International Journal on Emerging Technologies **10**(2): 167-173(2019)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

Data Integration and Data Privacy through "Pay-As-You-Go" Approach

E. Laxmi Lydia[1], R. PandiSelvam[2], R. Saranya[3], U.S. Kirutikaa[4], M. Ilayaraja[5], K. Shankar[6], Andino Maseleno[7]
[1]Associate Professor, Vignan's Institute of Information Technology(A),
Department of Computer Science and Engineering, Visakhapatnam, Andhra Pradesh, India.
[2]Assistant Professor & Head, PG Department of Computer Science, Ananda College, Devakottai, India.
[3]Assistant Professor, Department of Computer Science,
Dr. Umayal Ramanathan College for Women, Karaikudi, India.
[4]Assistant Professor, Department of Information Technology,
Dr. Umayal Ramanathan College For Women, Karaikudi, India.
[5]School of Computing Kalasalingam Academy of Research and Education, Krishnankoil, India.
[6]School of Computing Kalasalingam Academy of Research and Education, Krishnankoil, India.
[7]Institute of Informatics and Computing Energy, University Tenaga Nasional, Malaysia.

(Corresponding author: E. Laxmi Lydia)
(Received 03 May 2019, Revised 10 July 2019 Accepted 17 July 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Data Analytics has taken important and demanding problems in the research areas such as computer science, biology, medicine, finance, and homeland security. This research paper has resolved the problem of Entity resolution(ER) which recognizes the database records, which referred to the same real-world entity. The latest explosion of data made ER a impeach problem in a large range of applications. This paper proposed a scalable ER approach, used on-board datasets. Our latest approaches are simple because they consider either the entire ER process or the function, which are matching, and merging records as a black box procedure and used in a large range of ER applications. Pay-as-you-go approach for ER was a limit on the resources (e.g., work, runtime). This made the maximum progress as possible as required. This paper suggests scalable ER methods and new ER functionalities that have been not studied in the previous. Entity Resolution as a black-box operation provides general mechanisms which be used across applications. Further, the issue of managing information leakage, where one must try to avoid important bits of data from resolved by Entity Resolution, to sage against the loss of data privacy. As more of our sensitive data gets unprotected to various merchants, health care providers, employers, social sites and so on, there is a large chance that an adversary can "connect the dots" and piece together our data, which leads to even more damage of privacy. Thus to measure the quantifying data leakage, we use "disinformation" as a device which containing data leakage.

Keywords: Data Analytics, Data Integration, Data Privacy, Entity Resolution(ER), ER techniques.

I. INTRODUCTION

Since large amount of data is available for the analysis, scalable integration techniques playing an important role. At the same time, the latest privacy issues arise where sensitive data can be easily is inferred from a large amount of data. The two closely major related problems are identified with the analytics: data integration [21] and data privacy, "pay-as-you-go" approach for ER to maximize the progress of ER with a small amount of work. The problem of incremental ER, is not the one time process, but is continuously improved; as the data, schema, and applications better understand. The obstacles of joint ER with large datasets of various entity types are resolved together and the issue of ER with inconsistencies.

The objectives with prospective to data Integration keeps ER results updated when the ER logic is used go contrast records evolves time and again. A malleable, modular resolution framework where available ER algorithm developed for a given record type can be endeavour in and used in concert with another ER algorithm [8-9]. Suggested methods for efficiently generating hints and investigating of how ER algorithms cab is used hints to enlarge the number of records. Disallow inconsistencies in ER solution using Negatives rules of ER. The objectives with prospective to data privacy [18] provide effective algorithms for computing data leakage and emulate their achievement and scalability. Suggested mechanisms a disinformation technique [10-11] for entity resolution in order to manage data leakage is to develop a model which captures the privacy of loss relative to the target person, on a regular scale from 0 to 1.

II. LITERATURE SURVEY

Blocking strategies centre around improving the general runtime of ER where the records are isolated into potentially covering blocks, and the blocks are settled each one in turn [5-6].

