EFFICIENT PRIVACY PRESERVATION FOR MICRO-DATA APPLICATION USING CROSS-BUCKET WITH PROGRESSIVE DUPLICATE DETECTION MODEL

S. Menaka,
M.Phil Research Scholar,
Department Of Computer Science,
SSM College of Arts and Science,
Komarapalayam,Tamilnadu,India.

M.M. Kavitha,
Assistant Professor,
Department Of Computer Science,
SSM College of Arts and Science,
Komarapalayam,Tamilnadu,India.

Abstract: In cross bucket generalization, divide micro-data into equivalence teams and buckets. First, it provides separate protection for identity and sensitive values, and therefore the level of protection may be flexibly adjusted supported actual demands. Second, the sizes of equivalence teams and buckets are reduced as way as potential by solely satisfying the protection needs, that avoid the over shielding for identity and cut back data loss. Most duplicates are detected by progressive duplicate detection. Rather than reducing the general time required to end the whole method, progressive approaches try and scale back the typical time when that a replica is found. To discover the duplicity with less time of execution and additionally while not perturbing the dataset quality, strategies like concurrent Progressive blocking and concurrent Progressive Neighborhood are used. Progressive sorted neighborhood technique conjointly known as CC-PSNM is employed during this model for locating or sleuthing the duplicate during a parallel approach.

Keywords: Data mining, Data bucketization, Concurrent Progressive Blocking, Concurrent sorted neighborhood, Privacy Data, Generalization, Cross Bucket.

I.INTRODUCTION

A huge quality of organizations issue microdata for a cause like demographic and health analysis. So as to guard individual privacy, best-known identifiers (e.g., Name and Social Security Number) should be removed. Additionally, this method should account for the chance of mixing bound different attributes with external knowledge to unambiguously establish people. For instance, a personal may well be "re-identified" by connection the free information with (public) information on Age, Sex, and postcode. K-anonymity has projected to scale back the chance of this sort of attack. The first goal of k-anonymization is to shield the privacy of the people to whom the information pertains. However, subject to the present constraint, it's necessary that the free information stay as "useful" as attainable. Various cryptography models are projected within the literature for k-anonymization, and sometimes the "quality" of the revealed knowledge is settled by the model that's used.

Many organizations are progressively publishing the microdata- record that contains unaggregated data regarding people. These tables will embrace medical, elector registration, census, and client information. Microdata may be a valuable supply of data for the allocation of public funds, medical analysis, and analytic thinking. However, if people are often unambiguously known within the data then the personal Data (Such as their medical condition) would be disclosed, and this is often unacceptable. To avoid the identification of records in microdata, unambiguously characteristic data like names and Social Security numbers are far away from the table. However, this initial cleansing still doesn't make sure the privacy of people within the information. Sets of attributes (like sex, DOB, and postal code within the example above) which will be joined with external information to unambiguously establish people within the population are known as quasi-identifiers.

To counter linking attacks victimization quasi-identifiers, Samarati and Sweeney projected a meaning of privacy referred to as k-anonymity. A record of tables in which the k anonymity is satisfied, if each record within the tables is interchangeable from a minimum of k one different records with relevance every pack of quasi identifier element; such a record of table is named a k-anonymous table. Hence, for every combining of values inside the k-anonymous table, that split those values. This ensures that people cannot be unambiguously known by linking attacks.

Compared with information dissemination in pre-aggregated or applied mathematics forms, the discharge of microdata offers vital benefits in terms of data convenience that creates it significantly appropriate for circumstantial analyses during a form of domains like public health and population studies. However, the discharge of microdata raises privacy considerations of unveiling non-public data of people. Easy deidentification (i.e., removing express identifiers, e.g., name, SSN) has shown to be inadequate to guard an individual's privacy.

One's different attributes (so-called quasi identifiers, like age, code and date of birth) are sometimes required for information analyses, and so are kept once deidentification. This typically permits individuals' sensitive data to be unconcealed once microdata are coupled with publically obtainable data through

quasi-identifiers. The techniques to deal with this privacy downside are k-anonymization and then l-diversity. Through domain generalization, they guarantee that although publically obtainable data is coupled with records during a microdata information, a sensitive attribute worth will at the most be associated with a bunch of no less than k entities (and having no less than ` distinct sensitive attribute values), rather than to a selected individual.

