

HEALTH DATA PRIVACY USING K-ANONYMITY AND COMPARATIVE ANALYSIS OF K-ANONYMIZATION ALGORITHMS

Prepared by:

Munilakshmi G J – 20BCI0190

ACKNOWLEDGEMENT

we overcome with humility and gratitude to express our sincere appreciation to everyone who has assisted us in turning these concepts—far beyond their level of simplicity—into something tangible.

We would like to express our sincere gratitude to our renowned teacher, Mrs. Anbarasi, who provided us with the fantastic opportunity to complete this project. She also assisted us in conducting extensive research, which allowed us to learn a great deal of new information. We are very appreciative of her. Without the assistance and direction of MY parents and friends, no endeavor at any level can be successfully completed.

Table of contents:

S.NO.	TOPIC	PG. NO.
	Abstract	4
1	1.1 scope	5
	1.2 Introduction	5-6
	1.3 Literature survey	7-12
2	2.1 proposed system	13
	2.2 software requirements	13
3	3.1 results & discussion	14-30
	3.2 conclusion	30
	3.3 references	31-32

Abstract

The enormous amount of information that is being gathered on people has made it more difficult to secure their privacy. With the use of an ideal de-identification algorithm that meets the k -anonymity condition, this research seeks to protect health data. Due to the fact that privacy-preserving Data Publishing is a current subject of research, numerous anonymization techniques have been put forth. Additionally, this study seeks to give a classification and analysis of several privacy-preserving anonymization methods including Datafly, Incognito, and Mondrian.

Section 1

1.1 Scope

The vast amount of data being collected about individuals has brought new challenges in protecting their privacy. Also, The COVID-19 epidemic is putting health systems and hospitals under unprecedented stress, and their IT departments are also struggling with key skill and manpower shortages as they fight relentless cyberattacks. The scope of this project is to aim at protecting health data using an optimal de-identification algorithm that satisfies the k-anonymity criterion. this study also seeks to give a classification and analysis of several privacy-preserving anonymization methods including Datafly, Incognito, and Mondrian.

1.2 Introduction

Currently, annual increases in data volumes are exponential. The privacy of the people who are the data's subjects is violated when this data is shared and distributed. One of the most crucial challenges in the processing of huge data is privacy protection. To draw conclusions about a person's identification, one might mix various datasets or have information about their background. Re-identification of a person is accomplished by connecting characteristics, or quasi-identifiers, such as gender, date of birth, or ZIP code.

De-identification of data beforehand or as soon as possible is one method to aid health research and reduce some of the issues. K-anonymity is a widely used de-identification criterion. Each record in a dataset must match at least one other $k-1$ record according to this criterion in order for the possibly identifying variables to be used. With the use of an ideal de-identification algorithm that meets the k-anonymity condition, this research seeks to protect health data.

Additionally, it can be challenging to identify and choose the best algorithm for a specific publication scenario given the abundance of algorithms available and the paucity of data regarding their performance. Since they are primarily concerned with proving the

advantages of the proposed algorithm over some of the previously proposed ones, the initial experimental evaluations of these algorithms are typically constrained. Therefore, we think there is a significant need to expand the current evaluations of these anonymization algorithms to include a wider range of experimental setups, and a comparison of the algorithms' effectiveness and efficiency is required.

1.3 Literature survey

TITLE	AUTHOR & YEAR	PURPOSE	DATASETS USED	METHODS/ALGORITHMS
Comparison and Analysis of Anonymization Techniques for Preserving Privacy in Big Data.	Johny Antony P, 2017	Comparative analysis of algorithms	adult data set	Comparative analysis was done on k-anonymity, l-diversity, t-closeness, differential privacy, and slicing. experiments are conducted on a machine with Intel ® Core TM i5-2120 CPU @ 3.30 GHZ, 4 GB RAM, Window 7, JAVA –JDK 8.0.
A Globally Optimal k-Anonymity Method for the De-Identification of Health Data	KHALED EL EMAM, PHD, FIDA KAMAL DANKAR, PHD, ROMEO ISSA, MS, ELIZABETH JONKER, BA, DANIEL AMYOT, PHD, ELISE COGO, ND, JEAN-PIERRE CORRIVEAU, PHD, MARK WALKER, MS, MD, SADRUL CHOWDHURY, MS, REGIS VAILLANCOURT, BPHARM, PHARMD, TYSON ROFFEY, BA, JIM BOTTOMLEY, BSCH, MHA, 2022	Comparative analysis and de-identification algorithm	-	Analysis done on three existing k-anonymity algorithms, Datafly, Samarati, and Incognito, on six public, hospital, and registry datasets for different values of k and suppression limits. The new OLA algorithm, whose objective is to find optimal node in the lattice.

TITLE	AUTHOR & YEAR	PURPOSE	DATA SETS USED	METHODS/ALGORITHMS
A Systematic Comparison and Evaluation of k-Anonymization Algorithms for Practitioners	Vanessa Ayala-Rivera* , Patrick McDonagh**, Thomas Cerqueus* , Liam Murphy*, 2017	Comparative analysis	Adult census dataset, Synthetic Dataset, real dataset,	a systematic comparison of three well-known k-anonymization algorithms to measure their efficiency and their effectiveness. Algorithms like datafly, incognito, Mondrian are used.
A Comparative Study of Data Anonymization Techniques	Suntherasvaran Murthy, Asmidar Abu Bakar, Fiza Abdul Rahim, Ramona Ramli 2019	Comparative Study	collected data from e-commerce sites may profile clients based on their previous searches and purchases	Generalization Suppression Distortion Swapping Masking
A Study on k-anonymity, l-diversity, and t-closeness Techniques focusing Medical Data	Keerthana Rajendran, Manoj Jayabalan, Muhammad Ehsan Rana 2017	Comparative study	the medical field in chronological order (based on published date).	1.k-Anonymity .Generalization .Suppression 2. l-Diversity 3.t-Closeness

TITLE	AUTHOR & YEAR	PURPOSE	DATA SETS USED	METHODS/ALGORITHMS
l-Diversity: Privacy Beyond k-Anonymity	ASHWIN MACHANAVAJJHALA, DANIEL KIFER, JOHANNES GEHRKE, and MUTHURAMAKRISHNAN VENKITASUBRAMANIAM	Identified 2 attacks in k-anonymity and proposed a powerful privacy criterion called l-diversity.	Lands End and Adult databases	Bayes-Optimal Privacy, since it involves modeling background knowledge as a probability distribution over the attributes and uses Bayesian inference techniques to reason about privacy
Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression	Pierangela samrati, latanya sweeny	Addressed the problem of releasing person-specific data while, at the same time, safeguarding the anonymity of individuals.	-	Illustrated how k-anonymity is provided by using generalization and suppression techniques. Introduced the concept of minimal generalization.
Data privacy preservation algorithm with k-anonymity	Waranya Mahanan ¹ · W. Art Chaovalitwongse ² · Juggapong Natwichai ³	proposed a data privacy preservation heuristic algorithm on IGH data. The algorithm is developed from the observations on the anonymous property of the problem structure that can eliminate the privacy constraints consideration	Jester and T-drive datasets	Optimal k-anonymity on IGH, Extended-OIGH algorithm.