Entity goals include contrasting records and deciding whether they allude to the same entity or not [2-3]. The vast majority of the work can be categorized as one of the ER models we consider, match based clustering and distance-based clustering [4]. While the ER writing centres on improving the precision or runtime execution of ER, they, for the most part, accept a fixed rationale

for reconciling records. As far as we could possibly know, our work is the first to consider the ER result update issue when the rationale for goals itself changes. Measures dependent on ER have proposed to evaluate the measure of delicate data that has discharged to the general population [1].

Whang, (2012) examines how can maximize the advancement of ER with a restricted measure of work utilizing "hints," which give data on records that are probably going to allude to a similar genuine element. A hint can be spoken to in different configurations (e.g., a gathering of records dependent on their probability of matching), and ER can utilize this data as a rule for which records to think about first. We present a group of strategies for building indications proficiently and methods for utilizing the insights to expand the quantity of coordinating records distinguished utilizing a constrained sum of work. Utilizing real data sets, we show the potential increases in our compensation as-you-go approach contrasted with running ER without utilizing indications [7].

The examinations are identified with the issue of disinformation. We expect that a "specialist" have some touchy data that the "adversary" is attempting to get. For instance, a camera organization (the specialist) may covertly be building up its new camera model, and a client (the foe) might need to know ahead of time the point-by-point specs of the model. The operator will probably spread false data to "weaken" what is known by the foe. We model the enemy as an Entity Resolution (ER) process that pieces together accessible data. They formalize the issue of finding the disinformation with the most noteworthy advantage given a constrained spending plan for making the disinformation and propose productive calculations for taking care of the issue. They at that point assess our disinformation arranging calculations on genuine and engineered information and look at the heartiness of existing ER calculations. Largely, our disinformation strategies can be utilized as a system for testing ER heartiness [7].

The P4P system looks to contain ill-conceived utilization of individual data that has discharged to a foe. For various kinds of data, universally useful systems are proposed to hold control of the information [12].

Clustering methods that are vigorous against clamor has contemplated widely in the Past. The vast majority of the work proposes clustering algorithms that locate the correct groups within the sight of superfluous clamor. Conversely, we adopt a contrary strategy where they probably work deliberately confound the ER calculation for the objective element however much as could reasonably be expected [13].

Various works propose effective similitude joins. Our pay-as-you-go methods improve hindering by additionally misusing the requesting of record sets as indicated by their probability of coordinating to deliver the best moderate ER results [14].

The notable bunching issue is to settle by K-implies, which is one of the least difficult solo learning calculations. Accept the K number of bunches for characterizing a given network processor in a basic and simple way. K-means clustering does not ensure for the ideal arrangement as the presentation depends on underlying K-centroids. Along these lines, the proposed framework utilizes the apportioning bunching, state, K-centroids clustering [15].

With the persistent work of Lydia *et.al* [15] Disparateness group condition is made alongside the properties of an asset, for example, asset type, preparing speed, and the memory. To stay away from the planning delay, the framework needs to shape a group utilizing the K-centroids clustering. Depending upon higher needs, the hub will move to the group [16].

Thilagam *et al* (2018) included cloud computing and distributed computing with hierarchical productivity [20]. Scientists applied testing approaching for supply users to accomplish all security and protection issues like providing data confidentiality, integrity, and availability. This has achieved cloud services [22] with reasonable cost, speed, productivity, performance, Reliability, pay per utilize, and workload. Furthermore, multi-cloud services are implemented [25].

In this paper, we considered two firmly related issues inside examination: information reconciliation and information protection, "pay-as-you-go" approach for ER [9] where we explore how to boost the advancement of ER with a restricted measure of work. Issue of steady ER [8], ER may not be a one-time process, and continuously improved, with respect to information, mapping, and application. The issue of joint ER where numerous datasets of various element types are settled together [10] and the issue of ER with irregularities [11].

In this paper, we first spread the issue of substance goals (ER), which recognizes database records that allude to a similar genuine element. The on-going blast of information has now made ER a difficult issue in a wide scope of uses. They proposed adaptable ER strategies implemented in huge datasets. Our strategies are general since they consider either the whole ER process or the capacities for coordinating and blending records as a discovery task and this would be able to utilize in a wide scope of ER applications. We likewise proposed a compensation as-you-go approach for ER to give a point of confinement in assets (e.g., work, runtime) we endeavour to gain the extreme ground conceivable.

