Detection of Data Leaks through Large Scale Distributed Query Processing using Machine Learning

Kiranmai MVSV[1]
Assistant Professor (C) and Research Scholar
Department of CSE, UCEK, JNTUK
Kakinada, India

D Haritha[2]
Professor
Department of CSE, UCEK, JNTUK
Kakinada, India

***Abstract*—With the growth in the distributed data processing and data being the fuel for each of the processes, the query processes of the data are expected to be significantly lower. Hence, the distribution of the data is highly expected and during the distributing of the data, the chances for data leakage increases to a significant extend. The data leakage problems are not generally caused by intentional errors, rather this is caused by the higher visibility of the data over multiple clusters. Henceforth, the detection process is also very critical. Many of the parallel research attempts have demonstrated various methods for the detection and as well as the prevention methods. The works in the direction of the detection of the data leaks are highly dependent either on the historical information of the leaks or depends on the contextual importance of the data. In both the cases, the outcomes of the detection process accuracy cannot be ensured. In the other hand, the preventive measures can also turn into a reactive process for detection by reversing the principles proposed in these research outcomes, but the computational complexities are significantly higher. Thus, this work proposes a novel strategy for detection of the data leakages after the data distribution during the query processing events. This work proposes an initial Occurrence Based Rule Set Extraction method using Adaptive Threshold for generating the rulesets, further for reducing the time complexity and reducing the loss of dataset attribute information, this work introduces yet another algorithm for Dynamic Inference-based Rule Set Reduction. After the inferences are generated, finally this work deploys the Attribute Subset Equivalence-based Leak Detection mechanism for final detection of the clusters with data leaks. This work demonstrates nearly 89% accuracy for the detection process.**

Keywords*—Distributed query processing; distributed data leak; data leak detection; attribute subset equivalence; dynamic inference; adaptive threshold model introduction

I. Introduction

Leakage of the data during centralized or distributed query processing environment is one of the primary concerns in the recent cyber security domain. Number of professionals and researchers have aimed to define the problem of data leakage in the past decade and aimed to provide solutions considering the higher number of cases are reported for data leakage, which leads to data breaches. As per the reports by Breach Level index [1], a gigantic volume of data, as close to 4.5 Billion, are compromised only in 2018.

The empirical study by C. Missaoui et al. [2] have critically analysed the data leakage situations and formulated the correct sequence of events, which might lead to the data breach. The common reason for data leakage, as per this study, showcased that the use of unused data stored in the centralized or distributed storage solutions leads to data leakage. The example also demonstrated that the data, which is left unattended or not in motion, must be revoked or should be distributed with a specified life time and expiry to be restricted for reuse.

Yet another case study presented by IBM showcased that the cost of recovery of stolen or leaked data can be as high as 3.86 Billion dollars [3].

Thus, the demand for data protection and data leakage detection is one of the primary demands from the current research trends and must be addressed. Also, the parallel research outcomes have demonstrated that the security concerns are higher in case of the distributed data sources for the obvious reasons. Thus, this work focuses on the distributed query processing environments to detect the data leaks.

The rest of the work is furnished such as in Section – II, the fundamentals of the distributed query processing is discussed and understood, in Section – III, the outcomes from the parallel research attempts are analysed, in Section – IV, the problem is formulated using mathematical model, in Section – V, the proposed solution is again formulated using mathematical modelling method, in Section – VI, the proposed algorithms are furnished and elaborated, in Section – VII, the obtained results are analysed, in Section – VIII, the comparative analysis is presented and in Ssection – IX, the final conclusion of the research is presented.

II. Distributed Query Processing Fundamentals

In this section of the work, the fundamental of distributed query processing method is discussed in order to realize the recent outcomes from the parallel research attempts.

Assuming that, every query Q is the collection of relations and predicates as R and P respectively. This relation can be formulated as,

$$Q \rightarrow \sum R . \sum P \tag{1}$$

Or for example, the above relation can be re-written as,

$$Q \rightarrow [R_1 \cap R_2] \cup R_3 \tag{2}$$

Also, assuming that the relation R, is distributed over the cluster set C[], where each and every cluster can be identified as C_X for "n" number of clusters. Thus, can be represented as,

$$C[] \leftarrow \sum_{x=1}^{n} C_x \tag{3}$$

And,

$$R[] \rightarrow C[] \tag{4}$$

Further, assuming the data distribution is as follows,

$$\{R_1, R_2, R_3\} :: \{\langle C_1, C_2, C_3 \rangle, \langle C_2, C_3 \rangle, C_3\} \tag{5}$$

During the query processing for distributed systems, the primary objective is to find the allocation and minimum distribution of the relations. As,

$$\{R_1, R_2, R_3\} :: \{C_3\} \tag{6}$$

Further, the optimization possibilities for predicates can also be identified and performed. Once the optimization task is completed, the result of the query can be return to the queue buffer.