TITLE	AUTHOR & YEAR	PURPOSE	DATA SETS USED	METHODS/ALGORITHMS
An Extensive Study on Data Anonymization Algorithms Based on K-Anonymity	Ms. Simi M S1 , Mrs. SankaraNayaki K2 , Dr.M.Sudheep Elayidom3	proposed the three best algorithms along with their efficiency and effectiveness.	-	Incognito Algorithm, Samarati's Algorithm, Sweeney's Algorithm-Datafly.
Investigation on Privacy Preserving using K-Anonymity Techniques	B.Santhosh Kumar, T.Daniya, N.Sathya, R.Cristin	Improvising the K-anonymity, the privacy of data can be executed. These can be a change the face of implantation of privacy if used properly. The prime focus is to perform an in depth analysis of all the prevailing schemes designed by the scholars for attaining a best possible solutions.	-	If data is collected from medical survey or records, that particular data can be implemented by using K-anonymity as represented below i.e. by dividing categorically and by decreasing granularity.

TITLE	AUTHOR & YEAR	PURPOSE	DAT A SETS USED	METHODS/ALGORITHMS
Privacy Preserving method for covid-19 Related Geo spatial technology.	Rohan Iyer ^{1 a} , Regina Rex ^{1 b} , Kevin P. McPherson ^{1 c} , Darshan Gandhi ¹ , Aryan Mahindra ¹ , Abhishek Singh ¹² and Ramesh Raskar ¹²	To prevent Leakage of Covid-19 information from cyber criminals	COVID -19 database	spatial K anonymity, contact tracing spatial privacy
Comparative analysis of anonymization techniques	Dilpreet Kaur Arora ¹ , Divya Bansal ² and Sanjeev Sofat ³ , 2014	Research on Anonymization techniques to protect the sensitive data of customers stored by companies.	IT company database	Anonymization,k-anonymity,l-diversity,t-closeness
Privacy Preservation in Online Social Networks Using Multiple-Graph-Properties-Based Clustering	Rupali Gangarde ^{1,*} , Amit Sharma ² , Ambika Pawar ¹ , Rahul Joshi ¹ and Sudhanshu Gonge ¹	Ensuring different level of privacy of information of users using social networks	-	Anonymization,k-anonymity,l-diversity,t-closeness,privacy preservation
A Globally Optimal k-Anonymity Method for the De-Identification of Health Data	KHALED EL EMAM, PHD, FIDA KAMAL DANKAR	To develop and evaluate a new globally optimal de-indentification algorithm that satisfy the k-anonymity suitable for health database	health dataset	Optimal lattice anonymization, k-anonymity, datafly, samarati

TITLE	AUTHOR & YEAR	PURPOSE	DATA SETS USED	METHODS/ALGORITHMS
Protecting Privacy When Disclosing Information	V. Khanaa ¹ , R. Udayakumar ² , 2012	To check whether the database inserted with the tuple is still k-anonymous without letting the other owners to know the content of the tuple and database respectively.	confidential database	k-anonymity

section 2

2.1 proposed system

Two goals are the primary focus of this endeavor. One is to compare several anonymization systems for privacy preservation, such as Datafly, Incognito, and Mondrian, and then offer a classification of them. It can be challenging to discover and choose the best algorithm given a certain publishing scenario because there are so many algorithms available and so little is known about their effectiveness. Therefore, we firmly believe that in order to evaluate the effectiveness and efficiency of the algorithms, a comparative analysis must be conducted.

Using an ideal de-identification technique that meets the k-anonymity condition, the proposed system also attempts to safeguard health data. Currently, annual increases in data volumes are exponential. The privacy of the people who are the data's subjects is violated when this data is shared and distributed. De-identification of data beforehand or as soon as possible is one method to aid health research and reduce some of the issues. K-anonymity is a widely used de-identification criterion.

2.2 software requirements

- Jupyter notebook to execute python codes
- ARX software to implement the different algorithms(l-diversity, t-closeness, suppression, generalization techniques)
- Amnesia Anonymization Tool to show the generalization graph

Section 3

3.1 Results & discussion

3.1.1 Mondrian algorithm for medical insurance

With all the data being collected every day of our lives, protecting sensitive information and maintaining data utility are both crucial in today's environment. By appropriately generalizing or suppressing the quasi-identifiers in the dataset, our study intends to discover and implement an anonymization strategy that would guard against linkage attacks and security and privacy legislation violations. The Mondrian Algorithm was selected by us in order to use Python to apply k-anonymization to our dataset. For two single-dimensional models, the approach uses a greedy search algorithm that enables more desirable anonymizations than more conventional exhaustive optimum algorithms (N., 2018). Furthermore, it permits

multidimensional modeling is ideal for our particular dataset. However, since the values of the sensitive attribute were shuffled and the quasi-identifiers were generalized, the Mondrian Algorithm technique significantly reduced the utility of the data. Although k-mean clustering would result in a higher data utility, the Mondrian Algorithm provides us with the data privacy we require because our dataset contains a lot of unfiltered and unmasked information that may be leaked and used by an adversary.

Data set

The medical cost (insurance) dataset from Kaggle is the one that our group has selected for this project (Choi, 2018). This dataset has 1338 rows of distinct persons and the accompanying columns to show the various factors that can influence an individual's insurance costs. Age, sex, BMI, children, smoker, region, and charges are some of the columns that are present in this dataset. Age, sex, and location are just a few of the quasi-identifiers that these columns display about a person. The BMI, children, smokes, and charges that are connected to an individual would be considered sensitive data since they can be used

against an individual if their identity is revealed and not adequately anonymized.

