DataShield: Implementing (k, m, t)-Anonymity for Transactional Datasets

Arwaz Khan*, Aryam Shrivastava*, Joshua Joy†
*Department of Computer Science and Engineering, NIT Delhi, India
†Department of Computer Science and Engineering (Analytics), NIT Delhi, India

***Abstract*—This project presents a transactional data privacy system based on an enhanced privacy model called (k, m, t)-Anonymity. Developed using Python technologies such as Flask, Pandas, and NumPy, the system offers robust privacy guarantees for uploaded datasets. It employs Genetic Algorithm (GA)-based clustering to group records effectively, enforces t-closeness to protect sensitive attribute distributions, and applies km-anonymity through vertical partitioning to preserve structural privacy. Through a web-based interface, users can upload CSV files, configure privacy parameters (k, m, t), and download the resulting anonymized data. The platform combines advanced privacy techniques with ease of use, making it a practical solution for secure data publishing.**

***Index Terms*—k-anonymity, m-anonymity, t-closeness, transactional data, data privacy, anonymization, clustering**

I. Introduction

With the rise of data-driven systems, transactional datasets—such as purchase logs and health records—are increasingly exposed to privacy risks. Traditional methods like *k*-anonymity reduce identity disclosure but fall short against background knowledge and skewed data distributions.

To address these limitations, this project introduces the (k, m, t)-Anonymity model, which combines:

- ***k*-Anonymity:** Prevents identity disclosure.
- ***m*-Itemset Anonymity:** Hides sensitive item co-occurrences.
- ***t*-Closeness:** Limits attribute disclosure via distribution control.

A Genetic Algorithm (GA)-based approach is used for clustering and vertical partitioning, balancing privacy with utility. The solution is deployed as a Python-Flask web app, enabling users to upload data, set privacy parameters, and download anonymized results for secure, real-world use.

II. Objective

The objective of this project is to develop an enhanced privacy-preserving framework for transactional data using the (k, m, t)-anonymity model. The project addresses key research gaps and implements practical solutions as outlined below:

- **Research Gap:** Traditional *k*-anonymity is insufficient against inference and attribute disclosure attacks in transactional datasets.
- **Gap in Item Co-occurrence Protection:** Frequent or rare itemsets can still lead to re-identification even after *k*-anonymization.
- **Distribution Disclosure:** Skewed sensitive attribute distributions increase vulnerability, not addressed well by *l*-diversity.
- **Objective:** Integrate *k*-anonymity, *m*-itemset anonymity, and *t*-closeness into a unified framework to mitigate identity and attribute disclosure.
- **Use Genetic Algorithm (GA):** Efficiently cluster transactions to satisfy privacy constraints with minimal information loss.
- **Apply Vertical Partitioning:** Enforce *m*-itemset anonymity by ensuring frequent itemsets appear in at least k records.
- **Maintain Attribute Distribution:** Enforce *t*-closeness using KL-divergence to protect against attribute inference.
- **Web Interface Deployment:** Provide a user-friendly, Python Flask-based platform for uploading data, setting privacy parameters, and downloading anonymized output.

III. Motivation

Transactional datasets often contain sensitive information that can lead to privacy breaches if not properly anonymized. While *k*-anonymity protects against identity disclosure, it is insufficient against attribute linkage and homogeneity attacks. To address these gaps, *m*-itemset anonymity and *t*-closeness provide stronger safeguards by hiding rare item combinations and preserving sensitive attribute distributions. The integrated (k, m, t)-anonymity model offers a comprehensive solution for protecting both identity and sensitive information in real-world data publishing.

IV. Dataset Description

To evaluate the effectiveness of the (k, m, t)-anonymity model, the DATASHIELD system was tested on both real-world and synthetic transactional datasets:

- **BMS-WebView1:** Click-stream dataset with 59,601 transactions and 497 items, used to evaluate sparse transaction performance [1].
- **INFORMS Dataset:** Healthcare dataset with 102,578 records and 561 items, representing high-dimensional clinical data [2].
- **Synthetic Dataset (Synthea):** Generated using Synthea with 500,000 records and 1,293 unique items to simulate realistic healthcare scenarios [3].

- **Synthetic Dataset (Faker):** Created using the `Faker` library to generate 500,000 fake health profiles for testing anonymization logic [4].

These datasets offer diverse characteristics to test the privacy, utility, and scalability of the proposed system.

V. Architecture and Workflow

The (k, m, t)-anonymity framework follows these core steps:

1) **Data Upload:** Users upload a transactional dataset with quasi-identifiers and sensitive items.
2) **Parameter Input:** Values for k, m, and t are provided by the user.
3) **Preprocessing:** The dataset is cleaned and item types are identified.
4) **Clustering (GA):** Transactions are grouped using a Genetic Algorithm to satisfy k and t constraints.
5) **Vertical Partitioning:** Enforces km-anonymity by splitting itemsets across clusters.
6) **t-Closeness Adjustment:** Ensures sensitive item distribution in clusters matches the global distribution.
7) **Output:** An anonymized dataset is generated and made available for download.