We proposed versatile ER strategies and new ER functionalities that have not been concentrated previously. We likewise see ER as a discovery task and give general methods that utilize crosswise over applications. Next, we present the issue of overseeing data spillage, where one must attempt to keep significant bits of data from being opted by ER, to make preparations for loss of information protection. As a greater amount of our delicate information is presented to an assortment of traders, medicinal services suppliers, businesses, social locales, etc., there is a higher possibility that an enemy can "come to an obvious conclusion" and sort out our data [24], prompting significantly more loss of security. We propose a measure for evaluating data spillage and use "disinformation" as an instrument for containing data spillage.

A. Methodology

Proposed Model: Due to the very large datasets, the process has become very expensive to compute and compare records. For instance, the collection of people's profiles over social media tends to millions of records that need to be resolved. In such a case, a comparison of any pair of records needs a logical applicational tool. The most recovery process of this situation is resolved by using an ER within a limited amount of time. Entity Resolution [23] for the performance of results has identified an approach named "Pay-as-you-go". Its main intention is to work for obtaining faster results for similar records that end to real-world entities. In addition, it unifies the entire structure of potential evaluation. Fig. 1 demonstrates the flowchart of methodology steps in brief for Data Integration and Data Privacy.

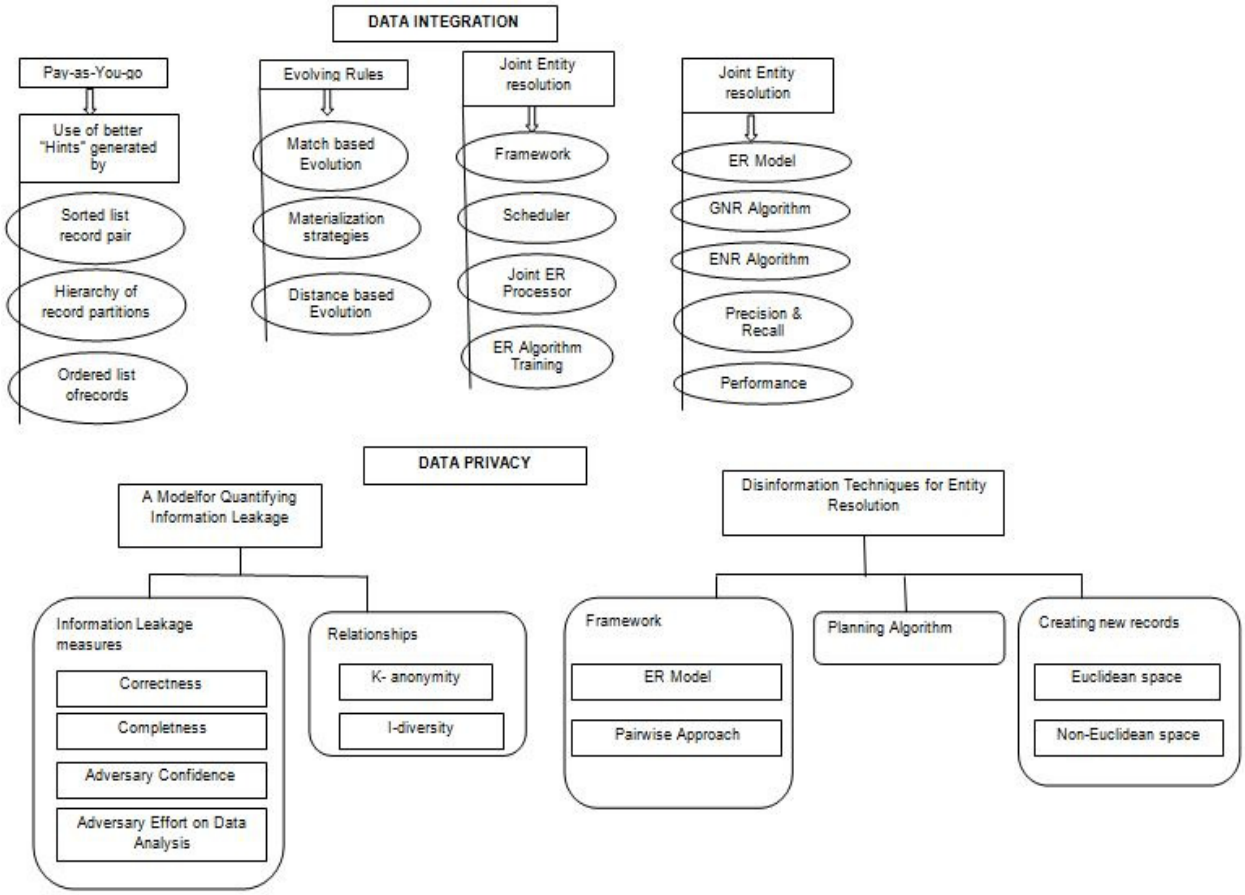

Fig. 1. Flow chart of methodology steps.

Adapted Methodology

- A standardized technique using ER constructs blocking that improves the partial ER result for scaling. This formalizes pay-as-you-go.
- There are three forms of hints
  - Most informative, Least Compact form of a hint for a sorted list of record pairs.
  - The moderately informative, moderately compact form of a hint for Hierarchy of Partitions.
  - Most compact, least informative form of the sorted list of records.
- Every hint form proposes effective techniques for developing ER algorithms to maximize its quality of ER and minimize no. of record comparisons.
- This paper has extended its approaches using multiple hints.
- On applying good quality of ER, results are fast. We have compared shopping data and hotel data from Yahoo.

Design Structure: The proposed model using ER for "Pay-as-you-go" is elaborated clearly using the general model as follows:

Entity Resolution Model: An ER algorithm considers a set of records as input (E), which is a real-world entity. The partitioned input is the output figured from group of records. Suppose, the output for G = {{r1, r3}, {r2}, {r4, r5, r6}} indicates that records r1 and r3 shows only one entity, r2 uniquely represent an independent entity. In some cases, the output is based on the before resolution, when we originally represent the input in

partitions like e.g., {{r1}, {r2}, {r3}, {r4}, {r5}, {r6}}. E(R) [t] denotes ER algorithm with 'E' at 'R' time.

Pay-as-you-go Model: In this model [17], a visionary approach i.e., candidate pairs are ordered likely for the match. The ER process implements the record correlations to math more likely pairs. This corresponds to the approximate and most efficient pairs for ordering.

Furthermore, suppose we need to place 6 records into 2 blocks. The first block containsr1, r2, and r3, while the second block contains r4, r5, and r6. The implied candidate pairs are {r1 - r2, r1 - r3, r2 - r3, r4 - r5 ….}. On applying the ER algorithm, the input pairs first consider all the pairs in the first block arbitrarily and compare them r5 - r6, later compare the next most likely pair r2 - r3. Nevertheless, when one block fits in memory at a time, we consider each block separately i.e., firstly we compare the pairs in the block by descending then move on to the second block. There the main intention is to achieve match pairs quickly using candidate pairs. On the other side, the ER algorithm will develop an output partition approximate to the result.

From the first condition, we aim is obtained by generating higher-quality ER outcomes. From the second condition, the pay-as-you-go approach will generate the same quality of results without using any hint forms. Through comparative study, the candidate pairs are matched approximately through an auxiliary data structure. Fig. 2 demonstrates the three forms of hints:

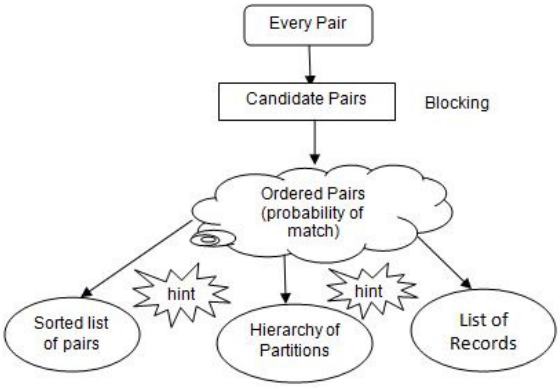

Fig. 2. "Pay-as-You-Go Framework".

Record Pairs with Sorted List: Here the hint form maintains the list of record pairs and ranks of the pair matches. ER algorithm uses functions like distance or match functions.

The distance function, d(r, s) finds the distance between the records r and s; smaller distance shows more likely real-world entities.

The match function, m(r, s) also represents the same real-world entity. This function will also use distance function like if d(r, s) <T and other conditions are true. T represents the threshold.