The issue with this dataset is that several of the categories, such as age, and gender, are still available to be connected to other databases and used to identify the individual because various quasi-identifiers have not been generalized or suppressed. In addition, there could be a number of legal repercussions, including a HIPAA violation, if there is a leak in this database.

```
In [7]: import pandas as pd
import matplotlib as plt
import seaborn as sns

url = 'insurance.csv'
df = pd.read_csv(url)
df
```

Out[7]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

Fig 1.1

```
In [8]: df['sex'] = df['sex'].map({'male':0, 'female':1})
df['smoker'] = df['smoker'].map({'yes':1, 'no':0})
df.head()
```

Out[8]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	1	southwest	16884.92400
1	18	0	33.770	1	0	southeast	1725.55230
2	28	0	33.000	3	0	southeast	4449.46200
3	33	0	22.705	0	0	northwest	21984.47061
4	32	0	28.880	0	0	northwest	3866.85520

```
In [10]: print(df.children.value_counts())
sns.heatmap(df.corr());
```

```
0    574
1    324
2    240
3    157
4     25
5     18
Name: children, dtype: int64
```

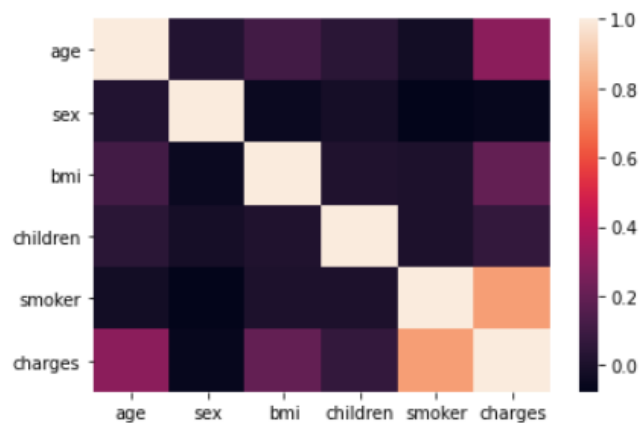


Fig 1.2

```
In [11]: #changing the data type of the variables in our dataset to categorical
categorical = set((
    'sex',
    'smoker',
    'region',
))
for name in categorical:
    df[name] = df[name].astype('category')
df.dtypes
```

```
Out[11]: age          int64
sex          category
bmi          float64
children     int64
smoker       category
region       category
charges      float64
dtype: object
```



```
In [12]: def get_spans(df, partition, scale=None):
        """
        :param df: the dataframe for which to calculate the spans
        :param partition: the partition for which to calculate the spans
        :param scale: if given, the spans of each column will be divided
                       by the value in `scale` for that column
        :returns: The spans of all columns in the partition
        """
        spans = {}
        for column in df.columns:
            if column in categorical:
                span = len(df[column][partition].unique())
            else:
                span = df[column][partition].max()-df[column][partition].min()
            if scale is not None:
                span = span/scale[column]
            spans[column] = span
        return spans
```

```
In [19]: full_spans = get_spans(df, df.index)
        full_spans
```

```
Out[19]: {'age': 46,
          'sex': 2,
          'bmi': 37.17,
          'children': 5,
          'smoker': 2,
          'region': 4,
          'charges': 62648.554110000005}
```

The Mondrian Algorithm's span function determined the maximum and minimum values for numerical columns and the variety of possible values for categorical variables. To assist in dividing the data into manageable portions, this function is computed for each column.

```
In [18]: def split(df, partition, column): ## divides values by median
        """
        :param df: The dataframe to split
        :param partition: The partition to split
        :param column: The column along which to split
        :returns: A tuple containing a split of the original partition
        """
        dfp = df[column][partition]
        if column in categorical:
            values = dfp.unique()
            lv = set(values[:len(values)//2])
            rv = set(values[len(values)//2:])
            return dfp.index[dfp.isin(lv)], dfp.index[dfp.isin(rv)]
        else:
            median = dfp.median()
            dfl = dfp.index[dfp < median]
            dfr = dfp.index[dfp >= median]
            return (dfl, dfr)
```

The split function takes our data and the provided partition of the frame as input and produces two partitions that divide the given partition into two different columns for values above and below the median.

The partitioned dataset was anonymized using K-anonymity as the next step. $K=3$ was used in this process. We set the sensitive values to be ["charges"] and the featured values to be ["age," "BMI," and "children."

```
In [17]: def is_k_anonymous(df, partition, sensitive_column, k=3):
        """
        :param df: The dataframe on which to check the partition.
        :param partition: The partition of the dataframe to check.
        :param sensitive_column: The name of the sensitive column
        :param k: The desired k
        :returns: True if the partition is valid according to our k-anonymity criteria, False otherwise.
        """
        if len(partition) < k:
            return False
        return True

def partition_dataset(df, feature_columns, sensitive_column, scale, is_valid):
    """
    :param df: The dataframe to be partitioned.
    :param feature_columns: A list of column names along which to partition the dataset.
    :param sensitive_column: The name of the sensitive column (to be passed on to the `is_valid` function)
    :param scale: The column spans as generated before.
    :param is_valid: A function that takes a dataframe and a partition and returns True if the partition is valid.
    :returns: A list of valid partitions that cover the entire dataframe.
    """
    finished_partitions = []
    partitions = [df.index]
    while partitions:
        partition = partitions.pop(0)
        spans = get_spans(df[feature_columns], partition, scale)
        for column, span in sorted(spans.items(), key=lambda x: -x[1]):
            lp, rp = split(df, partition, column)
            if not is_valid(df, lp, sensitive_column) or not is_valid(df, rp, sensitive_column):
                continue
            partitions.extend((lp, rp))
            break
        else:
            finished_partitions.append(partition)
    return finished_partitions
```

```
In [20]: # we apply our partitioning method to three columns of our dataset, using "charges" as the sensitive attribute
feature_columns = ['age', 'bmi', 'children']
sensitive_column = 'charges'
finished_partitions = partition_dataset(df, feature_columns, sensitive_column, full_spans, is_k_anonymous)

# we get the number of partitions that were created
len(finished_partitions)
```

Out[20]: 347

```
In [21]: import matplotlib.pyplot as plt
import matplotlib.patches as patches
```