Henceforth, this fundamental understanding of the distributed query processing shall help in realizing the parallel research outcomes, which are discussed in the next section of the work.

III. Parallel Research Outcomes: Survey

After the fundamental understanding of the distributed query processing and chances for the data leakages, in this section of the work, the parallel research outcomes are discussed.

In the recent years, multiple organizations have aimed to provide the complete solution for the detection and up to some extend prevention of the data leakages problems. The primary working principles of these systems are to manual perform exhaustive search with the previously leaked data on the existing shared data for finding the leakage. The point to be mentioned here is that, the data leakage may not be an intentional issue every time. Many of the times, it is been observed that, the wrong distribution of the data attributes leads to the data leakages. Thus, the search option of the previous information may demonstrate machine learning characteristics, but eventually leads to failure in case of insufficient previous or historical data; thus, the other approaches getting popularity in the practice.

One of the major benchmarks in the domain of data leakage detection was the research outcome by P. Papadimitriou et al. [4]. This work demonstrates the data leakage detection by introducing watermarking methods to identify the source of leakage. However, this method was highly criticised by many other researchers due to the higher complexity of the computational models. Also, the size of the actual data, which is distributed, increases to a significant extend because of the replication for each watermarking information and the second issue was that the digital watermarking process is fragile as because of the pre-processing methods used by many algorithms can lead to the loss of watermarks and making the complete process again vulnerable.

The other parallel research outcome by L. Cheng et al. [5] has demonstrated another method to data leakage problem solution. This method demonstrates an approach to classify the content based on the sensitivity of the information and manage the distribution of the higher sensitive data with maximum care. This work is also criticised by the parallel researchers due to the facts that, firstly, the data sensitivity also depends on the context of the data, which is highly variable in all the instances of the query, and secondly, the higher time complexity and chances of low data visibility is also a challenge.

Further, in the complete other direction from these solutions, the work by S. Liu et al. [6] have demonstrated and listed the principles of data leakage preventions. This direction, in contract to the detection of the data leakage, promotes the prevention methods. Elaborating this fact, the work by B. Hauer et al. [7] have also showcased the functional rules for making the data leakage preventions. The set of mentioned rules can also be reversed to identify the leakage sources. Nevertheless, it is natural to realize that, the computational time complexity can be very high for this detection method and also for the distributed environments, this method is prune to errors. The next method is one of the extensions, proposed by T. Malderle et al. [8]. This work elaborates the mechanisms for collecting evidences of the data leakage and further validates the evidences against the prevention principles.

The data leakages are not only limited into the scope of centralized data, rather also extended in higher scale for distributed data. The work by J. Schütte et al. [9] has elaborated on the challenges of data leakage for distributed mobility devices.

The work by S. Trabelsi et al. [10,11,12] provides the summery of the challenges and failure points of each of the above-mentioned mechanisms.

Henceforth, in order to provide the solution to data leakage detection, in the next section of the work, the mathematical model of the actual problem is furnished for better identification of the solution possibilities.

IV. Problem Formulation

After the fundamental understanding of the query processing and the review of the parallel research outcomes, in this section of the work the research problem is formulated.

Assuming that the complete data schema is denoted as DSC[] and every attribute is denoted as AR_X. Thus, for n number of attributes, the total relationship can be formulated as,

$$DSC[] \leftarrow \sum_{i=0}^{n} AR_i \tag{7}$$

Also, one of the attributes in the total attribute set must be identified as the class variable and can be denoted as AR_C. This can be formulated as,

$$\prod_{i=0}^{n} AR_i \cup \prod_{j=1}^{n-1} AR_j \rightarrow AR_C \tag{8}$$

It is natural to realize that the class variable or the class attribute can also be retrieved using other attributes as well and can be denoted as,