We also have estimator function e(r, s) is less valuable to compute when compared to distance and match functions. Mostly ER uses match function.

Theoretically, the lists of recorded pairs are increase by e value. In practice, they are not explicitly and fully initiated. When some constant number of pairs after giving threshold, ER algorithm will generate request by initiating pairs " on-demand" to the request of the list of pairs. The obtained complexity is O (N_2) for the hint.

Generation and practicing Application Estimates: Generally, to build an application-specific estimate function for computation at cheap and check for the functions as if the distance function performance needs to compute and merge the similarities of considered attributes most significantly.

To build the hint form, e(r, s) for any n number of record pairs, every pair needs to be inserted into a heap data structure to estimate with smallest pairs. Later we try to remove all the ascending estimate pairs. In some cases, if we try to obtain top estimates, we remove entries until threshold distance is obtained with a limited number of pairs otherwise until ER stops requesting for pairs.

For an alternative case, suppose the estimates maps into distances in one Dimensional, heap data will be reduced. That is, e(r, s) needs to identify the price attributes of records. Every record that entered into the memory closest neighborsis recognized and the price difference is checked. The smallest estimate pair is verified in the heap by checking next neighbors and finally, we again re-upload the records with a new estimate and save all the records in heap at the order of O (|R|2).

No Availability of Application Estimate: There may be some cases where no known application-specific estimate functions used. In such cases, a generic but rough estimate related to sampling is selected. This will not always give better results.

The most natural way to work on expensive function (d) is to estimate the distance from all small subset of record pairs and later to the other pairs of records. Here the distance need not be absolute. Sampling technique uses distance function through estimate distances.

If Sample S is represented as a subset of R records, original distances between S and R, among S, are measured. S is always smaller than all the number of records R. For instance, if |R| has 1000 records and |S| has 10, then the actual distance is calculated by

$$\frac{\frac{10}{2} + 990 \times 10}{\frac{1000}{2}} = \frac{9945}{499500} \approx 2\%$$

For any given estimation of real distances, with two records p and sum of square difference q among d (p, t) and d (t, q), we have

$$e(p,q) = \sum_{t \in Q}^{n} (d(p,t) - (t,q))^2$$

We observe that the sample set has a quality of estimation. For the last case, |S| has the same pair of records. Thus, the sample records are circulated within R with desired approximate possibilities.

Record Partitions for Hierarchy: Records are partitioned based on the hierarchical order to achieve possible formats for hints. Partitions are performed according to the match records at different levels of partitions using the ER. It follows bottom-most level clustering records as inputs. These partitions are stored arbitrarily using higher hierarchy order.

Generation: while building partition hierarchy, various approaches are developed. Here sorted records rely on hint forms that point to application estimates. These partitions can also be performed depending on hash functions and inexpensive distance functions. Following is the algorithm1 for partition hierarchy using sorted records hint form:

Step1: Consider list of sorted records and list of thresholds i.e, Sorted = {r1, r2 ...}; T = {T1, T2 ...TL}

Step2: hint H = {P1 ...PL}

Step3: Initialize partitions P1… PL

Step4: for r € Sorted do

Step5: for Tj, € sorted do

Step6: if r.prev.exists () ^KeyDistance (r.key, r.prev.key) <=T_j then r is added to P_j the new cluster or else

Step7: initiate new cluster P_j that contains r

Step8: return {P1 ...PL}

The above stepwise algorithm1 shows the creation of partition hierarchy by setting different thresholds using hint depending on key distance value. All the thresholds values that are initiated are user-defined for different levels of hierarchy. Suppose, let us assume three records [tom, tommy, tombo] and set threshold values as T1=2 and T2=3. Basing on the edit distance, the key distance among the records is estimated (they convert one string to another if necessary operations like inserts and deletes).

Now when the algorithm first reads tom, it adds to new cluster P1 and P2, then read tommy. From here, the comparison starts as it has the previous record. Calculate the distance, check it with the threshold T1, and add it to the new cluster. Likewise, readtombo, calculate edit distance, check with the previous records, and add to the cluster. The obtained partitions are P1 = fftomg, ftommyg, ftombogg and P2 = {{tom, tommy}, {tombo}}. This shows the perfection of algorithm1.