The next step is to aggregate each of the columns in order to create the final anonymized dataset after we have established an anonymized dataset. The data before we used these anonymization strategies comprised many more columns and details about each of the individuals, as shown in the graphic below.

```
In [22]: def build_indexes(df):
    indexes = {}
    for column in categorical:
        values = sorted(df[column].unique())
        indexes[column] = { x : y for x, y in zip(values, range(len(values)))}
    return indexes

def get_coords(df, column, partition, indexes, offset=0.1):
    if column in categorical:
        sv = df[column][partition].sort_values()
        l, r = indexes[column][sv[sv.index[0]]], indexes[column][sv[sv.index[-1]]]+1.0
    else:
        sv = df[column][partition].sort_values()
        next_value = sv[sv.index[-1]]
        larger_values = df[df[column] > next_value][column]
        if len(larger_values) > 0:
            next_value = larger_values.min()
        l = sv[sv.index[0]]
        r = next_value
    # we add some offset to make the partitions more easily visible
    l -= offset
    r += offset
    return l, r

def get_partition_rects(df, partitions, column_x, column_y, indexes, offsets=[0.1, 0.1]):
    rects = []
    for partition in partitions:
        xl, xr = get_coords(df, column_x, partition, indexes, offset=offsets[0])
        yl, yr = get_coords(df, column_y, partition, indexes, offset=offsets[1])
        rects.append(((xl, yl),(xr, yr)))
    return rects

def get_bounds(df, column, indexes, offset=1.0):
    if column in categorical:
        return 0-offset, len(indexes[column])+offset
    return df[column].min()-offset, df[column].max()+offset

# we calculate the bounding rects of all partitions that we created
indexes = build_indexes(df)
column_x, column_y = feature_columns[:2]
rects = get_partition_rects(df, finished_partitions, column_x, column_y, indexes, offsets=[0.0, 0.0])
```

```
In [23]: def agg_categorical_column(series):
    return ','.join(set(series))

def agg_numerical_column(series):
    return [series.mean()]
```

```
In [24]: def build_anonymized_dataset(df, partitions, feature_columns, sensitive_column, max_partitions=None):
    aggregations = {}
    for column in feature_columns:
        if column in categorical:
            aggregations[column] = agg_categorical_column
        else:
            aggregations[column] = agg_numerical_column

    rows = []
    for i, partition in enumerate(partitions):
        if i % 100 == 1:
            print("Finished {} partitions...".format(i))
        if max_partitions is not None and i > max_partitions:
            break

        grouped_columns = df.loc[partition].agg(aggregations, squeeze=False)

        sensitive_counts = df.loc[partition].groupby(sensitive_column).agg({sensitive_column : 'count'})
        values = grouped_columns.to_dict()
        for sensitive_value, count in sensitive_counts[sensitive_column].items():
            if count == 0:
                continue
            values.update({
                sensitive_column : sensitive_value,
                'count' : count,
            })
            rows.append(values.copy())
    return pd.DataFrame(rows)
dfn = build_anonymized_dataset(df, finished_partitions, feature_columns, sensitive_column)

Finished 1 partitions...
Finished 101 partitions...
Finished 201 partitions...
Finished 301 partitions...
```

```
In [25]: feature_columns
```

```
Out[25]: ['age', 'bmi', 'children']
```

```
In [26]: dfn = build_anonymized_dataset(df, finished_partitions, feature_columns, sensitive_column)
```

```
Finished 1 partitions...
Finished 101 partitions...
Finished 201 partitions...
Finished 301 partitions...
```

```
In [27]: df.sort_values(feature_columns+[sensitive_column])
```

```
Out[27]:
```

	age	sex	bmi	children	smoker	region	charges
172	18	0	15.960	0	0	northeast	1694.79640
250	18	0	17.290	2	1	northeast	12829.45510
359	18	1	20.790	0	0	southeast	1607.51010
1212	18	0	21.470	0	0	northeast	1702.45530
1033	18	0	21.565	0	1	northeast	13747.87235
...
603	64	1	39.050	3	0	southeast	16085.12750
418	64	0	39.160	1	0	southeast	14418.28040
199	64	1	39.330	0	0	northeast	14901.51670
768	64	1	39.700	0	0	southwest	14319.03100
534	64	0	40.480	0	0	southeast	13831.11520

```
1338 rows × 7 columns
```

Fig 1.3

```
In [28]: df.region.value_counts()
```

```
Out[28]: southeast    364
         northwest    325
         southwest    325
         northeast    324
         Name: region, dtype: int64
```

Before :

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

Fig 1.4

After:

	age	sex	bmi	children	smoker	region	charges
172	18	0	15.960	0	0	northeast	1694.79640
250	18	0	17.290	2	1	northeast	12829.45510
359	18	1	20.790	0	0	southeast	1607.51010
1212	18	0	21.470	0	0	northeast	1702.45530
1033	18	0	21.565	0	1	northeast	13747.87235
...
603	64	1	39.050	3	0	southeast	16085.12750
418	64	0	39.160	1	0	southeast	14418.28040
199	64	1	39.330	0	0	northeast	14901.51670
768	64	1	39.700	0	0	southwest	14319.03100
534	64	0	40.480	0	0	southeast	13831.11520

Fig 1.5

Several changes were made to the dataset after the Mondrian Algorithm was used to construct K-Anonymity in Python in order to correctly mask the dataset and safeguard privacy while maintaining the usefulness of the data.