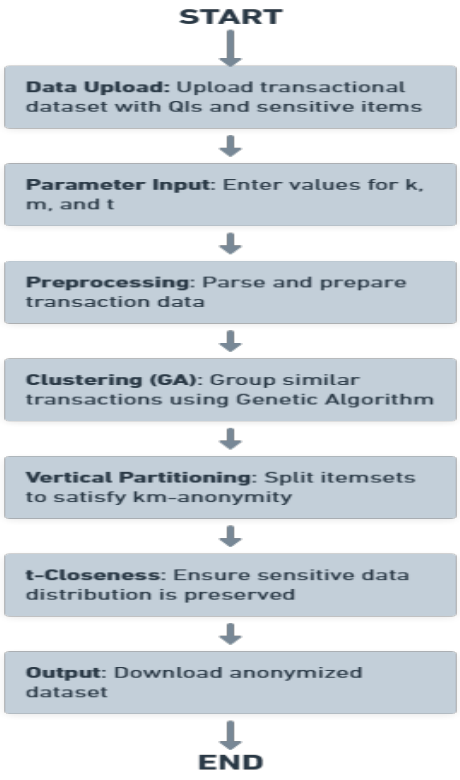

Fig. 1: Workflow Architecture for (k, m, t)-Anonymity-Based Transactional Data Privacy

VI. Implementation

A. Anonymization Workflow

The anonymization process in DATASHIELD follows a two-phase approach using Genetic Algorithm-based clustering and VerPart vertical partitioning. The process ensures that each cluster meets the requirements of (k, m, t)-anonymity. The generalization in this system is implicit through clustering and vertical partitioning rather than explicit value mapping.

TABLE I: Phases in (k, m, t)-Anonymity Implementation

Phase	Description
Data Upload	User uploads a transactional CSV dataset via the web interface.
Parameter Input	User specifies values for k, m, and t to define privacy thresholds.
Preprocessing	Dataset is parsed to separate sensitive and non-sensitive items.
Clustering (GA)	Genetic Algorithm clusters transactions based on similarity and t-closeness.
Fitness Calculation	Clusters are evaluated using similarity score and KL-divergence deviation from t.
Selection & Crossover	Chromosome pairs are selected (e.g., nearest neighbor) and crossed to generate better clusters.
Vertical Partitioning (VerPart)	Non-sensitive items are partitioned to satisfy km-anonymity by removing item associations.
Anonymized Output	The final dataset is anonymized while preserving structure and minimizing information loss.

B. Key Functions in the Anonymizer Code

TABLE II: Functions in the DATASHIELD Anonymization System

Function	Description
load_dataset	Loads the uploaded transactional CSV file and parses into itemsets.
extract_items	Identifies sensitive and non-sensitive items from each transaction.
initialize_population	Randomly creates initial clusters (chromosomes) of at least size k.
fitness_function	Computes fitness score based on transaction similarity and deviation from t.
crossover_chromosomes	Applies crossover operations (e.g., single-point) to improve population fitness.
verpart_partitioning	Applies vertical partitioning to itemsets in each cluster to enforce km-anonymity.
kl_divergence	Calculates KL-divergence to evaluate how close sensitive item distributions are to global distribution.
export_anonymized_data	Outputs the anonymized transactional data in CSV format for download.

VII. Results and Output

The DATASHIELD system was evaluated on a synthetic healthcare dataset with 500,000 records containing attributes such as Name, Age, Gender, Pincode, Disease, and Medication.

The anonymization process preserved all records while masking quasi-identifiers such as Name, Gender, and Pincode.

Algorithm 1 (k, m, t)-Anonymity Based Transactional Data Anonymization

Require: Dataset D, parameters k, m, t
Ensure: Anonymized dataset D'

```
Preprocess D to identify sensitive and non-sensitive items
Initialize population of clusters using Genetic Algorithm
for each generation do
    Evaluate fitness of each cluster using similarity and KL-divergence
    Select parent clusters based on fitness
    Apply crossover to generate new clusters
    Apply mutation to maintain diversity
    Replace least-fit clusters with new ones
end for
For each cluster in best solution:
    Apply VerPart to achieve m-itemset anonymity
    Adjust sensitive item distribution to enforce t-closeness
Combine clusters into final anonymized dataset D'
return D'
```

Sensitive attributes like Disease and Medication were retained. As shown in Fig. 2, approximately 62.5% of the total attribute values were masked to ensure compliance with the (k, m, t) model using parameters $k = 5$, $m = 2$, and $t = 0.12$.