The privacy goal of k-anonymization and then l-diversity is appropriate for categorical sensitive attributes, like the sickness attribute in a very patient record table, wherever completely different attribute values are incomparable. In follow, besides categorical attributes, several perspective attributes in microdata db are numerical, e.g., salary amount, investment losses or gains, white blood corpuscle count, etc. Applying existing privacy goals of k-anonymity and then ℓ-diversity is usually not decent to shield numerical attributes. For instance, even once l-diversity is glad, if the earnings values fall in a very slender varies, sensitive data is discovered. Thus, it's necessary to outline new secure goals for the protecting the numerical sensitive element in microdata. Two notions, k-anonymity and then ℓ-diversity, are planned to live the degree of privacy preservation. K-anonymity should still permit associate degree person to infer the sensitive worth of a private with extraordinarily high confidence. Hence, they adopt l-diversity that provides stronger privacy protection.
Hence, they adopt ℓ-diversity that provides stronger privacy protection.

Publication microdata permits analyses and policy-makers to research the information and learn necessary information benefiting the society as an entire, like the factors inflicting bound diseases, effectiveness of a medication or treatment, and social-economic patterns which will guide the formulation of effective public policies. In different words, publication microdata leads to utility gain for the society as a full. However, as microdata contain specific data concerning people, publication microdata might conjointly end in privacy loss for people whose data is revealed. Thence before the microdata are often created public, one should make that the privacy loss is restricted to a suitable level. This is often usually done via anonymization that transforms the microdata to enhance the privacy. As a result of anonymization makes information inexact and/or distorted, it conjointly causes losses in potential utility gain, compared with the case of publication the unanonymized microdata.

Data generalization is related to only if there's taxonomy for every public attributes. Also, generalized information cannot be simply analyzed by normal ways. As an example, to reduce the data loss in native cryptography, price the worth} "Engineer "could be generalized into the high level value "Professional "in some records whereas it remains unchanged in alternative records. Consequently, "Professional "and "Engineer "cannot be treated as 2 distinct values for investigation, that makes it not possible to use any normal count primarily based data processing ways like Naive Beyes classifiers. Data bucketization doesn't have the higher than issues as result of it doesn't generalize domain values.

In bucketization schemes use the same confidentiality setting for all precarious values. for instance, the Anatomy algorithmic program in uses l-diversity to specify the privacy criterion that the frequency of HIV and respiratory disease in a very bucket isn't any over 1/l . Suppose that HIV is additional sensitive than respiratory disease, wherever respiratory disease happens additional often times than HIV. ℓ-diversity should be set consistent with the sensitivity of HIV, i.e., an oversized l so 1/l is tiny enough for HIV. However, typically this setting isn't satiable for respiratory disease that features a higher frequency within the data. Another disadvantage of preceding bucketization scheme is that a similar size for all buckets, i.e., either l or l + one in. Within the example of HIV and respiratory disease, l that's set consistent with the foremost sensitive HIV is often terribly massive. A bigger bucket size suggests that less association of a record with its original sensitive value within the bucket, thus, additional info loss.

II. RELATED WORKS

Benjamin C. M. Fung [1] described that the collection of digital information by government, corporation, and individual has created tremendous opportunities for knowledge and information based decision making. Driven by mutual advantages, or by rules that need certain information to be revealed, there is a necessity for the exchange and publication of information among numerous parties. Information in its original type, however, generally contains sensitive information regarding individuals, and publish such information can violate individual privacy. The present observation in information publishing depends primarily on policies and tips on what sorts of information may be revealed, and agreements on the employment of revealed information. This approach alone could result in excessive information distortion or inadequate protection. Privacy protecting Data publishing (PPDP) provides strategies and tool to publish helpful data whereas preserving data privacy. Recently PPDP have received appreciable attention in analysis communities, and lots of approaches are projected for various data publishing situations. During this survey, they're going to consistently summarize and valuate completely different approaches to PPDP study the challenges in sensible information publishing, clarify the variations and needs that distinguishes PPDP from different related issues and propose future analysis directions.