3.1.2 Anonymizing categorical data using pandas

The data set used is spreadspoke_scores.csv

1	schedule	schedule	schedule	schedule	team_home	score_home	score_away	team_away	team_favc	spread_favorite	over_under	stadium	stadium_name	weather_temperature	weather_velocity	weather_humidity
2	#####	1966	1	FALSE	Miami Dol	14	23	Oakland Raiders				Orange Bo	FALSE	83	6	71
3	#####	1966	1	FALSE	Houston C	45	7	Denver Broncos				Rice Stadiu	FALSE	81	7	70
4	#####	1966	1	FALSE	San Diego	27	7	Buffalo Bills				Balboa Sta	FALSE	70	7	82
5	#####	1966	2	FALSE	Miami Dol	14	19	New York Jets				Orange Bo	FALSE	82	11	78
6	#####	1966	1	FALSE	Green Bay	24	3	Baltimore Colts				Lambeau f	FALSE	64	8	62
7	#####	1966	2	FALSE	Houston C	31	0	Oakland Raiders				Rice Stadiu	FALSE	77	6	82
8	#####	1966	2	FALSE	San Diego	24	0	New England Patriots				Balboa Sta	FALSE	69	9	81
9	#####	1966	1	FALSE	Atlanta Fal	14	19	Los Angeles Rams				Atlanta-Fu	FALSE	71	7	57
10	#####	1966	2	FALSE	Buffalo Bil	20	42	Kansas City Chiefs				War Mem	FALSE	63	11	73
11	#####	1966	1	FALSE	Detroit Lic	14	3	Chicago Bears				Tiger Stadi	FALSE	67	7	73
12	#####	1966	1	FALSE	Pittsburgh	34	34	New York Giants				Pitt Stadiu	FALSE	64	5	70
13	#####	1966	1	FALSE	San Franci	20	20	Minnesota Vikings				Kezar Stad	FALSE	60	25	75
14	#####	1966	1	FALSE	St. Louis C	16	13	Philadelphia Eagles				Busch Mer	FALSE	72	5	70
15	#####	1966	1	FALSE	Washingtc	14	38	Cleveland Browns				RFK Memc	FALSE	65	8	52
16	9/16/1966	1966	2	FALSE	Los Angele	31	17	Chicago Bears				Los Angele	FALSE	72	10	52
17	9/18/1966	1966	3	FALSE	Buffalo Bil	58	24	Miami Dolphins				War Mem	FALSE	61	3	67
18	9/18/1966	1966	2	FALSE	Cleveland	20	21	Green Bay Packers				Cleveland	FALSE	62	6	57
19	9/18/1966	1966	2	FALSE	Dallas Cow	52	7	New York Giants				Cotton Bo	FALSE	72	9	82

Fig 2.1

This is the data to be anonymized.

```
In [2]: import pandas as pd
```

```
In [3]: def clean_df(df,cols):
        for col_name in cols:
            keys = {cats: i for i,cats in enumerate(df[col_name].unique())}
            df[col_name] = df[col_name].apply(lambda x: keys[x])
        return df
```

```
In [9]: df = pd.read_csv('spreadspoke_scores.csv')
        print(df.head())
```

	schedule_date	schedule_season	schedule_week	schedule_playoff	\
0	9/2/1966	1966	1	False	
1	9/3/1966	1966	1	False	
2	9/4/1966	1966	1	False	
3	9/9/1966	1966	2	False	
4	9/10/1966	1966	1	False	

	team_home	score_home	score_away	team_away	\
0	Miami Dolphins	14.0	23.0	Oakland Raiders	
1	Houston Oilers	45.0	7.0	Denver Broncos	
2	San Diego Chargers	27.0	7.0	Buffalo Bills	
3	Miami Dolphins	14.0	19.0	New York Jets	
4	Green Bay Packers	24.0	3.0	Baltimore Colts	

	team_favorite_id	spread_favorite	over_under_line	stadium	\
0	NaN	NaN	NaN	Orange Bowl	
1	NaN	NaN	NaN	Rice Stadium	
2	NaN	NaN	NaN	Balboa Stadium	
3	NaN	NaN	NaN	Orange Bowl	
4	NaN	NaN	NaN	Lambeau Field	

	stadium_neutral	weather_temperature	weather_wind_mph	weather_humidity	\
0	False	83.0	6.0	71.0	
1	False	81.0	7.0	70.0	
2	False	70.0	7.0	82.0	
3	False	82.0	11.0	78.0	
4	False	64.0	8.0	62.0	

	weather_detail
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN

Fig 2.2

We will anonymize the following columns and save the anonymous data in the anonymous_data.csv file

```
In [5]: cols = ['team_home', 'team_away', 'stadium']
```

```
In [6]: df = clean_df(df, cols)
```

```
In [7]: df.to_csv('anonymous_data.csv')
```

```
In [8]: cf = pd.read_csv('anonymous_data.csv')
print(cf.head())
```

	Unnamed: 0	schedule_date	schedule_season	schedule_week	schedule_playoff	\
0	0	9/2/1966	1966	1	False	
1	1	9/3/1966	1966	1	False	
2	2	9/4/1966	1966	1	False	
3	3	9/9/1966	1966	2	False	
4	4	9/10/1966	1966	1	False	

	team_home	score_home	score_away	team_away	team_favorite_id	\
0	0	14.0	23.0	0	NaN	
1	1	45.0	7.0	1	NaN	
2	2	27.0	7.0	2	NaN	
3	0	14.0	19.0	3	NaN	
4	3	24.0	3.0	4	NaN	

	spread_favorite	over_under_line	stadium	stadium_neutral	\
0	NaN	NaN	0	False	
1	NaN	NaN	1	False	
2	NaN	NaN	2	False	
3	NaN	NaN	0	False	
4	NaN	NaN	3	False	

	weather_temperature	weather_wind_mph	weather_humidity	weather_detail
0	83.0	6.0	71.0	NaN
1	81.0	7.0	70.0	NaN
2	70.0	7.0	82.0	NaN
3	82.0	11.0	78.0	NaN
4	64.0	8.0	62.0	NaN