Hypothesis: Algorithm1 returns a valid hint

Demonstration: The lower level partition and higher-level partitions are varied depending on the thresholds. All records are split and organized in the sorted list. As the higher level partitions are particulate than lower-level partition. The sorted records are given as input with hint H. The time complexity for iterating all records and all thresholds are O(L-|R|).

-The relevance of the Paper to the work already going on in the organization: None

-Implementation arrangements proposed for the Paper (linkages and management structure)

III. RESULT ANALYSIS

This Paper would experiment on a comparison-shopping dataset provided by Yahoo! Shopping and a hotel dataset provided by Yahoo! Travel. It would evaluate the following ER algorithms: SN, HCB, HCBR, ME, HCDS, and HCDC. These algorithms are implemented in Java, and our experiments were run on a 2.4GHz Intel(R) Core 2 processor with 4GB of RAM. The detailed software information's are as following Table 1. Since Hadoop and Mahout are worked with Java locally, it would be simple for interoperability between the parts created with Java.

Table 1: Software Languages.

Programming Language Referred	Java
Platform used	Only tested in GNU/Linux
Preferred IDE	NetBeans IDE 6.0.1
UML Software Documentation	Umbrello UML Modeler 2.0.3

Considering this reality, Java was picked as the programming language. Umbrello UML Modeler has a basic yet effective arrangement of demonstrating instruments, because of which was utilized for UML Documentation. NetBeans IDE was picked as formative IDE accounting to its rich arrangement of elements and simple GUI Builder tool.

Datasets: a collection of *shopping dataset*s considered by Yahoo, which contains millions of shopping records on a regular basis. Online shopping data, queries from customers are maintained and recorded. Each record may contain many attributes depending on the item such as its price, its name, and category. In this paper, randomly 3000 shopping records with a subset of one million people records are selected for experimental study.

The same procedure is also applied for the *hotel dataset* by yahoo. Travels that go across the world by their needs stay in hotels. Yahoo maintains the hotel address and keeps records of the customer details. In this paper, randomly 3000 hotel records with a subset of one million people records located in the United States are selected for experimental study.

The 3K datasets from shopping and hotel need to fit in memory; the 1 million shopping dataset did not fit in memory, which is stored using the disk.

Implemented Match Rules: Following is the Table1that gives the match rules provided in our experiments. Among the three columns, type column represents the match rules that are Boolean or distance, the data column represents the dataset from where the data is collected i.e., shopping or hotel records, the third column represents the match rules that are applied on datasets. The table clearly mentions the Boolean match rules for both shopping and hotel datasets and distance match rules for both shopping and hotel datasets.

For the shopping datasets, we see that B_1^S compares the names and categories of two shopping records using Boolean match rule while B_2^S compares the names and prices of shopping records using Boolean match rule.

For the hotel data, we see that B_1^Hcompares the states, cities, zip codes, and names of two hotel recordsusing Boolean match rule. TheB_2^H rule compares the states, cities, zip codes, and street addresses of two hotel records using Boolean match rule. The following Table 2 shows the type of the data, considered data and applied to match rules.

Table 2: Match Rules.

TYPE	DATA	MATCH RULES
Boolean	Shopping	$B_1^S : p_{ti} \bigwedge p_{ca}$ $B_2^S : p_{ti} \bigwedge p_{pr}$
Boolean	Hotel	$B_1^H : p_{st} \bigwedge p_{ci} \bigwedge p_{zi} \bigwedge p_{na}$ $B_2^H : p_{st} \bigwedge p_{ci} \bigwedge p_{zi} \bigwedge p_{sa}$
Distance	Shopping	$D_1^S : Jaro_{ti}$ $D_1^S : Jaro_{ti}$ changes randomly within 5%
Distance	Hotel	$D_1^H : Jaro_{na} + 0.05 \times Equals_{ci}$ $D_2^H : Jaro_{na} + 0.05 \times Equals_{zi}$

For the shopping data, we see that D_1^S measures the Jaro distance between the names of two shopping records using Distance match rule while D_2^S randomly alters the distance of D_1^S by a maximum ratio of 5% using Distance match rule. The Jaro distance will return a value within the range of [0, 1], and provides higher values for closer records.