Kristen Lefevre [2] represented that K Anonymity have projected as a mechanism for shielding protection in microdata publishing. This paper proposed a brand new multidimensional model that provides a further flexibility degree is not available in existing single dimensional approaches. Usually this flexibility results in higher quality anonymization as measured each by common purpose metrics and a lot of specific symbols of query responsibility. Optimum multidimensional anonymization is hardness of NP like previous optimum k anonymity problems. However, they introduce an easy greedy approximation algorithmic rule, and experimental results show that this greedy algorithmic rule frequently results in additional fascination of anonymity than complete

optimum principle for two, single dimensional models.

Ashwin Ma Chanavajjhala [3] represented that publication data concerning individuals while not revealing sensitive data regarding them is a crucial drawback. A definition for privacy referred to as k anonymity has achieved quality. During a k anonymity dataset, each report is identical from a minimum of k 1 alternative records with relevancy sure "identifying" attributes. During this thesis they show with two simple attack that a k anonymity dataset has some refined, however severe privacy issues. First, they show that a hacker will find the values of privacy attributes once there's very small diversity in those privacy attributes. Second, hacker usually has domain knowledge, and that they show that k-anonymity doesn't guarantee privacy against hackers exploiting the background knowledge.

Qing Zhanga [4] represented that privacy could be a serious concern once microdata need to be discharged for ad hoc analyses. The goal of previous privacy protection approaches is that the ℓ diversity and k anonymity that is appropriate just for categorical sensitive attributes. Since applying those on to numeric sensitive identifier that is salary could lead to undesirable data discharge. They proposes privacy goal to raised capture the requirement of privacy protection for a sensitive numerical attributes. Complementing the need for privacy is that they have to be compelled with ad hoc combination analyses over microdata. Already presented generalization based anonymization approaches cannot respond combination queries with proper accuracy. They present a common framework of permutations based are mostly anonymization to support accuracy for corresponding aggregate query and shown that for a similar grouping permutation based technique will continuously answer aggregate queries a lot of exactly than that of generalization based approaches. They additionally propose many criteria for optimizing permutation to accurate answering of combination queries and to develop efficient algorithm for each and every criterion.

Xiaokui Xiao [5] described a unique technique anatomy for business enterprise sensitive information. Anatomies free all the quasi identifiers and receptive value directly in two different separate tables. Combined with certain grouping method follows these approaches protect privacy and capturing an oversized quantity of correlation within the microdata. They designed a linear time formula for the computation of anatomized tables that adapt the l diversity privacy demand and minimize the error of reconstructing the microdata. Intensive experiments ensure that our technique permits considerably more effective information analysis than the conventional publication method based on generalization. Specifically anatomy permits mixture reasoning with average error below 100 pc which is less than error obtained from a generalized table by orders of magnitude.

Tiancheng Li [6] explained that in information publishing, anonymization techniques like bucketization and generalization are designed to supply privacy protection. Within the meanwhile, they cut back the effectiveness of the information. it is necessary to think about the exchange between privacy and effectiveness. During a thesis, Shmatikov and Brickell proposed a study methodology by compare the security gain with utility the gain resulted from anonymization the information, and ended that even modest security gains need virtually completes the demolition of data mining utility. This conclusion looks to challenge previous work on information anonymization. During this thesis, they analyze the elemental characteristic of security and utility, and show that it's inappropriate to directly compare privacy with utility. Then observes the security utility tradeoff in information publish is same as the risk return exchange in financial asset, and finally, they evaluate the method on the fully developed datasets from the machine learning depository. Our results clarify many common misconceptions about information utility and supply information publishers' helpful guidelines on selecting the proper tradeoff between privacy and utility.

Peng Wanga [7] represented that bucketization is the Data anonymization method for publication sensitive information. The proposal is to cluster records into tiny buckets to ambiguous the record level association between privacy data and identifying information. Compared to the standard generalization technique, bucketization doesn't need a classification of attribute values, therefore is applicable to additional information sets. A disadvantage of existing bucketization method is that the equivalent privacy setting and equivalent bucket size, which regularly leads to a non feasible privacy goal or excessive data loss if sensitive values have variable sensitivity.

Steven Euijong Whang [8] represented Entity resolution is that the drawback of identifying that reports in a database consult with an equivalent entity. In observe, several applications have to be compelled to resolve massive data sets efficiently, however the ER doesn't need result to be actual. For instance, individual's information from the online might merely be large to fully resolve with an inexpensive quantity of work. As another example, time period applications might not be ready to tolerate any ER process that takes longer than a certain quantity of your time. This thesis investigates however they'll maximize the progress of ER with a restricted quantity of work using "hints," that provide data on records that are probably to refer to a comparable to real world thing. A touch are often described in varied format (e.g., a grouping of records supported their probability of matching), and ER will use this data as a tenet that records to check first. They introduce a family of techniques for constructing hints efficiently and techniques for using the hints to maximize the number the quantity the quantity of matching records identified using a restricted amount of work.