Fig 2.3

3.1.3 ARX Tool

Step 1: Inserting Data set in the software

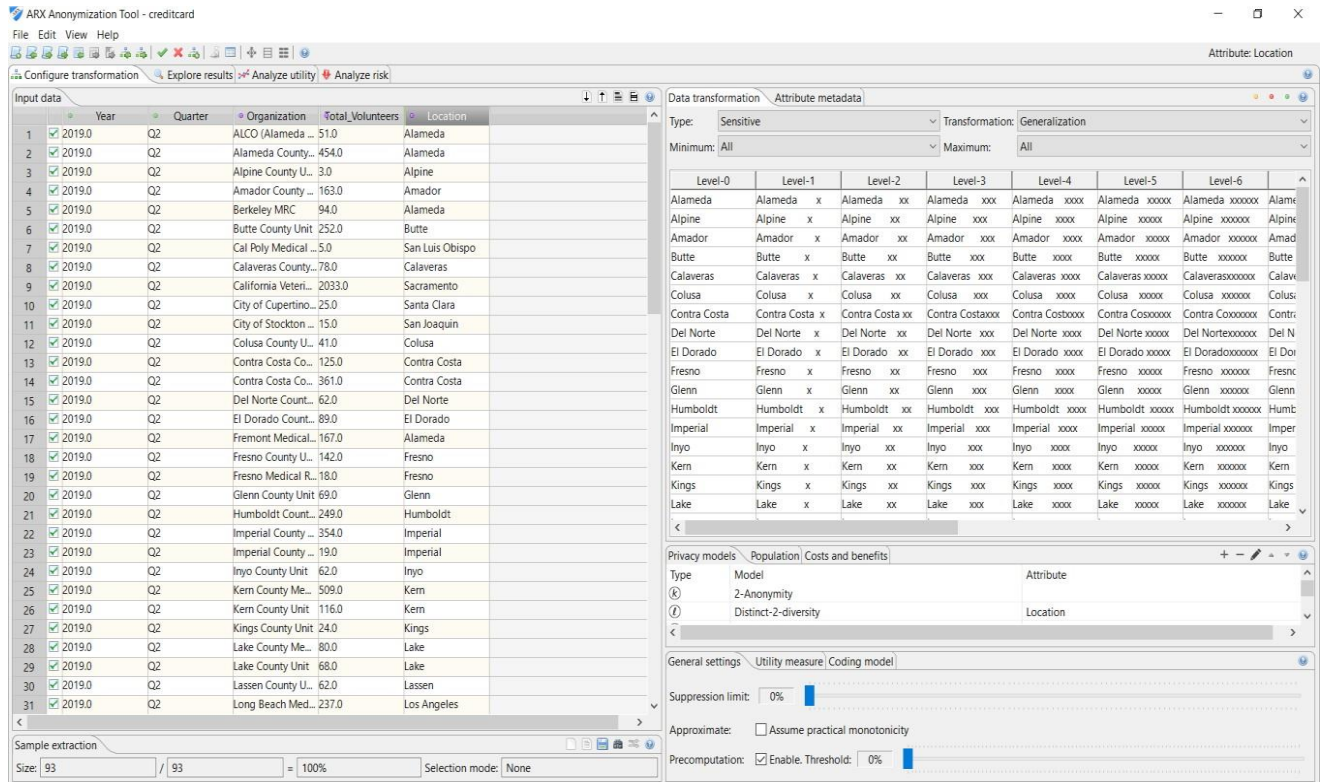


Fig 3.1

Step2: Based on the level algorithm get changed and privacy will be high

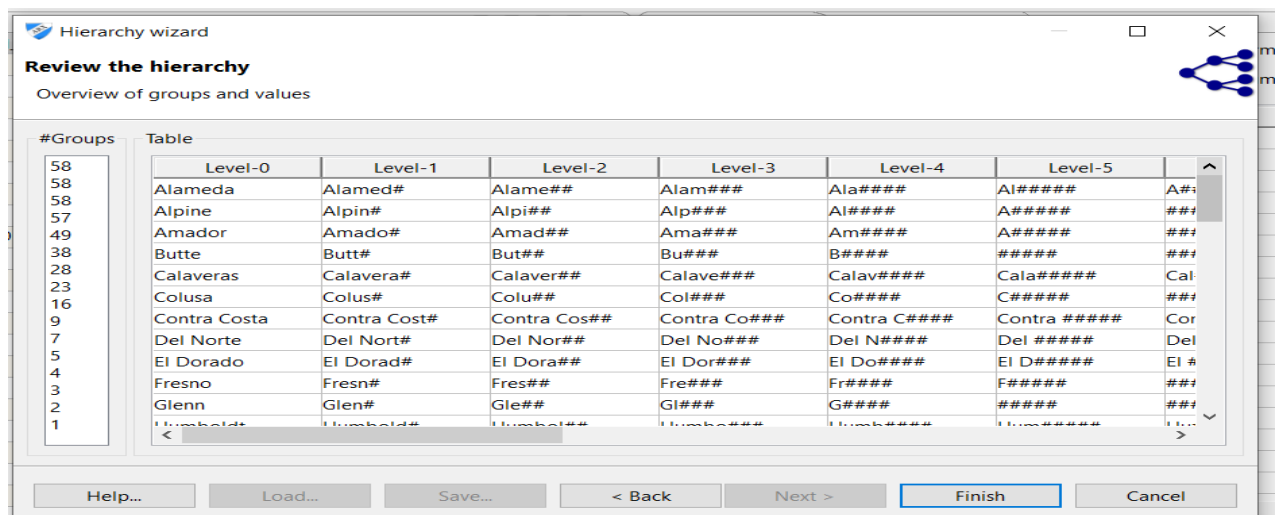


Fig 3.2

Step 3: Risk analysis table

Quarter, Total_Volunteers	86.02151%	99.62599%
Year, Total_Volunteers	86.02151%	99.62599%
Organization, Location	100%	100%
Organization, Total_Volunteers	100%	100%
Quarter, Organization	100%	100%
Total_Volunteers, Location	100%	100%
Year, Organization	100%	100%
Year, Quarter, Location	62.36559%	98.87798%
Year, Quarter, Total_Volunteers	86.02151%	99.62599%
Organization, Total_Volunteers, Location	100%	100%
Quarter, Organization, Location	100%	100%
Quarter, Organization, Total_Volunteers	100%	100%
Quarter, Total_Volunteers, Location	100%	100%
Year, Organization, Location	100%	100%
Year, Organization, Total_Volunteers	100%	100%
Year, Quarter, Organization	100%	100%
Year, Total_Volunteers, Location	100%	100%
Quarter, Organization, Total_Volunteers, Location	100%	100%