$$\prod_{x=0}^{n} AR_x \cup \prod_{y=1}^{n-1} AR_y \rightarrow AR_C \tag{9}$$

In case of a distributed query processing environment, the data is expected to be distributed over multiple clusters, denoted as C[] and each and every cluster can be identified as, C_i. Thus, this relation for k number of clusters can be identified as,

$$C[] \leftarrow \sum_{i=1}^{k} C_i \tag{10}$$

The data is expected to be distributed over the clusters from the initial schema sets and can be realized as,

$$\prod_{i=1}^{n} DSC[i] \Rightarrow \prod_{j=1}^{k} C[j] \tag{11}$$

This can be re-written as,

$$\prod_{i=1}^{n} AR_i \Rightarrow \prod_{j=1}^{k} C[j] \tag{12}$$

It is often to be realized that during the query processing, the similar information can be generated from different attributes and the data leaks can happen.

For example, as the attribute sets AR_i, AR_x and AR_j, AR_y can contain similar information, thus these sets if become part of same clusters during data distribution, then the data leakage is obvious and cannot be prevented.

$$\langle AR_i, AR_x \rangle \neq C_\alpha \tag{13}$$

And

$$\langle AR_j, AR_y \rangle \neq C_\beta \tag{14}$$

Or,

$$C_\alpha \neq C_\beta \tag{15}$$

Henceforth, detection of the data leakage from the distributed schema is the identified problem to be solved. In the next section of this work, the proposed mathematical model for the data leak detection is proposed.

V. Parallel Research Outcomes: Survey

After the fundamental understanding of the parallel research outcomes and the formulation of the problem, in this section of the work, the mathematical model for solution is elaborated.

Assuming that the complete data schema is denoted as DSC[] and every attribute is denoted as AR_X. Thus, for n number of attributes, the total relationship can be formulated as,

$$DSC[] \leftarrow \sum_{i=0}^{n} AR_i \tag{16}$$

Also, one of the attributes in the total attribute set must be identified as the class variable and can be denoted as AR_C. This can be formulated as,

$$\prod_{i=0}^{n} AR_i \cup \prod_{j=1}^{n-1} AR_j \rightarrow AR_C \tag{17}$$

Further analysing the item set frequency for each data item sets, the ruleset, R[] can be generated for "d" number rules and each rule in the ruleset can be considered as R_X. Thus, this relation can be formulated as,

$$R[] \leftarrow \sum_{i=1}^{d} R_i \tag{18}$$

Also, assuming that two different rules, R_X and R_Y, implies the same class variable data instance, AR_C, as,

$$R_X \rightarrow AR_C \tag{19}$$

And,

$$R_Y \rightarrow AR_C \tag{20}$$

However, the rules can contain different attribute sets, as,

$$R_X = \langle AR_i, AR_{i+1}, AR_{i+2}, ...AR_n \rangle \tag{21}$$

Or,

$$R_X = AR'[] \tag{22}$$

And,

$$R_Y = \langle AR_j, AR_{j+1}, AR_{j+2}, ...AR_m \rangle \tag{23}$$

Or,

$$R_Y = AR''[] \tag{24}$$

Finally, in order to detect the data leak based on the large-scale distributed query processing, both the attribute sets, must not coexist on the same cluster, as formulated as,

$$(AR'[], AR''[]) \rightarrow C_{\alpha} \tag{25}$$

If the above situation is detected, then the data leakage can occur, and the security challenges can be increased.

Further, in the next section of this work, based on the problem formulation, the proposed algorithms are furnished and elaborated.

VI. Proposed Algorithms

Furthermore, in this section of the work, based on the proposed mathematical model of the solution in the previous section, the proposed algorithms are furnished.

The first algorithm is designed to generate the rulesets from the given schema and the dataset items. The algorithm is furnished here:

Algorithm - 1: Occurrence Based Rule Set Extraction using Adaptive Threshold Model (**OBRSE-ATM**)

Input: {Read the schema definition for attribute sets, A[] and Class variable, ARC}
Output: {Rulesets, FR}
Process:
Step - 1. Accept the list of attributes as A[]
Step - 2. Accept the class variable as ARC
Step - 3. For each item sets
  a. For each attribute value as A[i]
    i. If A[i] and A[i+1] generates ARC[i]
    ii. Then, count the number of occurrence as O[i] and Rule[i] = A[i] and A[i+1]
  b. End
Step - 4. For each occurrence as O[i]
  a. Calculate the mean as OM = {Sum(O[])}/{Count(O[])}
  b. Calculate the position of the OM as O[k]
  c. Calculate the adaptive threshold as AT = K/{Count(O[])}
Step - 5. For each occurrence as O[j]
  a. If O[j] > OM * AT
  b. Then, Accept R[j] as final rule and add to FR[i]
Step - 6. Report FR[]