For the hotel data, we see that D_1^H sums the Jaro distance between the names of two records and the Equality distance between the cities of two records weighted by 0.05 using Distance match rule. Here it defines the Equality distance to return 1 if two values are the same and 0 if they are not the same using Distance match rule.

The D_2^H rule sums the Jaro distance between names with the Equality distancebetween the zip codes of two records weighted by 0.05. As a result, the D_1^H distance canalter by at most the constant 0.05.This paper implements ER algorithms like SN, HCB, HCBR, ME, HCDS, and with HCDC. They mostly focus on cluster ER models rather rule evolution.

Following is the Table3that summarizes the algorithms for ER and Rule Evolution. The HC_{DS} and HCDC distanced-based clustering algorithms terminate when the minimum distance between clusters is smaller than the threshold 0.95 (recall that closer records have higher Jaro + Equality distances). Although the ME and HC_{DC} algorithms do not satisfy the RM property, we can still use Algorithm 7 to efficiently produce new ER results with a small loss inaccuracy. Notice that, although GI, Algorithm 3 is not efficient because of the way 'ME algorithm' extracts all records from the input partition Pi (without exploiting any of the clusters in Pi) and sorts them again. Both the C_{DS} and HC_{DC} algorithms use Algorithm 7 adjusted for the distance-based clustering model.

Table 3 explains the ER algorithms and its corresponding Rule Evolution algorithms used for testing.

Table 3: ER and Rule Evolution algorithms tested.

ER algorithm	Rule Evolution algorithm used
SN	Algorithm for SN
HC_B	Algorithm 3
HC_{BR}	Algorithm 7
ME	Algorithm 7
HC_{DS}	Algorithm 7
HC_{DC}	Algorithm 7

IV. CONCLUSION

The research paper focused on challenging problems in data analytics, data integration, and data Privacy with the implementation of Entity Resolution, which is resolved by the ER techniques at large databases. This paper introduced the implementation of the "pay-as-you-go" approach and algorithms related to the ER. It limits the run-time of the resources and handles the information leakage of data to provide data privacy. The research paper proposed various measures to estimate the information leakage and tools for "disinformation".

V. FUTURE SCOPE

For the future enhancement of the present work, it can be implemented on complex problems, using the different datasets of various Social media website like Google, Facebook, Twitter, Instagram so on, focusing on the data privacy and data integration using Entity Resolution.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1]. Whang, S.E., Garcia-Molina, H., (2012). A model for quantifying Information Leakage, *In Jonker W., Petkovic M. (Eds) Secure Data Management. SDM*, *Springer*, Vol. **7482**: pp 25-44.

[2]. Sanderson, Mark & Bruce Croft, W. (2012). The history of information Retrieval research. *Proceedings of the IEEE-PIEEE 100*:1444-1451. 10.1109/JPROC.2012.2189916.

[3]. Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven E. Whang, and Jennifer Widom, (2009). Swoosh: a generic approach to entity resolution, *International Journal on Very Large Data Bases.,* **18**(1): 255–276

[4].Bhattacharya, I., & Getoor, L. (2004). Iterative record linkage for cleaning and integration, *Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery- DMKD'* 04.pp:11-18, DOI: 10.1145/1008694.1008697

[5]. Steorts, R.C., Ventura, S.L., Sadinle, M. Fienberg, S.E. (2014). A Comparison of Blocking methods for record linkage. *International Conference on Privacy in Statistical Databases*, Vol. **8744**, pp: 253-268.

[6]. McCallum, A., Nigam, K., & Ungar, L.H. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on*