Fig 3.3

Almost here we achieved 100% risk-free Data

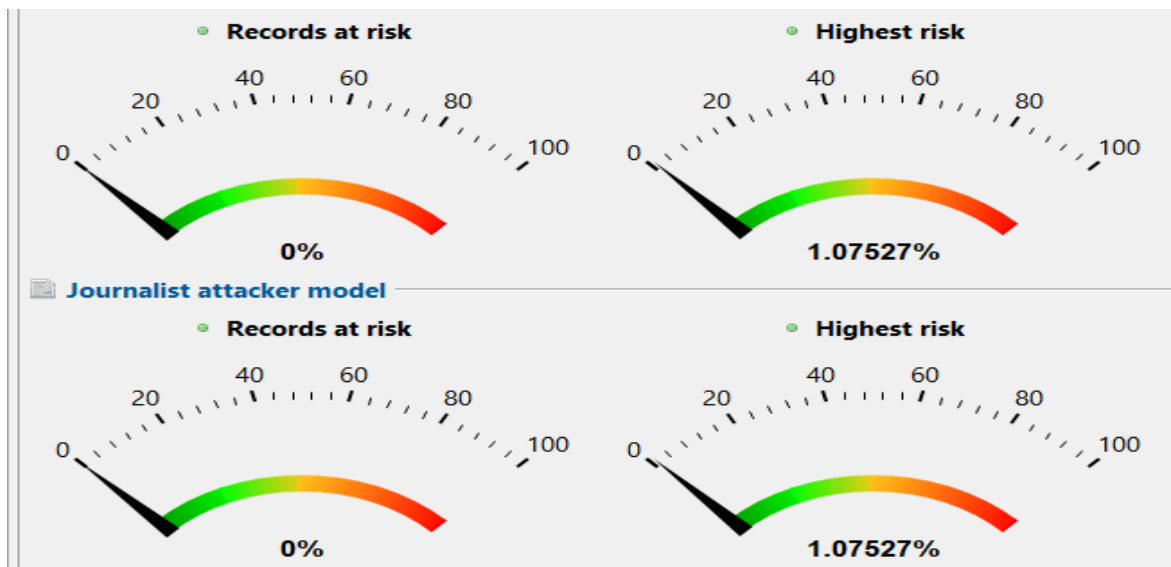


Fig 3.4

Step 4: The algorithm we used

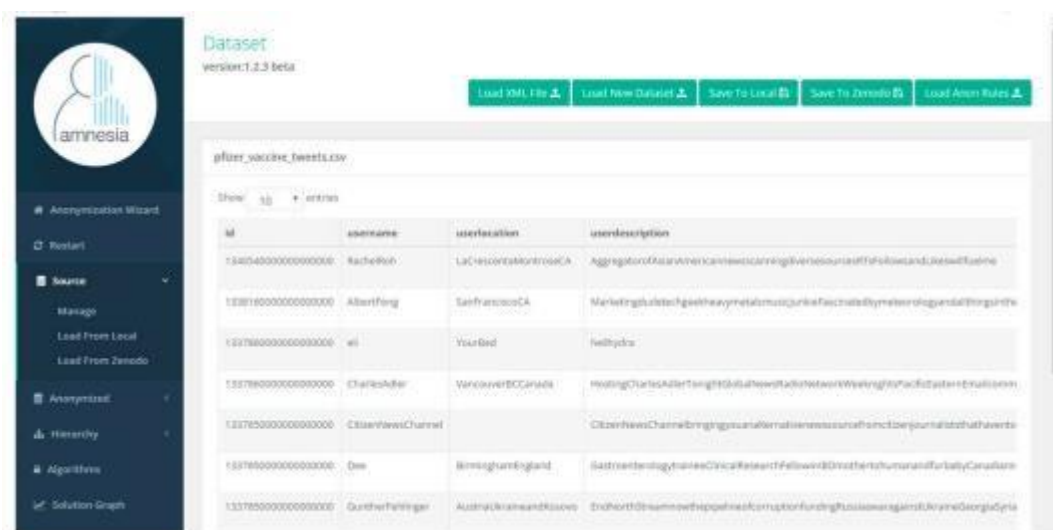
Privacy models			Population	Costs and benefits	
Type	Model	Attribute			
(k)	2-Anonymity				
(l)	Distinct-2-diversity	Location			

Privacy models			Population		Costs and benefits			
Type	Model	Attribute						
(t)	0.001-Closeness (equal ground-distance)	Location						
(l)	Distinct-2-diversity	Total_Volunteers						

Fig 3.5

3.1.4 Amnesia Tool Graph

Step 1: Dataset uploaded in Amnesia



id	username	userlocation	userdescription
1320540000000000000	RachelRob	LaCrescentMontrealCA	Aggregator of American news scanning diverse sources off the web and, I believe, some
1328160000000000000	AlbertFong	SanFranciscoCA	Marketing strategist with heavy metal music taste and a love for the arts and the
1327980000000000000	eli	YonkersNY	Healthcare
1327980000000000000	CharlesAdler	VancouverBCCanada	Marketing Charles Adler is a digital marketing expert with a focus on social media and
1327950000000000000	CitizenNewsChannel		CitizenNewsChannel brings you the latest news and information from the streets of the
1327950000000000000	Osw	BirminghamEngland	Software engineer and a former research fellow at the University of Birmingham
1327950000000000000	GunterFehlinger	AustriaAustriaandRussia	Software engineer and a former research fellow at the University of Birmingham