To instigate the best principles dependent on a given perception, RULES family start by choosing (isolating) a seed guide to assemble a standard, condition by condition. The standard that covers the best models and the least negative models are picked as the best principle of the present seed model. It permits the best guideline to cover some negative guides to deal with the expansion adaptability and lessen the over fitting issue and uproarious information in the standard enlistment.

The second algorithm is designed to reduce the redundant rule sets and build the final reduced rulesets.

This articulation expresses that at whatever point over the span of some legitimate inference the given premises have been gotten, the predetermined end can be underestimated too. The specific conventional language that is utilized to portray the two premises and ends relies upon the real setting of the determinations.

Algorithm - 2: Dynamic Inference-based Rule Set Reduction (**DI-RSR**)

Input: {Rulesets, FR}
Output: {Reduced Rulesets, FRR}
Process:
Step - 1. Accept the rule sets as FR
Step - 2. For each FR[i]
  a. Generate the attribute set as AR[j]
  b. If AR[j] is subset of AR[j]
  c. Then, remove the AR[j] and FR[i]
  d. Else, Keep FR[i] into FRR[k]
Step - 3. Build the final rule set as FRR[]

The final algorithm is built for detecting the leaks in the distributed query situations and as furnished here.

Algorithm - 3: Attribute Subset Equivalence-based Leak Detection (**ASLD**)

Input: {Final Rulesets, FRR and Cluster Sets, CS}
Output: {Leaked Clusters, LC}
Process:
Step - 1. Accept the final rule sets as FRR[]
Step - 2. Accept the cluster distributions CS[]
Step - 3. For each FRR[i]
  a. Identify the attribute sets as AR[j]
  b. For each AR[j]
    i. If AR[j] infers to ARC[k] and AR[j+1] infers to ARC[k]
    ii. Then, Check AR[j] and AR[j+1] cluster allocation
      1. If AR[j] -> CS[r] and AR[j+1] -> CS[r]
      2. Then, detect data leakage at CS[r] and CS[r]->LC[]
      3. Else, Continue
Step - 4. Report the final leaked clusters as LC[]

Quite a bit of science is grounded in the investigation of equivalences, and request relations. Cross section hypothesis catches the scientific structure of request relations. Despite the fact that equality relations are as omnipresent in arithmetic as request relations, the mathematical structure of equivalences isn't too known as that of requests.

The working flow of the proposed algorithms as a framework is furnished here [Fig. 1].

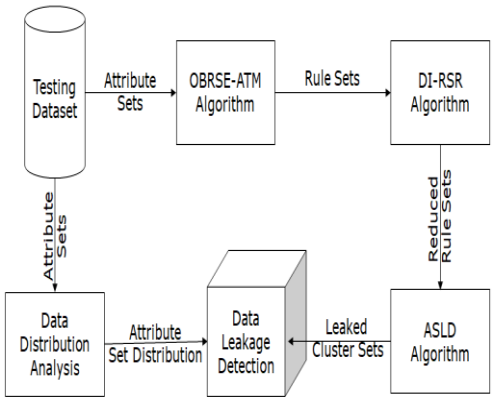

Fig. 1. Working Model of the Proposed Algorithms.

The obtained results from the proposed algorithms are highly satisfactory and discussed in the next section of the work.

VII. Result and Discussion

After the detailed discussed on the mathematical model and the proposed algorithms, in this section of the work, the obtained output from the algorithms are discussed here.

Firstly, the rule extraction results are discussed [Table I].

TABLE I. Rule Extraction Results

Test Number	Number of Rules Extracted	Time Complexity (Sec)
Test Run - 1	329	8.065
Test Run - 2	321	8.301
Test Run - 3	303	8.199
Test Run - 4	326	8.802

The number of extracted rules is the indications of the detailed understanding and deep consideration of all the attribute sets from the dataset. The greater number of rules in this phase of the result defines that most of the attributes are considered and further detection of the leaks can be performed more efficiently.

The results are analysed graphically here [Fig. 2, Fig. 3].