TABLE III: Anonymization Summary

Property	Value
Original Records	500,000
Anonymized Records	500,000
Masking Percentage	62.5%
Preserved Attributes	Disease, Medication
Masked Attributes	Name, Gender, Pincode, Doctor, Hospital
Anonymization Time	$\sim$ Under 1 minute

Additionally, Fig. 3 shows that increasing k values led to a reduction in retained records, highlighting the privacy-utility trade-off.

Figures:

- Fig. 2: Pie chart showing percentage of masked vs unmasked data.
- Fig. 3: Line chart showing record retention as k increases.

VIII. Challenges

Implementing the (k, m, t)-anonymity model posed several challenges:

- **Privacy-Utility Trade-off:** Achieving strong privacy without excessive data distortion.
- **Variable-Length Transactions:** Difficulty in forming consistent equivalence classes.
- **Parameter Tuning:** Selecting optimal k, m, and t values to balance risk and utility.
- **Skewed Sensitive Distributions:** Enforcing t-closeness becomes harder with imbalanced data.

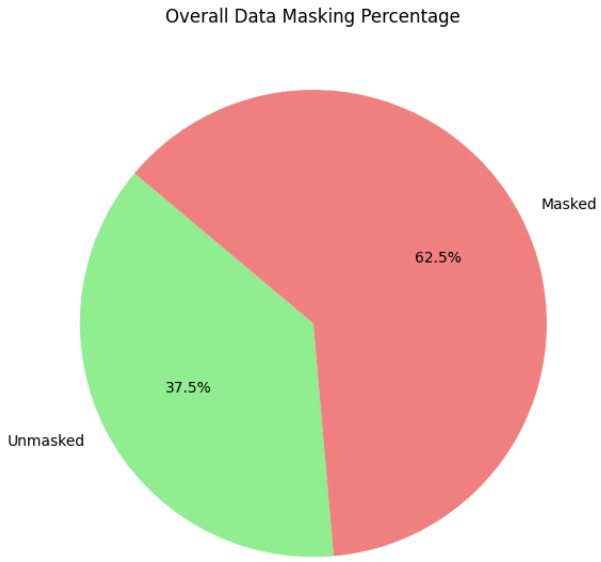

Fig. 2: Overall Data Masking Percentage

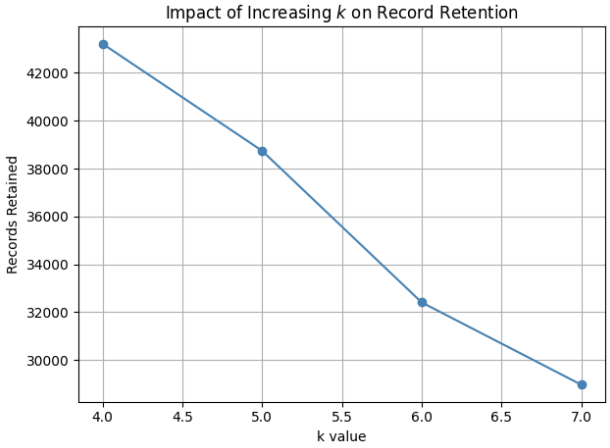

Fig. 3: Impact of Increasing k on Record Retention

- **Computational Overhead:** Genetic Algorithm and clustering increase runtime for large datasets.
- **Sparse Itemsets:** Rare items reduce generalization effectiveness and increase suppression.
- **Partitioning Complexity:** Ensuring m-itemset anonymity while preserving data semantics.
- **Scalability:** Performance issues arise with high-volume transactional data.
- **Output Validation:** Ensuring the final data remains accurate and usable post-anonymization.

IX. Future Scope

Future improvements to the DATASHIELD system may include:

- **Adaptive Parameters:** Dynamically tuning k, m, and t based on data context.

- **Real-Time Anonymization:** Supporting live anonymization for streaming data.
- **Scalability:** Using parallel or distributed methods for large datasets.
- **ML Integration:** Applying anonymized data in privacy-preserving machine learning.
- **User Controls:** Allowing users to define custom privacy settings.

REFERENCES

[1] BMS-WebView1 dataset. Available at: http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php

[2] INFORMS Data Mining Challenge dataset. Available at: https://sites.google.com/site/informsdataminingcontest/data

[3] Walonoski, Jason, et al. "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record." *Journal of the American Medical Informatics Association* 25.3 (2018): 230-238.

[4] Gérard, J. (2014). Faker: Python package for generating fake data. https://faker.readthedocs.io/en/master/

[5] Puri, Vartika, Parmeet Kaur, and Shelly Sachdeva. "(k, m, t)-anonymity: Enhanced privacy for transactional data." *Concurrency and Computation: Practice and Experience* 34.18 (2022): e7020.

[6] L. Sweeney, "k-anonymity: A model for protecting privacy," Int. J. Uncertainty, Fuzziness Knowl.-Based Syst., vol. 10, no. 5, pp. 557–570, 2002.