Knowledge discovery and data mining (pp. 169-178). ACM.
[7]. Whang, S.E., & Garcia-Molina, H. (2013). Disinformation techniques for entity resolution, *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management- CIKM'13*. pp:715-720, ISBN: 978-1-4503-2263-8. Doi:10.1145/2505515.2505636.
[8]. Whang, S.E. and Hector Garcia-Molina, (2010). "Entity resolution with evolving rules", *PVLDB*, **3**(1): 1326–1337.
[9]. Whang, S.E., Marmaros, D. and Garcia-Molina, H. (2012). Pay-as-you-go entity resolution, *IEEE Transactions on Knowledge and Data Engineering*, **25**(5): 1111-1124.
[10]. Whang, S.E., & Garcia-Molina, H. (2012). Joint entity resolution, *In IEEE International Conference on Data Engineering*.doi: 10.1109/icde.2012.119
[11]. Whang, S.E., Benjelloun, O., and Garcia-Molina,H (2009) Generic entity resolution with negative rules, *VLDB Journal*, **18**(6): 1261–1277.
[12]. Agawam, G., Bawa, M., Ganesan, P., Garcia-Molina, H., Kenthapadi, K., Mishra, N. Motwani, R., Srivastava, U., Thomas, D., Widom, J., and Xu, Y., (2004). Vision Paper: Enabling Privacy for the paranoids. *In VLDB*, pp: 708–719.
[13]. Xu, R., & Wunsch, D., (2005). Survey of clustering algorithms, *IEEE Transactions on Neural Networks*, **16**(3): 645–678.
[14]. Arasu, A., Ganti, V., & Kaushik, R. (2006, September). Efficient exact set-similarity joins. In *Proceedings of the 32nd international conference on Very large data bases* (pp. 918-929). VLDB Endowment.
[15]. Laxmi, C.V.T.E.V. & Somasundaram, K. (2015). 2HARS: Heterogeneity-Aware Resource Scheduling In Grid Environment Using K-Centroids Clustering and PSO Techniques, *International Journal of Applied Engineering Research,* **10**: 18047-18062.
[16]. Lydia, L., Swarup, M. Ben (2016). A Disparateness-Aware Scheduling using K-Centroids Clustering and PSO Techniques in Hadoop Cluster, *International Journal of Advanced Network, Monitoring, and controls*, **1**(2): 34-46.
[17]. Anand, V., Moiz Qyser, Ahmed Abdul (2014). A Comparative study of secure search protocols in Pay-as-you-go Clouds, *International Journal of Research in Engineering and Technology*, **03**(05).
[18]. Anil Kumar N., Althaf Rahaman, S.K., Girija, K., (2018). An approach towards data security in organizations by avoiding data breaches through standards of DLP, I*nternational Research Journal of Engineering and Technology(IRJET),* **5**(10), pp: 580-584.
[19]. Anandan, R., Bhyrapuneni, S., Kalaivani, K., Swaminathan, P., (2018). A Survey on big data analytics with deep learning in the text using machine learning mechanisms, *International Journal of Engineering & Technology*, **7**(2.21): 335-338.
[20]. Thilagam, T., Arthi, K., Amuthadevi, C., (2018). A Survey on Security and Privacy issues in cloud computing, *International Journal of Engineering & Technology*, **7**(2.4): 88-92.
[21]. Dong, X. L., & Rekatsinas, T. (2018). Data Integration and Machine Learning. *Proceedings of the 2018 International Conference on Management of data - SIGMOD'18ACM,* pp: 1645-1650 Doi:10.1145/3183713.3197387.
[22]. Elsayed, M., & Zulkernine, M. (2019). Offering Security diagnosis as a service for cloud SaaS applications. *Science Direct, Journal of Information Security and Applications,* **44**, 32-48. Doi: 10.1016/j.jisa.2018.11.006.
[23]. Kooli, N., Robin, A., &Pigneul, E., (2018). Deep Learning-based approach for Entity Resolution in Databases. *Intelligent Information and Databases systems,* pp.3-12, DOI: 10.1007/978-3-319-75420-8_1.
[24]. Blazquez, D., Domenech, J., (2018). Big Data Sources and methods for social and economic analyses, *Science Direct, Technological Forecasting & Social Change,* **130**: 99-113.
[25]. Alshammari, M.M., Alwan, A.A., Nordin, A., Abualkishik, A.Z., (2019). Disaster Recovery with minimum replica plan for reliability checking in multi-cloud, *Procedia Computer science,* **130**: 247-254.

How to cite this article: Lydia, E.L., Pandi Selvam, R., Saranya, R., Kirutikaa, U.S., Ilayaraja, M., Shankar, K. and Maseleno, A. (2019). Data Integration and Data Privacy through "Pay-As-You-Go" Approach. *International Journal on Emerging Technologies*, **10**(2): 167-173.