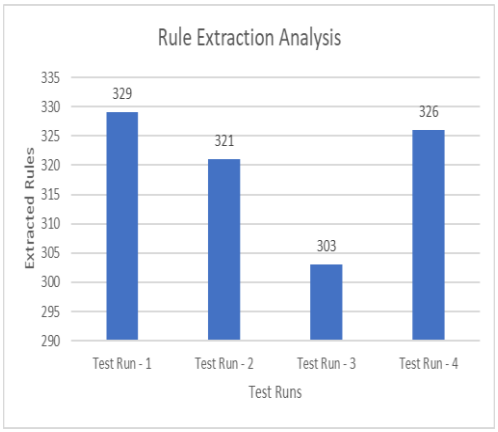

Fig. 2. Rule Extraction Analysis.

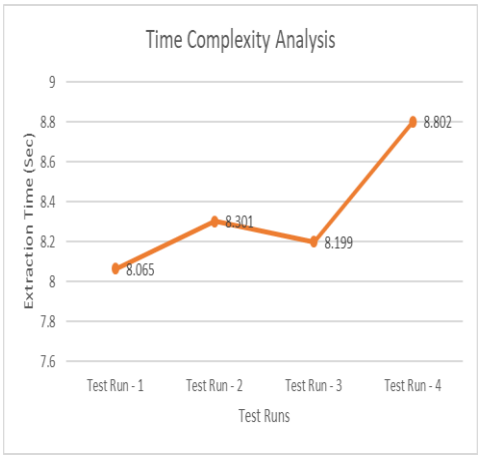

Fig. 3. Rule Extraction Time Complexity Analysis.

However, a greater number of rules can lead to higher time complexity for detection of data leakages. Thus, the reduction of the rulesets is highly expected here. Also, during the rule set reduction process, the point of caution is to maintain the attribute inference properties and relations, which will be helpful in deep detection of the data leakages. Henceforth, the rule set reduction results are discussed here [Table II].

TABLE II. Rule Reduction Results

Test Number	Number of Rules Extracted	Number of Rules after Reduction	Percentage of Reduction (%)	Time Complexity (Sec)
Test Run - 1	329	136	58.66	0.118
Test Run - 2	321	133	58.57	0.120
Test Run - 3	303	126	58.42	0.121
Test Run - 4	326	135	58.59	0.164

The results are analysed graphically here [Fig. 4, Fig. 5, Fig. 6].

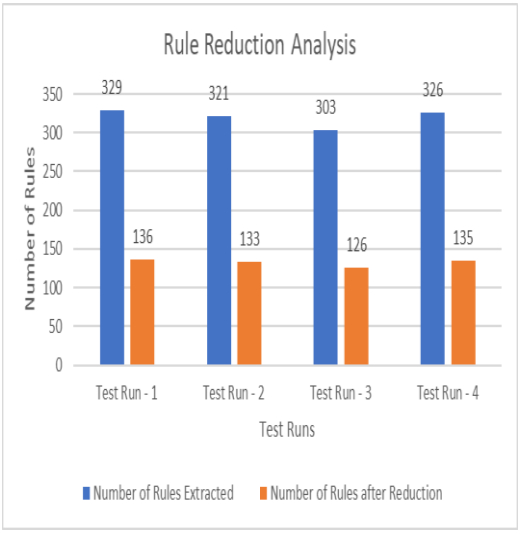

Fig. 4. Rule Reduction Analysis.

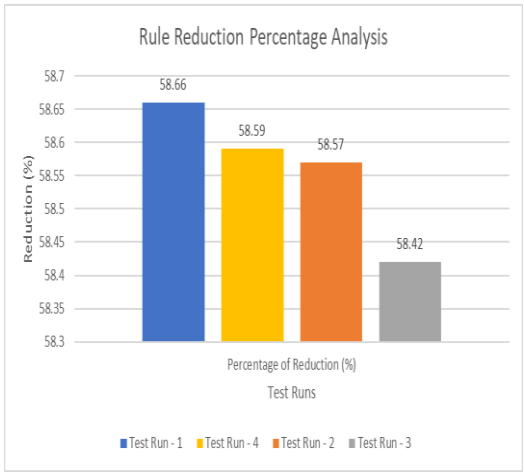

Fig. 5. Rule Reduction Percentage Analysis.