Jayant Madhavan [9] represented a novel data integration design, PAYGO, that is impressed by the idea spaces and emphasize pay as you go records management as suggests that for achieve web scaled data integration. However, author is witnessing an increase each within the large amount of structured data's on the online and within the diversity of the structures in which these data are stored. A typical characteristic of those collections of structured data is that they yield heterogeneousness at scales unseen before.

Mauricio A [10] represented the problem of integration of multiple database of information regarding common entities is often encounter in the KDD and the decision support applications in massive business and government organizations. The problem they study is commonly known as Merge/Purge problem and it is difficult to resolve each in balance and accuracy.

III. METHODOLOGY

Duplicate detection is that the method of finding several representations of same world entities. In present, duplicate detection strategies have to be compelled to process ever big datasets in ever limit time maintains the qualitative of a datasets becomes progressively difficult. This thesis presents two unique, progressive duplicate detection process that considerably increase the potency of catching duplicates if the process time is limited: They maximize the gain of the general method among the time accessible by reporting most results a lot of previous ancient approaches.

Comprehensive experiments show that progressive process will twin the efficiency eventually of old time duplicate detection and considerably improve upon connected work. Data are among the foremost vital assets of an organization. However because of data changes and sloppy information entry, failure like duplicate entries may occur, creating data cleansing and especially duplicate detection indispensable. The Progressive duplicate detection finding most duplicate pairs before within a detection method. Rather than reducing the time required to complete the whole method, progressive approaches try and cut back the common time when that a reproduction is found. Early terminations, specially, then yields a lot of complete results on a progressive algorithmic rule than on any ancient approach.

A. RECORD ADDITION

In this module, the records are additional for the given columns. The records might contain any information. The records are saved in 'Patient Health Record' and 'Test Data' in the table.

B. ATTRIBUTE CHOICE FOR DUPLICATION IDENTIFYING

In this module, those columns are chosen for locating duplicates in records.

C. INPUT PARAMETER FRAME FOR PSNM/PB

1. Input Parameters (D, K, W, I, N)

In this module, input for the algorithmic rule PSNM is chosen. The algorithmic rule takes 5 input parameters: D could be a relevancy the information, that has not been loaded from disk however. The sorting key K defines the attribute or attributes combination that ought to be utilized in the sorting step. W specifies the most window size, which corresponds to the window size of the standard sorted neighborhood technique. Once exploitation early termination, this parameter will be set to an optimistically high default value. Parameter I defines the extension interval for the progressive iterations. N is that the variety of records.

2. Input Parameters (D, K, R, S, N)

In this module, input for the Algorithm PB is selected. The algorithm takes five input parameters: D is a reference to the data, which has not been loaded from disk yet. The sorting key K defines the attribute or attributes combination that should be used in the sorting step. R specifies the maximum block range, S Block Size and N Total No. of Records. When using early termination, this parameter can be set to an optimistically high default value.

D. PROGRESSIVE SORTED NEIGHBORHOOD TECHNIQUE ALGORITHMIC RULE

The PSNM technique calculates associate applicable partition size pSize, i.e., the most range of records that in memory, victimization the demoralized sampling perform calcPartitionSize(D) in Line 2: If the information is scan from a information, the perform will calculate the dimensions of a record from the information sorts and match this to the out there main memory. Otherwise, it takes a sample of records and estimates the dimensions of a record with the biggest values for every field.