Fig 4.1

Step 2: Hierarchy Username

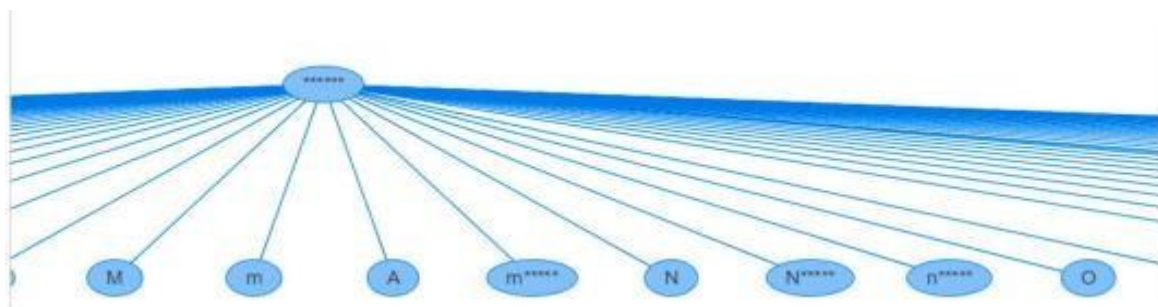


Fig 4.2

Step 3: Hierarchy ID

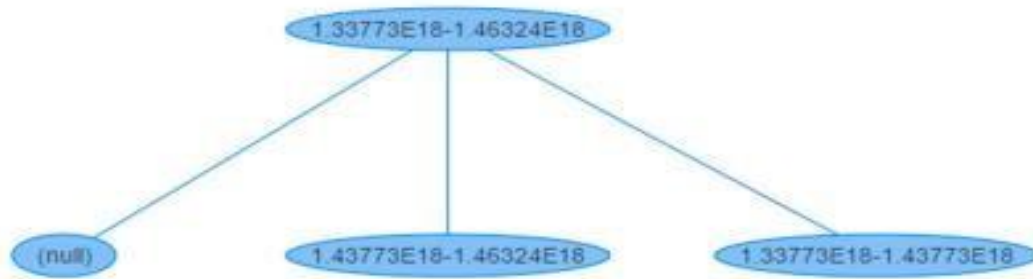


Fig 4.3

Step 4: Hierarchy Location

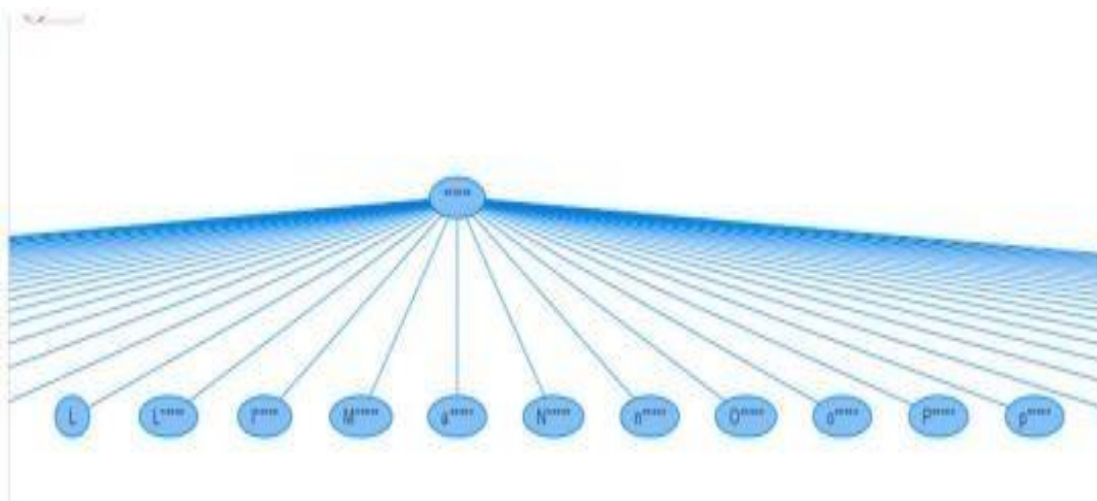


Fig 4.4

the quasi-identifiers were made more generic. We could have improved our model by making sure that the sensitive data value "charges" was consistent with the original dataset and did not have its values shuffled. Due to the fact that all other identifiable characteristics were either repressed or generalized, this increases the utility of the data while maintaining privacy.

3.3 references

- [1] Ayala-Rivera, Vanessa, et al. "A systematic comparison and evaluation of k-anonymization algorithms for practitioners." *Transactions on data privacy* 7.3 (2014): 337-370.
- [2] Johny Antony, P., and Dr Antony Selvadoss Thanamani. "Comparison and Analysis of Anonymization Techniques for Preserving Privacy in Big Data." *Advances in Computational Sciences and Technology ISSN* (2017): 0973-6107.
- [3] El Emam, Khaled, et al. "A globally optimal k-anonymity method for the de-identification of health data." *Journal of the American Medical Informatics Association* 16.5 (2009): 670-682.
- [4] Murthy, Suntherasvaran, et al. "A comparative study of data anonymization techniques." *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. IEEE, 2019.
- [5] Rajendran, Keerthana, Manoj Jayabalan, and Muhammad Ehsan Rana. "A study on k-anonymity, l-diversity, and t-closeness techniques." *IJCSNS* 17.12 (2017): 172.
- [6] Machanavajjhala, Ashwin, et al. "l-diversity: Privacy beyond k-anonymity." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007): 3-es.
- [7] Samarati, Pierangela, and Latanya Sweeney. "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression." (1998).
- [8] LeFevre, Kristen, David J. DeWitt, and Raghuram Ramakrishnan. "Mondrian multidimensional k-anonymity." *22nd International conference on data engineering (ICDE'06)*. IEEE, 2006.
- [9] Shin, Moonshik, et al. "Electronic medical records privacy preservation through k-anonymity clustering method." *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*. IEEE, 2012.

- [10] Belsis, Petros, and Grammati Pantziou. "A k-anonymity privacy-preserving approach in wireless medical monitoring environments." *Personal and ubiquitous computing* 18.1 (2014): 61-74.
- [11] Mahanan, Waranya, W. Chaovalitwongse, and Juggapong Natwichai. "Data privacy preservation algorithm with k-anonymity." *World Wide Web* 24.5 (2021): 1551-1561.
- [12] Simi, M. S., K. Sankara Nayaki, and M. Sudheep Elayidom. "An extensive study on data anonymization algorithms based on k-anonymity." *IOP Conference Series: Materials Science and Engineering*. Vol. 225. No. 1. IOP Publishing, 2017.
- [13] Kumar, B. Santhosh, et al. "Investigation on privacy preserving using K-anonymity techniques." *2020 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2020.
- [14] Iyer, Rohan, et al. "Spatial K-anonymity: A privacy-preserving method for COVID-19 related geospatial technologies." *arXiv preprint arXiv:2101.02556* (2021).
- [15] Li, Hongtao, et al. "(a, k)-Anonymous scheme for privacy-preserving data collection in IoT-based healthcare services systems." *Journal of Medical Systems* 42.3 (2018): 1-9.
- [16] Rashid, Asmaa H., and Abd-Fatth Hegazy. "Protect privacy of medical informatics using k-anonymization model." *2010 The 7th International Conference on Informatics and Systems (INFOS)*. IEEE, 2010.
- [17] Zhu, Yan, and Lin Peng. "Study on k-anonymity models of sharing medical information." *2007 International Conference on Service Systems and Service Management*. IEEE, 2007.
- [18] Omran, Esraa, Albert Bokma, and Shereef Abu-Almaati. "A k-anonymity based semantic model for protecting personal information and privacy." *2009 IEEE International Advance Computing Conference*. IEEE, 2009.
- [19] Hu, Xinping, et al. "K-anonymity based on sensitive tuples." *2009 First international workshop on database technology and applications*. IEEE, 2009.
- [20] J. J. Panackal, A. S. Pillai and V. N. Krishnachandran, "Disclosure risk of individuals: A k-anonymity study on health care data related to Indian population," 2014 International Conference on Data Science & Engineering (ICDSE), 2014, pp. 200-205, doi: 10.1109/ICDSE.2014.6974637.