *et al.*, "Building a privacy web with spider - secure pipeline for information de-identification with end-to-end encryption," in *2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS)*, 2025, pp. 1–3. DOI: 10.1109/COMSNETS63942.2025.10918872.

[9] K. El Emam, F. K. Dankar, R. Issa, *et al.*, "A Globally Optimal k-Anonymity Method for the De-Identification of Health Data," *Journal of the American Medical Informatics Association*, vol. 16, no. 5, pp. 670–682, 2009, ISSN: 10675027. DOI: 10.1197/jamia.M3144.

[10] F. Prasser, R. Bild, J. Eicher, H. Spengler, F. Kohlmayer, and K. A. Kuhn, "Lightning: Utility-driven anonymization of high-dimensional data," *Transactions on Data Privacy*, vol. 9, no. 2, pp. 161–185, 2016, ISSN: 20131631.

[11] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowldege-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002, ISSN: 02184885. DOI: 10.1142/S021848850200165X.

[12] T. Meurers, R. Bild, K. M. Do, and F. Prasser, "A scalable software solution for anonymizing high-dimensional biomedical data," *GigaScience*, vol. 10, no. 10, pp. 1–13, 2021, ISSN: 2047217X. DOI: 10.1093/gigascience/giab068.

[13] S. Garfinkel, J. Near, A. Dajani, P. Singer, and B. Guttman, "De-identifying government datasets:" National Institute of Standards and Technology, Tech. Rep., Sep. 2023. DOI: 10.6028/NIST.SP.800-188.

[14] R. Motwani and Y. Xu, "Efficient Algorithms for Masking and Finding Quasi-Identifiers," *Vldb '07*, pp. 23–28, 2007.

[15] L. Sweeney, "Simple Demographics Often Identify People Uniquely," *Data Privacy Working Paper 3*, pp. 1–34, 2000.

[16] A. Meyerson and R. Williams, "On the complexity of optimal k-anonymity," *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, vol. 23, pp. 223–228, 2004. DOI: 10.1145/1055558.1055591.

[17] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-Domain K-Anonymity," *ACM SIGMOD International Conference on Management of Data*, 2005.

[18] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, and K. A. Kuhn, "Flash: Efficient, stable and optimal k-anonymity," *Proceedings - 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust and 2012 ASE/IEEE International Conference on Social Computing, SocialCom/PASSAT 2012*, pp. 708–717, 2012. DOI: 10.1109/SocialCom-PASSAT.2012.52.