Algorithm 1. Progressive Sorted Neighborhood

```
Require: dataset reference D, sorting key K, window size
    W, enlargement interval size I, number of records N
 1: procedure PSNM(D, K, W, I, N)
 2:     pSize ← calcPartitionSize(D)
 3:     pNum ← ⌈N/(pSize − W + 1)⌉
 4:     array order size N as Integer
 5:     array recs size pSize as Record
 6:     order ← sortProgressive(D, K, I, pSize, pNum)
 7:     for currentI ← 2 to ⌈W/I⌉ do
 8:         for currentP ← 1 to pNum do
 9:             recs ← loadPartition(D, currentP)
10:             for dist ∈ range(currentI, I, W) do
11:                 for i ← 0 to |recs| − dist do
12:                     pair ← ⟨recs[i], recs[i + dist]⟩
13:                     if compare(pair) then
14:                         emit(pair)
15:                         lookAhead(pair)
```

In Line 3, the algorithmic rule calculates the quantity of necessary partitions pNum, whereas considering a partition overlap of W - one record to slip the window beyond their margin. Line four defines the order-array that stores the order of records with respect to the given key K. By storing solely record IDs during this array; they assume that it is unbroken in memory. To carry the particular records of a present partition, PSNM maintains a recs array in a Line five.

In the Line 6, PSNM kinds the datasets D by basic key K. The classifying is finished using progressive sorting method. Afterwards, PSNM linearly will increase the windows size from two to the most windows size W in the steps I (Line 7). During this manner, promising close neighbors are chosen initial and fewer promising far-away neighbors presently. For every of those progressive iterations, PSNM reads the whole dataset once. Since the load method is finished partition-wise, PSNM consecutive repeat (Line 8) and masses (Line 9) all partitions.

To method a load the partition, PSNM initial iterates overall report RD (Rank Distances) dist they are among the present window interval current I. For I= one this can be only 1

distance, particularly the report rank distance of the present main-iteration. In Line 11, PSNM then iterates all records within the current partition to check them to their dist-neighbor. The comparison is executed exploitation the compare (pair) function in Line13. If this performs returns "true", a replica has been found and may be emitted.

E. PROGRESSIVE BLOCKING

In this section, datasets references D, key identifier K, R as maximum block range, S as block size and N as record number are given as input. The algorithm accepts five input parameters: The dataset reference D specifies the dataset to be cleaned and the key attribute or key attribute combination K defines the sorting. The parameter R limits the maximum
Block range, which is the maximum rank-distance of two blocks in a block pair, and S specifies the size of blocks. Finally, N is size of input dataset.

Algorithm 2. Progressive Blocking

```
Require: dataset reference D, key attribute K, maximum
    block range R, block size S and record number N
 1: procedure PB(D, K, R, S, N)
 2:     pSize ← calcPartitionSize(D)
 3:     bPerP ← ⌊pSize/S⌋
 4:     bNum ← ⌈N/S⌉
 5:     pNum ← ⌈bNum/bPerP⌉
 6:     array order size N as Integer
 7:     array blocks size bPerP as ⟨Integer, Record[ ]⟩
 8:     priority queue bPairs as ⟨Integer, Integer, Integer⟩
 9:     bPairs ← {⟨1, 1, _⟩, . . . ,⟨bNum, bNum, _⟩}
10:     order ← sortProgressive(D, K, S, bPerP, bPairs)
11:     for i ← 0 to pNum − 1 do
12:         pBPs ← get(bPairs, i · bPerP, (i + 1) · bPerP)
13:         blocks ← loadBlocks(pBPs, S, order)
14:         compare(blocks, pBPs, order)
15:     while bPairs is not empty do
16:         pBPs ← {}
17:         bestBPs ← takeBest(⌊bPerP/4⌋, bPairs, R)
18:         for bestBP ∈ bestBPs do
19:             if bestBP[1] − bestBP[0] < R then
20:                 pBPs ← pBPs ∪ extend(bestBP)
21:         blocks ← loadBlocks(pBPs, S, order)
22:         compare(blocks, pBPs, order)
23:         bPairs ← bPairs ∪ pBPs
24: procedure compare(blocks, pBPs, order)
25:     for pBP ∈ pBPs do
26:         ⟨dPairs, cNum⟩ ← comp(pBP, blocks, order)
27:         emit(dPairs)
28:         pBP[2] ← |dPairs| / cNum
```

In the Lines six to eight, pb then defines the 3 main data structures: the order-array, that stores the ordered list of record IDs, the blocks-array, that holds this partition of the blocked report, and also the bPairs-list, that store the all newly evaluated block pairs. Through, a BP (block pair) is illustrating as a ternary of (blockNr1, blockNr2, duplicates Per Comparison).