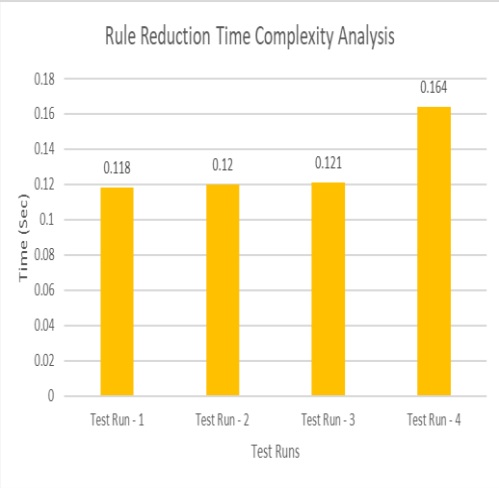

Fig. 6. Rule Reduction Time Complexity Analysis.

Once the rulesets are reduced and the inference properties are extracted, the actual distribution of the data must be analysed as the number of attributes distributed over the test clusters are the key points for detection of the leakages. Hence, the data distribution is analysed here [Table III].

TABLE III. DATA DISTRIBUTION ANALYSIS RESULTS

Test Number	Number of Clusters Detected	Number of Attributes (Mean) stored	Time Complexity (Sec)
Test Run - 1	5	4	0.023
Test Run - 2	4	7	0.022
Test Run - 3	4	7	0.022
Test Run - 4	5	6	0.023

Further after the analysis of the data sets, which are distributed over multiple clusters, finally the data leakages are detected, and the result is furnished here [Table IV].

TABLE IV. DATA LEAKAGE DETECTION ANALYSIS RESULTS

Test Number	Number of Clusters Detected	Number of Clusters with Leakage	Time Complexity (Sec)	Detection Accuracy (%)
Test Run - 1	5	2	0.010	89.55
Test Run - 2	4	3	0.012	89.55
Test Run - 3	4	3	0.009	89.55
Test Run - 4	5	4	0.018	89.32

It is natural to realize that, the higher accuracy of the detection process leads to higher security of the data distribution during the distributed query processing.

The results are analysed graphically here [Fig. 7, Fig. 8].

Henceforth, with the detailed analysis of the obtained results, in the next section of this work, the results are compared with the parallel research outcomes.

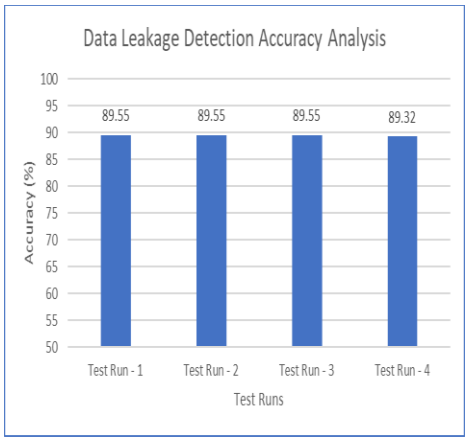

Fig. 7. Data Leakage Accuracy Analysis.

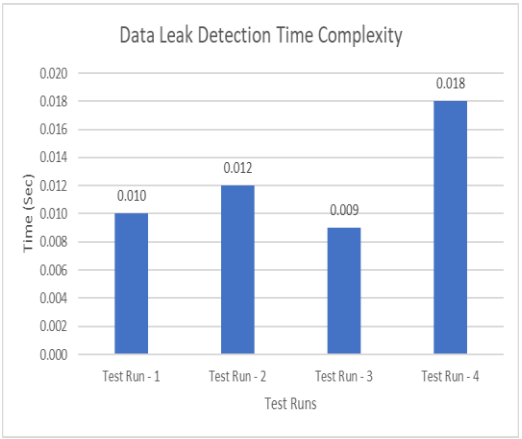

Fig. 8. Data Leakage Time Complexity Analysis.

VIII. COMPARATIVE ANALYSIS

In order to establish the believe that the proposed model is better performing than the other parallel research outcomes, the comparative analysis is most important. Thus, in this section of the work, the proposed algorithms are compared with the parallel research outcomes [Table V].

Hence it is natural to realize that, the proposed algorithms have outperformed the other parallel research attempts. The detailed reason for this achieved benefits are discussed in the previous sections of this work.

Further, with the analysis of the results and comparative analysis, in the next section of the work, the research conclusion is presented.