The bPairs-list is enforced as a priority queue, as a result of the algorithmic rule often reads the highest parts from this list. Within the following Line ten, the lead algorithmic rule sorts the dataset exploitation the progressive Magpie Sort algorithmic rule. Afterwards, the Lines eleven to fourteen loaded the all the blocks partition-wise from the disk to execute the comparisons among every block. When the preprocessing, the lead algorithmic rule starts increasingly extending the foremost promising block pairs (Lines fifteen to 23).

F. PROPOSED SYSTEM (PSNM AND PROGRESSIVE BLOCKING)

In this module, the algorithms square measure same as module four and five however the records are split and every window partition is given to totally different systems (resources i.e., computers) in order that the method is executed in parallel.

It des

IV. RESULTS AND PLANNING

In order to evaluate the proposed methodology, the experimental research is developed using C# and it outputs a list of SQL statements with the fusion and propagation. The created SQL code consists of Update, Insert and Delete commands that are compatible with the SQL ordinary.

Table1. Duplicate Detection Model– Cross-Bucket Generalization

No of Records	Duplicate Records	Execution Time (Sec)
20	5	5
50	20	5
75	34	10
100	42	10
150	50	12
225	70	15
250	82	20
500	113	30

It describes the experimental dataset for the Cross-Bucket Generalization methodology. It contains the number of reports taken for the process, number of duplicate reports find and the execution time for the duplication finding process.

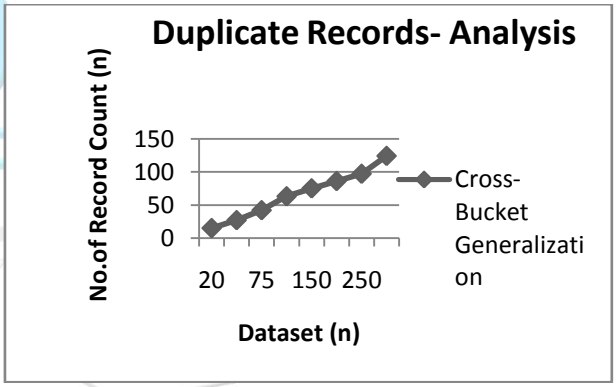

Figure1. Duplicate Detection Model – Cross-Bucket Generalization

It describes the experimental dataset for the Cross-Bucket Generalization methodology. It shows the number of records taken for the process and number of duplicate records find for the duplication finding process.

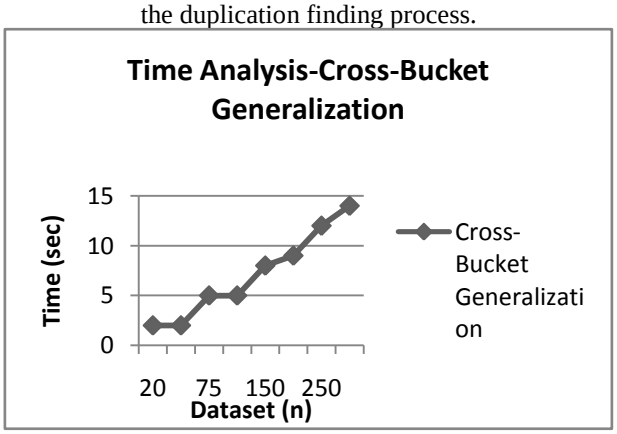

Figure2. Time Analysis Model – Cross-Bucket Generalization

It describes the experimental time analysis for Cross-Bucket Generalization methodology. It shows the number of records

taken for the process, number of record and execution time for the duplication finding process.

Table2. PSNM Model – Proposed Methodology

No of Records	Duplicate Records	Execution Time (Sec)
30	15	2
50	27	2
100	42	5
125	63	5
170	75	8
200	86	9
250	97	12
500	124	14

It describes the experimental dataset for the PSNM Model. It contains the number of report taken for the development, number of duplicate records find and the execution time for the duplication finding process.

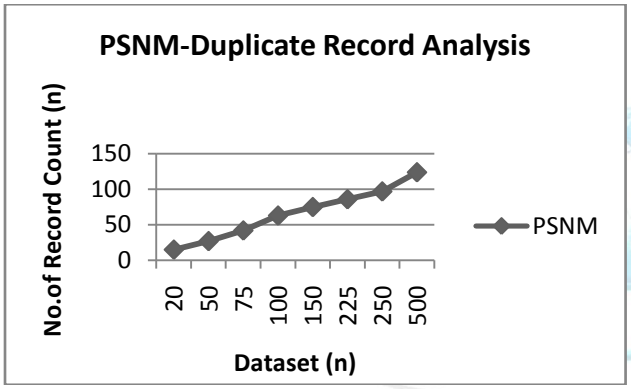

Figure3. Duplicate Record Detection –PSNM Model

The following Figure 4.3 describes the experimental duplicate detection for the proposed PSNM model. It shows the number of records taken for the process and number of duplicate records find for the duplication finding process.

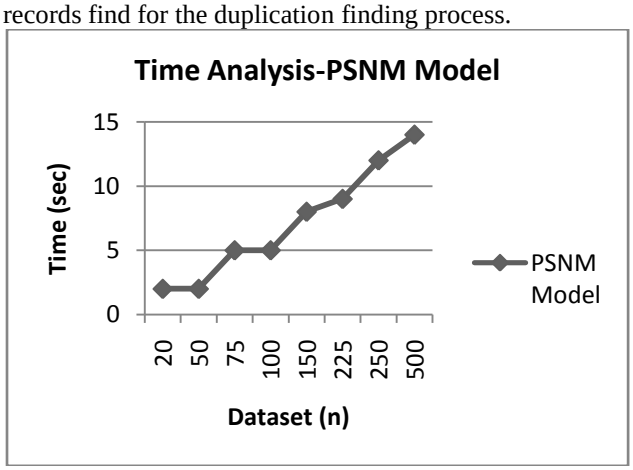

Figure4. Time Analysis- PSNM Model

It describes the experimental Time analysis for the PSNM Model. It contains the number of records taken for the process and the execution time for the duplication finding process.

V. CONCLUSION

In this thesis, efficiency of duplication detection is increased better than cross bucket system. Here introduced the PSNM (progressive sorted neighborhood process) and progressive blocking. Both algorithms will increase the ability of duplicate detection for situations with restricted execution time; they dynamically modification the ranking of comparison candidates based on intermediate results to execute promising comparisons first and fewer promising comparisons later. In projected system a completely unique quality measure for progressivity that integrates seamlessly with cross bucket measures. It uses multiple kind keys parallel to interrupt their progressive iteration. By analyze the intermediate results; every approach dynamically ranks the various kind keys at runtime, drastically easing the key selection drawback. It's believed that nearly all the system objectives that are planned at the commencements of the software package development are web with and therefore the implementation method of the thesis is completed. A trial run of the system has been created and is giving sensible results the procedures for process is easy and regular order. The method of preparing plans been ignored which could be thought of for any modification of the applying.

REFERENCES

[1]. Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu: "Privacy-Preserving Data Publishing: A Survey of Recent Developments ," on 22 May 2014.

[2]. K. LeFevre, D.J. DeWitt, and R. Ramakrishna, "Mondrian multidimensional k-anonymity," in Proc. IEEE Int. Conf. Data Eng., 2006.

[3]. A. Machanavajjhala, J. Gehrke, and D. Kifer, "L-diversity: Privacy beyond k-anonymity," in Proc. IEEE Int. Conf. Data Eng., 2006.

[4]. Q. Zhang, N. Koudas, D. Srivastava, and T. Yu, "Aggregate query answering on anonymized tables," in Proc. Int. Conf. Data Eng.,2007.

[5]. X. Xiao and Y. Tao, "Anatomy: Simple and effective privacy preservation," in Proc. 32nd Int. Conf. Very Large Data Bases, 2006.

[6]. T. Li and N. Li, "On the tradeoff between privacy and utility in data publishing," in Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining., 2009.

[7]. P. Wang, K. Wang, A. W. Fu, and R. C. W. Wong, "Generalized bucketization scheme for flexible privacy settings," Inf. Sci., vol.348, 2016.

[8]. Steven Euijong Whang, David Menestrina, Georgia Koutrika, "Entity Resolution with Iterative Blocking," July 2009.

[9]. Jayant Madhavan, Shawn R. Jeffery, Shirley Cohen, Xin (Luna) Dong, David KO, Cong Yu, Alon Halevy "Web-Scale Data Integration: You Can Only Afford To Pay As You Go," 2007.

[10]. Mauricio A. Hernandez, Salvatore J. Stolfo "The Merge / Purge Problem for Large Databases," Columbia University, New York.