TABLE V. COMPARATIVE ANALYSIS

Proposed Methods	Year	Fundamental Mechanism	Detection Accuracy (%) [Mean]	Time Complexity (Sec) [Mean]
Appcaulk by J. Schütte et al. [9]	2014	Distributed Data & Taint Tracking	80	0.37
Proactive Warning by T. Malderle et al. [8]	2018	Centralized Data & Historical Data Analysis	88	0.40
Monitoring Methods by S. Trabelsi et al. [10]	2019	Centralized Data & Cost-Based Analysis	85	0.24
OBRSE-ATM, DI-RSR & ASLD Method [12]	2020	Distributed Data, Adaptive Threshold Model, Dynamic Inference & Subset Equivalence Analysis	89.49	0.01

IX. CONCLUSION

The distributed query processing is an essential part of today computing and the challenges of the distributed query processing is the leakage of the data. The data leakage can lead to critical security issues. Thus, this work identifies the solutions to detect the data leaks, which can further be used to ensure data distribution carefully. As mentioned in the previous sections of this work, the detection of the data leakages is highly difficult and demands a deep machine learning based approach. Thus, in order to solve the data leakage detection this work demonstrates a step by step process as initially the data relation between the data set attributes are extracted in form of rulesets. During the further processing, it is been observed that, due to higher number of rulesets in the system, the computational complexity is increasing to a greater extend, thus, this work again deploys a novel mechanism for reduction of rulesets. The deployed algorithm for rule reduction is designed carefully not to lose any inference properties. The rule reduction algorithm demonstrates a nearly 50% reduction in rulesets. Further, the data distribution is analysed, and the number of clusters are detected. Finally, this work deploys yet another novel machine learning based algorithm for detection of the data leakages based on data information equivalence and demonstrates a nearly 89% accuracy. The algorithms designed in this case are highly generic and can be applied for any data distribution scenarios and can be considered as a benchmark in this field of research for making the distributed query processing domain safer and faster.

REFERENCES

[1] Data Breach Index, https://breachlevelindex.com/.

[2] C. Missaoui, S. Bachouch, I. Abdelkader, S. Trabelsi, "Who Is Reusing Stolen Passwords? An Empirical Study on Stolen Passwords and Countermeasures", International Symposium on Cyberspace Safety and Security, pp. 3-17, 2018, October.

[3] Cost of a Data Breach Study: Global Overview, July 2018.

[4] P. Papadimitriou, H. Garcia-Molina, "Data leakage detection", IEEE Transactions on knowledge and data engineering, vol. 23, no. 1, pp. 51-63, 2011.

[5] L. Cheng, F. Liu, D. D. Yao, "Enterprise data breach: causes challenges prevention and future directions", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 7, no. 5, 2017.

[6] S. Liu, R. Kuhn, "Data loss prevention", IT professional, vol. 12, no. 2, 2013.

[7] B. Hauer, "Data and information leakage prevention within the scope of infor-mation security", IEEE Access, vol. 3, pp. 2554-2565, 2015.

[8] T. Malderle, M. Wübbeling, S. Knauer, A. Sykosch, M. Meier, "Gathering and analyzing identity leaks for a proactive warning of affected users", Proceedings of the 15th ACM International Conference on Computing Frontiers, pp. 208-211, 2018.

[9] J. Schütte, D. Titze, J. M. De Fuentes, "Appcaulk: Data leak prevention by injecting targeted taint tracking into android apps", 2014 IEEE 13th International Conference on Trust Security and Privacy in Computing and Communications, pp. 370-379, 2014.

[10] S. Trabelsi, "Monitoring Leaked Confidential Data," 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), CANARY ISLANDS, Spain, 2019, pp. 1-5.

[11] A.Sindhura, J.Rajeshwar, M.V.Narayana , M.Ram Babu, "An Effective Semantic Web Knowledge Processing Mechanism by Using an Adaptive Swarm Intelligence Technique for Ontology (ASITO)", International Journal of Engineering Trends and Technology, Volume 69 Issue 3, 195-200, March 2021, ISSN: 2231 – 5381.

[12] Niladri Shekar Dey, Purnachand Kollapudi, M V Narayana, I Govardhana Rao, "An Automated Framework for Detecting Change in the Source Code and Test Case Change Recommendation", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 11, No. 8, 2020, pp.270-280, ISSN : 2156-5570 (Online), 2158-107X (Print).